



**UNIVERSITÀ
DI TRENTO**
**Dipartimento di
Ingegneria e Scienza dell'Informazione**

Progetto: TRENTOUS

Sviluppo

Doc. Name: *D3-TRENTOUS-Sviluppo*

Doc. Number: *D3-V3.1*

INDICE

Scopo del documento.....	2
1. User Stories.....	3
User Story 1: Login.....	3
User Story 2: Stato dei servizi.....	3
User Story 3: Registrazione Utente.....	4
User Story 4: Segnalazione Problemi.....	4
User Story 5: Sondaggi su misura.....	5
User Story 6: Lista Preferiti.....	5
User Story 7: Commenti.....	6
User Story 8: Ricerca.....	6
User Story 9: Registrazione GdS.....	7
User Story 10: Avvisi.....	7
User Story 11: Form su misura per il GdS.....	8
User Story 12: Registrazione e Verifica del GdS (SuperAdmin).....	8
User Story 13: Controllo Globale (SuperAdmin).....	9
2. User Flow.....	9
3.0 Web APIs.....	12
4.0 Implementazione.....	21
4.1 Repository Organization.....	22
4.2 Branching strategy e organizzazione del lavoro.....	23
4.3 Dependencies.....	23
4.4 Database.....	24
4.5 Testing.....	29
5. Front-End.....	38
5.1 Front-End per l'Utente.....	38
5.2 Front-End per il GdS.....	51
5.3 Front-End per il SuperAdmin.....	67
5.4 Schermate di Errore.....	74
6. Deployment.....	76

Scopo del documento

Il presente documento riporta tutte le informazioni necessarie per lo sviluppo di una parte dell'applicazione Trentus. In particolare, presenta tutti gli artefatti necessari per realizzare le funzioni di segnalazione servizi e visualizzazione dello stato degli stessi con la WebApp Trentus.

Partendo dalla descrizione delle User Stories, il documento prosegue con gli User Flow, la presentazione delle web APIs del servizio web, l'organizzazione del codice, il modello dati e ulteriori informazioni sul testing. Infine, una sezione dedicata al front-end e una agli aspetti di deployment

1. User Stories

Nel presente capitolo vengono riportate le User Stories del progetto Trentus, partendo dai requisiti funzionali individuati nel D1.

User Story 1: Login

Come Utente, voglio poter effettuare il login con credenziali personali (email e password) ed accedere ai servizi di Trentus in modo sicuro e univoco.

Criteri di Accettazione:

- L'utente può accedere con email e password registrate.
- Se le credenziali sono errate, viene mostrato un messaggio di errore.
- Se le credenziali sono corrette, il sistema reindirizza l'utente alla homepage.

Task:

1. **Implementare il form di login**
 - Implementare il form di login con pulsanti e textfield funzionanti per l'accesso con credenziali.
2. **Validazione delle credenziali.**
 - Verificare che le credenziali siano corrette e contenute nel database
 - Assicurarsi che l'utente che accede correttamente sia reindirizzato alla Home Page dell'applicazione
 - Assegnare all'utente il token di autenticazione
3. **Gestione degli errori di autenticazione.**
 - Implementare dei messaggi di errore specifici per problemi relativi al login.

User Story 2: Stato dei servizi

Come Utente, voglio visualizzare lo stato dei servizi, in modo da sapere se sono operativi o presentano problemi.

Criteri di Accettazione:

- La piattaforma mostra una lista di servizi con stato aggiornato.
- La piattaforma indica il numero di segnalazioni ricevute dal servizio
- La piattaforma indica i dettagli del malfunzionamento e i commenti degli utenti.

Task:

1. Creare l'interfaccia per la lista servizi.
 - Implementare nel front-end l'interfaccia con la lista dei servizi, l'interfaccia dei dettagli di un servizio, con grafici che riportino i dati di segnalazione (come numero di segnalazioni o motivi di segnalazione) e una sezione con i commenti degli altri utenti.
2. Implementare il sistema di aggiornamento dei dati.
 - Implementare la logica nel back-end che permette di ricevere i dati delle segnalazioni ed i commenti.
 - Aggiornare i contenuti del front-end.

User Story 3: Registrazione Utente

Come Utente, voglio potermi registrare alla piattaforma, in modo da poter segnalare problemi e ricevere notifiche.

Criteri di Accettazione:

- Il modulo di registrazione raccoglie email e password.
- La pagina si occupa di verificare che le credenziali siano valide.

Task:

1. Creare form di registrazione.
 - Implementare nel front-end il form di registrazione, con textfield e tasti per l'inserimento di email e password
2. Implementare validazione dati.
 - Implementare nel back-end la logica che serve per verificare la correttezza dei dati forniti
 - Implementare nel back-end la logica che serve per inserire i dati, se corretti, nel database
3. Gestire attivazione account.
 - Assicurarsi che l'account appena registrato e confermato sia capace di loggarsi e sfruttare a pieno le potenzialità dell'applicazione

User Story 4: Segnalazione Problemi

Come Utente, voglio poter segnalare un malfunzionamento di un servizio, in modo che venga risolto.

Criteri di Accettazione:

- L'utente può segnalare un problema tramite il bottone dedicato.
- Il sistema raccoglie dettagli e categorie del problema.
- Il sistema raccoglie un eventuale commento dell'utente.
- La segnalazione viene salvata e visualizzata pubblicamente.

Task:

1. Creare interfaccia per segnalazioni.
 - Implementare nel front-end un'interfaccia che permetta all'utente di segnalare il malfunzionamento di un servizio, specificando le problematiche con commenti e con un form
2. Implementare raccolta dati e invio al database.
 - Implementare nel back-end la logica che serve a prendere i dati forniti dall'utente per mandarli al database.

User Story 5: Sondaggi su misura

Come Utente, voglio poter specificare il tipo di problema con un menù dedicato, in modo che la segnalazione sia più precisa.

Criteri di Accettazione:

- L'utente può selezionare il problema da un menu.
- Il menù varia a seconda del servizio segnalato.
- Il sistema salva la scelta insieme alla segnalazione.

Task:

- Creare menu dinamico per selezione problemi.
- Collegare menu a database delle segnalazioni.
- Implementare salvataggio dei dati.

User Story 6: Lista Preferiti

Come Utente, voglio poter aggiungere servizi ai preferiti, in modo da monitorarli facilmente.

Criteri di Accettazione:

- L'utente può aggiungere/rimuovere un servizio dai preferiti.
- La lista preferiti è accessibile rapidamente.

Task:

1. Creare funzionalità di aggiunta/rimozione preferiti.
 - Implementare nel front-end un tasto che permette l'aggiunta o la rimozione di un servizio dalla lista dei preferiti
2. Implementare salvataggio nel database.
 - Implementare nel back-end la logica necessaria per salvare le liste dei preferiti di tutti gli utenti
3. Creare interfaccia per la lista preferiti.
 - Implementare nel front-end una schermata che raccolga solo i servizi preferiti dell'utente

User Story 7: Commenti

Come Utente, voglio poter commentare una segnalazione per fornire dettagli aggiuntivi.

Criteri di Accettazione:

- L'utente registrato può aggiungere un commento.
- Il commento viene salvato e visualizzato nella segnalazione.

Task:

1. Creare modulo di commento.
 - Implementare nel front-end un textfield che permetta all'utente di inserire un commento (facoltativo) ad una segnalazione
2. Implementare gestione e salvataggio commenti.
 - Implementare nel back-end la logica che permette all'applicazione di salvare i commenti
3. Mostrare commenti sotto le segnalazioni.
 - Implementare nel front-end una sezione dei dettagli del servizio che mostri i commenti degli utenti, dai più recenti ai meno recenti

User Story 8: Ricerca

Come Utente, voglio poter cercare un servizio, in modo da verificarne rapidamente lo stato.

Criteri di Accettazione:

- L'utente può cercare un servizio con una barra di ricerca.
- Il sistema mostra risultati pertinenti.

Task:

1. Implementare barra di ricerca.
 - Implementare nel front-end una barra di ricerca che permetta all'utente di scriverci dentro
2. Collegare ricerca al database dei servizi.
 - Implementare nel back-end la logica che permette all'applicazione di fornire il risultato cercato dall'utente

(NOTA: nella versione finale del sito questa funzione è stata scartata a causa del poco tempo disponibile. Abbiamo deciso di scartare questa precisa funzione poiché non la consideravamo così importante rispetto alle altre, in quanto la lista dei servizi presente sul database è abbastanza piccola, quindi l'implementazione di una barra di ricerca sarebbe stata poco utile. Si veda sezione Criticità del D4)

User Story 9: Registrazione GdS

Come GdS, voglio registrare il mio servizio sulla piattaforma, in modo da poter interagire con gli utenti.

Criteri di Accettazione:

- Il GdS compila un modulo con dati aziendali.
- La richiesta viene inviata al SuperAdmin per approvazione.

Task:

1. Creare form di registrazione GdS.
 - Implementare nel front-end un form di registrazione servizio che contenga le credenziali richieste
2. Implementare invio richiesta al SuperAdmin.
 - Implementare nel back-end la logica che serve per gestire la richiesta e mandarla per la revisione del SuperAdmin
 - Implementare nel front-end una schermata dell'applicazione che consenta al SuperAdmin di verificare i dati di registrazione e di accettare o rifiutare la registrazione di un servizio
3. Gestire approvazione e creazione profilo.
 - Implementare nel back-end la logica necessaria affinché, in caso di approvazione del servizio, questo venga inserito nel database dell'applicazione e reso visibile a tutti gli utenti

User Story 10: Avvisi

Come GdS, voglio poter avvisare gli utenti di interruzioni o manutenzioni, in modo da tenerli informati.

Criteri di Accettazione:

- Il GdS può creare un avviso con dettagli e durata.
- L'avviso viene mostrato agli utenti.

Task:

1. Creare modulo per pubblicazione avvisi.
 - Implementare nel front-end una pagina per l'invio di avvisi agli utenti
2. Implementare visibilità avvisi sui servizi.
 - Implementare nel front-end una sezione del sito che mostri gli avvisi mandati dai GdS dei servizi preferiti

User Story 11: Form su misura per il GdS

Come GdS, voglio mettere a disposizione un form semplice e conciso per permettere agli utenti di segnalare in maniera precisa le problematiche riscontrate durante l'utilizzo del servizio.

Criteri di Accettazione:

- Il GdS può creare un form associato al proprio servizio.
- Il form è semplice, conciso e facilmente compilabile dall'utente.
- Le risposte del form vengono salvate insieme alla segnalazione del problema.

Task:

1. Creare l'interfaccia per la gestione dei form su misura.
 - Implementare nel front-end una schermata che permetta al GdS di creare una checklist, da far compilare all'utente, con i possibili problemi riscontrati
2. Collegare il form alle segnalazioni del servizio.
 - Implementare nel back-end la logica necessaria per inviare i dati dei form di segnalazione creati dal GdS
3. Testare il flusso di compilazione e salvataggio dei dati.
 - Verificare che i form di segnalazione vengano salvati e mostrati correttamente all'utente finale.

User Story 12: Registrazione e Verifica del GdS (SuperAdmin)

Come SuperAdmin, voglio verificare e registrare un GdS, creandone il profilo e il servizio associato, in modo da garantire l'affidabilità dei servizi presenti sulla piattaforma.

Criteri di Accettazione:

- Il SuperAdmin può visualizzare le richieste di registrazione inviate dai GdS.
- Il SuperAdmin può approvare o rifiutare una richiesta.
- In caso di approvazione, il profilo del GdS viene creato.

Task:

1. Creare l'interfaccia per la gestione delle richieste di registrazione dei GdS.
 - Implementare nel front-end una schermata che permetta al SuperAdmin di vedere tutte le richieste di aggiunta servizio arrivate
2. Implementare il flusso di approvazione/rifiuto.
 - Inserire nel front-end i bottoni di approvazione/rifiuto per i servizi
 - Inserire nel back-end la logica necessaria per far arrivare le richieste di aggiunta servizio ai SuperAdmin.
 - Inserire nel back-end la logica necessaria per far arrivare l'approvazione o il rifiuto dell'aggiunta di un nuovo servizio da parte del SuperAdmin

User Story 13: Controllo Globale (SuperAdmin)

Come SuperAdmin, voglio avere accesso completo al database e agli strumenti di gestione della piattaforma, in modo da poter mantenere la WebApp pulita, sicura e performante.

Criteri di Accettazione:

- Il SuperAdmin ha accesso completo ai dati dei servizi.
- Il SuperAdmin può gestire e modificare i dati in caso di necessità.

Task:

1. Implementare un'interfaccia di controllo dedicata al SuperAdmin.
 - Inserire nel front-end una Home Page del SuperAdmin con completo accesso ai dati dei servizi
 - Inserire nel front-end una schermata per la modifica dei dati dei servizi
 - Implementare nel back-end la logica necessaria per la modifica globale dei dati dei servizi

2. User Flow

In questa sezione riportiamo gli User Flow per i ruoli di Utente, Gestore del Servizio (GdS) e SuperAdmin.

Nell'*Immagine 1* è riportata la legenda, necessaria per la comprensione degli User Flows.

L'*Immagine 2* descrive l'intera esperienza utente disponibile per gli Utenti. L'Utente, accedendo al sito, viene subito portato alla HomePage di Trentus. Da qui, l'utente non loggato può visualizzare la lista dei servizi, i servizi più segnalati ed i dettagli e i commenti per ciascun servizio. L'utente non loggato, cliccando sull'icona delle impostazioni, viene portato alla pagina di Login per accedere con un account o creare uno nuovo. Quando l'Utente si autentica, potrà segnalare i servizi, compilando il form di segnalazione e lasciando (eventualmente) un commento. L'Utente loggato potrà inoltre impostare alcuni servizi come "preferiti", potendoli filtrare e ricevendo notifiche e sondaggi dai GdS di tali servizi. L'Utente può accedere al contenuto di tali notifiche, e può compilare i sondaggi, attraverso la barra delle notifiche. Infine, l'Utente, cliccando sul tasto impostazioni presente sulla Navbar superiore, potrà fare il logout.

L'*Immagine 3* descrive l'intera esperienza utente disponibile per i Gestori di un Servizio (GdS). Il primo accesso al sito per un GdS avviene sulla pagina di login. Qui, il GdS, potrà loggare usando le sue credenziali (se già accettate da un SuperAdmin), oppure potrà registrarsi, facendo richiesta di aggiunta del suo servizio sulla piattaforma. Una volta loggato, il GdS verrà portato alla HomePage GdS di Trentus. Sulla sua HomePage, il GdS potrà verificare tutti i dati di segnalazione, attraverso semplici grafici, e i commenti degli utenti. Il GdS può cliccare sul tasto "Modifica", accedendo così ad una pagina in cui potrà modificare i dati del proprio servizio, come il nome, la descrizione, lo stato del servizio ed il logo. Il GdS può cliccare sul tasto "Invia Avviso", accedendo così ad una pagina in cui potrà mandare un avviso agli utenti che hanno messo il proprio servizio tra i preferiti, compilando un messaggio che indica il problema, il motivo del problema stesso e la sua durata. Il GdS può cliccare sul tasto "Invia Sondaggio", accedendo così ad una pagina di creazione sondaggi, in cui il GdS potrà creare sondaggi servendosi di checklists, domande a risposta multipla o domande a risposta aperta. Alla sinistra della schermata home per i GdS vengono riportati i sondaggi mandati agli utenti. Cliccando sui sondaggi mandati, il GdS potrà visionare i risultati degli stessi, contenenti le risposte degli utenti. Infine, cliccando sul tasto impostazioni presente nella Navbar superiore, il GdS potrà fare il logout.

L'*Immagine 4* descrive l'intera esperienza utente disponibile per i SuperAdmin. Il SuperAdmin, accedendo a Trentus, viene portato alla pagina di login a lui dedicata. Non è possibile registrarsi in quanto le credenziali per i SuperAdmin vengono generate separatamente dal sistema. Una volta loggato, il SuperAdmin accederà alla HomePage per i SuperAdmin, contenente la lista di tutti i servizi registrati su Trentus. Il SuperAdmin, accedendo alla schermata in dettaglio di un singolo servizio, può cliccare sul tasto "Modifica" per modificarne i dettagli, come il logo, il nome, la descrizione o lo stato del servizio. Alla sinistra della schermata, il SuperAdmin può trovare le richieste di aggiunta servizio provenienti dai GdS appena registrati. Cliccandoci sopra, il SuperAdmin verrà portato su una

pagina in cui è possibile visualizzare i dettagli forniti dal GdS che vuole aggiungere il proprio servizio su Trentus. Da lì, il SuperAdmin può decidere se accettare la richiesta o se rifiutarla.



Immagine 1: Legenda

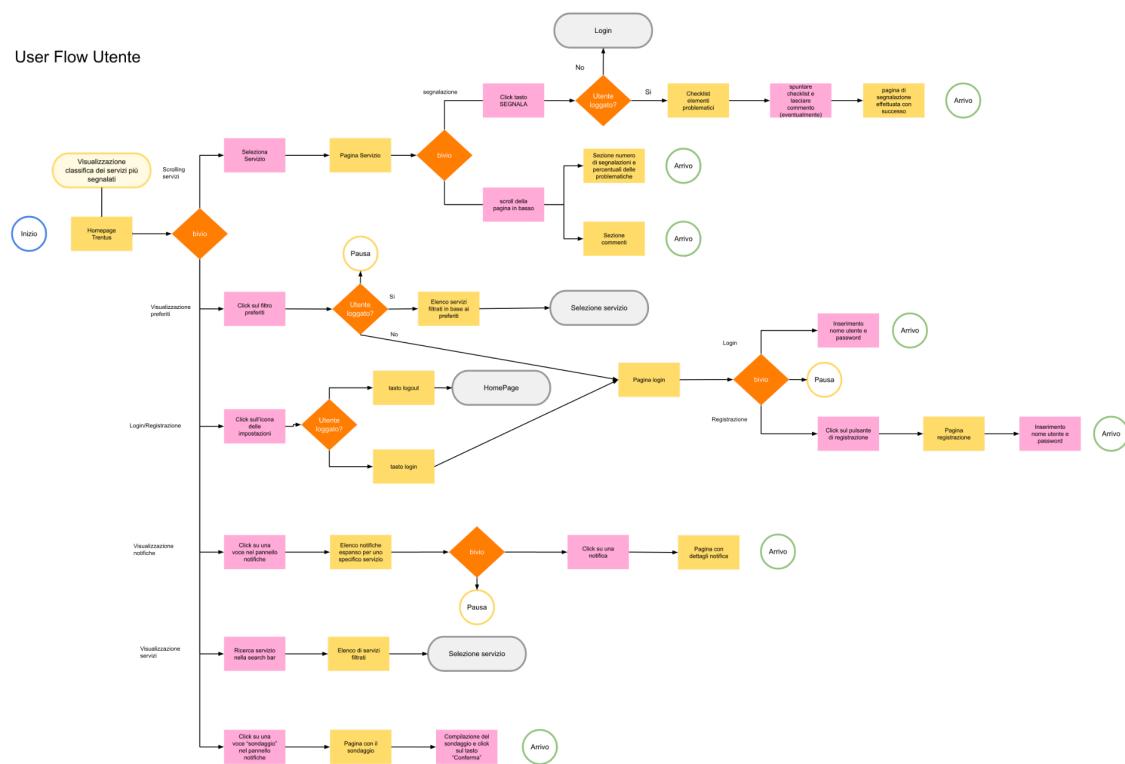


Immagine 2: User Flow per l'Utente

User Flow GdS

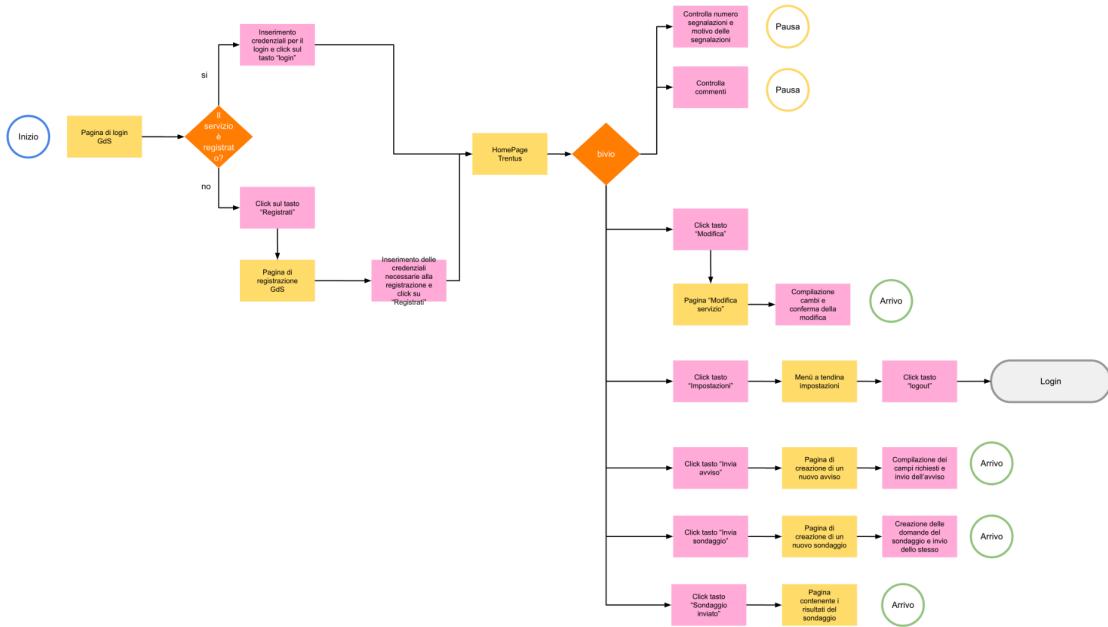


Immagine 3: User Flow per il Gestore del Servizio

User Flow SuperAdmin

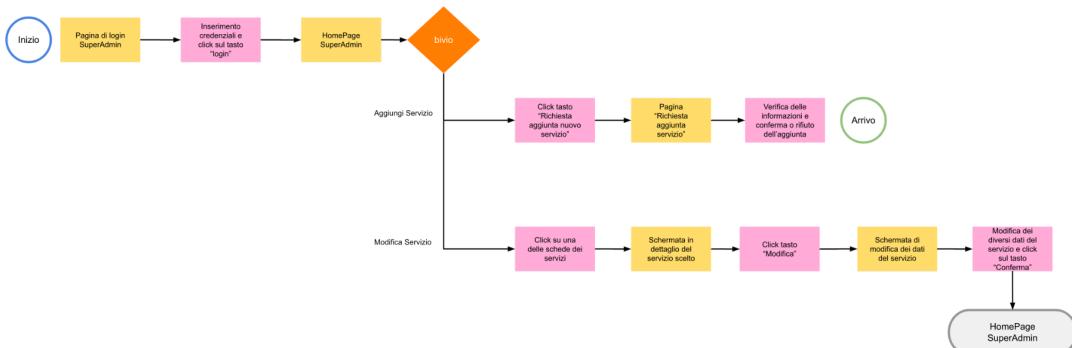


Immagine 4: User Flow per il SuperAdmin

3.0 Web APIs

Questa qui presentata è una versione ritagliata dello swagger completo, che si può trovare su [swagger.yaml](#)

Molte cose sono omesse poiché ridondanti:

- Tutti i codici HTML che non sono 200
- Tutti i required nei parametri in path
- Alcune descrizioni
- il parametro "authorization" nell'header, che fa riferimento al JWT
- SuperAdmin nei bearerAuth dove non è richiesto (perché può sempre accedere)

Gli schemi dei componenti, anche questi scritti e modificati con il progredire del progetto, sono stati alleggeriti, descrivendo solo il tipo della variabile.

