

Progetto:

# SatisTrento

Titolo del documento:

## Sviluppo

Document Info

Doc. Name	D3-satisTrento-sviluppo	Doc. Number	D3 V0.1
Description	Documento di sviluppo dell'applicazione		

### Authors

Facchini Luca	245965
Prigione Luca	242880
Faa Enrico	243889



# Indice

<b>Scopo del documento</b>	<b>1</b>
<b>1 <i>User Stories</i></b>	<b>2</b>
<b>2 <i>User Workflow</i></b>	<b>13</b>
<b>3 <i>Web APIs</i></b>	<b>15</b>
<b>4 <i>Implementazione</i></b>	<b>30</b>
4.1 Repository Organization . . . . .	30
4.2 <i>Branching Strategy</i> e organizzazione del lavoro . . . . .	31
4.3 <i>Dependency</i> . . . . .	31
4.4 <i>Database</i> . . . . .	32
4.5 Testing . . . . .	34
<b>5 <i>FrontEnd</i></b>	<b>35</b>
5.1 <i>Home Page</i> . . . . .	35
5.2 <i>Login</i> . . . . .	36
5.3 Funzionalità Analista . . . . .	37
5.4 Funzionalità Sondaggista . . . . .	40
<b>6 <i>Deployment</i></b>	<b>43</b>



## Scopo del documento

Il presente documento riporta le informazioni relative allo sviluppo del progetto SatisTrento. In particolare, viene descritta la parte di visualizzazione della homepage, la visualizzazione da parte di tutti gli utenti delle informazioni riguardanti le circoscrizioni/quartieri, la gestione dei sondaggi da parte degli utenti comunali, la visualizzazione della “soddisfazione media” e la visualizzazione dei dettagli di un sondaggio. Viene inoltre descritta la sezione di analisi per gli addetti comunali.

Il documento è strutturato partendo dalle *User Stories* ed gli *User Workflow* per poi passare alla descrizione degli *endpoint* delle *web APIs* e a come il codice è stato organizzato all’interno della *repository* del progetto. Si terminerà il documento con una descrizione della parte di *frontend* sviluppata ed una descrizione della tecnica di *deployment* utilizzata.

# 1 *User Stories*

Vengono di seguito riportate le *User Stories* (US) identificate per la parte che si è andata a implementare del progetto SatisTrento. Le US sono state individuate a partire dai requisiti funzionali (RF) descritti dal documento di analisi dei requisiti (D1).

## User Story 1: Visualizzare gli attributi

Come utente voglio poter visualizzare gli attributi demografici e riguardanti la soddisfazione della città, così da poter avere delle informazioni dettagliate della città

### Criteri di accettazione:

- Tutti gli utenti possono visualizzare gli attributi demografici e riguardanti la soddisfazione di Trento

### TASKS - User Story 1:

#### 1. Implementare richiesta dei dati al database

- Creare l'API necessaria per richiedere al database i dati degli attributi di Trento
- Configurare l'applicazione per utilizzare l'API per richiedere i dati degli attributi di Trento

#### 2. Mostrare gli attributi a schermo

- Creare un'interfaccia che mostri i valori degli attributi
- Impostare l'interfaccia per mostrare gli attributi nel punto giusto sullo schermo

#### 3. Testare che gli attributi vengano visualizzati correttamente

- Verificare che i valori degli attributi siano corretti
- Verificare che i valori degli attributi siano collocati nel posto giusto sullo schermo

## User Story 2: Visualizzare la mappa suddivisa per zone

Come utente voglio visualizzare la mappa del comune di Trento divisa per zone, così da poter distinguere la divisione delle varie aree della città e il grado di soddisfazione medio che vi è in esse

### Criteri di accettazione:

- L'utente può vedere la mappa di Trento
- La mappa è suddivisa in zone ben definite
- Ogni zona è colorata in base al suo grado di soddisfazione medio

### TASKS - User Story 2:

#### 1. Implementare la mappa

- Aggiungere all'interfaccia la mappa di Trento

#### 2. Dividere la mappa in zone

- Implementare la divisione della mappa secondo i confini delle zone

#### 3. Configurare la colorazione delle zone

- Associare a ogni valore di soddisfazione media un colore

- Implementare la logica che colora le zone in base alla soddisfazione

#### 4. Testare la corretta visualizzazione delle zone

- Verificare che la mappa sia inserita nell'interfaccia correttamente
- Verificare che la mappa sia suddivisa correttamente
- Verificare che i colori delle zone siano corretti

## User Story 3: Spostare il focus della mappa

Come utente voglio poter spostare il focus della mappa, così da poter cambiare l'area geografica che sto visualizzando

### Criteri di accettazione:

- L'utente può spostare il focus centrale della mappa

### TASKS - User Story 3:

#### 1. Implementare lo spostamento del focus

- Fare in modo che sia possibile spostare il focus della mappa
- Modificare la posizione del focus della mappa in base all'input dell'utente

#### 2. Testare lo spostamento del focus

- Verificare che il focus della mappa si sposti con l'input dell'utente
- Verificare che il focus della mappa si sposti nella direzione corretta

## User Story 4: Ingrandire/rimpicciolire la mappa

Come utente voglio poter ingrandire o rimpicciolire la mappa, così da poter avere una visione più chiara dell'area geografica che sto visualizzando

### Criteri di accettazione:

- L'utente può ingrandire la mappa
- L'utente può rimpicciolire la mappa

### TASKS - User Story 4:

#### 1. Implementare il sistema di modifica della grandezza della mappa

- Fare in modo che sia possibile impostare la grandezza della mappa
- Modificare la grandezza della mappa in base all'input dell'utente

#### 2. Testare la modifica della grandezza della mappa

- Verificare che la mappa si ingrandisca e si rimpicciolisca con l'input dell'utente

## User Story 5: Selezionare le zone

Come utente voglio poter selezionare una zona della città, così da poter ricevere informazioni più dettagliate riguardanti quella zona

**Criteri di accettazione:**

- L'utente può selezionare una zona
- L'utente può deselectare una zona

**TASKS - User Story 5:****1. Implementare la selezione**

- Implementare la logica di selezione di una zona e deselectare se viene selezionata di nuovo

**2. Testare il sistema di selezione**

- Verificare che sia possibile selezionare una zona
- Verificare che sia possibile deselectare la zona selezionata

**User Story 6: Visualizzare gli attributi delle zone**

Come utente voglio poter visualizzare gli attributi demografici e riguardanti la soddisfazione della zona che ho selezionato. Così da poter avere accesso alle informazioni specifiche che cerco

**Criteri di accettazione:**

- Quando una zona è selezionata, l'utente può vedere gli attributi demografici e riguardanti la soddisfazione di quella zona
- Quando nessuna zona è selezionata, l'utente può vedere gli attributi demografici e riguardanti la soddisfazione di Trento

**TASKS - User Story 6:****1. Implementare richiesta dei dati delle zone al database**

- Creare l'API necessaria richiedere al database i dati di una zona
- Configurare l'applicazione per utilizzare l'API per richiedere i dati della zona selezionata