Una vista più dettagliata degli schemi si può trovare su [swagger.comp.yaml](#)

La gestione delle API è probabilmente stato il documento che è più cambiato nel corso dello sviluppo del progetto, rivisitato molteplici volte, di cui si possono vedere versioni precedenti in [SwaggerOld](#). Le prime versioni sono state scritte a mano.

L'ultima versione invece è stata adattata per essere anche visualizzata da browser su /api-docs per essere gestita, oltre che come piano di progetto, come documentazione. Questa viene generata leggendo estratti delle versioni scritte a mano dello swagger in commenti posti sulle rotte ed importando alla fine del documento lo schema di tutti i componenti sopra elencato.

Nota! Per evitare ripetizioni, lo schema dei componenti non sarà inclusa in questa sezione, ma si potrà trovare nella parte **4.4: Database**.

```
openapi: 3.0.0
info:
  version: 2.0.0
  title: Trentus API
  description: Documentazione delle API di Trentus
servers: (dopo il deployment)
- url: http://localhost:8080
  description: Server Locale
paths:
  "/servizi/{servizio_id}/avvisi/":
    get:
      description: Restituisce tutti gli avvisi
      parameters:
        - name: servizio_id
          in: path
          type: string
      responses:
        '200':
          content:
            application/json:
              schema:
                type: array
                items:
```

```
        "$ref": "#/components/schemas/Avviso"
post:
  description: Crea un avviso ed invia una mail agli utenti interessati
  parameters:
    - name: servizio_id
      in: path
      type: string
  security:
    - BearerAuth:
      - gds
  requestBody:
    content:
      application/json:
        schema:
          "$ref": "#/components/schemas/Avviso"
"/servizi/{servizio_id}/avvisi/{avviso_id}":
get:
  description: Restituisce un avviso in base all'id
  parameters:
    - name: servizio_id
      in: path
      type: string
    - name: avviso_id
      in: path
      type: string
  responses:
    '200':
      content:
        application/json:
          schema:
            "$ref": "#/components/schemas/Avviso"
"/servizi/{servizio_id}/sondaggi__":
get:
  description: Restituisce i sondaggi compilabili del servizio
  parameters:
    - name: servizio_id
      in: path
      type: string
  responses:
    '200':
      content:
        application/json:
          schema:
            type: array
            items:
              "$ref": "#/components/schemas/Sondaggio"
  security:
    - BearerAuth:
      - gds
      - utente
```

```
post:
  description: Crea un sondaggio
  parameters:
  - name: servizio_id
    in: path
    type: string
  responses:
  '201':
    description: Created
  security:
  - BearerAuth:
    - gds
  requestBody:
    content:
      application/json:
        schema:
          "$ref": "#/components/schemas/Sondaggio"
"/servizi/{servizio_id}/sondaggi/{sondaggio_id}":
get:
  description: Restituisce un sondaggio
  parameters:
  - name: servizio_id
    in: path
    type: string
  - name: sondaggio_id
    in: path
    type: string
  responses:
  '200':
    content:
      application/json:
        schema:
          "$ref": "#/components/schemas/Sondaggio"
  security:
  - BearerAuth:
    - gds
    - utente
post:
  description: Rilascia un feedback
  parameters:
  - name: servizio_id
    in: path
    type: string
  - name: sondaggio_id
    in: path
    type: string
  responses:
  '201':
    description: Created
  security:
```

```
- BearerAuth:  
  - utente  
requestBody:  
  content:  
    application/json:  
      schema:  
        "$ref": "#/components/schemas/Feedback"  
"/servizi/{servizio_id}/sondaggi/{sondaggio_id}/risultati":  
get:  
  description: Restituisce i risultati di un sondaggio  
  parameters:  
    - name: servizio_id  
      in: path  
      type: string  
    - name: sondaggio_id  
      in: path  
      type: string  
  responses:  
    '200':  
      content:  
        application/json:  
          schema:  
            type: array  
            items:  
              "$ref": "#/components/schemas/Feedback"  
  security:  
    - BearerAuth:  
      - gds  
"/servizi/{servizio_id}/segnalazioni/form":  
get:  
  description: Restituisce la form di segnalazione compilabile per un servizio  
  parameters:  
    - name: servizio_id  
      in: path  
      type: string  
  responses:  
    '200':  
      content:  
        application/json:  
          schema:  
            "$ref": "#/components/schemas/Servizio/properties/Form"  
post:  
  description: Genera la form di segnalazione compilabile per un servizio  
  parameters:  
    - name: servizio_id  
      in: path  
      type: string  
  responses:  
    '201':  
      description: Created
```

```
security:
- BearerAuth:
  - gds
requestBody:
  content:
    application/json:
      schema:
        "$ref": "#/components/schemas/Servizio/properties/Form"
"/servizi/{servizio_id}/segnalazioni/compila":
  post:
    description: Salva i dati del form di segnalazione del servizio compilato dall
parameters:
- name: servizio_id
  in: path
  type: string
responses:
  '201':
    description: Created
security:
- BearerAuth:
  - utente
requestBody:
  content:
    application/json:
      schema:
        "$ref": "#/components/schemas/Segnalazione"
"/servizi/{servizio_id}/segnalazioni/commenti":
  get:
    description: Restituisce i commenti delle segnalazioni di un servizio
parameters:
- name: servizio_id
  in: path
  type: string
responses:
  '200':
    content:
      application/json:
        schema:
          type: array
          items:
            "$ref": "#/components/schemas/Segnalazione/properties/Commento"
  '201':
    description: Created
"/servizi/{servizio_id}/segnalazioni/istogramma":
  get:
    description: Restituisce dati per la creazione del grafico di frequenza
segnalazioni
parameters:
- name: servizio_id
  in: path
```

```
type: string
responses:
  '200':
    content:
      application/json:
        schema:
          "$ref": "#/components/schemas/Segnalazione/properties/Grafico"
"/servizi/{servizio_id}/segnalazioni/areogramma":
  get:
    description: Restituisce dati per la creazione del grafico di problemi
riscontrati
    parameters:
      - name: servizio_id
        in: path
        type: string
    responses:
      '200':
        content:
          application/json:
            schema:
              "$ref": "#/components/schemas/Segnalazione/properties/Grafico"
"/gds/login":
  post:
    description: Login GdS
    responses:
      '200':
        description: OK
    requestBody:
      content:
        application/json:
          schema:
            "$ref": "#/components/schemas/Login"
"/gds/registrazione":
  post:
    description: Invia una richiesta di registrazione per un GdS
    responses:
      '201':
        description: Created
    requestBody:
      content:
        application/json:
          schema:
            "$ref": "#/components/schemas/RichiestaGdS"
"/gds/servizio":
  get:
    description: Restituisce il servizio associato al GdS
    responses:
      '200':
        content:
          application/json:
```

```
schema:
    "$ref": "#/components/schemas/Servizio"
security:
- BearerAuth:
  - gds
"/utente/login":
post:
  description: Login Utente
  responses:
    '200':
      description: OK
  requestBody:
    content:
      application/json:
        schema:
          "$ref": "#/components/schemas/Login"
"/utente/":
get:
  description: Restituisce i dati di un utente
  parameters:
  responses:
    '200':
      content:
        application/json:
          schema:
            "$ref": "#/components/schemas/Utente"
  security:
- BearerAuth:
  - utente
"/utente/registrazione":
post:
  description: Registra un nuovo utente
  responses:
    '201':
      description: Created
  requestBody:
    content:
      application/json:
        schema:
          "$ref": "#/components/schemas/Utente"
"/utente/preferiti":
get:
  description: Restituisce i preferiti di un utente
  parameters:
  responses:
    '200':
      content:
        application/json:
          schema:
            "$ref": "#/components/schemas/Utente/properties/PREFERITI"
```

```
security:
- BearerAuth:
  - utente
"/servizi/":
get:
  description: Restituisce tutti i servizi
  responses:
    '200':
      content:
        application/json:
          schema:
            type: array
            items:
              "$ref": "#/components/schemas/Servizio"
"/servizi/classifica":
get:
  description: "Classifica dei servizi più problematici"
  responses:
    '200':
      content:
        application/json:
          schema:
            type: array
            items:
              type: object
              properties:
                servizio_id:
                  type: string
                segnalazioni:
                  type: integer
"/servizi/{servizio_id}":
get:
  description: Restituisce servizio in base al suo id
  parameters:
  - name: servizio_id
    in: path
    type: string
  responses:
    '200':
      content:
        application/json:
          schema:
            "$ref": "#/components/schemas/Servizio"
"/servizi/{servizio_id}/modifica":
post:
  description: Servizio ON
  parameters:
  - name: servizio_id
    in: path
    type: string
```

```
responses:
  '201':
    description: Created
requestBody:
  content:
    application/json:
      schema:
        "$ref": "#/components/schemas/Servizio"
security:
- BearerAuth:
  - gds
"/controlpanel/login":
post:
  description: Login Superadmin
  responses:
    '200':
      description: OK
  security:
- BearerAuth:
  - superadmin
  requestBody:
    content:
      application/json:
        schema:
          "$ref": "#/components/schemas/Login"
"/controlpanel/confermagds/lista":
get:
  description: Ottieni richieste di registrazione per un GdS
  parameters:
  responses:
    '201':
      description: Created
  security:
- BearerAuth:
  - superadmin
"/controlpanel/confermagds/{richiesta_id}":
get:
  description: Conferma la richiesta di registrazione per un GdS
  parameters:
- name: richiesta_id
  in: path
  type: string
  responses:
    '201':
      description: Created
  security:
- BearerAuth:
  - superadmin
```

```
components:  
  securitySchemes:  
    BearerAuth:  
      type: apiKey  
      name: Authorization  
      in: header  
      description: Bearer token per accedere alle API  
  security:  
    - BearerAuth: [gds, utente, superadmin]
```

4.0 Implementazione

Abbiamo scelto VueJS per il frontend principalmente per il materiale e il supporto fornito durante il corso, inoltre Giovanni aveva già alcune conoscenze basilari del framework.

Abbiamo deciso inoltre di utilizzare alcuni plugin essenziali che ci sono tornati molto utili nello sviluppo, come vue-router, per gestire le rotte lato frontend, e vuex, per la creazione e la gestione dello store.

Abbiamo scelto NodeJS ed Express per il back-end poichè queste erano state proposte a lezione, nonostante nessuno di noi li avesse mai utilizzati

Anche mongoDB è stato scelto principalmente per le motivazioni sopracitate, ma durante il suo utilizzo ci siamo resi subito conto della flessibilità che offre rispetto ad altri DBMS che adottano un approccio relazionale classico molto più rigido.

Insomnia è stato utilizzato perché al tempo in classe non avevamo ancora visto Jest e ci serviva un metodo per testare il corretto funzionamento delle API dal lato back-end, e questo è quello che abbiamo reputato più appropriato.

4.1 Repository Organization

Il codice del progetto (<https://github.com/ldrunitn/Trentus>) è organizzato secondo la seguente struttura:

```
Trentus
└── Back-end                                //cartella contenente il backend
    ├── controllers                           //cartella controller del backend
    ├── middleware                            //cartella contenente il middleware
    ├── models                               //
    └── public                                //cartella contenente immagini
        └── routes                             //cartella contenente le routes express
    ├── schemas                             //cartella contenente gli schemi di mongoose
    └── Documenti                            //cartella contenente i documenti del progetto
    └── Frontend                             //cartella contenente il frontend
        ├── public                            //
        └── src                                //cartella file sorgenti del frontend
            ├── assets                           //cartella contenente le immagini/icone del sito
            ├── components                      //cartella contenente i componenti di vue
            │   ├── admin                         //componenti per le view dell'admin
            │   └── sidebarLeft                  //componenti per la sidebar sinistra dell'admin
            ├── classification                //componenti per la sidebar destra
            ├── icons                          //
            ├── navbar                         //componenti per la navbar
            └── services                       //componenti per la view del servizio
            └── router                         //cartella contenente il vue router
            └── store                           //cartella contente i file di vuex, divisi per ruolo
                └── modules                     //
                    ├── admin                   //
                    ├── gds                     //
                    └── user                     //
            └── views                           //cartella contenente le view, divise per ruolo
                ├── admin                   //
                ├── gds                     //
                └── user                     //
```

4.2 Branching strategy e organizzazione del lavoro

Lo sviluppo ha visto una collaborazione attiva tra i membri del gruppo. Ci siamo divisi il lavoro in questo modo:

- Giovanni Sbalchiero si è occupato di aiutare nello sviluppo iniziale del back-end.
Ha poi lavorato sul front-end e del collegamento di quest'ultimo con il back-end.
- Giuseppe Conte si è occupato dello sviluppo del front-end, con più enfasi sulla parte grafica e sullo styling delle pagine
- Luca Da Riz si è occupato nello sviluppo delle API e del back-end ed il suo mantenimento e aggiornamento secondo le necessità del front-end.

In totale abbiamo eseguito 188 commit, distribuiti tra i membri in questo modo:

- Giovanni Sbalchiero: 87 commit
- Giuseppe Conte: 33 commit
- Luca da Riz: 68 commit

Lo sbilanciamento tra il numero di commit è dovuto dal fatto che Giuseppe, non avendo mai toccato gli strumenti di sviluppo usati per il progetto, ha passato un po' di tempo nell'apprendimento di questi ultimi. Il suo coinvolgimento nello sviluppo, infatti, è più in là nel tempo rispetto alla creazione della repository. Inoltre, essendosi occupato della parte grafica, la maggior parte delle commit di Giuseppe è composta da grossi pezzi di codice.

Nel conteggio dei commit di Giovanni, c'è da considerare che gli ultimi 10/15 push sono stati fix di dettagli, spesso di poche righe di codice.

Luca, per organizzare i problemi riscontrati più onerosi, ha anche utilizzato la funzionalità degli issues, e la creazione di pull request per la correzione di queste.

A livello di repository abbiamo optato per una strategia di Feature Branch Workflow. Per lo sviluppo di ogni piccola feature, abbiamo lavorato su una singola branch dedicata. Con il continuo dello sviluppo, le varie feature più piccole venivano unite al main, per mantenere la repository principale aggiornata.

Per quanto riguarda l'elenco delle branch utilizzate, molte sono state eliminate dopo le relative pull requests, per questo motivo, invece di riportare un elenco incompleto non soddisfacente, preferiamo segnalarvi il link di github, in cui sono visibili le branch rimanenti:
<https://github.com/lrunitn/Trentus/branches/active>

4.3 Dependencies

Il progetto npm si basa sui seguenti moduli esterni:

frontend:

└── @vitejs/plugin-vue@5.2.0:	componente che usa vite per gestire i plugin
└── axios@1.7.8:	usato per effettuare le chiamate alle API
└── chart.js@4.4.6:	usato per la visualizzazione dei grafici
└── dotenv@16.4.7:	usato per gestire le variabili globali
└── js-cookie@3.0.5:	usati per l'utilizzo di cookies nell'applicazione
└── primeicons@7.0.0:	usate per visualizzare alcune icone
└── tailwind@4.0.0:	usato per dare lo stile al sito
└── tailwindcss@3.4.15:	usato per dare lo stile al sito
└── vite@5.4.11:	componente usato per gestire i progetti vue
└── vue-chartjs@5.3.2:	usato per gestire i grafici di chart.js con vue
└── vue-router@4.4.5:	usato per gestire le rotte con vue
└── vue@3.5.13:	framework frontend su cui si basa tutta l'applicazione
└── vuex-persistedstate@4.1.0:	usato per rendere persistenti alcuni attributi dello store
└── vuex@4.0.2:	usato per creare e gestire lo store

backend:

└── bcrypt@5.1.1:	usati per cifrare la password dell'utente
└── bcryptjs@2.4.3:	usati per cifrare la password dell'utente
└── body-parser@1.20.3:	usato per convertire i body delle richieste ed estrarre i campi
└── cors@2.8.5:	usato per permettere a tutti gli utenti di connettersi al sito
└── dotenv@16.4.5:	usato per gestire tutte le variabili globali dell'applicazione
└── express@4.21.2:	framework su cui si basa tutto il backend
└── http-errors@1.6.3:	usato per restituire gli errori alle richieste http
└── jsonwebtoken@9.0.2:	usato per generare e verificare i token nell'autenticazione
└── mongoose@8.9.5:	usato per interfacciarsi con mongoDB
└── sharp@0.33.5:	usato per ridimensionare le immagini caricate dagli utenti
└── multer@1.4.5-lts.1:	usato per salvare i file mandati dagli utenti
└── nodemailer@6.10.0:	usato per la gestione della mail
└── swagger-ui-express	usato per creare la pagina api-docs
└── uuid-v4@0.1.0	usato per la gestione delle immagini
└── uuidv4@6.2.13	usato per la gestione delle immagini

4.4 Database

Per la gestione dei dati utili all'applicazione abbiamo definito varie principali strutture dati tramite l'utilizzo di mongoose. faremo qui riferimento agli schemi dei componenti alleggeriti dello swagger. (vedi sezione **3.1 API**).

Abbiamo 2 principali componenti per gestire i ruoli:

- **Utente:**

Utilizzato per rappresentare la classe Utente e la classe SuperAdmin. La distinzione tra i due è fatta da un flag *checkSuperAdmin* abilitato a vero se l'utente è un superadmin. Oltre agli identificatori (email e password in hash) ogni utente può anche avere un array di Servizio.id per rappresentare i servizi preferiti.

- **GdS:**

Utilizzato per rappresentare la classe GdS. Anche lui ha gli identificatori (email e password in hash), ma al contrario dell'utente, possiede anche l'id del servizio che gestisce.