**2. Creare l'interfaccia di visualizzazione degli attributi**

- Configurare l'interfaccia grafica per mostrare gli attributi di una zona quando viene selezionata
- Configurare l'interfaccia per tornare a mostrare gli attributi di Trento quando nessuna zona è selezionata

**3. Testare la visualizzazione delle zone**

- Verificare che quando una zona è selezionata i suoi attributi vengono mostrati nell'interfaccia
- Verificare che quando nessuna zona è selezionata l'interfaccia mostra gli attributi di Trento
- Verificare che i valori degli attributi mostrati siano corretti
- Verificare che gli attributi delle zone siano nel posto giusto sullo schermo

**User Story 7: Cambiare la tipologia di zone**

Come utente voglio poter cambiare la tipologia di zona con la quale si può interagire sulla mappa, così da poter ricevere informazioni riguardanti l'area geografica di preferenza senza complicare l'utilizzo della mappa

**Criteri di accettazione:**

- L'utente può cambiare la divisione per zone da quartieri a circoscrizioni
- L'utente può cambiare la divisione per zone da circoscrizioni a quartieri



## TASKS - User Story 7:

### 1. Creare l'interfaccia di selezione

- Creare l'interfaccia che permette all'utente di selezionare la tipologia di zone da visualizzare

### 2. Implementare il cambiamento di tipologia di zona

- Fare in modo che, quando la tipologia di zona scelta cambia, la mappa viene ricaricata per essere divisa nel modo giusto

### 3. Testare lo scambio della tipologia di zona

- Verificare che sia possibile selezionare la tipologia di zona
- Verificare che la mappa venga ricaricata con la divisione corretta
- Verificare che, una volta cambiata la tipologia di zona, gli attributi delle zone mostrati dall'interfaccia siano corretti

## User Story 8: Visualizzare la tabella

Come utente analista voglio poter visualizzare le zone della città attraverso una tabella, così da poter visualizzare le zone geografiche attraverso il mio ordinamento di preferenza

### Criteri di accettazione:

- L'utente analista può decidere di visualizzare la tabella
- Gli altri utenti non possono visualizzare la tabella

## TASKS - User Story 8:

### 1. Creare l'interfaccia di selezione

- Creare l'interfaccia che permette all'utente analista di decidere di visualizzare la mappa o la tabella

### 2. Creare la tabella

- Creare l'interfaccia che mostra i dati delle zone nella forma di una tabella

### 3. Testare la visualizzazione della tabella

- Verificare che l'utente analista possa decidere di visualizzare la tabella
- Verificare che la tabella mostri i dati delle zone in modo corretto
- Verificare che, una volta cambiata la tipologia di zona, la tabella continui a mostrare gli attributi in modo corretto
- Verificare che, una volta selezionata la visualizzazione in tabella, sia possibile tornare alla visualizzazione della mappa

## User Story 9: Login

Come utente non loggato voglio poter accedere al mio account tramite Nome Utente e Password, così da poter avere accesso alle funzionalità fornite al mio account

### Criteri di accettazione:

- L'utente non loggato può inserire Nome Utente e Password
- Se le credenziali sono valide, l'utente può accedere al suo account
- Se le credenziali non sono valide, un messaggio di errore appropriato viene presentato

## TASKS - User Story 9:

### 1. Creare il form delle credenziali

- Creare l'interfaccia che permette all'utente di inserire il proprio Nome Utente, la propria Password, e di inviare le credenziali per eseguire l'accesso

### 2. Implementare i controlli di autenticazione

- Implementare la logica di controllo sulle credenziali per verificare se sono valide
- Implementare la logica che determina quale messaggio di errore presentare se le credenziali non sono valide

### 3. Aggiungere i messaggi di errore

- Creare il messaggio di errore che viene presentato se la password è errata
- Creare il messaggio di errore che viene presentato se il Nome Utente non corrisponde a nessun utente nel sistema
- Creare il messaggio di errore che viene presentato se Nome Utente o Password sono mancanti

### 4. Testare il funzionamento del login

- Verificare che il form delle credenziali funzioni correttamente
- Verificare che sia possibile accedere a un account inserendo le credenziali corrette
- Verificare che i messaggi di errore giusti vengono presentati se le credenziali non sono valide

## User Story 10: Logout

Come utente loggato voglio poter scollegarmi dall'account al quale sono attualmente collegato, così da non permettere ad altre persone di usare il mio account o da potermi collegare a un altro account

### Criteri di accettazione:

- L'utente loggato può scollegarsi dal proprio account
- Appena scollegato dal proprio account, l'utente viene reindirizzato alla pagina principale dell'applicazione

## TASKS - User Story 10:

### 1. Implementare l'uscita dall'account

- Creare l'interfaccia che permette all'utente di uscire dal proprio account
- Implementare il ricaricamento della pagina principale una volta eseguito il logout

### 2. Testare il funzionamento del logout

- Verificare che l'uscita dall'account viene effettuata correttamente
- Verificare che, una volta eseguito il logout, viene ricaricata la pagina principale
- Verificare che, una volta eseguito il logout, non è possibile accedere alle funzionalità riservate all'utente loggato

## User Story 11: Visualizzare i sondaggi

Come sondaggista voglio poter visualizzare i vari sondaggi con il nome ed il relativo stato di completamento, così da poter distinguere lo stato dei vari sondaggi e così sapere quali di questi sono stati completati, quali sono da finire, quali da modificare e quali da eliminare

### **Criteri di accettazione:**

- L'utente sondaggista può visualizzare il nome dei propri sondaggi
- L'utente sondaggista può visualizzare lo stato di completamento dei propri sondaggi

### **TASKS - User Story 11:**

#### **1. Implementare richiesta dei sondaggi al database**

- Creare l'API necessaria per richiedere al database la lista dei sondaggi appartenenti a un certo utente sondaggista
- Configurare l'applicazione per utilizzare l'API per richiedere i sondaggi appartenenti all'utente che è loggato

#### **2. Creare l'interfaccia di visualizzazione dei sondaggi**

- Creare l'interfaccia grafica che mostra all'utente la lista dei propri sondaggi, e mostri per ognuno il relativo nome e stato di completamento

#### **3. Testare il funzionamento della visualizzazione dei sondaggi**

- Verificare che l'interfaccia mostri tutti i sondaggi appartenenti all'utente sondaggista che è loggato
- Verificare che le informazioni sui sondaggi mostrate (nome e stato di completamento) siano corrette

## **User Story 12: Creare sondaggi**

Come sondaggista voglio poter creare nuovi sondaggi, così da poter cominciare a raccogliere voti

### **Criteri di accettazione:**

- L'utente sondaggista può creare sondaggi che vengono immagazzinati nel database
- L'utente sondaggista può inserire il nome di ogni sondaggio che crea
- Se l'utente sondaggista prova a creare un sondaggio senza nome, un messaggio di errore appropriato viene presentato

### **TASKS - User Story 12:**

#### **1. Implementare la creazione di sondaggi nel database**

- Creare l'API necessaria per creare un nuovo sondaggio e inserirlo nel database
- Configurare l'applicazione per utilizzare l'API per creare un sondaggio

#### **2. Creare il form per creare un sondaggio**

- Creare l'interfaccia che permette all'utente sondaggista di inserire il nome del sondaggio e di crearlo

#### **3. Creare i messaggi di errore**

- Creare il messaggio di errore che viene presentato se si cerca di creare un sondaggio senza nome

#### **4. Testare la creazione dei sondaggi**

- Verificare che l'utente sondaggista possa inserire il nome del sondaggio e crearlo attraverso il form
- Verificare che quando un utente sondaggista crea un sondaggio esso viene memorizzato nel database
- Verificare che il messaggio di errore giusto viene mostrato quando un utente sondaggista cerca di creare un sondaggio senza nome

## User Story 13: Salvare e continuare sondaggi

Come sondaggista voglio poter salvare i miei sondaggi e continuarli in un secondo momento, così da poter lavorare sui sondaggi quando voglio

### Criteri di accettazione:

- L'utente sondaggista può salvare un suo sondaggio in corso
- L'utente sondaggista può continuare un suo sondaggio salvato
- Se un sondaggio non è in corso, l'utente sondaggista non può più continuarlo

### TASKS - User Story 13:

#### 1. Creare l'interfaccia di gestione dei sondaggi

- Creare l'interfaccia che permette all'utente sondaggista di selezionare un sondaggio in corso e continuarlo
- Aggiungere all'interfaccia un pulsante che permette all'utente sondaggista di salvare il sondaggio
- Impostare l'interfaccia per essere accessibile solo per i sondaggi che sono ancora in corso

#### 2. Testare il salvataggio dei sondaggi

- Verificare che l'utente sondaggista possa salvare un sondaggio tramite il pulsante nell'interfaccia

#### 3. Testare la continuazione dei sondaggi

- Verificare che l'utente sondaggista possa continuare un sondaggio in corso, e che i dati del sondaggio non cambino da quando è stato salvato
- Verificare che l'utente sondaggista non possa continuare sondaggi che non sono più in corso

## User Story 14: Eliminare sondaggi

Come sondaggista voglio poter eliminare i miei sondaggi, così da poter rimuovere sondaggi sbagliati o creati per errore

### Criteri di accettazione:

- L'utente sondaggista può eliminare un suo sondaggio che è ancora in corso
- Se un sondaggio non è in corso, l'utente sondaggista non può più eliminarlo

### TASKS - User Story 14:

#### 1. Implementare l'eliminazione dei sondaggi dal database

- Creare l'API necessaria per eliminare un sondaggio dal database
- Configurare l'applicazione per utilizzare l'API per eliminare il sondaggio selezionato dall'utente sondaggista

#### 2. Creare il pulsante di eliminazione del sondaggio

- Aggiungere all'interfaccia di gestione del sondaggio un pulsante per eliminare il sondaggio

#### 3. Testare l'eliminazione dei sondaggi

- Verificare che il pulsante di eliminazione presente nell'interfaccia di gestione sondaggio funzioni correttamente
- Verificare che quando un sondaggio viene eliminato esso viene permanentemente rimosso dal database

## User Story 15: Completare sondaggi

Come sondaggista voglio poter completare i miei sondaggi in corso, così da poter aggiungere al sistema i dati che ho raccolto

### Criteri di accettazione:

- L'utente sondaggista può completare i suoi sondaggi in corso
- Una volta completati, i sondaggi non vengono più considerati in corso

### TASKS - User Story 15:

#### 1. Creare il pulsante di completamento del sondaggio

- Aggiungere all'interfaccia di gestione del sondaggio un pulsante per completare il sondaggio

#### 2. Testare il completamento dei sondaggi

- Verificare che il pulsante di completamento presente nell'interfaccia di gestione sondaggio funzioni correttamente
- Verificare che quando un sondaggio viene completato esso non sia più considerato in corso dal sistema

## User Story 16: Visualizzazione dei voti

Come sondaggista voglio poter visualizzare le statistiche parziali dei voti nei miei sondaggi in corso, così da poter monitorare l'andamento del sondaggio e lo svolgimento corretto dei voti

### Criteri di accettazione:

- L'utente sondaggista può vedere la lista dei voti nei suoi sondaggi in corso
- L'utente sondaggista può vedere il numero totale di voti nei suoi sondaggi, l'età media parziale dei voti nei suoi sondaggi, e il numero di voti nei suoi sondaggi divisi per quartiere

### TASKS - User Story 16:

#### 1. Aggiungere la lista dei voti all'interfaccia

- Aggiungere all'interfaccia di gestione del sondaggio la lista dei voti presenti nel sondaggio

#### 2. Implementare la logica delle statistiche parziali

- Implementare la logica di calcolo dell'età media dei voti presenti nel sondaggio, tenendo conto dei voti senza età
- Implementare la logica che suddivide i voti presenti nel sondaggio in base al loro quartiere e produce il numero totale dei voti per ogni quartiere

#### 3. Aggiungere le statistiche parziali all'interfaccia

- Aggiungere all'interfaccia di gestione del sondaggio il numero totale dei voti nel sondaggio
- Aggiungere all'interfaccia di gestione del sondaggio l'età media parziale dei voti nel sondaggio
- Aggiungere all'interfaccia di gestione del sondaggio il numero totale di voti per ogni quartiere

#### 4. Testare la visualizzazione dei voti

- Verificare che la lista dei voti, il numero totale dei voti, l'età media parziale, e il numero di voti totale per ogni quartiere vengano mostrati nell'interfaccia di gestione del sondaggio
- Verificare che la lista dei voti contenga tutti i voti presenti nel sondaggio
- Verificare che i valori del numero totale di voti, dell'età media parziale, e del numero di voti totale per ogni quartiere siano corretti

## User Story 17: Aggiungere voti

Come sondaggista voglio poter aggiungere un voto a un mio sondaggio in corso, così da permettere al cittadino di esprimere il proprio grado di soddisfazione rispetto alle attività del comune per la zona nella quale abita e per la sua città

### Criteri di accettazione:

- L'utente sondaggista può aggiungere il voto di un cittadino a un suo sondaggio in corso inserendo il quartiere del cittadino che sta votando e, opzionalmente, l'età.
- Il cittadino può compilare il form di voto per esprimere il suo grado di soddisfazione
- I voti aggiunti ai sondaggi vengono memorizzati nel database
- Se il sondaggista prova ad aggiungere un voto senza inserire il quartiere del cittadino, un messaggio di errore appropriato viene presentato

### TASKS - User Story 17:

#### 1. Implementare l'aggiunta di voti al database

- Creare l'API necessaria per creare un nuovo voto, aggiungerlo ad un sondaggio esistente, e inserirlo nel database
- Configurare l'applicazione per utilizzare l'API per aggiungere il voto creato al sondaggio in corso