Abbiamo invece 6 componenti che sono le fondamenta della webapp:

- **Avviso:**

Ogni avviso ha un titolo, un corpo, per spiegare il problema, un enumeratore tipo, per classificare il problema tra "manutenzione", "problemi" e "down", ed ovviamente l'id del servizio a cui è collegato

- **Segnalazione:**

Le segnalazioni sono identificate dal servizio a cui sono collegate e dall'utente che l'ha inviata. Sono memorizzate nel database tramite un array di oggetti che rappresentano le risposte dell'utente per ogni domanda

- **Sondaggio:**

Ogni sondaggio ha un titolo, un corpo, che è composto da un array di oggetti, ognuno rappresentante una domanda, e l'id del servizio a cui fa riferimento

- **Feedback:**

Ogni feedback è individuato da 3 identificatori: il sondaggio compilato, il servizio a cui fa riferimento, e l'utente che lo ha compilato. Contiene poi un array di oggetti che rappresentano le varie risposte dell'utente

- **Servizio:**

Il servizio è l'oggetto più grande del database. Contiene varie informazioni riguardanti se stesso e l'azienda che lo eroga, il suo stato, un enumeratore che può essere "on", "manutenzione", "problemi" e "down". Infine il form, che non è che un array di oggetti con il tipo di opzione e la descrizione

- **GdSRequest:**

Contiene tutte le sopracitate informazioni riguardanti il servizio ed il GdS, insieme ad un flag confermata per indicare che il servizio descritto è stato approvato dal SuperAdmin e creato.

```
components:  
schemas:  
  Avviso:  
    type: object  
    required:  
      - titolo  
      - corpo  
      - tipo  
      - servizio_id  
    properties:  
      titolo:  
        type: string  
      corpo:  
        type: string  
      tipo:  
        type: string  
        enum:  
          - manutenzione  
          - problemi  
          - down  
      inizio:  
        type: string  
      fine:  
        type: string  
    servizio_id:  
      type: string  
Feedback:  
  type: object  
  required:  
    - risposte  
    - utente_id  
    - sondaggio_id  
    - servizio_id  
  properties:  
    risposte:  
      type: array  
      items:  
        type: object  
    utente_id:  
      type: string  
    sondaggio_id:  
      type: string  
    servizio_id:  
      type: string  
GdS:  
  type: object  
  required:  
    - email  
    - passwordHash
```

```
- servizio
properties:
  email:
    type: string
  passwordHash:
    type: string
  servizio:
    type: string
RichiestaGdS:
  type: object
  required:
    - email
    - passwordHash
    - titolo
    - azienda
    - foto
  properties:
    email:
      type: string
    passwordHash:
      type: string
    titolo:
      type: string
    azienda:
      type: string
    url:
      type: string
    foto:
      type: string
    descrizione:
      type: string
    confermata:
      type: boolean
Segnalazione:
  type: object
  required:
    - risposte
    - utente_id
    - servizio_id
  properties:
    risposte:
      type: array
      items:
        type: object
Commento:
  type: string
  utente_id:
    type: string
  servizio_id:
    type: string
```

```
Grafico:  
    type: object  
Servizio:  
    type: object  
    required:  
        - titolo  
        - azienda  
        - foto  
    properties:  
        titolo:  
            type: string  
        azienda:  
            type: string  
        url:  
            type: string  
        foto:  
            type: string  
        descrizione:  
            type: string  
    Stato:  
        type: string  
    enum:  
        - manutenzione  
        - problemi  
        - down  
        - on  
Form:  
    type: array  
    items:  
        type: object  
Sondaggio:  
    type: object  
    required:  
        - titolo  
        - corpo  
        - servizio_id  
    properties:  
        titolo:  
            type: string  
        corpo:  
            type: array  
            items:  
                type: object  
        servizio_id:  
            type: string  
Utente:  
    type: object  
    required:  
        - email  
        - passwordHash
```

```
properties:  
  email:  
    type: string  
  passwordHash:  
    type: string  
  adminCheck:  
    type: boolean  
  Preferiti:  
    type: array  
    items:  
      type: string
```

4.5 Testing

Il testing delle API è stato gestito con Insomnia. Si può trovare un file importabile nella repository all'url: [insomnia.json](#).

Tutte le API hanno già un corpo impostato, che è possibile testare, o modificare per testare casi limite. Alcune API richiedono un ordine di esecuzione, come ad esempio azioni che richiedono un login, che ha già associato degli script che impostano direttamente le variabili d'ambiente necessarie ad altre API. Qui sotto la tabella dei casi definiti e testati.

Come detto i casi sono tra loro legati dall'ordine di esecuzione. Per rendere queste dipendenze meno verbose, verrà solo scritto il numero test case necessario da eseguire prima di poter eseguire il test desiderato. Sarà ricorsivo, quindi se al test N.3 serve il test N.2 ed al test N.2 serve il test N.1, anche al test N.3 servirà il test N.1.

Tutti gli errori riguardanti i permessi per certe azioni ed i ruoli che le riguardano sono alla fine della tabella, nelle cui note si potrà trovare tutti i test in cui possono essere incontrati.

Tutti i casi con un URL dinamico, quindi dal N.13 al N.30, non dovrebbero fallire, se non nel caso in cui venga inserito l'URL manualmente o modificato dal lato dell'utilizzatore, oppure si tenti di accedere ad una risorsa che nel database non esiste più o non è mai esistita, cosa normalmente non avviene.

*Nelle note può apparire “utilizza il token”. In questo caso l'api funzionerà solo se il corretto utilizzatore è loggato, ed il JWT appropriato è presente. Se lo è, la motivazione per cui l'api fallisce non può essere dovuta all'autenticazione, a meno che non sia stato toccato il JWT. La seconda tabella alla fine mostra i risultati attesi per problemi di autenticazione e di ruoli. N.B. il token SuperAdmin è il passe-partout dei token.

**Nelle note può apparire “non può fallire”. Questo perché vengono considerate alcune azioni che non possono “teoricamente” fallire se non per un errore del server, e per questo non viene descritto alcun test case che ne riguardi un'esecuzione problematica.

***Tutti i test case dal N.13 al N.30 richiedono un corretto <Servizio.id> nell'URL. Si consideri sempre un <Servizio.id> valido ed esistente per tutte queste api. Nella tabella è scritto solo per i primi test, per le altre è sottinteso

N.	Desc. test case	Test data	Precondizioni	Dipende da	Risultato atteso	Note
1	Registrazione utente	<email> in formato corretto <password>	<email> non esistente nel sistema		Viene creato l'account specificato, il sistema risponde con: "Utente registrato con successo"	
1.1	Registrazione utente già esistente	rispettosa delle politiche di sicurezza	<email> già inserita nel sistema	N.1	L'account non viene creato, viene mostrato l'errore: "Email già registrata"	
2	Login utente	<email> in formato corretto <password>	coppia <email> <password> presente nel sistema	N.1	Viene creato un JWT con il ruolo "utente" nel client locale e viene mostrato il messaggio: "Login effettuato"	
2.1	Login dati errati	rispettosa delle politiche di sicurezza	<email> presente ma <password> sbagliata	N.1	L'utente non viene loggato, viene mostrato l'errore: "Password errata"	
2.2	Login utente inesistente		coppia <email> <password> non presente nel sistema		L'utente non viene loggato, viene mostrato l'errore: "Email non corretta"	
3	Ottenere dati utente	<i>Non necessario</i>		N.2	Vengono restituiti i dati dell'utente.	Utilizza il token* <i>Utente</i>
4	Ottenere preferiti utente	<i>Non necessario</i>		N.2	Vengono restituiti i preferiti dell'utente. Il contenuto può essere vuoto	Utilizza il token* <i>Utente</i>
5	Registrazione GdS	<email> e <password> in formato corretto + dati della <richiestaGdS> ben formattati	<email> non presente tra le richieste già presenti nel sistema		Viene inviata al server la richiesta di registrazione del GdS e mostrato il messaggio: "Richiesta inviata con successo"	

5.1	Registrazione GdS già richiesto	<email> e <password> in formato corretto + dati della <richiestaGdS> ben formattati	<email> presente tra le richieste già presenti nel sistema	N.5	La richiesta non viene spedita, viene mostrato il messaggio: "Richiesta già spedita"	
6	Login GdS		coppia <email> <password> presente nel sistema + <richiestaGdS> accettata	N.5, N.11	Viene creato un JWT con il ruolo "GdS" nel client locale e viene mostrato il messaggio: "Login effettuato"	
6.1	Login dati errati	<email> in formato corretto <password>	<email> presente ma <password> sbagliata	N.5, N.11	Il GdS non viene loggato, viene mostrato l'errore: "Password errata"	
6.2	Login richiesta non ancora approvata	rispettosa delle politiche di sicurezza	coppia <email> <password> presente nel sistema ma <richiestaGdS> non accettata	N.5	Il GdS non viene loggato, viene mostrato l'errore: "GdS inesistente"	
6.3	Login inesistente		coppia <email> <password> non presente nel sistema			
7	Ottieni servizio associato al GdS	<i>Non necessario</i>		N.6	Restituisce il servizio associato al GdS	Utilizza il token* GdS
8	Registrazione SuperAdmin		Il SuperAdmin viene solo creato manualmente, non ha un form di registrazione.			
9	Login SuperAdmin		flag <checkSuperAdmin> attivo	N.8	Viene creato un JWT con il ruolo "SuperAdmin" nel client locale e viene mostrato il messaggio: "Login effettuato"	
9.1	Login dati errati	<email> in formato corretto <password> rispettosa delle politiche di sicurezza e	<email> presente ma <password> sbagliata	N.8	Il SA non viene loggato, viene mostrato l'errore: "Password errata"	
9.2	Login inesistente		coppia <email> <password> non presente nel sistema	N.8	Il SA non viene loggato, viene mostrato l'errore: "Email non corretta"	

9.3	Login account non SuperAdmin	<email> in formato corretto <password> rispettosa delle politiche di sicurezza	flag <checkSuperAdmin> non attivo per l'<email> inserita	N.2, N.6	Il SA non viene loggato, viene mostrato l'errore: "You shouldn't be here..."	
10	Ottieni richieste GdS	<i>Non necessario</i>		N.9	Viene restituita una lista di RichiestaGdS non ancora confermate	Utilizza il token* SA
11	Conferma richiesta GdS		<RichiestaGdS.id> esistente con flag <accettata> a false	N.9, N.5	Viene creato nel sistema il GdS ed il Servizio a lui associato. Viene mostrato il messaggio: "GdS creato con successo"	Utilizza il token* SA
11.1	Conferma richiesta già confermata	<RichiestaGdS.id> valido e presente nel path dell'URL	<RichiestaGdS.id> esistente con flag <accettata> a true	N.9, N.5, N.11	Il GdS non viene creato e viene mostrato il messaggio: "Richiesta non trovata"	Utilizza il token* SA
11.2	Conferma richiesta inesistente		<RichiestaGdS.id> non esistente	N.9	"Richiesta non trovata"	Utilizza il token* SA
12	Ottieni lista servizi	<i>Non necessario</i>			Viene restituita una lista di servizi del sito (può essere vuota)	
13	Ottieni servizio		<Servizio.id> esistente	N.11	Restituisce i dati del sito associato all'id nel URL	
13.1	Ottieni servizio con id errato	<Servizio.id> valido presente nel path dell'URL***	<Servizio.id> non esistente		Il servizio non viene restituito e viene mostrato il messaggio: "Servizio inesistente"	Questo può accadere in TUTTE le api sottostanti (N.13, N.30)
14	Salva servizio nei preferiti	<Servizio.id> valido ed esistente nel path dell'URL***		N.11, N.2	In base alla presenza o no del servizio dei preferiti dell'utente accade: "Servizio salvato" oppure "Servizio eliminato"	Utilizza il token* Utente

15	Modifica servizio	<Servizio.id> valido ed esistente nel path dell'URL*** + modifiche valide da apportare al servizio		N.6	Il servizio viene aggiornato e viene mostrato il messaggio: "Servizio aggiornato con successo"	Utilizza il token* GdS
15.1	Modifica servizio non andato a buon fine	<Servizio.id> valido ed esistente nel path dell'URL*** + modifiche mal formattate da apportare al servizio	Immagine logo con estensione non supportata	N.6	Il servizio non viene aggiornato e viene mostrato il messaggio: "Errore nella modifica del servizio"	Utilizza il token* GdS
15.2			campi lasciati vuoti	N.6		
15.3			opzioni invalide	N.6		
16	Classifica servizi problematici	<i>Non necessario</i>			Viene restituita la classifica dei servizi ordinate per numero di segnalazioni	
17	Crea avviso	<Titolo> e <Corpo> dell'avviso ben formattati campo <Motivazione> selezionato		N.6	L'avviso viene salvato nel sistema, le email vengono inviate agli utenti con il servizio nei preferiti e viene mostrato il messaggio: "Avviso creato ed email inviata"	Utilizza il token* GdS
17.1	Avviso creato ma email non inviate		Server SMTP non funzionante	N.6	L'avviso viene salvato e viene mostrato il messaggio: "Errore durante l'invio delle email"	Utilizza il token* GdS
17.2	Avviso creato ma data di inizio incoerente		Data di inizio dell'avviso impostata nel passato	N.6	L'avviso non viene salvato e viene mostrato il messaggio: "La data di inizio non deve essere nel passato"	Utilizza il token* GdS
17.3	Avviso creato ma data di fine incoerente		Data di fine dell'avviso impostata prima della data d'inizio	N.6	L'avviso non viene salvato e viene mostrato il messaggio: "La data di fine deve essere successiva alla data di inizio."	Utilizza il token* GdS

17.4	Crea avviso non andato a buon fine	<Titolo> e <Corpo> dell'avviso mal formattati o vuoti o campo <Motivazione> non selezionato		N.6	L'avviso non viene salvato e viene mostrato il messaggio: "Errore nella creazione dell'avviso"	Utilizza il token* GdS
18	Ottieni avvisi	<i>Non necessario</i>			Viene restituita una lista di avvisi del servizio (può essere vuota)	
19	Ottieni avviso		<Avviso.id> esistente	N.17	Viene restituito l'avviso con l'id associato	
19.1	Ottieni avviso del servizio sbagliato	<Avviso.id> valido presente nel path dell'URL	<Avviso.id> esistente ma <Servizio.id> errato	N.17	L'avviso non viene restituito, viene mostrato il messaggio: "L'avviso non è del servizio corretto"	
19.2	Ottieni avviso con id inesistente		<Avviso.id> non esistente nel sistema		L'avviso non viene restituito, viene mostrato il messaggio: "Errore nel recupero dei dati"	
20	Crea form			N.6	La form del servizio viene creata nel sistema e viene mostrato il messaggio: "Form creata con successo"	Utilizza il token* GdS
20.1	Crea form con dati vuoti	campi <Opzione> della form ben formattati e validi	campi <Opzione> non presenti oppure lasciati vuoti	N.6	La form del servizio non viene creata e viene mostrato il messaggio: "Opzioni non presenti; impossibile creare la form"	Utilizza il token* GdS
21	Ottieni form	<i>Non necessario</i>	<Servizio>.<form> compilata	N.20	Viene restituita la form associata al servizio	Utilizza il token* Utente Utilizza il token* GdS

21.1	Ottieni form inesistente	<i>Non necessario</i>	<Servizio>.<form> non ancora compilata		La form non viene restituita, viene mostrato il messaggio: "Form non trovata"	Utilizza il token* <i>Utente</i> Utilizza il token* <i>GdS</i>
22	Compila segnalazione		<Servizio>.<form> esistente ed ottenuta correttamente	N.21 e N.2	La segnalazione viene creata nel sistema e viene mostrato il messaggio: "Segnalazione creata con successo"	Utilizza il token* <i>Utente</i>
22.1	Compila segnalazione oltre il limite giornaliero	campi <Risposta> della segnalazione ben formattati e validi + commento (opzionale)	<Segnalazione> già compilata dall'utente per quel servizio nell'ultimo giorno	N.22	La segnalazione non viene creata e viene mostrato il messaggio: "Hai già compilato una segnalazione oggi."	Utilizza il token* <i>Utente</i>
22.2	Compila segnalazione con dati vuoti		nessun campo <Risposta> selezionato, oppure lasciato vuoto		La segnalazione non viene creata e viene mostrato il messaggio: "La form non è stata compilata"	Utilizza il token* <i>Utente</i>
23	Ottieni commenti	<i>Non necessario</i>			Viene restituita una lista di commenti associati a tutte le segnalazioni fatte al servizio (può essere vuota)	
24	Ottieni istogramma	<i>Non necessario</i>			Vengono restituiti i dati necessari per creare il grafico associato (può essere vuoto)	
25	Ottieni aerogramma	<i>Non necessario</i>			Vengono restituiti i dati necessari per creare il grafico associato (può essere vuoto)	
26	Crea sondaggio	campi <Opzione> della form dei tipi supportati, ben formattati e validi		N.6	Il sondaggio viene creato nel sistema e viene mostrato il messaggio: "Sondaggio creato con successo"	Utilizza il token* <i>GdS</i>

26.1	Crea sondaggio con dati vuoti	campi <Opzione> della form dei tipi supportati, ben formattati e validi	campi <Opzione> non presenti oppure lasciati vuoti	N.6	Il sondaggio non viene creato e viene mostrato il messaggio: "La form non è stata compilata"	Utilizza il token* GdS
26.2	Crea sondaggio con dati mal formattati	campi <Opzione> della form dei tipi non supportati, mal formattati o non validi		N.6	Il sondaggio non viene creato e viene mostrato il messaggio: "Tipo di opzione non valido"	Utilizza il token* GdS
27	Ottieni sondaggi	<i>Non necessario</i>			Viene restituita una lista di sondaggi associata al servizio	Utilizza il token* Utente Utilizza il token* GdS
28	Ottieni sondaggio		<Sondaggio.id> esistente	N.26	Viene restituito il sondaggio associato all'id nell'URL	Utilizza il token* Utente Utilizza il token* GdS
28.2	Ottieni sondaggio del servizio sbagliato	<Sondaggio.id> valido presente nel path dell'URL	<Sondaggio.id> non esistente nel sistema	N.26	Il sondaggio non viene restituito e viene mostrato il messaggio: "Il sondaggio non è del servizio corretto"	Utilizza il token* Utente Utilizza il token* GdS
28.1	Ottieni sondaggio inesistente		<Sondaggio.id> esistente ma <Servizio.id> errato		Il sondaggio non viene restituito e viene mostrato il messaggio: "Sondaggio non trovato"	Utilizza il token* Utente Utilizza il token* GdS
29	Compila sondaggio		<Sondaggio> esistente ed ottenuto correttamente	N.28, N.2	Il feedback viene creato nel sistema e viene mostrato il messaggio: "Feedback creato con successo"	Utilizza il token* Utente
29.1	Compila sondaggio con dati vuoti	campi <Risposta> della segnalazione dei tipi supportati, ben formattati e validi	nessun campo <Risposta> selezionato, oppure lasciato vuoto	N.28, N.2	Il sondaggio non viene creato e viene mostrato il messaggio: "Il sondaggio non è stato compilato"	Utilizza il token* Utente

29.2	Compila un sondaggio già compilato		<Sondaggio> già compilato dall'utente per quel servizio	N.29	Il sondaggio non viene creato e viene mostrato il messaggio: "Hai già compilato questo sondaggio"	Utilizza il token* Utente
29.3	Compila il sondaggio con dati mal formattati	campi <Risposta> della segnalazione dei tipi non supportati, mal formattati o non validi		N.28, N.2	Il sondaggio non viene creato e viene mostrato il messaggio: "Tipo di opzione non valido"	Utilizza il token* Utente
30	Ottieni risultati	<Sondaggio.id> valido presente nel path dell'URL			Viene restituita una lista di feedback rilasciati per il sondaggio associato all'id	

Nota, il caso 17.1 non riguarda alcun problema sui dati, bensì un problema server. E' presente nel test case poiché nel deployment apparirà alla creazione dell'avviso, poichè non abbiamo trovato il modo di abilitare il protocollo 'sendmail'

T1.1	Attore non loggato		<JWT> non presente nel sistema locale dell'attore	N.1 o N.5	All'attore non viene garantito l'accesso alla risorsa e viene presentato il messaggio:	Attore: Utente, GdS, SuperAdmin
T1.2	Attore con login obsoleto	<JWT> in formato valido	<JWT> scaduto	N.2 o N.6 o N.9	All'attore non viene garantito l'accesso alla risorsa e viene presentato il messaggio: "Token non valido"	
T3	Accesso ad un servizio senza i permessi giusti	<Attore> loggato correttamente	<JWT> non contenente il ruolo corretto per eseguire l'azione	N.2 o N.6	All'attore non viene garantito l'accesso alla risorsa e viene presentato il messaggio: "Accesso negato. Non hai i permessi per questa azione"	Utilizza il token* Utente Utilizza il token* GdS
T4	Accesso GdS ad un servizio che non possiede	<GdS> esistente, approvato, loggato correttamente <Servizio.id> valido ed esistente nell'URL	<GdS>.<Servizio> diverso dal <Servizio> presente nell' URL	N.6	All'attore non viene garantito l'accesso alla risorsa e viene presentato il messaggio: "Accesso negato. Non gestisci questo servizio"	Utilizza il token* GdS

5. Front-End

In questa sezione viene riportata l'intero front-end dell'applicazione, diviso in base ai 3 ruoli definiti in Trentus: Utente, GdS e SuperAdmin. Prima di riportare l'intero front-end dell'applicazione, descriverò le componenti grafiche presenti nella maggior parte delle schermate, a prescindere dal tipo di utente che le usa (nonostante ci siano delle piccole differenze in base al ruolo):

- **Navbar superiore:** La Navbar superiore, presente in tutte le schermate dell'applicazione (tranne le schermate di login e registrazione), permette all'utente di tornare alla home, cliccando sul logo di Trentus; essa permette inoltre di filtrare i servizi e di effettuare il login o il logout.
- **Sidebar di Sinistra:** La Sidebar di sinistra permette di visionare le notifiche e gli avvisi arrivati all'utilizzatore dell'applicazione (il tipo di notifica differisce in base al ruolo)
- **Sidebar di Destra:** La Sidebar di destra permette di visionare, indipendentemente dal ruolo, la top 5 dei servizi più segnalati sull'applicazione nelle ultime 24 ore.

5.1 Front-End per l'Utente

Per l'utente, il front-end fornisce le seguenti pagine:

- HomePage
- Pagina del servizio
- Pagina di segnalazione del servizio
- Pagina di compilazione sondaggio
- Pagina di visualizzazione avviso
- HomePage per utenti non loggati
- Pagina di login
- Pagina di registrazione

Nell'Immagine 1 troviamo l'Homepage dell'Utente loggato. In questa schermata troviamo una lista di servizi, rappresentati da carte. Queste, riportano il nome, il logo e lo stato del servizio. Sulle stesse, è presente un tasto "Preferiti" a forma di stella che, se cliccato, inserisce il servizio fra i servizi preferiti dell'utente, permettendogli di ricevere notifiche e sondaggi dal Gestore del Servizio e di filtrare la lista dei servizi.

Nell'Immagine 1a possiamo vedere come l'Utente loggato possa effettuare il logout, cliccando sull'icona delle Impostazioni.

The screenshot shows the Trentus user homepage. On the left, there's a sidebar with sections like Notifiche (Coven, UniInApp, Problemi sito web, Opera4U, Manutenzione sezione mensa), Sondaggi (In arrivo, Sondaggi sulla qualità dei nuovi distributori), and a search bar. The main area displays three service cards:

- TreC+**: Logo Tred, status "manutenzione" with a yellow dot, and a yellow star icon.
- Muoversi in Trentino**: Logo Muoversi, status "down" with a red dot, and a grey star icon.
- Opera4U**: Logo Opera4U, status "on" with a green dot, and a yellow star icon.

To the right, there's a sidebar titled "Classifica" containing five items with logos and signal counts:

- Poste Italiane (4 segnalazioni)
- TreC (3 segnalazioni)
- Muoversi (3 segnalazioni)
- Opera4U (2 segnalazioni)
- Poste Italiane (2 segnalazioni)

Immagine 1: Homepage dell'Utente

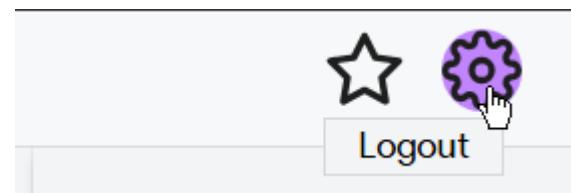


Immagine 1a: Click sul tasto impostazioni per l'utente

loggato

L'Immagine 2 ci mostra come i servizi vengono filtrati: cliccando l'icona della stella sulla Navbar (esclusiva degli Utenti loggati), infatti, ci verranno mostrati solo i servizi che sono stati impostati come "Preferiti" dall'Utente. Dalle immagini viste finora, possiamo notare che la Sidebar di sinistra dell'Utente loggato mostri tutte le notifiche ed i sondaggi mandati dai Gestori dei servizi preferiti.

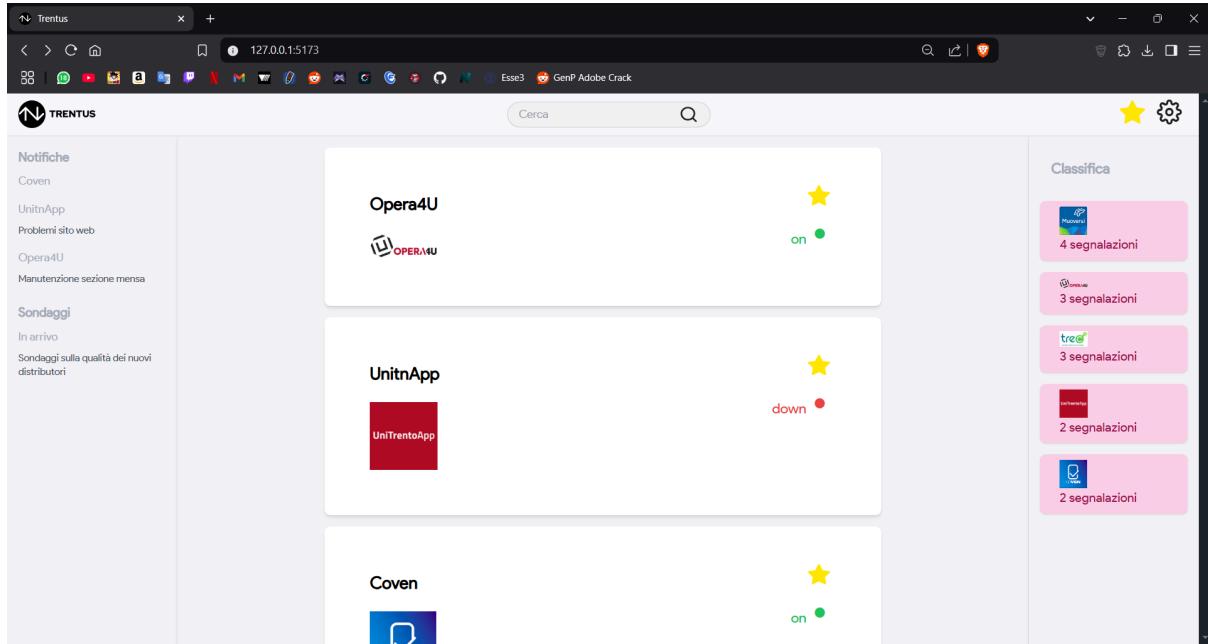


Immagine 2: Homepage dell'Utente con il filtro preferiti

Le Immagine 3 e 3a ci mostrano la pagina dei dettagli di un servizio, accessibile cliccando su una delle carte dei servizi della Homepage. Questa pagina riepiloga tutte le informazioni del servizio e riporta due grafici:

- **Istogramma:** Grafico che riporta il numero di segnalazioni ricevute dal servizio negli ultimi 10 giorni
- **Areogramma:** Grafico che riporta il motivo delle varie segnalazioni ricevute negli ultimi 2 giorni

Oltre ai grafici, su questa pagina è possibile visionare anche i commenti ricevuti da altri utenti che hanno già segnalato il servizio.

The screenshot shows a web browser window with the URL 127.0.0.1:5173/service/67a9e854d42bce60ca81b3e7. The page title is 'Trentus'. On the left, there's a sidebar with links like Notifiche, Coven, UnitnApp, Problemi sito web, Opera4U, Manutenzione sezione mensa, Sondaggi, In arrivo, and Sondaggi sulla qualità dei nuovi distributori. The main content area displays the service 'Muoversi in Trentino' by 'Trentino Trasporti'. It includes a logo, a brief description ('App per monitorare i trasporti'), a 'Segnala' button, and two charts. The first chart, 'Segnalazioni negli ultimi 10 giorni', is a bar chart showing signal counts from week 2-3 to 2-12. The second chart, 'Motivo delle segnalazioni', is a donut chart. A sidebar on the right, titled 'Classifica', lists other services with their signal counts: Muoversi (4), tree (3), tre (3), entro (2), and bus (2). Below the charts, there are several user comments in a scrollable box, all reporting issues with bus tracking.

Immagine 3: Pagina del servizio (1)

This screenshot is identical to the one above, showing the same service detail page for 'Muoversi in Trentino'. The layout, charts, and user comments are the same, demonstrating a consistent experience across different sessions.

Immagine 3a: Pagina del servizio (2)

Cliccando sul tasto “Segnala” accediamo alla schermata rappresentata dall’Immagine 4: Questa pagina ci permette di effettuare una segnalazione, in caso di malfunzionamento del servizio. Tale pagina mostra una check list, appositamente creata dal GdS per permettere agli utenti che segnalano di specificare cosa non funziona del servizio. Inoltre, in maniera del tutto opzionale, è possibile lasciare un commento per specificare ancora meglio quale sia stato il problema riscontrato. Le Immagini 4a e 4b mostrano rispettivamente il messaggio di errore ed il messaggio di conferma nell’invio della segnalazione. Il messaggio di errore viene evocato se la segnalazione viene lasciata vuota.

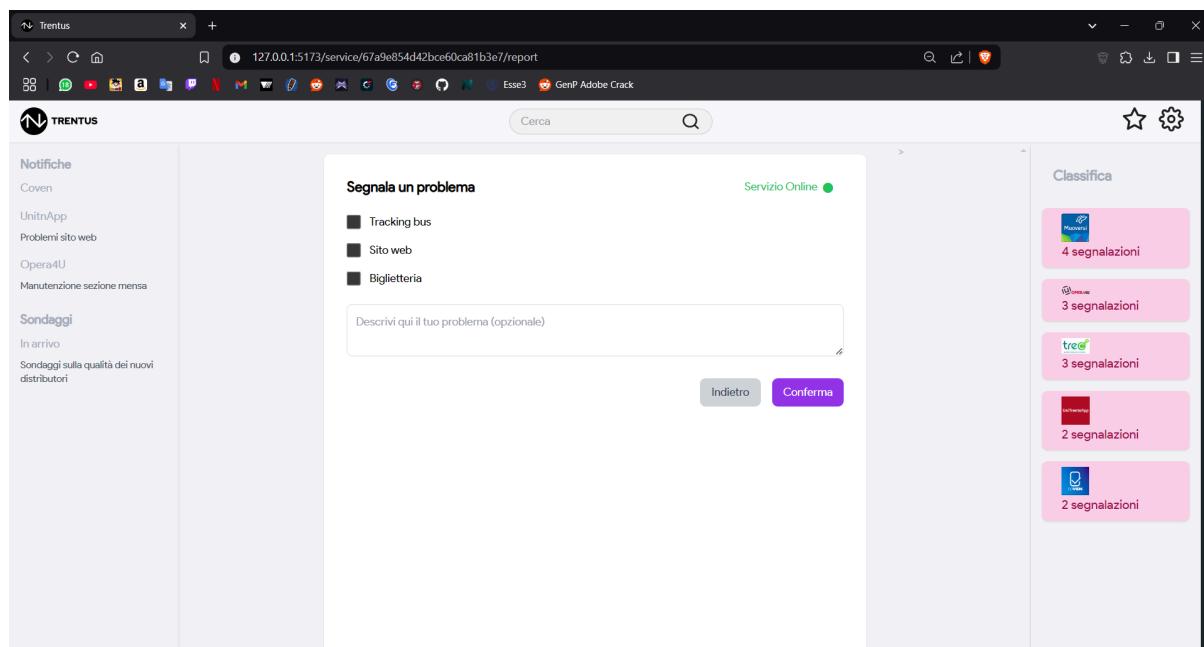


Immagine 4: Pagina di segnalazione del servizio

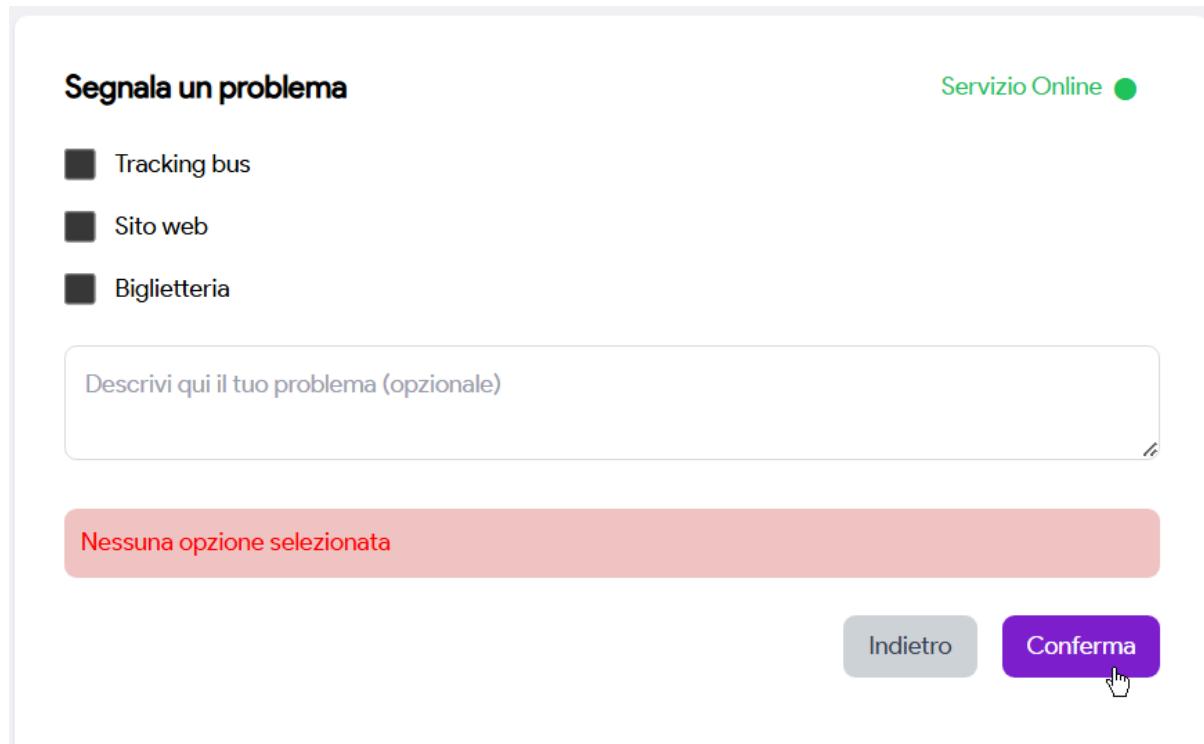


Immagine 4a: Errore nella segnalazione

Segnala un problema Servizio Online

Tracking bus
 Sito web
 Biglietteria

Non riesco ad aprire il sito!

Segnalazione mandata con successo

[Indietro](#) Conferma

Immagine 4b: Segnalazione eseguita correttamente

L'Immagine 5 mostra la schermata di compilazione sondaggio. Cliccando sulla notifica di nuovo sondaggio nella Sidebar di sinistra è infatti possibile accedere a questa schermata e iniziare a compilare i sondaggi inviati dai Gestori dei nostri servizi preferiti. Tali sondaggi possono essere formati da checkboxes, domande a risposta multipla o domande a risposta aperta. Le Immagini 5a e 5b mostrano i pop up che vengono evocati in caso di errore nella compilazione del sondaggio o in caso di conferma della sua conferma. L'errore viene evocato se il sondaggio viene lasciato vuoto o se era già stato compilato in precedenza.

The screenshot shows a survey titled "Sondaggi sulla qualità dei nuovi distributori" by "Coven". The survey includes a question about downloading the app ("Ho scaricato l'app Coven") with a checked checkbox. Another question asks what was tried ("Ho provato...") with three options: "Macchinette per le merendine" (unchecked), "Macchinette per il caffè" (checked), and "Distributore di acqua" (unchecked). A text input field asks for an opinion ("Cosa ne pensi delle nuove macchinette?") containing the text "Risposta". Below the input field are two buttons: "Indietro" and "Compila". On the right side, there is a sidebar titled "Classifica" showing five entries with logos and signal counts: "Postnord" (4), "Operai4U" (3), "tree" (3), "Poste Italiane" (2), and "Poste" (2). The URL at the bottom is 127.0.0.1:5173/service/67a9f207143861bd9dca4944/survey/67aa16af264fc204153424c.

Immagine 5: Pagina di compilazione sondaggio

This screenshot shows the same survey as above, but with a modal dialog box in the center stating "localhost:5173 dice" and "Sondaggio compilato con successo" with an "OK" button. The survey responses remain the same: the "Caffè" option is checked in the "Ho provato..." section, and the text input field contains "Il caffè non è proprio buonissimo". The sidebar "Classifica" shows updated signal counts: "Postnord" (5), "Operai4U" (3), "tree" (3), "Poste Italiane" (2), and "Poste" (2). The URL at the bottom is 127.0.0.1:5173/service/67a9f207143861bd9dca4944/survey/67aa16af264fc204153424c.

Immagine 5a: Sondaggio compilato correttamente

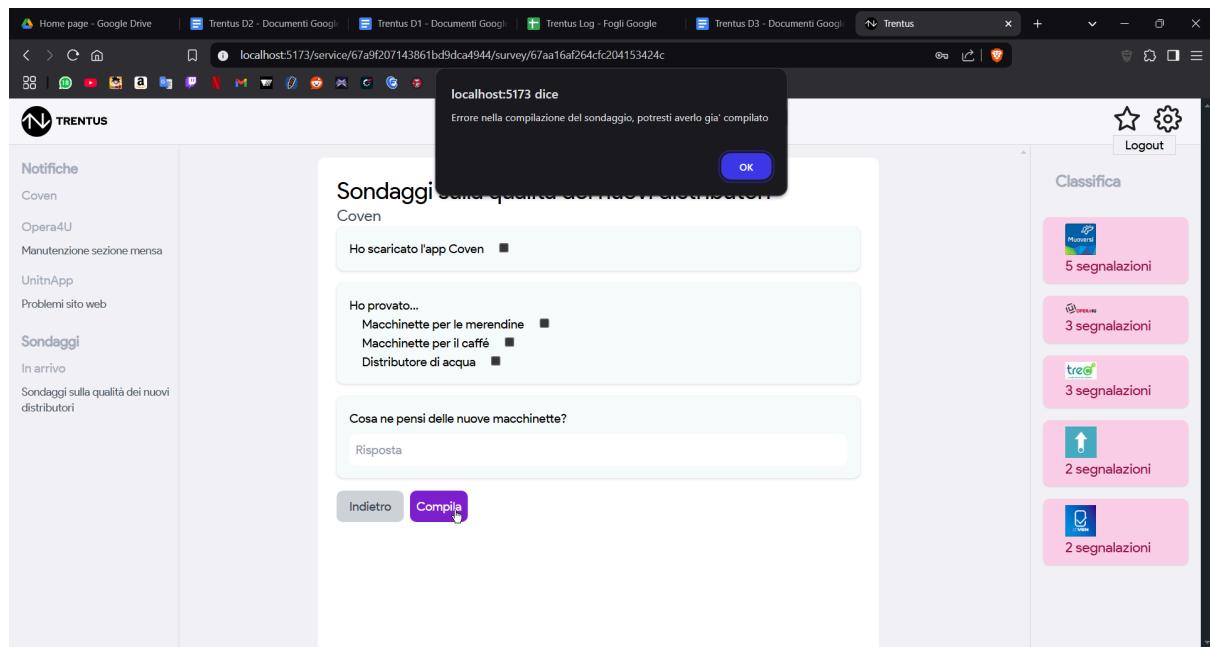


Immagine 5b: Sondaggio non compilato o già compilato

L'Immagine 6 riporta la schermata di visualizzazione notifica, accessibile cliccando sulla notifica presente sulla Sidebar di sinistra. Questa notifica è formata da un titolo, un messaggio, una motivazione e un lasso di tempo. La motivazione può essere di tre tipi:

- **Manutenzione:** Il servizio sarà inagibile per un periodo di tempo limitato per un processo di manutenzione
- **Problemi:** Una particolare funzione del servizio potrebbe essere inagibile o funzionare in maniera errata a causa di problemi nel sistema
- **Down:** La totalità del servizio è inagibile a causa di motivi ancora ignoti al team del servizio

The screenshot shows a web interface for monitoring service status. On the left sidebar, under 'Notifiche', 'Problemi sito web' is selected. The main content area displays a notice titled 'Problemi sito web' with the message: 'Motivazione: problema'. It states: 'Stiamo riscontrando delle problematiche al nuovo sito web, pertanto questo non sarà accessibile per i prossimi 3 giorni'. Below this, it says 'Periodo: 2025-02-10T00:00:00.000Z-2025-02-13T00:00:00.000Z'. A 'Indietro' button is at the bottom left. On the right, there's a 'Classifica' sidebar with five items, each with a logo and 'segnalazioni':

- Posteitaliane: 4 segnalazioni
- tree: 3 segnalazioni
- unica: 3 segnalazioni
- telecom: 2 segnalazioni
- tele2: 2 segnalazioni

Immagine 6: Pagina di visualizzazione notifica

L'Immagine 7 riporta la Homepage dell'Utente che non ha ancora effettuato il login. In caso di primo accesso a Trentus, questa è la prima schermata che viene mostrata all'utente. In questa speciale versione della Homepage dell'utente notiamo diverse differenze:

- La Sidebar left è vuota, in quanto, non avendo un account, non è possibile ricevere notifiche o sondaggi.
- Nella Navbar e sulle carte dei servizi scompare l'icona dei preferiti, in quanto, non avendo un account, non è possibile salvare un servizio fra i preferiti
- Nella Navbar, cliccando sul tasto delle impostazioni, apparirà l'opzione per effettuare il login (Immagine 7a)

L'Immagine 7b riporta invece la schermata dei dettagli di un servizio in caso di utente non loggato. In questo caso, scompare il tasto "Segnala", in quanto senza account non è possibile segnalare un servizio.