#### 2. Aggiungere il form di creazione di un voto

- Creare l'interfaccia che permette all'utente sondaggista di inserire il quartiere del cittadino che sta votando, l'età del cittadino (opzionalmente), e di iniziare il processo di aggiunta del voto

#### 3. Aggiungere il form di voto

- Creare l'interfaccia che permette all'utente di selezionare il suo grado di soddisfazione e confermare il suo voto

#### 4. Creare i messaggi di errore

- Creare il messaggio di errore che viene presentato quando il sondaggista prova ad aggiungere un voto senza inserire il quartiere di provenienza del cittadino

#### 5. Testare l'aggiunta di nuovi voti

- Verificare che l'interfaccia di creazione di un voto e il form di voto funzionino correttamente
- Verificare che sia possibile aggiungere un voto sia inserendo il quartiere e l'età del cittadino sia inserendo solo il quartiere
- Verificare che, se non si inserisce il quartiere, il messaggio di errore giusto viene presentato
- Verificare che, una volta aggiunto un voto a un sondaggio, esso viene memorizzato nel database

## User Story 18: Eliminare i voti

Come sondaggista voglio poter eliminare i voti di un sondaggio, così da poter correggere eventuali voti errati

### Criteri di accettazione:

- L'utente sondaggista può eliminare un voto di un suo sondaggio in corso

## TASKS - User Story 18:

### 1. Implementare l'eliminazione dei voti dal database

- Creare l'API necessaria per eliminare un voto dal database
- Configurare l'applicazione per utilizzare l'API per eliminare il voto selezionato dall'utente

### 2. Creare il pulsante di eliminazione dei voti

- Aggiungere a ogni voto nella lista di voti nell'interfaccia di gestione del sondaggio un pulsante per eliminare quel voto

### 3. Testare l'eliminazione dei voti

- Verificare che il pulsante di eliminazione presente nella lista dei voti funzioni correttamente
- Verificare che quando un voto viene eliminato esso viene permanentemente rimosso dal sistema

## User Story 19: Spostare il focus della tabella

Come analista voglio poter modificare il focus principale della tabella, così da poter visualizzare i dati appartenenti a zone non visibili in tabella

### Criteri di accettazione:

- L'utente analista può spostare il focus della tabella in alto e in basso

## TASKS - User Story 19:

### 1. Implementare lo spostamento della tabella

- Implementare la possibilità di spostare in alto o in basso il focus della tabella
- Modificare la posizione del focus della tabella in base all'input dell'utente

### 2. Testare lo spostamento del focus della tabella

- Verificare che il focus della tabella si sposti in alto e in basso con l'input dell'utente
- Verificare che il focus della tabella si sposti nella direzione corretta

## User Story 20: Interagire con la tabella

Come analista voglio poter interagire con la tabella, così da poter selezionare la zona che mi interessa

### Criteri di accettazione:

- L'utente analista può interagire con una riga della tabella e selezionare la relativa zona

## TASKS - User Story 20:

### 1. Implementare l'interazione della tabella

- Configurare l'interfaccia della tabella per selezionare una zona quando l'utente analista interagisce con la relativa riga della tabella

### 2. Testare l'interazione con la tabella

- Verificare che l'utente possa interagire con le righe della tabella per selezionare la relativa zona

## User Story 21: Visualizzare tutti gli attributi

Come analista voglio poter visualizzare un maggior numero di attributi riguardanti una data zona geografica, così da poter comprendere a fondo le motivazioni di un corrispettivo livello di soddisfazione

**Criteri di accettazione:**

- L'utente analista può vedere l'insieme completo di tutti gli attributi della zona selezionata

**TASKS - User Story 21:****1. Implementare la richiesta dei dati completi al database**

- Creare l'API necessaria per richiedere al database i dati di tutti gli attributi di una zona
- Configurare l'applicazione per richiedere i dati di tutti gli attributi quando l'utente analista seleziona una zona

**2. Testare la visualizzazione di tutti gli attributi**

- Verificare che tutti gli attributi, e non solo quelli demografici e riguardanti la soddisfazione media, vengono mostrati all'utente analista
- Verificare che i valori degli attributi mostrati siano corretti
- Verificare che i valori degli attributi siano collocati nel posto giusto sullo schermo

**User Story 22: Scegliere la categoria di attributi**

Come analista, quando sto visualizzando gli attributi di una zona, voglio vedere gli attributi suddivisi per categoria e selezionare quale categoria di attributi visualizzare, così da avere una visione chiara e organizzata dei dati

**Criteri di accettazione:**

- L'utente analista può selezionare la categoria di attributi che vuole visualizzare

**TASKS - User Story 22:****1. Suddividere gli attributi in categorie**

- Identificare le categorie di attributi
- Assegnare ogni attributo a una categoria

**2. Implementare la selezione della categoria di attributi**

- Creare l'interfaccia che permette all'utente analista di selezionare una delle categorie di attributi
- Configurare l'interfaccia di visualizzazione degli attributi per mostrare gli attributi appartenenti alla categoria selezionata dall'utente analista

**3. Testare la selezione della categoria di attributi**

- Verificare che l'utente analista possa selezionare le categorie di attributi
- Verificare che l'interfaccia di visualizzazione degli attributi mostri gli attributi della categoria selezionata in modo corretto



## 2 User Workflow

Nella seguente parte del documento verrà illustrato lo “*user flow*” per gli utenti del tipo: “non loggato”, “analista” e “sondaggista”.

**Utenti “non loggati”** Come possiamo osservare dalla figura 2.2, gli utenti non loggati possono visualizzare la *HomePage* della applicazione web, da questa possono visualizzare i dati generici della città. In alternativa alla pressione di una zona all’interno della mappa il sistema restituirà la pagina di dettaglio della zona selezionata dalla quale possono o tornare alla *HomePage* o visualizzare i dettagli. Dalla *HomePage* è possibile inoltre premere sul pulsante per il passaggio da “quartieri” a “circoscrizioni” e viceversa. La mappa presente nella *HomePage* è interattiva e permette di spostare la visuale e di ingrandire o rimpicciolire la mappa.

**Utenti “loggati”** Gli utenti loggati tramite un pulsante dedicato nella *HomePage* possono accedere alla pagina di *login* e, inserendo le proprie credenziali, se queste sono corrette allora in base al tipo di utente questo verrà reindirizzato alla propria pagina associata. Nel caso dell’utente “analista” verrà reindirizzato alla *HomePage* con le funzionalità di analisi, mentre l’utente “sondaggista” verrà reindirizzato alla pagina di gestione dei sondaggi. In entrambi i casi l’utente potrà effettuare il *logout* tramite un apposito menù a tendina sul quale è presente il proprio nome.