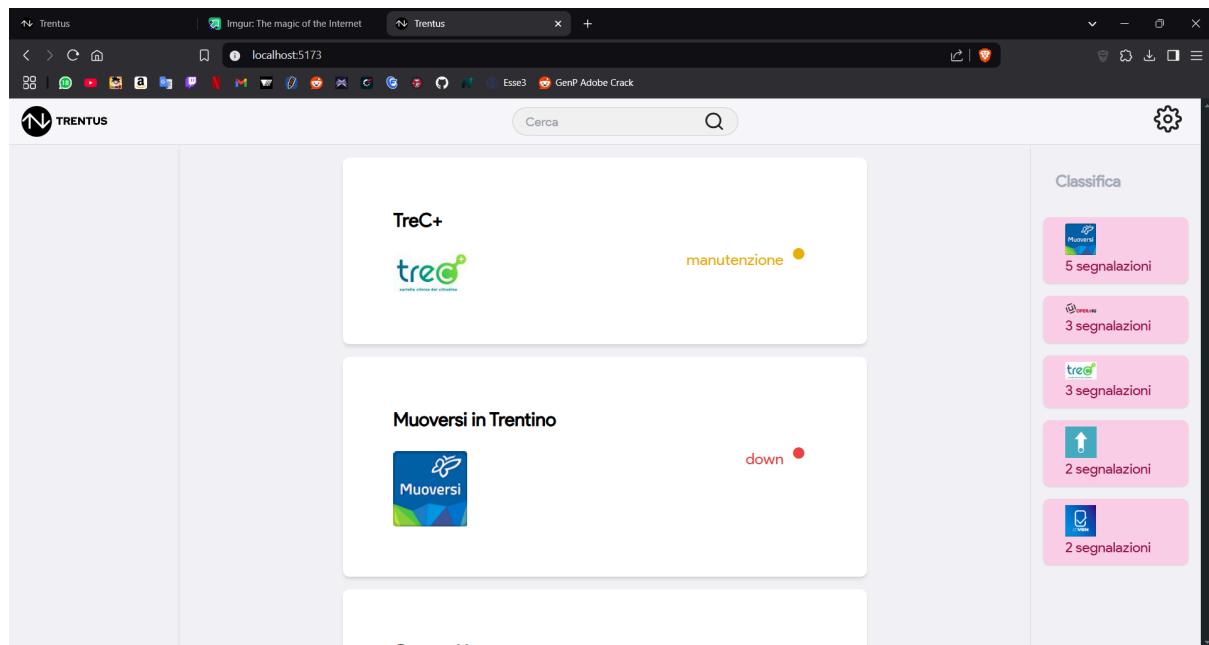


Immagine 7: Homepage dell'utente non loggato

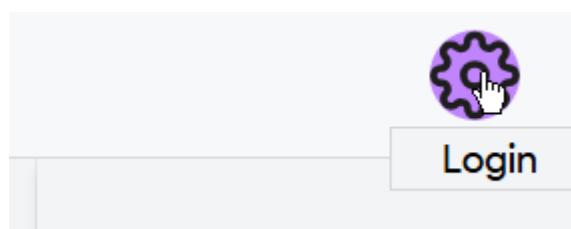


Immagine 7a: Click sul tasto impostazioni per l'utente non loggato

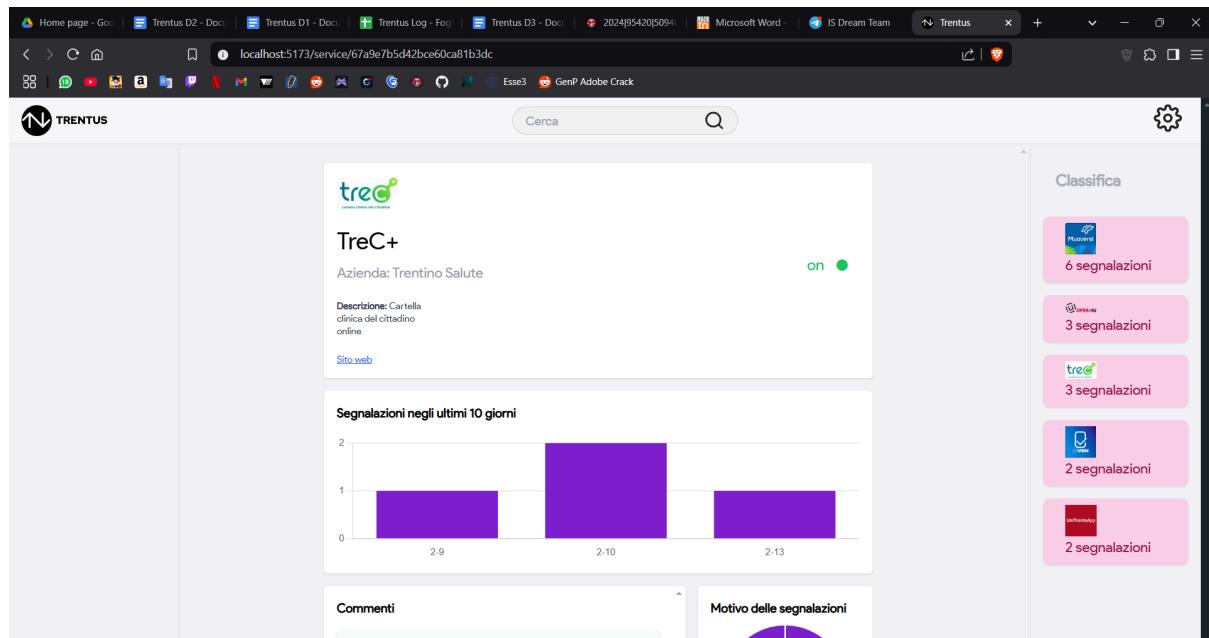


Immagine 7b: Pagina dei dettagli di un servizio per l'utente non loggato

L'Immagine 8 riporta la schermata di login per l'Utente. Le credenziali necessarie al login sono una email e una password registrate nel database. In caso di account mancante, è possibile andare alla pagina di registrazione, riportata nell'Immagine 9, attraverso il link "Registrati", sotto il form di login. Oltre al link per registrarsi, è presente anche un link per accedere alla pagina di login per i GdS, di cui parleremo più tardi.

L'immagine 8a ci mostra il messaggio di errore che viene evocato in caso di credenziali non valide o vuote.

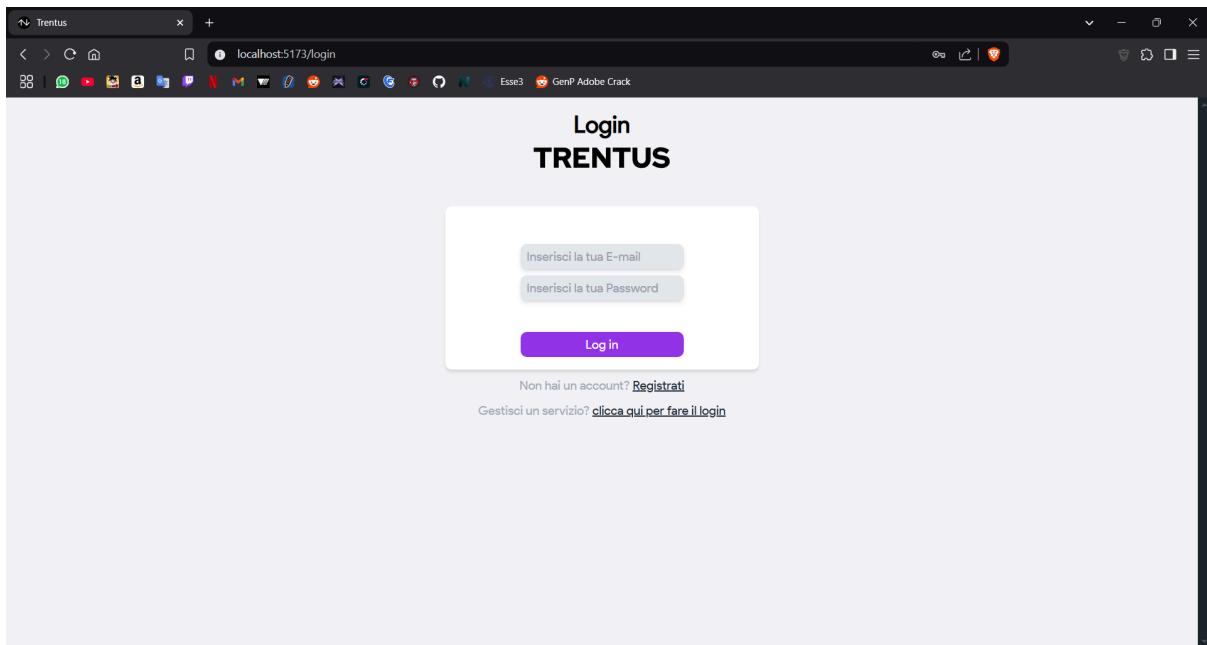


Immagine 8: Pagina di login per l'utente

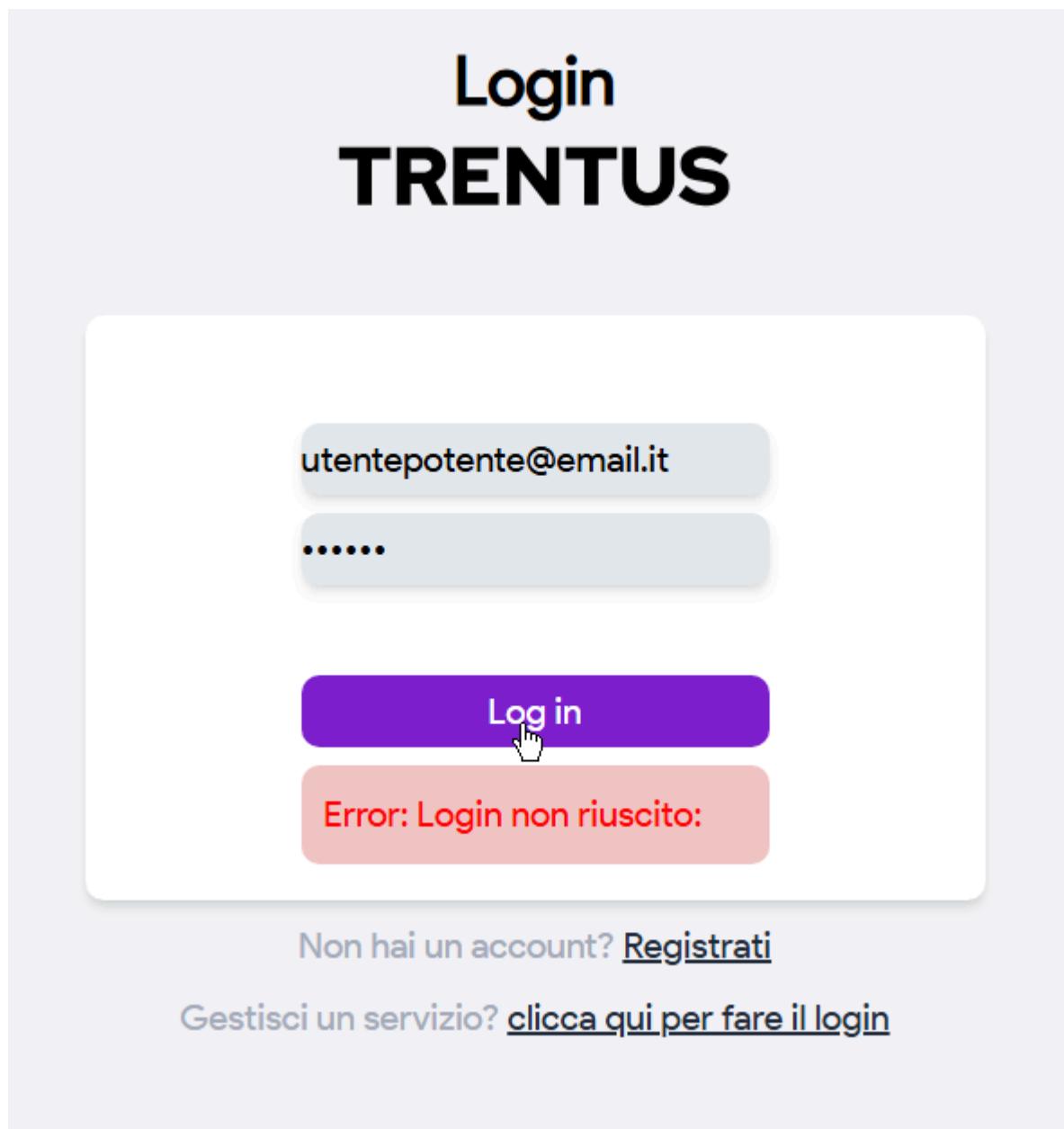


Immagine 8a: Login con credenziali errate

L'Immagine 9, come già detto poco prima, riporta la schermata di registrazione per l'Utente. Per registrarsi è richiesto di inserire una email ed una password, inserendola 2 volte per verificarne la corretta scrittura. L'Immagine 9a ci mostra il messaggio di errore che viene evocato in caso di password diverse, di credenziali vuote o già registrate.

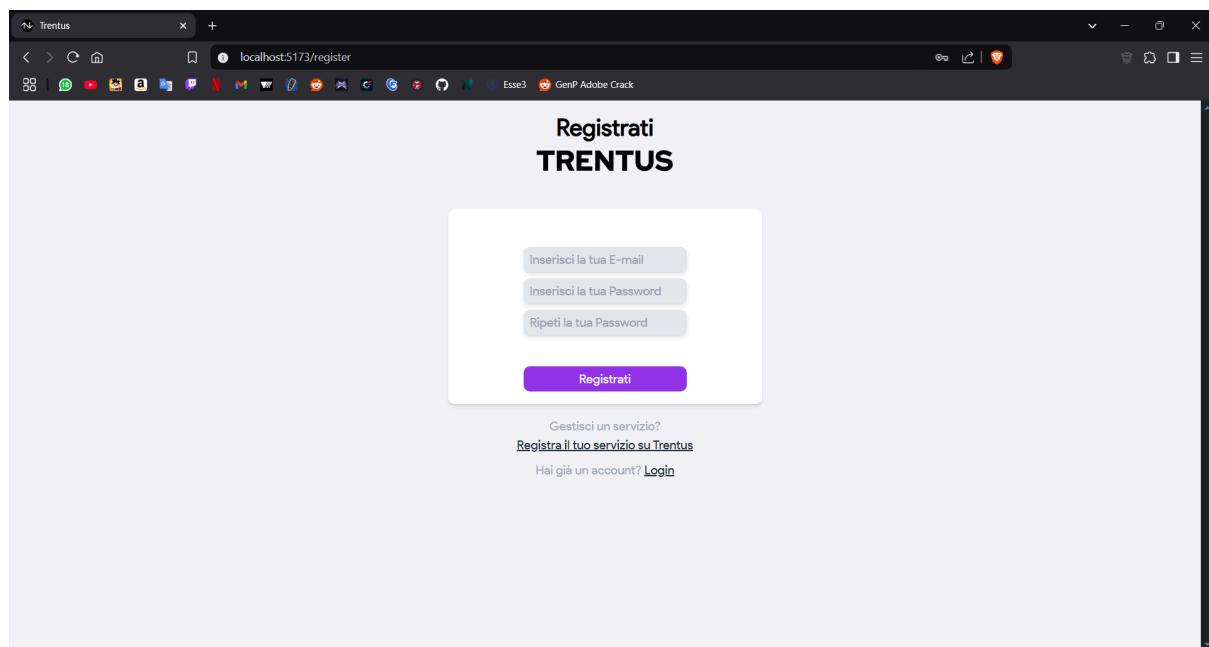


Immagine 9: Pagina di registrazione per l'utente

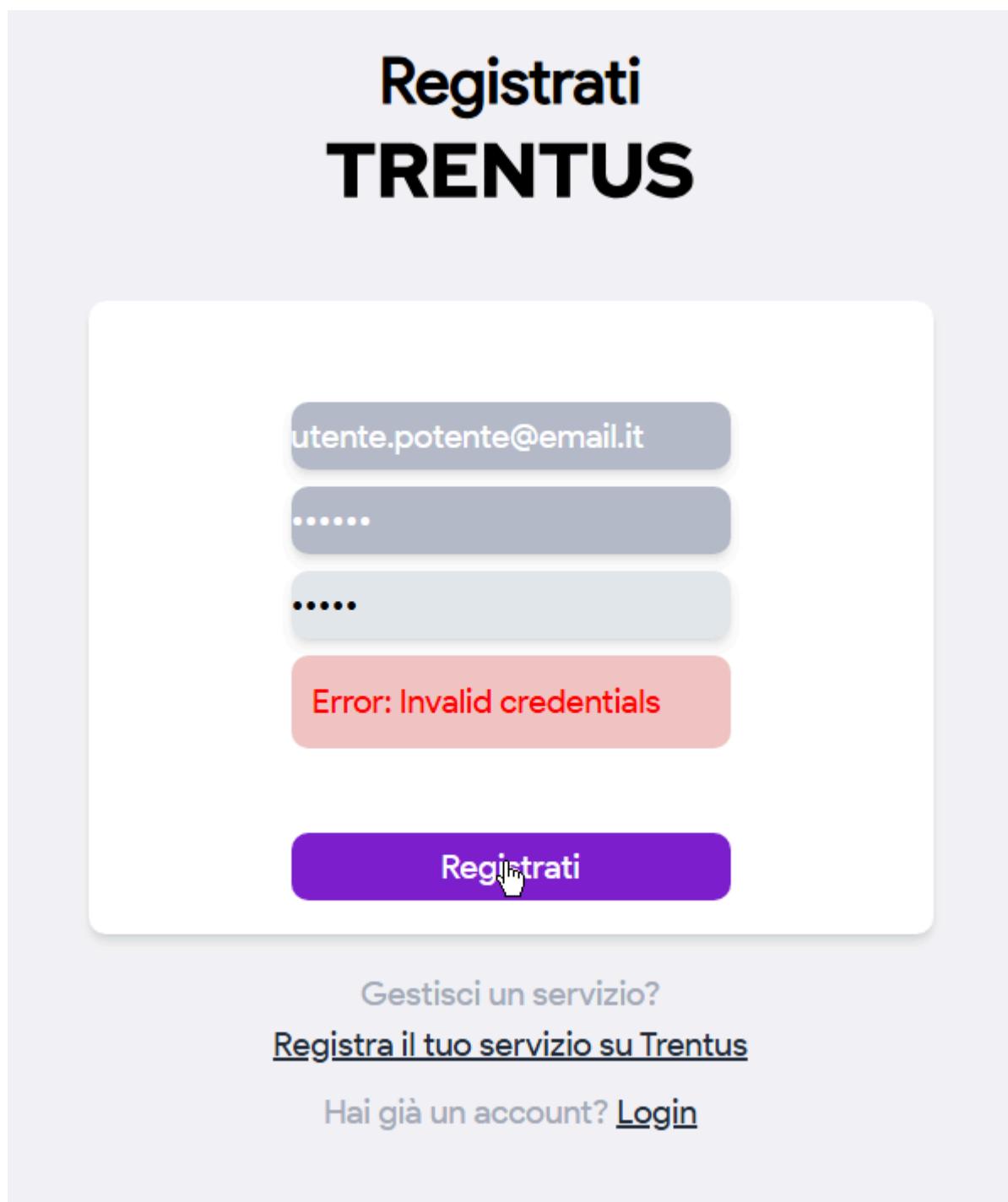


Immagine 9a: Registrazione con credenziali già registrate o nulle

5.2 Front-End per il GdS

Per il GdS, il front-end fornisce le seguenti pagine:

- Pagina di Login
- Pagina di registrazione
- Homepage GdS
- Pagina di modifica dei dettagli del servizio
- Pagina di creazione avviso (notifica)
- Pagina di creazione sondaggio
- Pagina di visualizzazione dei risultati del sondaggio

Nell'Immagine 10 troviamo la schermata di Login per il GdS. Come per l'utente, il GdS ha bisogno solo di una email e di una password per accedere. Al di sotto del form di login, sono presenti due link: un link per registrare un nuovo servizio (in caso non sia già stato fatto) e un link per accedere alla schermata di login per l'utente. L'Immagine 10a ci mostra il messaggio di errore che viene evocato in caso di login con credenziali errate o vuote.

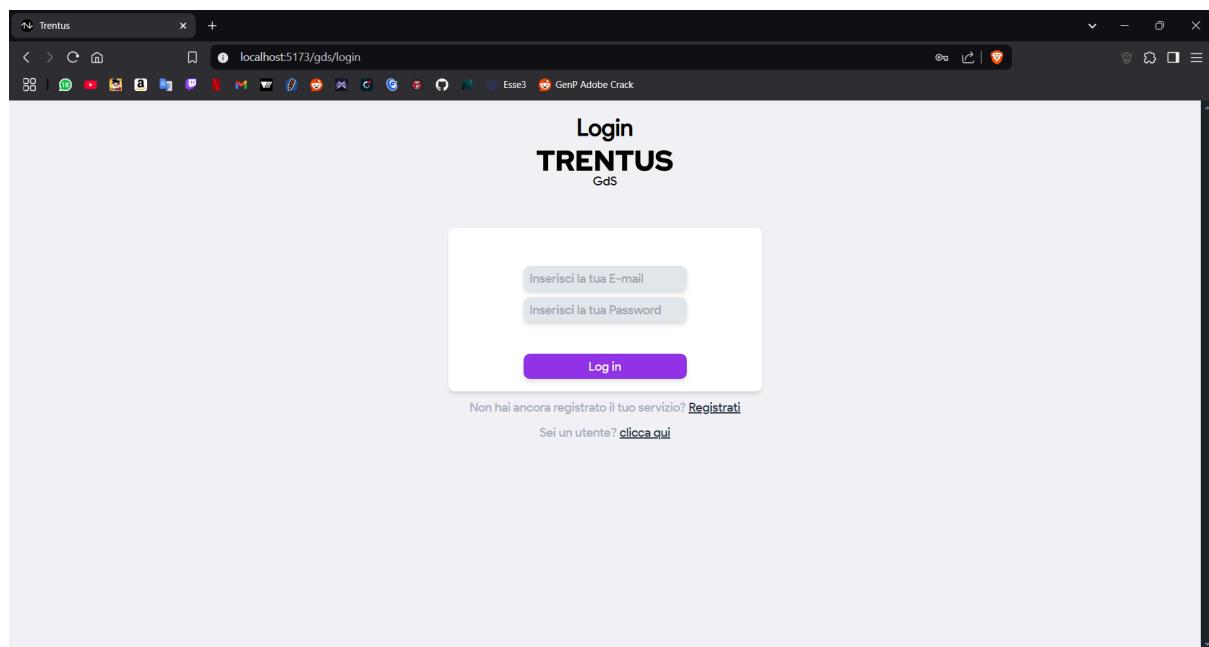


Immagine 10: Pagina di login per il GdS

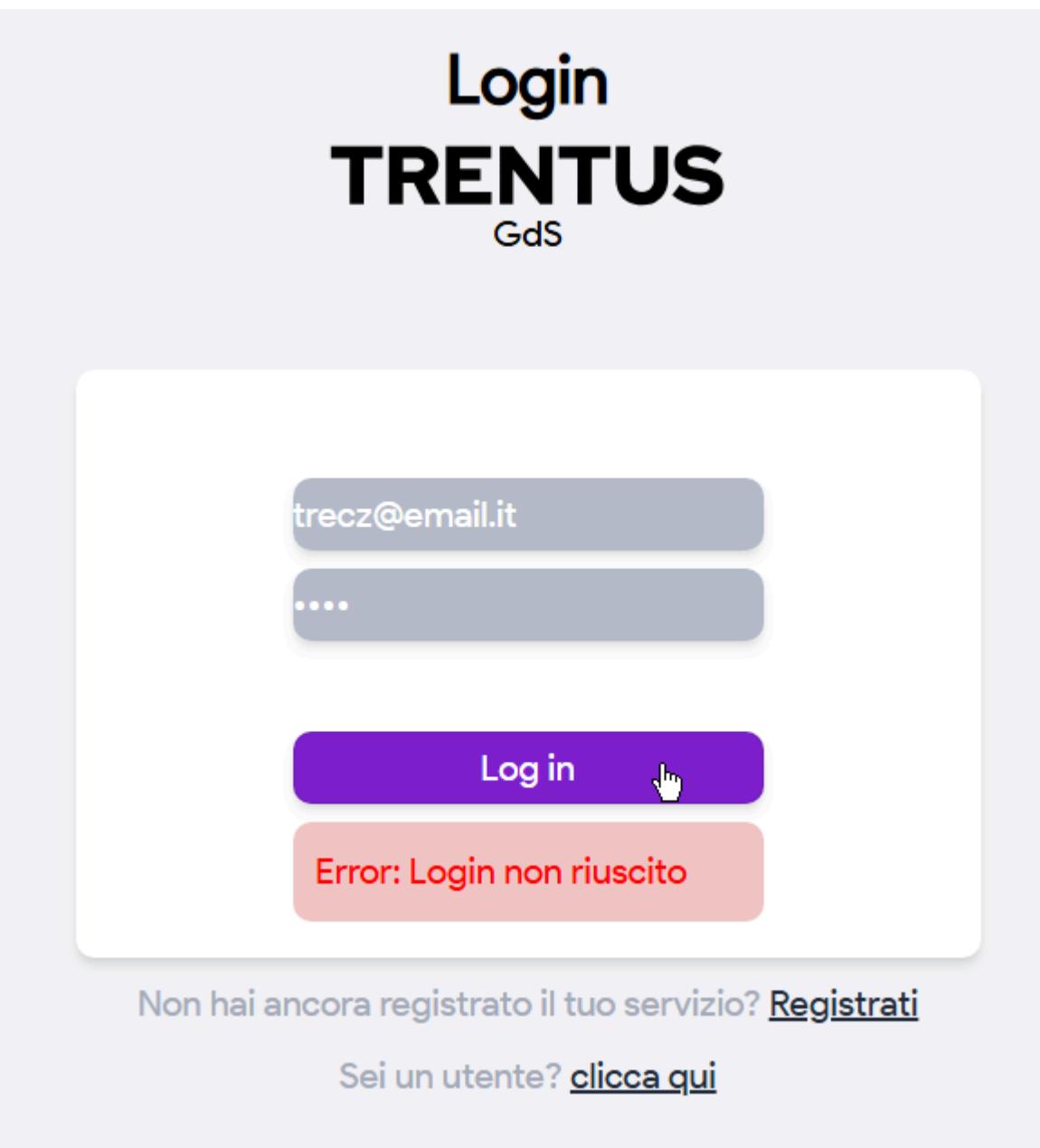


Immagine 10a: Login con credenziali errate o nulle

L'Immagine 11 raffigura la pagina di registrazione per il GdS. Al contrario dell'utente, il GdS ha altri parametri da compilare oltre email e password: bisogna infatti fornire il nome del servizio, il nome dell'azienda fornitrice del servizio, il link al sito web del servizio, una descrizione ed il logo del servizio. Una volta cliccato il tasto “Aggiungi”, la richiesta di registrazione servizio verrà inoltrata ai SuperAdmin, che avranno il compito di accettare il servizio o meno. In fondo al form di registrazione, è presente un link alla pagina di login per il GdS. L'Immagine 11a raffigura il pop-up di errore che viene evocato in caso di registrazione di un servizio già registrato o in caso di registrazione vuota. L'Immagine 11b raffigura invece il pop-up di errore che viene evocato in caso di logo non caricato.

Register to **TRENTUS**

Email

Password

Nome

Azienda

Sito web

Descrizione

Logo

Nessun file selezionato

Hai già registrato il tuo servizio? [Fai il Login](#)

Immagine 11: Pagina di registrazione servizio

localhost:5173 dice

Impossibile effettuare la registrazione.

OK

Email
trec@email.it

Password

Nome
trec

Azienda
trec

Sito web
trec.com

Descrizione
trec
~~~~~

Logo  
Scegli file italia[1].jpg

**Aggiungi**

Hai già registrato il tuo servizio? [Fai il Login](#)

The screenshot shows a registration form on a localhost:5173 page. At the top, a message box displays the error: 'localhost:5173 dice Impossibile effettuare la registrazione.' (localhost:5173 says It is impossible to perform the registration.) with an 'OK' button. Below this, the registration fields are filled with placeholder text: Email (trec@email.it), Password (\*\*\*\*), Nome (trec), Azienda (trec), Sito web (trec.com), Descrizione (trec  
~~~~~), and Logo (italia[1].jpg). A purple 'Aggiungi' button is at the bottom. A note at the bottom indicates that if registered, one can log in via a link.

Immagine 11a: Registrazione con credenziali già registrate

localhost:5173 dice

Seleziona un file prima di inviare.

OK

Email
trec@email.it

Password

Nome
trec

Azienda
trec

Sito web
trec.com

Descrizione
trec

Logo
Scegli file Nessun file selezionato

Aggiungi

Hai già registrato il tuo servizio? [Fai il Login](#)

Fai il Login'."/>

Immagine 11b: Registrazione errata per mancanza di logo

Le Immagini 12 e 12a raffigurano l'Homepage per il GdS. Questa schermata è molto simile alla schermata dei dettagli di un servizio per l'utente, con delle piccole differenze; sono infatti presenti 3 nuovi tasti al posto del tasto "Segnala" dell'interfaccia utente:

- **"Modifica"**: Questo tasto permette di accedere alla pagina di modifica del GdS (Immagine 13), che permette di modificare i dettagli del servizio
- **"Invia un avviso"**: Questo tasto permette di accedere alla pagina che permette ai GdS di mandare ai propri utenti delle notifiche (Immagine 14)
- **"Invia un sondaggio"**: Questo tasto permette di accedere alla pagina che permette ai GdS di creare e mandare ai propri utenti dei sondaggi (Immagine 15)

Oltre a questo, è presente, in fondo alla pagina, una sezione che permette la creazione e la modifica del form di segnalazione, che deve essere compilato da ogni utente che decide di segnalare una problematica del servizio (si noti come le domande presenti nell'Immagine 12a sono le stesse presenti nell'Immagine 4).

Da queste immagini possiamo notare un'ulteriore differenza fra le schermate riservate agli utenti e quelle riservate ai GdS: la Sidebar di sinistra del GdS riporta infatti tutte le notifiche e i sondaggi mandati ai propri utenti.

L'Immagine 12b ci mostra come, passando il cursore sopra le varie sezioni dell'Areogramma, sia possibile conoscere quali sono le sezioni del servizio più segnalate (funzionalità disponibile anche per gli altri ruoli). L'Immagine 12c, infine, ci mostra il messaggio di errore che viene riportato nel caso in cui si provi ad inserire una risposta vuota nel form di segnalazione.

The screenshot shows the GdS (Gestore dei Segnalazioni) homepage. On the left, there's a sidebar with links for 'Avvisi mandati' and 'Sondaggi Mandati'. The main content area features a card for the 'Muoversi in Trentino' entry. It includes the app logo, the name 'Muoversi in Trentino', the company 'Azienda: Trentino Trasporti', a description 'App per monitorare i trasporti', and three buttons: 'Modifica', 'Invia un avviso', and 'Invia un sondaggio'. Below this is a bar chart titled 'Segnalazioni negli ultimi 10 giorni' showing the number of reports for different date ranges. To the right is a 'Classifica' section listing various entries with their logos and the number of reports. At the bottom, there are sections for 'Commenti' and 'Motivo delle segnalazioni'.

Immagine 12: Homepage del GdS (1)

This screenshot shows the same GdS homepage as above, but with more content visible. The 'Commenti' section on the left contains several user reports, such as '@utente.potente Non riesco ad aprire il sito!' and '@test "Errore nel mostrare le rotte"'. To the right is a donut chart titled 'Motivo delle segnalazioni'. Below these are sections for 'Form di segnalazione' (listing options like 'Tracking bus', 'Sito web', and 'Biglietteria') and a text input field for adding new options. The right side also features the 'Classifica' section.

Immagine 12a: Homepage del GdS (2)

Motivo delle segnalazioni

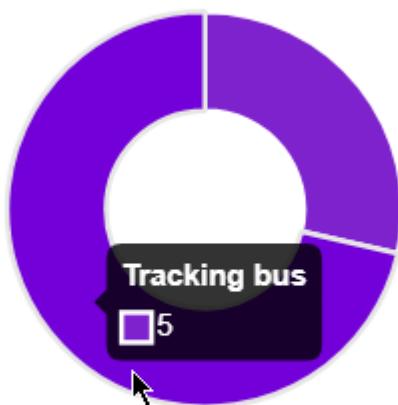


Immagine 12b: Visualizzazione dettagli dell'Areogramma

Form di segnalazione

lista di possibili opzioni:

- Tracking bus Elimina
- Sito web Elimina
- Biglietteria Elimina

Scrivi qui l'opzione da aggiungere

Opzione non presente

Aggiungi

Immagine 12c: Errore: aggiunta di un'opzione nulla per il form di segnalazione

Le Immagini 13 e 13a raffigurano la pagina, accessibile attraverso il tasto “Modifica” della Homepage, che permette al GdS di modificare i dati del proprio servizio. L’Immagine 13b raffigura il pop-up che viene evocato per confermare l’avvenuta modifica.

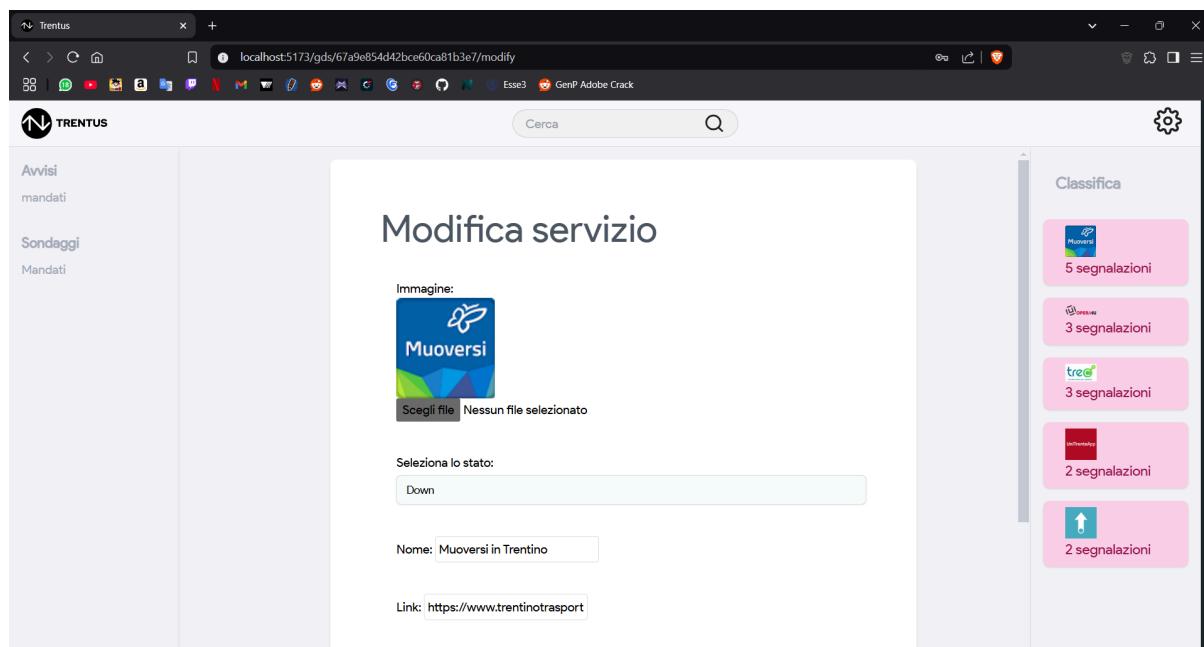


Immagine 13: Modifica dei dettagli e dello stato del servizio (1)

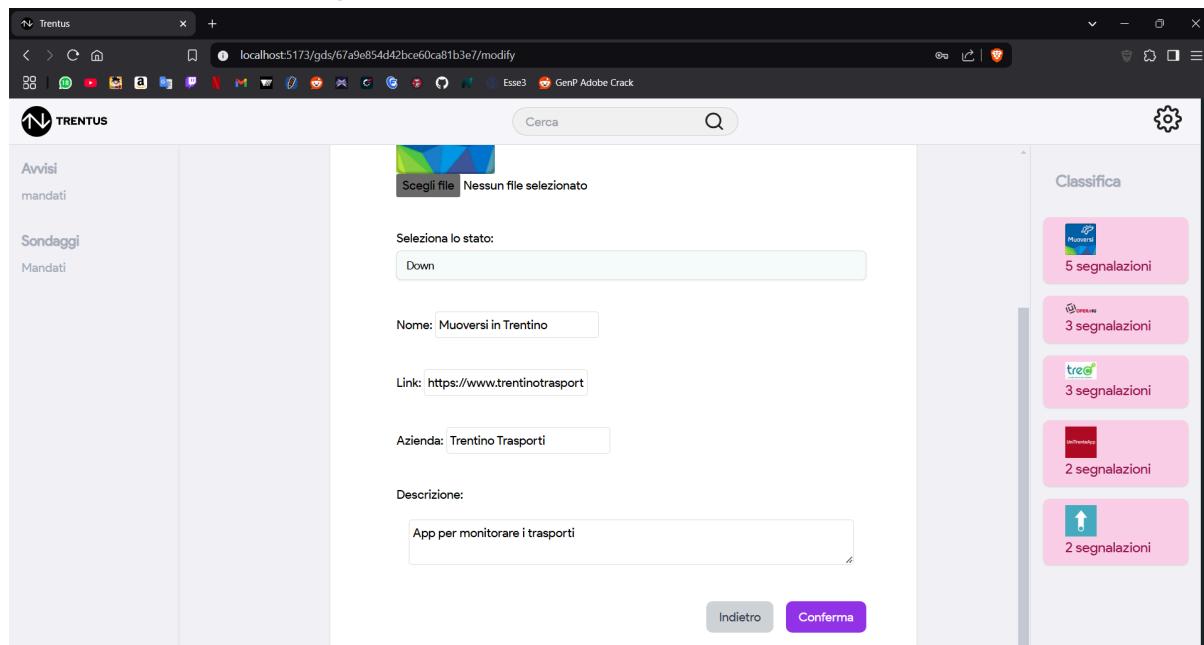


Immagine 13a: Modifica dei dettagli e dello stato del servizio (2)

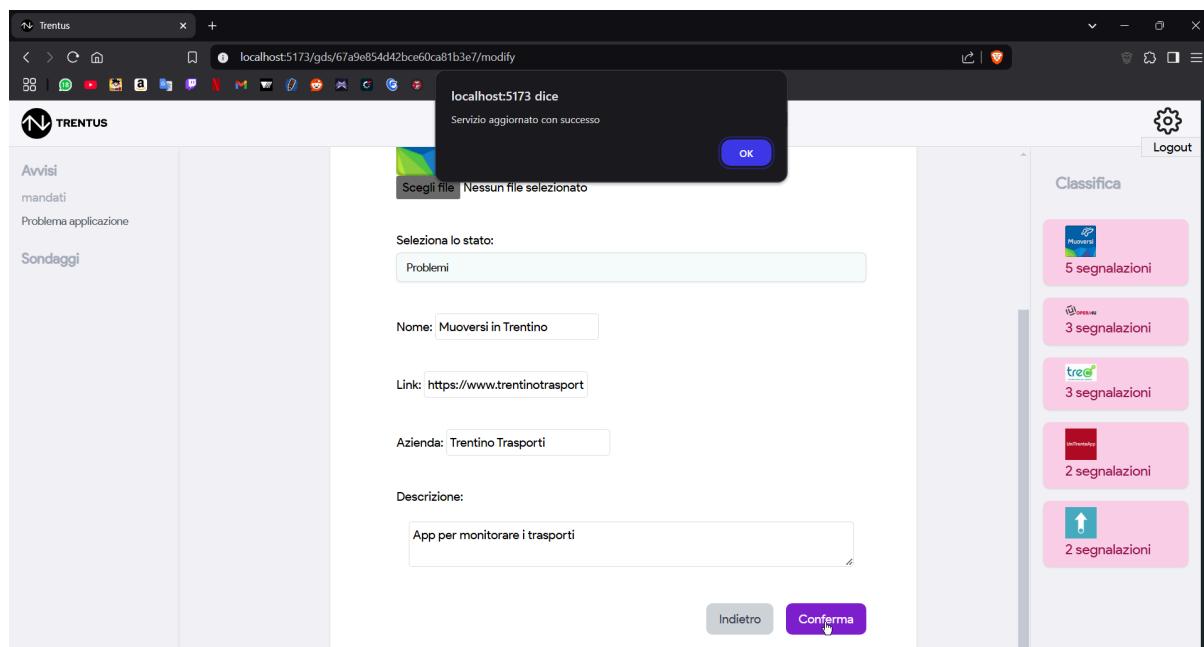


Immagine 13b: Dettagli del servizio modificati correttamente

L'Immagine 14 mostra la schermata, accessibile attraverso il tasto “Invia un avviso” della homepage, che permette ai GdS di inviare ai propri utenti delle notifiche. In questa sezione bisognerà inserire un titolo, una descrizione, il tipo di problema ed un lasso di tempo. L'Immagine 14a mostra il messaggio evocato in caso di corretto invio dell'avviso. Al contrario, l'Immagine 14b mostra il messaggio evocato in caso di avviso vuoto. Infine, l'Immagine 14c mostra la schermata dove è possibile vedere la notifica manda per intero, accessibile cliccando il titolo della notifica dalla Sidebar di sinistra.