**Utenti “analisti”** Gli utenti di tipo “analista” possono visualizzare la *HomePage* con le funzionalità di analisi, in particolare possono visualizzare i dati generici della città e, in alternativa alla pressione di una zona all’interno della mappa, il sistema restituirà la pagina di dettaglio della zona selezionata dalla quale possono o tornare alla *HomePage* o visualizzare i dettagli. Oltre alla possibilità di cambio di visualizzazione da “quartieri” a “circoscrizioni” e viceversa, l’analista può visualizzare i dati in modalità “tabella” premendo sul pulsante “mappa”/“tabella”. A differenza di tutti gli altri utenti alla pressione di una zona all’interno della mappa verrà visualizzata la pagina di dettaglio della zona selezionata per gli analisti, da questa possiamo o tornare alla *HomePage* o selezionando la categoria degli attributi visualizzare i dati della zona selezionata.

**Utenti “sondaggista”** Gli utenti di tipo “sondaggista” dalla pagina di gestione dei sondaggi possono visualizzare lo storico dei sondaggi oppure crearne uno inserendo il titolo di questo e premendo sul pulsante “Crea un nuovo sondaggio”, è anche possibile caricare un sondaggio precedentemente creato selezionando il file e premendo sul pulsante “Carica sondaggio”. Dalla pagina di gestione dei sondaggi è visualizzata una tabella con i sondaggi creati dall’utente, per ogni sondaggio non ancora “completato” è possibile continuare la compilazione premendo sopra questo, si apre dunque la pagina di gestione del sondaggio. Da questa pagina è possibile da una serie di pulsanti terminare il sondaggio e caricarlo per l’approvazione degli amministratori (col pulsante “Termina e Carica il Sondaggio”), premere sul pulsante “Salva ed Esci” per salvare i cambiamenti fatti al sondaggio e tornare alla pagina dei sondaggi, oppure premere sul pulsante “Elimina Sondaggio” e dopo una conferma il sondaggio verrà eliminato, anche in questo caso si verrà reindirizzati alla pagina dei sondaggi. Oltre a queste azioni è possibile eseguire le azioni di voto, questo compilando i dati del cittadino che sta per votare e premendo sul pulsante “Vota”, al termine delle operazioni si verrà reindirizzati alla pagina di gestione del sondaggio. Se sono stati commessi errori durante la compilazione del sondaggio è possibile cancellare un voto individuale.



Figura 2.1: User Workflow Legend

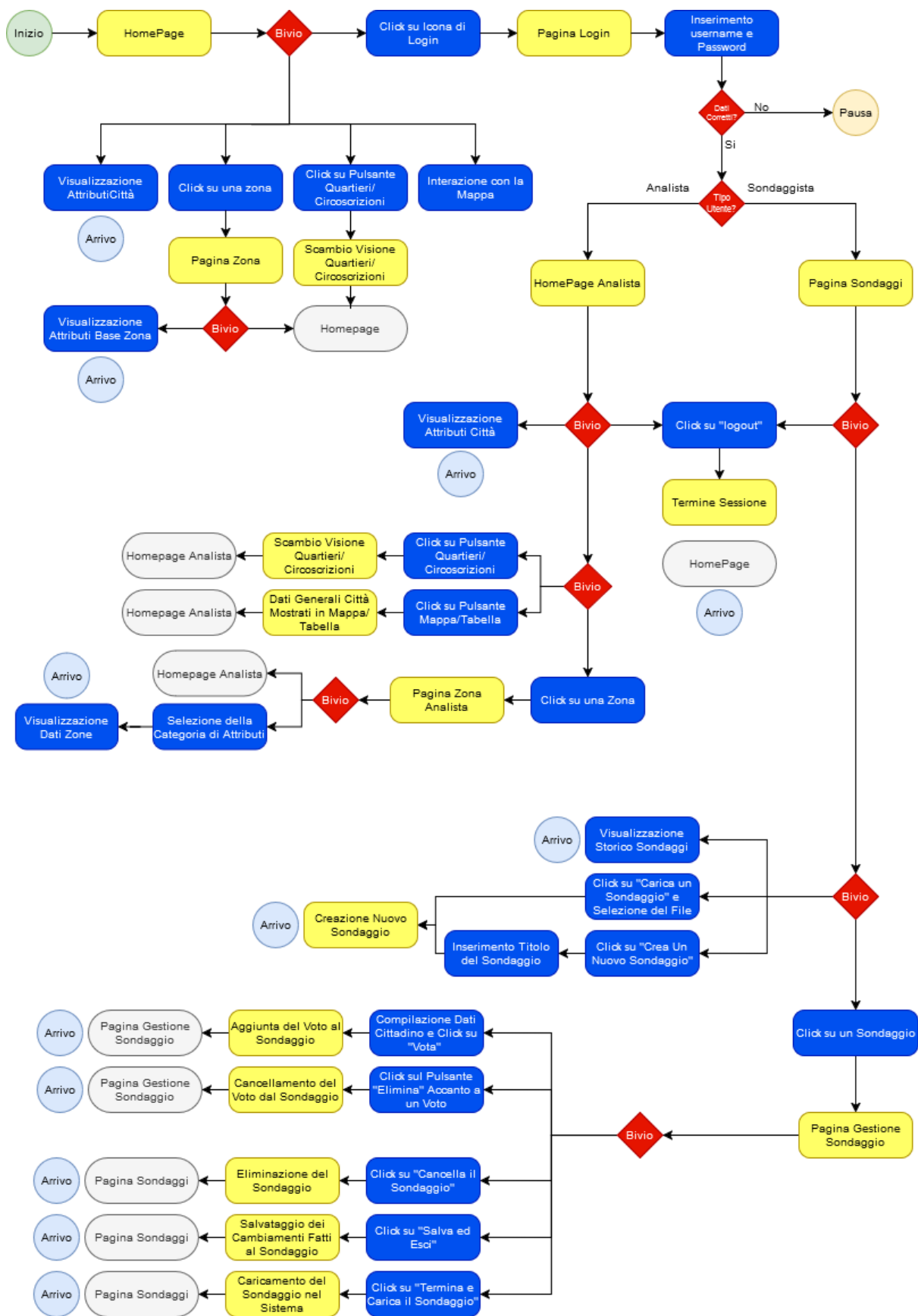


Figura 2.2: User Workflow

## 3 Web APIs

Le specifiche delle APIs sono state definite secondo le specifiche OpenAPI 3.0.3 e disponibili pubblicamente presso APIary al seguente indirizzo: <https://satistrento.docs.apiary.io>.

**Progettazione e note di design** In fase di progettazione si è scelto di ridurre al minimo la duplicazione degli attributi e delle relazioni tra le entità, in modo da rendere più semplice la manutenzione e l'aggiornamento delle APIs in futuro, dunque è presente un elevato numero di componenti ed ereditarietà tra questi. Inoltre si è seguito il paradigma degli "API link" per permettere una navigazione più semplice e intuitiva tra le risorse.