Trentus

localhost:5173/gds/67a9e854d42bce60ca81b3e7/sendalert

Cerca

Classifica

Muvveri 5 segnalazioni

Vodafone 3 segnalazioni

tre 3 segnalazioni

Trentus 2 segnalazioni

Indietro Invia

Immagine 14: Pagina di invio avvisi (notifiche per gli utenti che hanno il servizio tra i preferiti)

Avvisi mandati

Sondaggi Mandati

Titolo: Problema applicazione

Messaggio: Abbiamo riscontrato un problema all'applicazione, la funzione tracking autobus non sarà disponibile per

Tipo: Problemi

Inizio: 11/02/2025

Fine: 14/02/2025

Avviso mandato con successo

Indietro Invia

Muvveri 5 segnalazioni

Vodafone 3 segnalazioni

tre 3 segnalazioni

Trentus 2 segnalazioni

Indietro Invia

Immagine 14a: Avviso mandato correttamente

The screenshot shows a web-based application interface for 'TRENTUS'. On the left, there's a sidebar with navigation links: 'Avvisi' (selected), 'mandati', 'Problema applicazione', 'Sondaggi', and 'Mandati'. The main area has fields for 'Titolo' (Title) and 'Messaggio' (Message). Below these are dropdown menus for 'Tipo' (Type), 'Inizio' (Start Date), and 'Fine' (End Date). A red error message box at the bottom states 'Errore: parametri errati' (Error: wrong parameters). At the bottom right are two buttons: 'Indietro' (Back) and a purple 'Invia' (Send) button. To the right, there's a 'Classifica' (Classification) section with five pink boxes showing signal counts: 'Postnew' (5 segnalazioni), 'tre' (3 segnalazioni), 'Internship' (2 segnalazioni), and an upward arrow icon (2 segnalazioni).

Immagine 14b: Errore nell'invio dell'avviso (avviso vuoto)

This screenshot shows the same application interface as above, but the 'Problema applicazione' link in the sidebar is selected. The main content area displays a title 'Problema applicazione' and a sub-section 'Motivazione: problemi'. It contains a text message: 'Abbiamo riscontrato un problema all'applicazione, la funzione tracking autobus non sarà disponibile per i prossimi 3 giorni.' Below this is a timestamp: 'Periodo: 2025-02-11T00:00:00.000Z-2025-02-14T00:00:00.000Z'. At the bottom are 'Indietro' and 'Invia' buttons. The 'Classifica' section on the right is identical to the one in the previous screenshot.

Immagine 14c: Pagina di visualizzazione dell'avviso mandato

Le Immagini 15 e 15a raffigurano la schermata, accessibile attraverso il tasto “Invia un sondaggio” della Homepage, che permette ai GdS di inviare dei sondaggi ai propri utenti. Questa schermata permette infatti ai GdS di creare dei veri e propri form, garantendo la possibilità di inserire check boxes, domande a risposta chiusa e domande a risposta aperta. L’Immagine 15b raffigura il pop-up di errore, evocato in caso di sondaggio vuoto. L’Immagine 15c rappresenta il pop-up evocato per confermare il corretto invio del sondaggio.

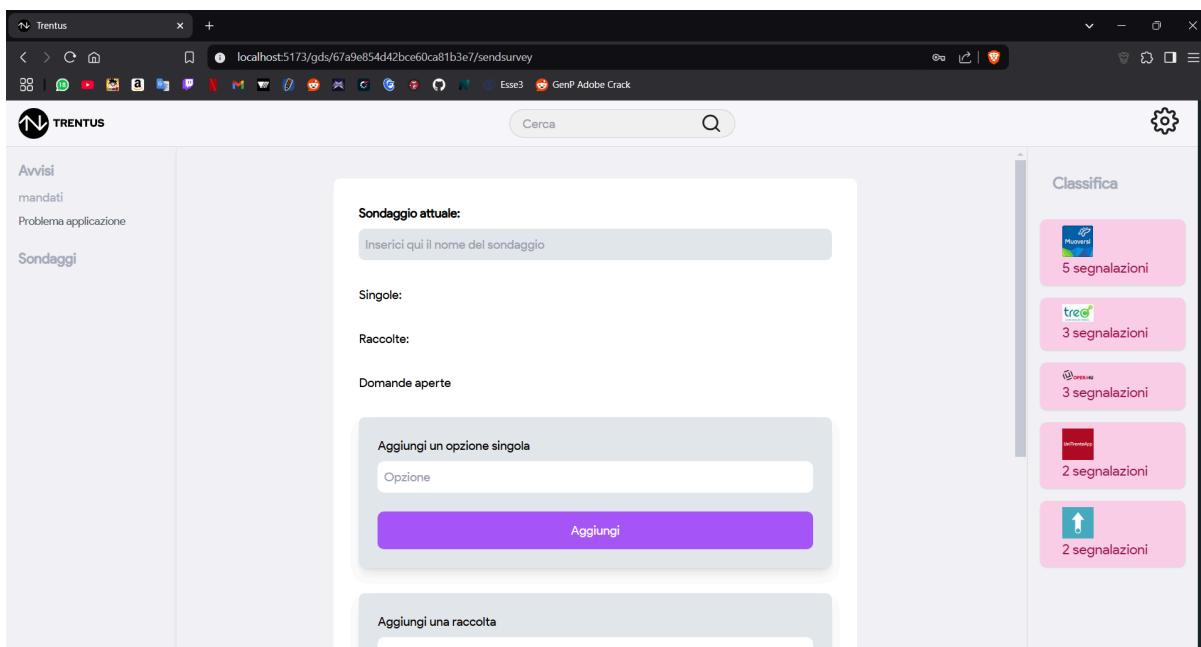


Immagine 15: Pagina di creazione e invio sondaggi (1)

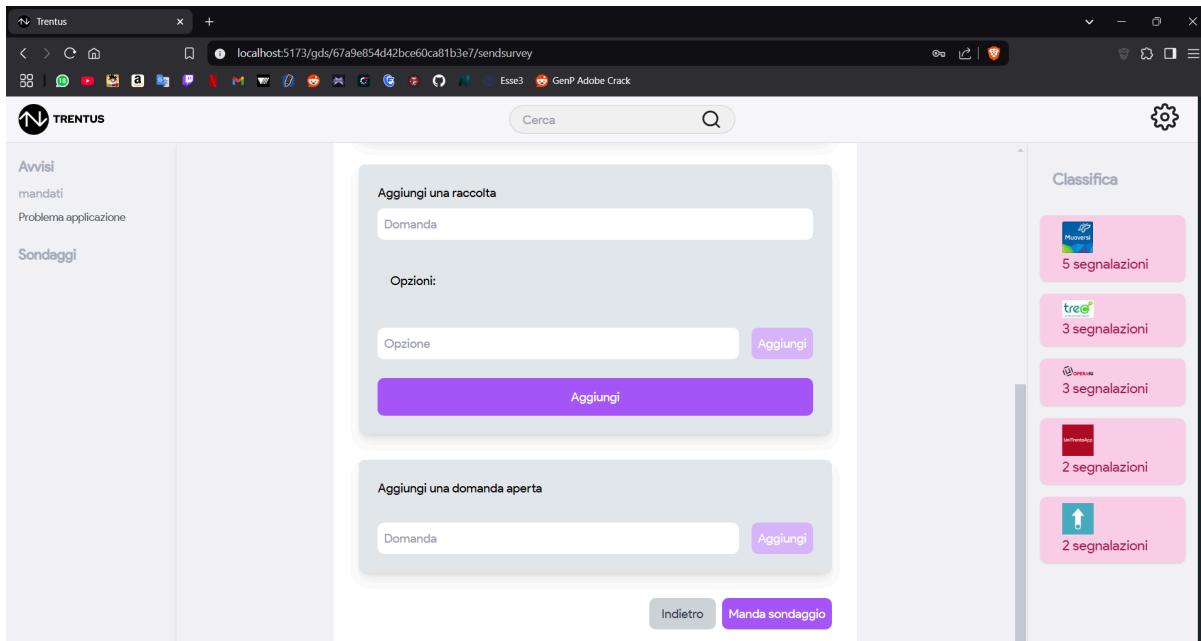


Immagine 15a: Pagina di creazione e invio sondaggi (2)

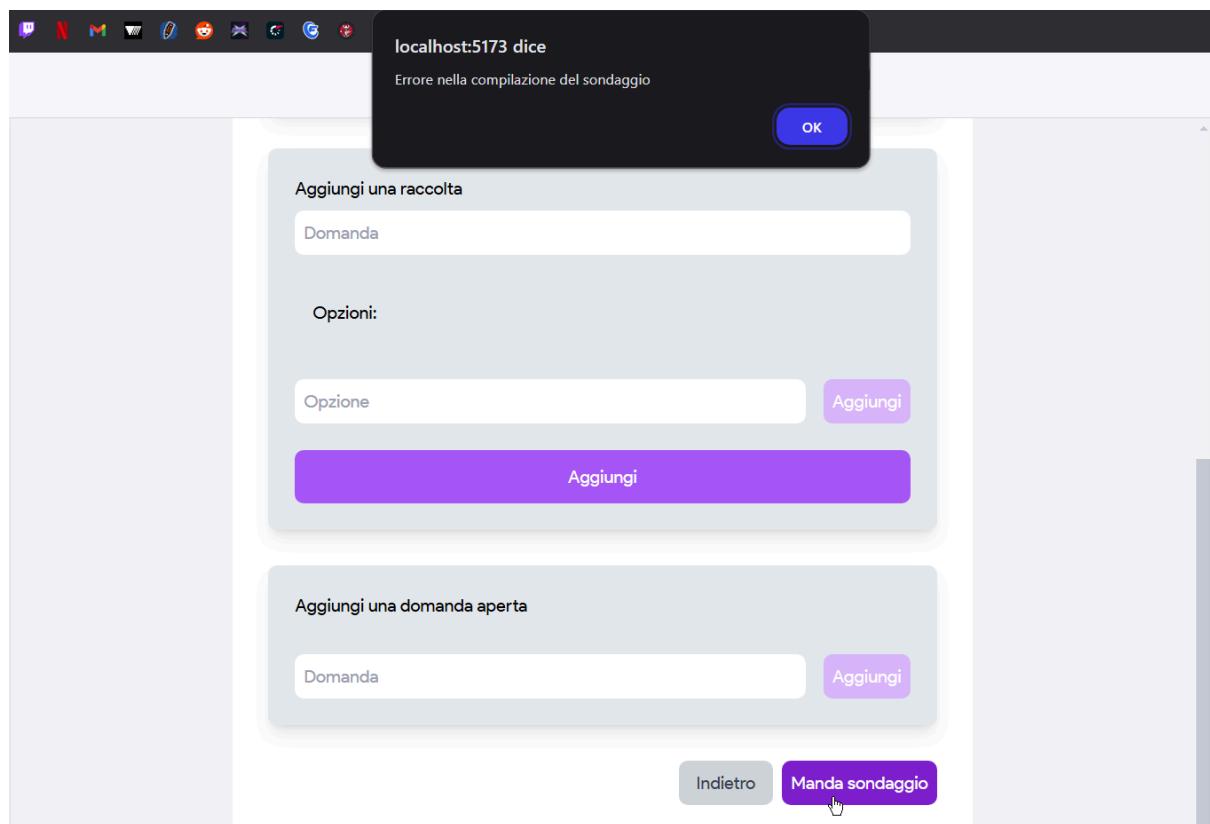


Immagine 15b: Errore nell'invio del sondaggio (sondaggio vuoto)

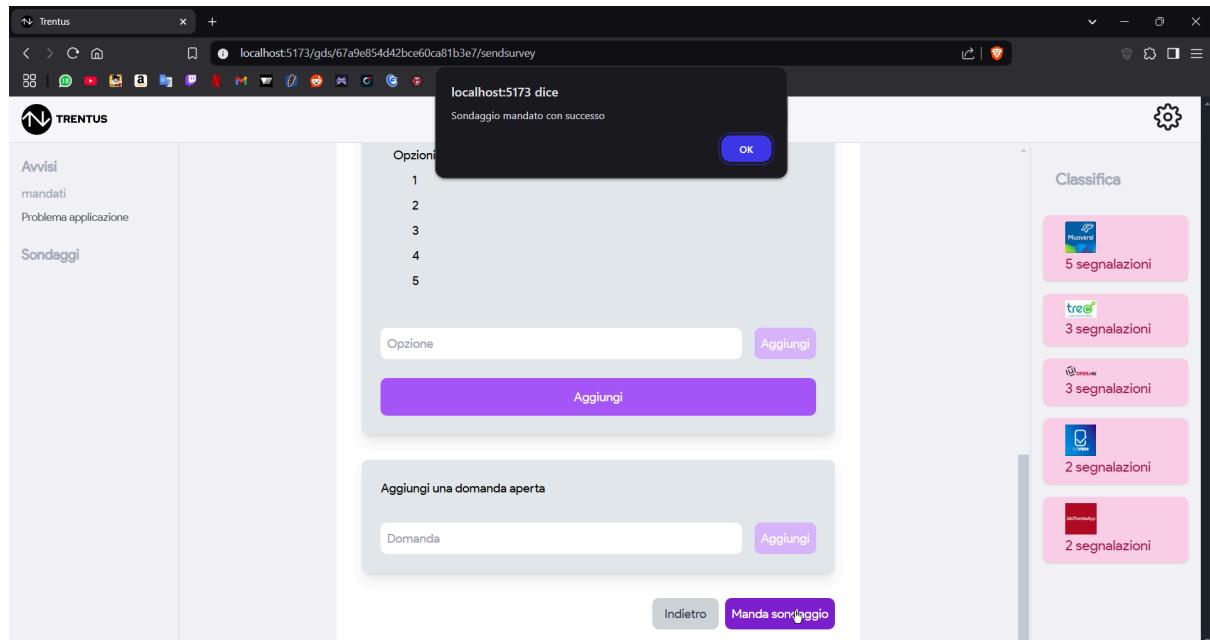


Immagine 15c: Sondaggio mandato correttamente

L'Immagine 16 raffigura la pagina contenente i risultati ottenuti dai sondaggi inviati. Questa schermata è raggiungibile cliccando sul titolo dei sondaggi mandati, presenti nella Sidebar di sinistra. Questa schermata mostra il numero di persone che hanno risposto al sondaggio e quante di queste abbiano spuntato una determinata opzione. Infine, troviamo la sezione contenente tutte le risposte alle eventuali domande a risposta aperta che il GdS può inserire nei suoi sondaggi.

The screenshot shows a web browser window titled 'Trentus'. The URL is 'localhost:5173/gds/service/67a9e854d42bce60ca81b3e7/survey/67ab25e1abe8d605b0005715'. The main content area displays a survey titled 'Sondaggio sulla qualità del servizio' with a note 'Numero risposte: 0'. It lists three closed-ended questions with zero responses each: 'Ho usato la funzione Mappe dell'app | 0', 'Ho usato la funzione di tracking dei bus dell'app | 0', and 'Ho usato la funzione orari dei bus dell'app | 0'. Below these is a section for 'Domande aperte' (Open questions) containing the text 'Facci sapere come valuti l'applicazione e cosa vorresti vedere in un futuro!'. A 'Indietro' (Back) button is at the bottom left. On the right, there's a sidebar titled 'Classifica' showing five entries with logos and signal counts: Huawei (5), tree (3), Vodafone (3), Internap (2), and an upward arrow icon (2). The sidebar also has a gear icon for settings.

Immagine 16: Pagina di visualizzazione risultati sondaggio

5.3 Front-End per il SuperAdmin

Per il SuperAdmin, il front-end fornisce le seguenti schermate:

- Pagina di login
- Homepage SuperAdmin
- Pagina del servizio
- Pagina di modifica del servizio
- Pagina di aggiunta nuovo servizio

Nell'Immagine 18 viene raffigurata la pagina di login per il SuperAdmin. Questa pagina è accessibile solamente tramite URL, non ci sono tasti che vi ci portano. In ogni caso, è comunque presente un link alla pagina di login per gli Utenti in caso di errore. Per il resto, la pagina di login è identica a quella del login dell'Utente. Per il SuperAdmin non esiste pagina di registrazione, in quanto le credenziali vengono create separatamente dal sito web.

Nell'Immagine 18a viene riportato il messaggio di errore in caso di credenziali errati o nulle.

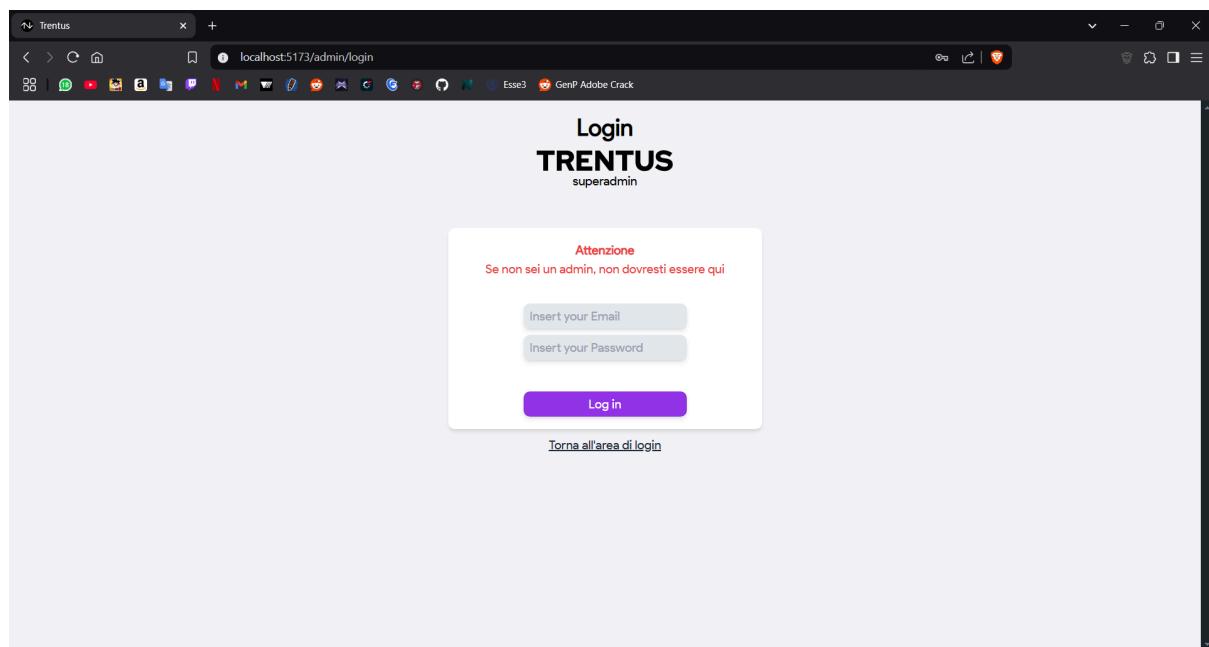


Immagine 18: Pagina di login per il SuperAdmin

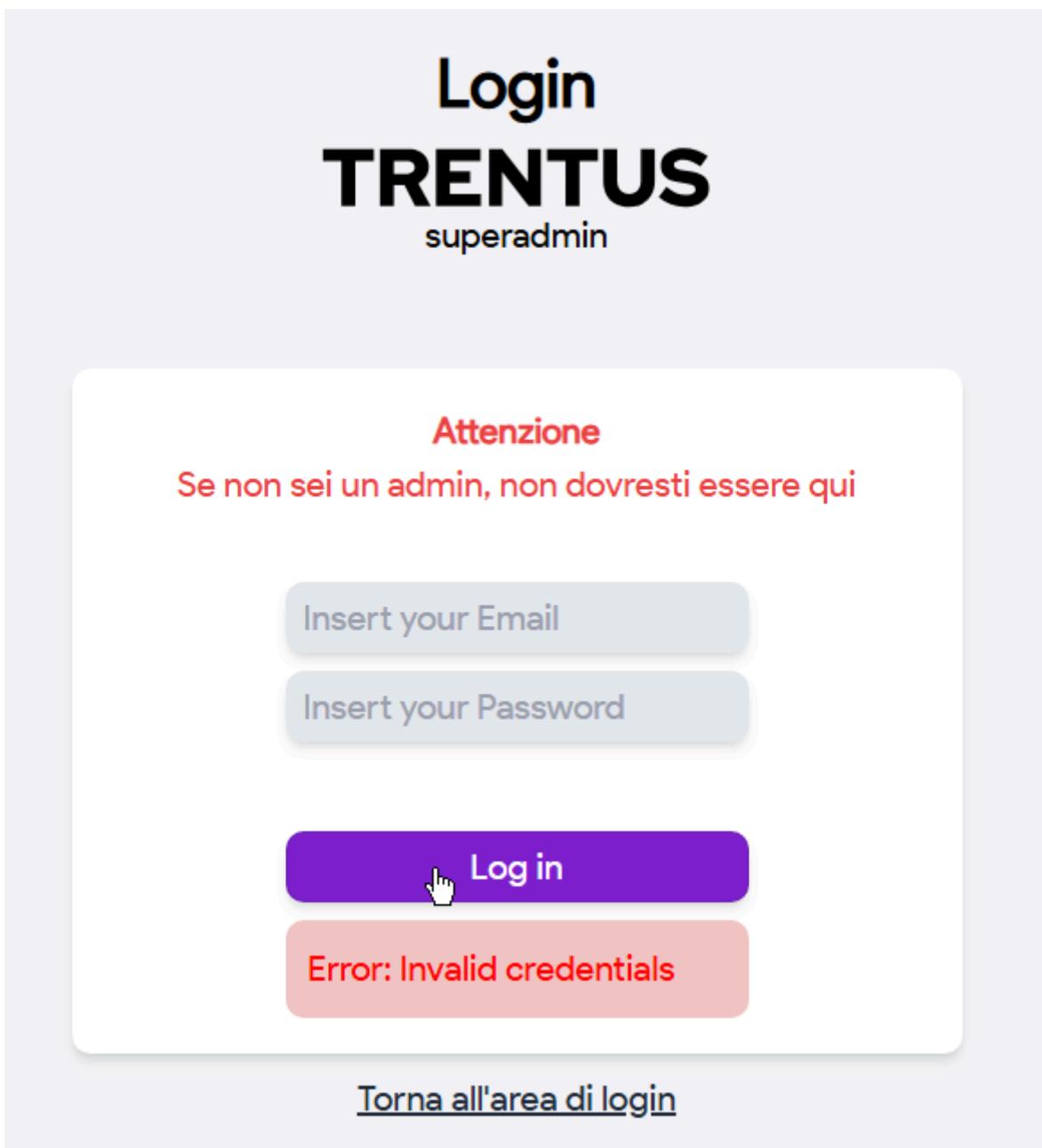


Immagine 18a: Errore nel login per il SuperAdmin (credenziali errate o nulle)

Nell'Immagine 19 viene raffigurata la Homepage per i SuperAdmin. Questa è simile a quella dell'Utente, ma con delle piccole differenze:

- Ogni riferimento ai preferiti è assente, il SuperAdmin non può avere servizi preferiti
- La Sidebar di sinistra, invece di riportare le notifiche, mostra le richieste di aggiunta servizio che nuovi GdS inviano al momento della registrazione.

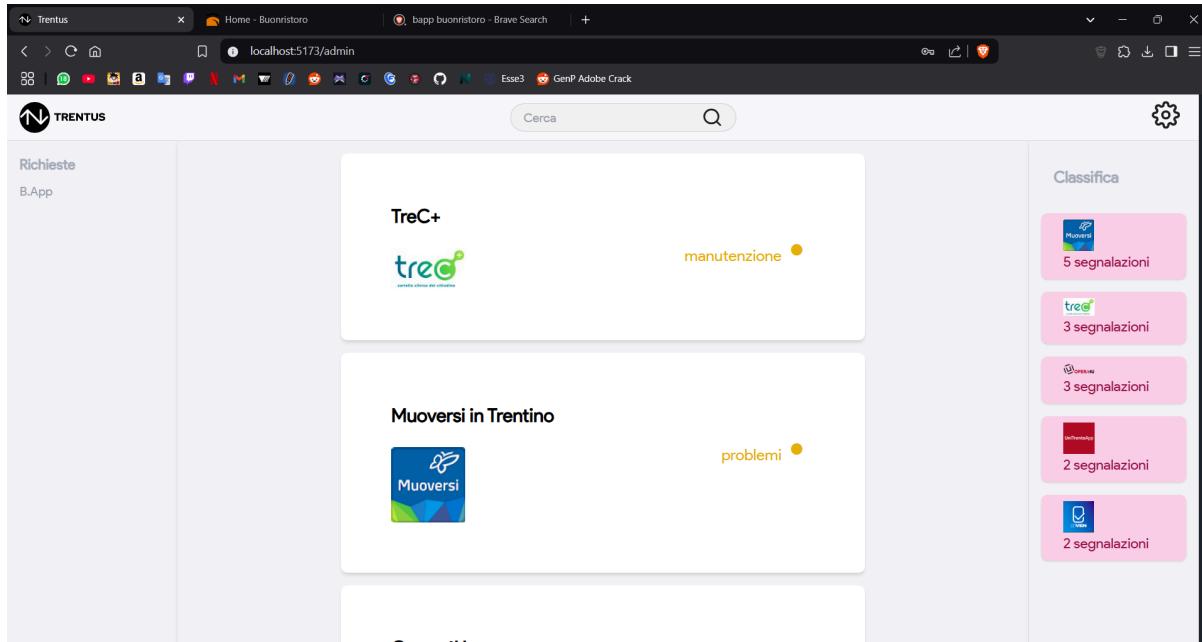


Immagine 19: Homepage del SuperAdmin

Le Immagini 20 e 20a mostrano la schermata di dettaglio di un servizio per il SuperAdmin. Questa è essenzialmente identica alla Homepage del GdS, con l'unica differenza che non è possibile mandare avvisi o sondaggi. Viene comunque conservata la possibilità di modificare i dettagli di un servizio, nelle stesse modalità che sono disponibili per il GdS (si veda Immagine 21 e 21a)

The screenshot shows a web browser window with the URL `localhost:5173/admin/service/67a9e7b5d42bce60ca81b3dc`. The page displays detailed information about the service 'TreC+' managed by 'Trentino Salute'. It includes a logo, a status indicator 'manutenzione ●', a 'Modifica' button, a bar chart showing reports over the last 10 days, and sections for comments and reporting reasons. A sidebar on the right lists other services with their logos and signal counts.

Immagine 20: Pagina dei dettagli del servizio per il SuperAdmin (1)

The screenshot shows the same web browser window with the same URL. This time, it displays a form for reporting issues. It includes a list of reported problems, a circular progress bar, a section for adding new report options, and a back button. The sidebar on the right remains the same.

Immagine 20a: Pagina dei dettagli del servizio per il SuperAdmin (2)

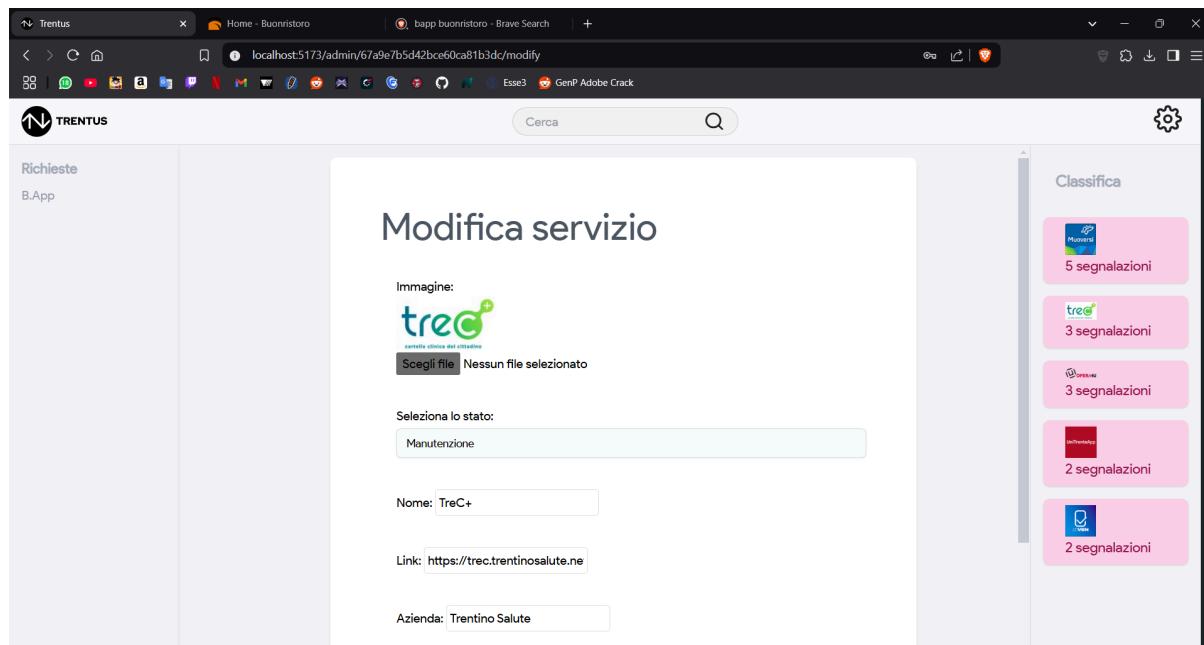


Immagine 21: Pagina di modifica dei dettagli del servizio per il SuperAdmin

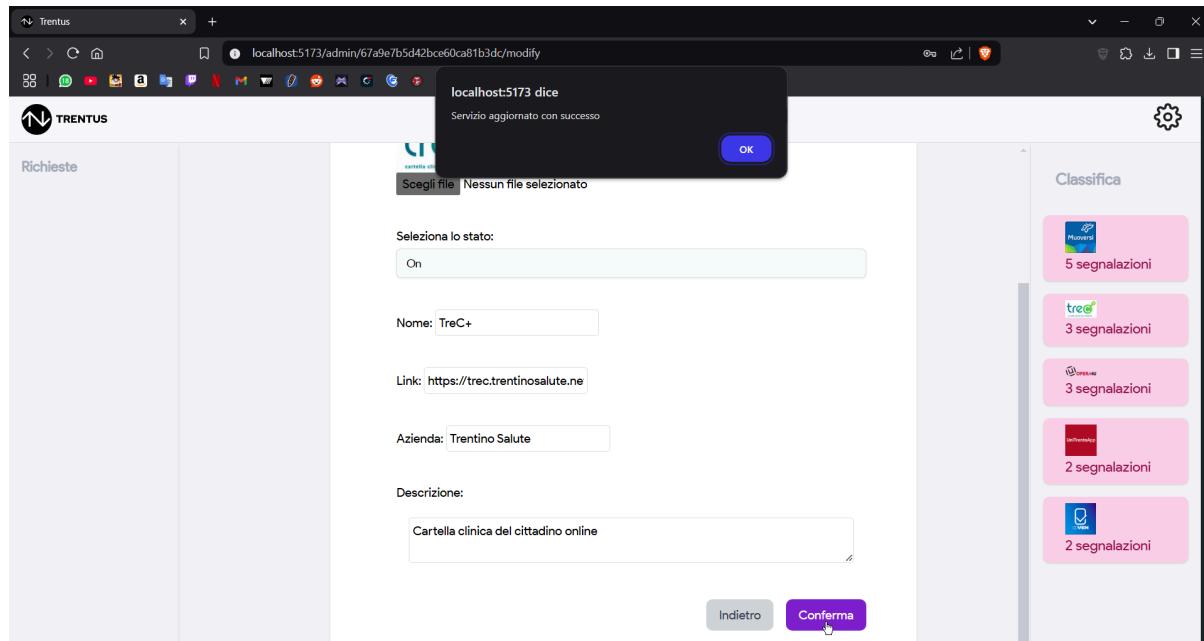


Immagine 21a: Modifica dei dettagli del servizio confermata

Le Immagini 22 e 22a mostrano la schermata che permette ai SuperAdmin di accettare o meno le richieste di registrazione di nuovi servizi all'interno di Trentus. Questa schermata riporta i dettagli forniti dal nuovo GdS al momento di registrazione. Dopo aver verificato che il servizio sia legittimo, il SuperAdmin potrà cliccare il tasto conferma per inserirlo sulla piattaforma. Infine, l'Immagine 22b raffigura il pop-up evocato per confermare la corretta registrazione del servizio su Trentus.

The screenshot shows a web application interface for managing service requests. On the left, there's a sidebar with a logo and navigation links for 'Richieste' and 'B.App'. The main content area has a title 'Richiesta di aggiunta servizio' and an ID 'Id: 67ab3200abe8d605b00057c6'. It displays a circular logo for 'B.APP' and the following details:

- Nome:** B.App
- Link:** <https://buonristoro.com/>
- Azienda:** Buonristoro Vending Group

To the right, there's a 'Classifica' sidebar listing various service providers with their signal counts:

- Munirail (5 segnalazioni)
- tree (3 segnalazioni)
- greenme (3 segnalazioni)
- Unibet (2 segnalazioni)
- Spazio (2 segnalazioni)

The URL in the browser bar is 'localhost:5173/admin/67ab3200abe8d605b00057c6/approve'.

Immagine 22: Pagina di verifica e accettazione delle richieste di aggiunta di un nuovo servizio (1)

This screenshot shows the second step of the service addition process. The interface is identical to the first one, displaying the same service details and classification sidebar. At the bottom, there are two buttons: 'Indietro' (Back) and 'Conferma' (Confirm). The URL in the browser bar is 'localhost:5173/admin/67ab3200abe8d605b00057c6/approve'.

Immagine 22a: Pagina di verifica e accettazione delle richieste di aggiunta di un nuovo servizio (2)

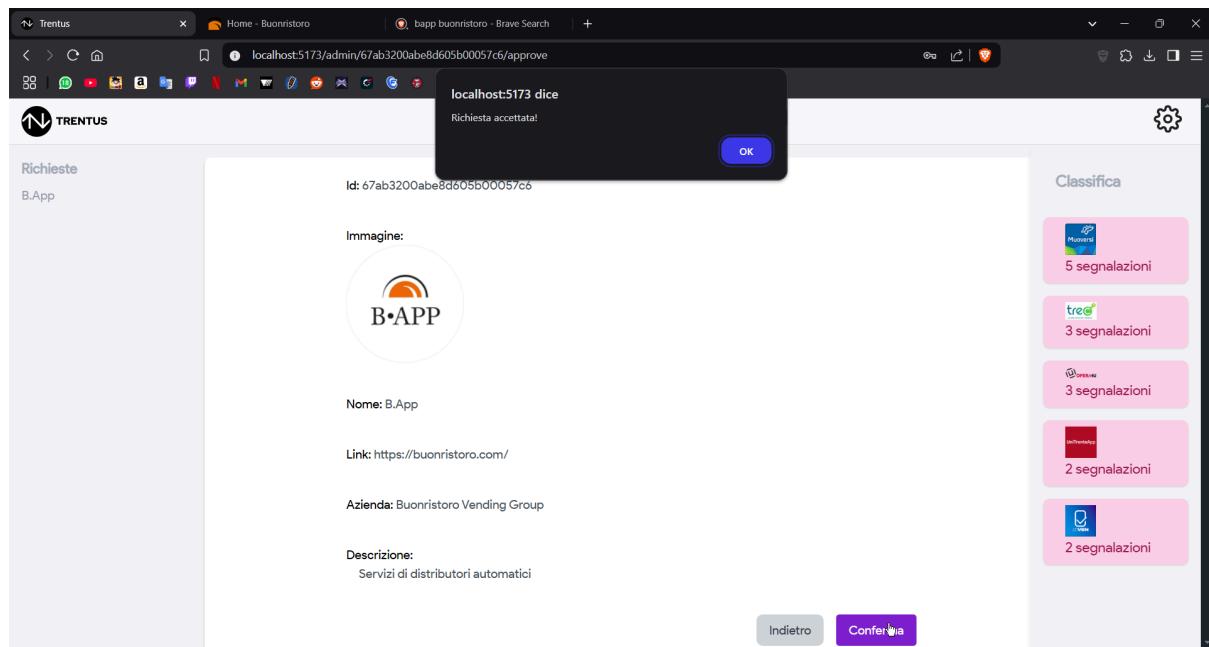


Immagine 22b: Conferma e accettazione del nuovo servizio

5.4 Schermate di Errore

L'Immagine 23 raffigura la schermata che viene mostrata in caso di accesso ad un link non previsto dell'applicazione. Per permettere a chi sbaglia di tornare a pagine conosciute, è disponibile un link alla Home dell'Utente.

L'Immagine 24 raffigura la schermata che viene mostrata se si prova ad entrare su una pagina in cui non si ha l'autorizzazione ad entrare (come, ad esempio, l'homepage del GdS se si è loggati come Utente).

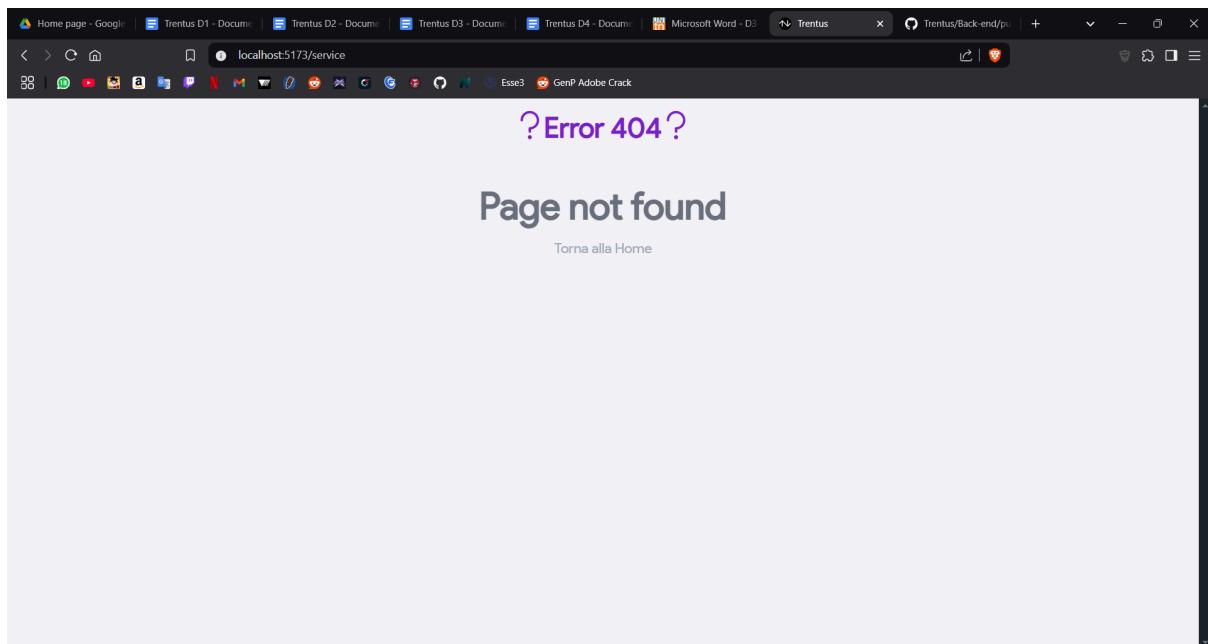


Immagine 23: Schermata di Error 404

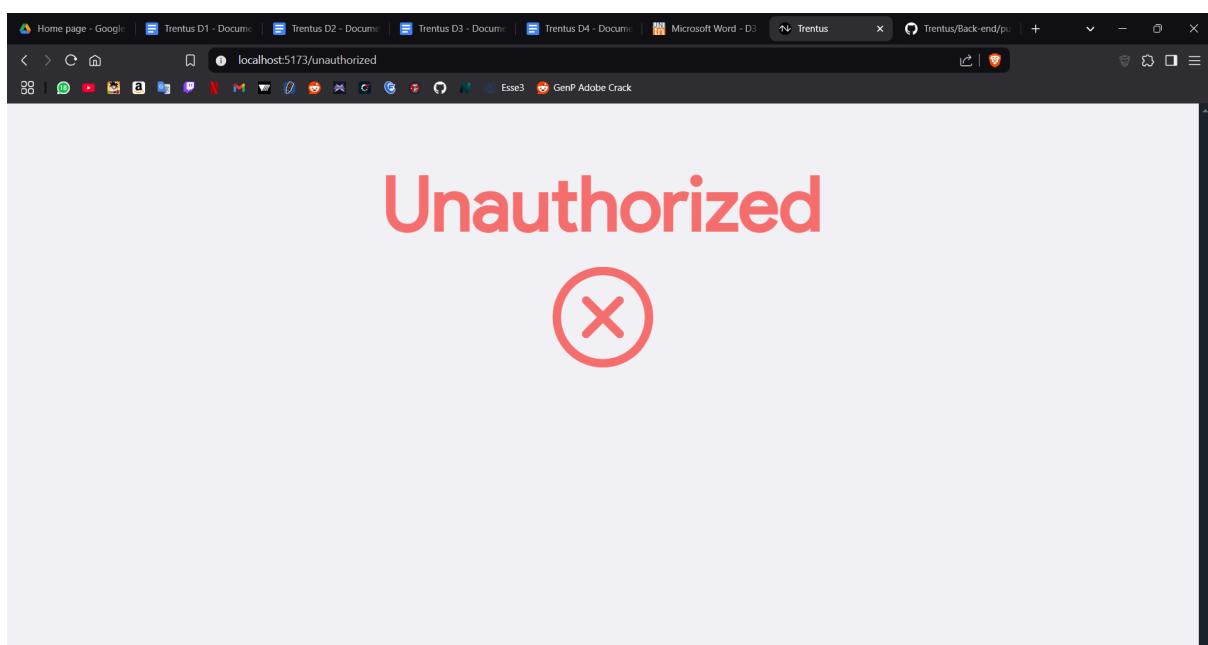


Immagine 24: Schermata di accesso a pagina non autorizzata

6. Deployment

Un'istanza del backend è ospitata sulla piattaforma render.com ed è disponibile al link <https://trentus.onrender.com>. Il frontend è deployato separatamente al link <https://trentus-frontend.onrender.com>. Per accedere al deploy dell'applicazione è possibile utilizzare i seguenti account:

- **Utente:** email: utente.potente@email.it / password: utente
- **GdS:** email: trentino.trasporti@email.it / password: trasporti
- **SuperAdmin:** email: proti@gmail.com / password: asd

Per qualsiasi problema con il deployment contattare:

- Giovanni Sbalchiero (Team Leader): giovanni.sbalchiero@studenti.unitn.it
- Giuseppe Conte: giuseppe.conte@studenti.unitn.it
- Luca da Riz: luca.dariz-1@studenti.unitn.it