**Specifiche** Le specifiche delle APIs sono disponibili al seguente indirizzo: <https://github.com/lucafano04/progettoComune/blob/main/APIdoc/api.yaml> e sono riportate di seguito.

```
1 openapi: 3.0.3
2 info:
3   title: Documentazione API progetto "SatisTrento"
4   description: Documentazione API per il progetto "SatisTrento" all'interno del progetto "100 progetti per
   ↳ il comune di Trento" per il corso di Ingegneria del Software a.a. 2024/2025 corso di laurea in
   ↳ Informatica presso l'Università degli Studi di Trento.
5   version: 0.0.1
6 servers:
7   - url: http://localhost:3000/api/v1
8     description: Development server for testing
9   - url: https://progettocomune.onrender.com/api/v1
10    description: Production server
11
12 components:
13   schemas:
14     DatiBase:
15       description: Oggetto contenente le informazioni di base di un quartiere.
16       type: object
17       properties:
18         popolazione:
19           type: integer
20           description: Numero di abitanti del quartiere.
21         superficie:
22           type: number
23           description: Superficie del quartiere in km^2.
24           format: double
25         serviziTotali:
26           type: integer
27           description: Numero totale di servizi presenti nel quartiere.
28         soddisfazioneMedia:
29           type: number
30           description: Soddisfazione media degli abitanti del quartiere.
31           format: double
32           minimum: 0
33           maximum: 100
34         interventiPolizia:
35           type: integer
36           description: Numero di interventi delle forze dell'ordine nel quartiere.
37         etaMedia:
38           type: number
39           description: Età media degli abitanti del quartiere.
40           format: double
41         ServiziGenerali:
42           description: Oggetto contenente le informazioni sui servizi presenti in un quartiere.
43           type: object
44           properties:
45             areeVerdi:
46               type: integer
47               description: Numero di aree verdi presenti nel quartiere.
48               minimum: 0
49             scuole:
50               type: integer
51               description: Numero di scuole presenti nel quartiere.
```

```

52         minimum: 0
53     serviziRistorazione:
54         type: integer
55         description: Numero di servizi di ristorazione presenti nel quartiere.
56         minimum: 0
57     localiNotturni:
58         type: integer
59         description: Numero di locali notturni presenti nel quartiere.
60         minimum: 0
61 Sicurezza:
62     description: Oggetto contenente le informazioni sulla sicurezza di un quartiere.
63     type: object
64     properties:
65         numeroInterventi:
66             type: integer
67             description: Numero di interventi delle forze dell'ordine nel quartiere.
68             minimum: 0
69         incidenti:
70             type: integer
71             description: Numero di incidenti nel quartiere.
72             minimum: 0
73         tassoCriminalita:
74             type: number
75             description: Tasso di criminalità del quartiere.
76             format: double
77             minimum: 0
78 Coordinate:
79     description: Array di coordinate che definiscono un quartiere e/o circoscrizione.
80     type: array
81     items:
82         description: Coordinate nel formato [latitude, longitude].
83         type: array
84         items:
85             type: number
86             format: double
87         minItems: 2
88         maxItems: 2
89     minItems: 1
90     uniqueItems: true
91 Soddisfazione:
92     type: "number"
93     description: "Soddisfazione media degli abitanti del quartiere."
94     format: "double"
95     minimum: 0
96     maximum: 100
97 CircoscrizioneBaseNC:
98     description: Oggetto contenente le informazioni di Base di una circoscrizione. (Senza coordinate)
99     type: object
100     properties:
101         self:
102             type: string
103             description: Identificatore univoco della circoscrizione.
104         nome:
105             type: string
106             description: Nome della circoscrizione.
107         soddisfazioneMedia:
108             "$ref": "#/components/schemas/Soddisfazione"
109 CircoscrizioneBase:
110     description: Oggetto contenente le informazioni di Base di una circoscrizione.
111     type: object
112     allOf:
113         - "$ref": "#/components/schemas/CircoscrizioneBaseNC"
114     properties:
115         coordinate:
116             "$ref": "#/components/schemas/Coordinate"
117 Circoscrizione:
118     description: Oggetto contenente le informazioni di una circoscrizione.
119     allOf:
120         - "$ref": "#/components/schemas/CircoscrizioneBase"
121         - "$ref": "#/components/schemas/DatiBase"
122     type: object
123     properties:
124         servizi:

```

```
125     "$ref": "#/components/schemas/ServiziGenerali"
126     sicurezza:
127     "$ref": "#/components/schemas/Sicurezza"
128 CircoScrizioneNC:
129     description: Oggetto contenente le informazioni di una circoscrizione. (Senza coordinate)
130     allOf:
131     - "$ref": "#/components/schemas/CircoScrizioneBaseNC"
132     - "$ref": "#/components/schemas/DatiBase"
133     type: object
134     properties:
135     servizi:
136     "$ref": "#/components/schemas/ServiziGenerali"
137     sicurezza:
138     "$ref": "#/components/schemas/Sicurezza"
139 QuartiereBaseNC:
140     description: Oggetto contenente le informazioni di base di un quartiere. (Senza coordinate)
141     type: object
142     properties:
143     self:
144     type: string
145     description: "Url della risorsa."
146     nome:
147     type: "string"
148     description: "Nome del quartiere."
149     circoscrizione:
150     "$ref": "#/components/schemas/CircoScrizioneBaseNC"
151     soddisfazioneMedia:
152     "$ref": "#/components/schemas/Soddisfazione"
153 QuartiereBase:
154     description: Oggetto contenente le informazioni di base di un quartiere.
155     type: object
156     allOf:
157     - "$ref": "#/components/schemas/QuartiereBaseNC"
158     properties:
159     circoscrizione:
160     "$ref": "#/components/schemas/CircoScrizioneBase"
161     coordinate:
162     "$ref": "#/components/schemas/Coordinate"
163 Quartiere:
164     description: Oggetto contenente le informazioni di un quartiere.
165     allOf:
166     - "$ref": "#/components/schemas/QuartiereBase"
167     - "$ref": "#/components/schemas/DatiBase"
168     type: object
169     properties:
170     servizi:
171     "$ref": "#/components/schemas/ServiziGenerali"
172     sicurezza:
173     "$ref": "#/components/schemas/Sicurezza"
174 QuartiereNC:
175     description: Oggetto contenente le informazioni di un quartiere.
176     allOf:
177     - "$ref": "#/components/schemas/QuartiereBaseNC"
178     - "$ref": "#/components/schemas/DatiBase"
179     type: object
180     properties:
181     servizi:
182     "$ref": "#/components/schemas/ServiziGenerali"
183     sicurezza:
184     "$ref": "#/components/schemas/Sicurezza"
185 User:
186     description: Oggetto contenente le informazioni di un utente.
187     type: object
188     properties:
189     self:
190     type: string
191     description: Url della risorsa.
192     email:
193     type: string
194     description: Email dell'utente.
195     nome:
196     type: string
197     description: Nome dell'utente.
```

```

198     cognome:
199         type: string
200         description: Cognome dell'utente.
201     ruolo:
202         type: string
203         enum:
204             - "amministratore"
205             - "analista"
206             - "sondaggista"
207             - "circoscrizione"
208         description: Ruolo dell'utente.
209     imageUrl:
210         type: string
211         description: Url dell'immagine profilo dell'utente. (gravatar)
212 Session:
213     description: Oggetto contenente le informazioni di una sessione.
214     type: object
215     properties:
216         token:
217             type: string
218             description: Token di autenticazione dell'utente.
219         user:
220             "$ref": "#/components/schemas/User"
221 Error:
222     description: Oggetto contenente le informazioni di un errore.
223     type: object
224     properties:
225         code:
226             type: integer
227             description: Codice dell'errore.
228         message:
229             type: string
230             description: Messaggio di errore.
231         details:
232             type: string
233             description: Dettagli dell'errore.
234         nullable: true
235 AddSondaggio:
236     description: Oggetto contenente le informazioni per aggiungere un sondaggio.
237     type: object
238     properties:
239         titolo:
240             type: string
241             description: Titolo del sondaggio.
242 SondaggioBase:
243     description: Oggetto contenente le informazioni di base di un sondaggio.
244     type: object
245     allOf:
246         - $ref: "#/components/schemas/AddSondaggio"
247     properties:
248         self:
249             type: string
250             description: URL della risorsa.
251         dataInizio:
252             type: string
253             description: Data di inizio del sondaggio.
254         format: date
255         isAperto:
256             type: boolean
257             description: Indica se il sondaggio è aperto (in corso) o chiuso.
258         statoApprovazione:
259             type: string
260             enum:
261                 - "approvato"
262                 - "in attesa"
263                 - "rifiutato"
264             description: Stato di approvazione del sondaggio.
265         sondaggista:
266             "$ref": "#/components/schemas/User"
267 ElencoSondaggi:
268     description: Array di oggetti contenenti le informazioni di base di un sondaggio (visualizzazione
269     ↳ sondaggista/admin).
270     oneOf:

```

```
270     - type: array
271       items:
272         "$ref": "#/components/schemas/SondaggioBase"
273     - type: array
274       items:
275         "$ref": "#/components/schemas/Sondaggio"
276 AddVoto:
277   description: Oggetto contenente le informazioni per aggiungere un voto.
278   type: object
279   properties:
280     eta:
281       type: integer
282       description: Età dell'utente che ha effettuato il voto. Può essere null se non specificata.
283       minimum: 0
284       nullable: true
285     voto:
286       type: integer
287       description: Voto assegnato al quartiere.
288       minimum: 1
289       maximum: 5
290     quartiere:
291       type: string
292       description: Identificatore univoco del quartiere.
293 Voto:
294   description: Oggetto contenente le informazioni di un voto.
295   type: object
296   properties:
297     self:
298       type: string
299       description: Identificatore univoco del voto.
300     dataOra:
301       type: string
302       description: Data e ora in cui è stato effettuato il voto.
303       format: date-time
304   allOf:
305     - "$ref": "#/components/schemas/AddVoto"
306 MediaVoti:
307   description: Oggetto semplice contenente la media dei voti di un quartiere.
308   type: object
309   properties:
310     media:
311       type: number
312       description: Media dei voti assegnati al quartiere.
313       format: double
314       minimum: 1
315       maximum: 5
316     quartiere:
317       type: string
318       description: Identificatore univoco della risorsa del quartiere.
319 Sondaggio:
320   description: Oggetto contenente le informazioni di un sondaggio.
321   allOf:
322     - "$ref": "#/components/schemas/SondaggioBase"
323   type: object
324   properties:
325     voti:
326       type: array
327       items:
328         "$ref": "#/components/schemas/Voto"
329     description: Array di oggetti contenenti i voti assegnati al sondaggio.
330     mediaVoti:
331       type: array
332       items:
333         "$ref": "#/components/schemas/MediaVoti"
334     description: Array di oggetti contenenti la media dei voti assegnati ai quartieri presenti nel
335       ↳ sondaggio (quelli non presenti non vengono visualizzati).
336 securitySchemes:
337   BearerAuth:
338     type: http
339     scheme: bearer
340     bearerFormat: JWT
341     description: Token di autenticazione fornito all'utente per accedere alle risorse specifiche
342     ↳ protette.
```

```
341
342 paths:
343   '/quartieri' :
344     description: "Risorsa che restituisce la lista dei quartieri di Trento."
345     get:
346       summary: "Ottieni lista dei quartieri"
347       description: "Ritorna una lista di tutti i quartieri della città di Trento."
348       parameters:
349         - name: deepData
350           in: query
351           description: "Se impostato a true, restituisce le informazioni complete di ogni quartiere."
352           required: false
353           schema:
354             type: boolean
355             default: false
356             allowEmptyValue: true
357         - name: coordinate
358           in: query
359           description: "Se impostato non impostato o impostato a true restituisce le coordinate di ogni
360             ↳ quartiere, altrimenti no."
361           required: false
362           schema:
363             type: boolean
364             default: true
365             allowEmptyValue: true
366       responses:
367         200:
368           description: "Risposta con la lista dei quartieri."
369           content:
370             application/json:
371               schema:
372                 oneOf:
373                   - type: array
374                     items:
375                       "$ref": "#/components/schemas/QuartiereBase"
376                   - type: array
377                     items:
378                       "$ref": "#/components/schemas/Quartiere"
379                   - type: array
380                     items:
381                       "$ref": "#/components/schemas/QuartiereNC"
382                   - type: array
383                     items:
384                       "$ref": "#/components/schemas/QuartiereBaseNC"
385       tags:
386         - Quartieri
387         - Dati Statici
388   '/quartieri/{id}' :
389     description: "Risorsa che restituisce le informazioni di un quartiere di Trento con le informazioni di
390       ↳ base e i servizi presenti."
391     parameters:
392       - name: id
393         in: path
394         description: "Identificatore univoco del quartiere."
395         required: true
396         schema:
397           type: string
398       - name: coordinate
399         in: query
400         description: "Se impostato a true, restituisce le coordinate del quartiere, altrimenti no.
401           ↳ Impostato di default a true."
402         required: false
403         schema:
404           type: boolean
405           default: true
406           allowEmptyValue: true
407     get:
408       summary: "Ottieni informazioni di un quartiere"
409       responses:
410         200:
411           description: "Risposta con le informazioni del quartiere."
412           content:
413             application/json:
```



```
411         schema:
412         oneOf:
413             - $ref: "#/components/schemas/Quartiere"
414             - $ref: "#/components/schemas/QuartiereNC"
415     400:
416         description: "ID proposto non valido."
417         content:
418             application/json:
419                 schema:
420                     "$ref": "#/components/schemas/Error"
421     404:
422         description: "Quartiere non trovato."
423         content:
424             application/json:
425                 schema:
426                     "$ref": "#/components/schemas/Error"
427     tags:
428         - Quartieri
429         - Dati Statici
430     '/circoscrizioni' :
431         description: "Risorsa che restituisce la lista delle circoscrizioni di Trento."
432         get:
433             summary: "Ottieni lista delle circoscrizioni"
434             description: "Returns a list of all the districts in the city of Trento."
435             parameters:
436                 - name: deepData
437                   in: query
438                   description: "Se impostato a true, restituisce le informazioni complete di ogni circoscrizione."
439                   required: false
440                   schema:
441                       type: boolean
442                       default: false
443                   allowEmptyValue: true
444                 - name: coordinate
445                   in: query
446                   description: "Se impostato non impostato o impostato a true restituisce le coordinate di ogni  
↳ circoscrizione, altrimenti no."
447                   required: false
448                   schema:
449                       type: boolean
450                       default: true
451                   allowEmptyValue: true
452             responses:
453                 200 :
454                     description: "Successful response"
455                     content:
456                         application/json:
457                             schema:
458                                 oneOf:
459                                     - type: "array"
460                                       items:
461                                           "$ref": "#/components/schemas/CircoscrizioneBase"
462                                     - type: "array"
463                                       items:
464                                           "$ref": "#/components/schemas/Circoscrizione"
465                                     - type: "array"
466                                       items:
467                                           "$ref": "#/components/schemas/CircoscrizioneBaseNC"
468                                     - type: "array"
469                                       items:
470                                           "$ref": "#/components/schemas/CircoscrizioneNC"
471             tags:
472                 - Circoscrizioni
473                 - Dati Statici
474     '/circoscrizioni/{id}' :
475         description: "Risorsa che restituisce le informazioni di una circoscrizione di Trento con le  
↳ informazioni di base e i servizi presenti."
476         parameters:
477             - name: id
478               in: path
479               description: "Identificatore univoco della circoscrizione."
480               required: true
481             schema:
```

```

482     type: string
483 - name: coordinate
484   in: query
485   description: "Se impostato a true, restituisce le coordinate della circoscrizione, altrimenti no.
486     ↳ Impostato di default a true."
487   required: false
488   schema:
489     type: boolean
490     default: true
491     allowEmptyValue: true
492 get:
493   summary: "Ottieni informazioni di una circoscrizione"
494   responses:
495     200:
496       description: "Risposta con le informazioni della circoscrizione."
497       content:
498         application/json:
499           schema:
500             oneOf:
501               - $ref: "#/components/schemas/Circoscrizione"
502               - $ref: "#/components/schemas/CircoscrizioneNC"
503     400:
504       description: "ID proposto non valido."
505       content:
506         application/json:
507           schema:
508             "$ref": "#/components/schemas/Error"
509     404:
510       description: "Circoscrizione non trovata."
511       content:
512         application/json:
513           schema:
514             "$ref": "#/components/schemas/Error"
515   tags:
516     - Circoscrizioni
517     - Dati Statici
518 '/generalInfo' :
519   description: "Risorsa che restituisce le informazioni generali della città di Trento."
520   get:
521     summary: "Ottieni informazioni generali"
522     description: "Restituisce le informazioni generali della città di Trento."
523     responses:
524       200 :
525         description: "Successful response"
526         content:
527           application/json:
528             schema:
529               type: "object"
530               properties:
531                 self:
532                   type: "string"
533                   description: "Url della risorsa."
534                 popolazione :
535                   type: integer
536                   description: "Numero di abitanti della città di Trento."
537                 superficie :
538                   type: number
539                   description: "Superficie della città di Trento in km^2."
540                   format: double
541                 etaMedia :
542                   type: number
543                   description: "Età media degli abitanti della città di Trento."
544                   format: double
545                 soddisfazioneMedia:
546                   type: number
547                   description: "Soddisfazione media degli abitanti della città di Trento."
548                   format: double
549                   minimum: 0
550                   maximum: 100
551   tags:
552     - Dati Statici
553 '/session':
554   description: "Risorsa che restituisce le informazioni della sessione dell'utente."

```

```
554 get:
555   summary: "Ottieni informazioni della sessione"
556   description: "Restituisce le informazioni della sessione dell'utente, se presente."
557   security:
558     - BearerAuth: []
559   responses:
560     200:
561       description: "Successful response"
562       content:
563         application/json:
564           schema:
565             "$ref": "#/components/schemas/User"
566     400:
567       description: "Sessione non valida."
568       content:
569         application/json:
570           schema:
571             "$ref": "#/components/schemas/Error"
572     401:
573       description: "Utente non autenticato."
574       content:
575         application/json:
576           schema:
577             "$ref": "#/components/schemas/Error"
578     404:
579       description: "Sessione non trovata."
580       content:
581         application/json:
582           schema:
583             "$ref": "#/components/schemas/Error"
584   tags:
585     - Utenti
586 post:
587   summary: "Crea una nuova sessione (aka login)"
588   description: "Crea una nuova sessione per l'utente. Altrimenti conosciuto come login."
589   requestBody:
590     required: true
591     content:
592       application/json:
593         schema:
594           type: object
595           properties:
596             email:
597               type: string
598               description: "Email dell'utente."
599             password:
600               type: string
601               description: "Password (codificata) dell'utente."
602   responses:
603     201:
604       description: "Sessione creata con successo."
605       content:
606         application/json:
607           schema:
608             "$ref": "#/components/schemas/Session"
609       headers:
610         Location:
611           description: "Url della risorsa creata."
612           schema:
613             type: string
614     400:
615       description: "Dati della sessione non validi."
616       content:
617         application/json:
618           schema:
619             "$ref": "#/components/schemas/Error"
620     401:
621       description: "Credenziali non valide."
622       content:
623         application/json:
624           schema:
625             "$ref": "#/components/schemas/Error"
626   tags:
```

```

627     - Utenti
628 delete:
629     summary: "Elimina la sessione (aka logout)"
630     description: "Elimina la sessione dell'utente revocando il token di autenticazione al server,
        ↳ altrimenti conosciuto come logout."
631 security:
632     - BearerAuth: []
633 responses:
634     204:
635         description: "Sessione eliminata con successo."
636     401:
637         description: "Sessione non trovata."
638         content:
639             application/json:
640                 schema:
641                     "$ref": "#/components/schemas/Error"
642 tags:
643     - Utenti
644 '/sondaggi' :
645     description: "Risorsa che restituisce la lista dei sondaggi dell'utente corrente, se amministratore
        ↳ allora restituisce tutti i sondaggi."
646 get:
647     summary: "Ottieni lista dei sondaggi"
648     description: "Restituisce la lista dei sondaggi dell'utente corrente. Se l'utente è amministratore,
        ↳ restituisce tutti i sondaggi."
649 security:
650     - BearerAuth: []
651 parameters:
652     - name: deepData
653       in: query
654       description: "Se impostato a true, restituisce le informazioni complete di ogni sondaggio."
655       required: false
656       schema:
657         type: boolean
658         default: false
659       allowEmptyValue: true
660 responses:
661     200:
662         description: "Successful response"
663         content:
664             application/json:
665                 schema:
666                     "$ref": "#/components/schemas/ElencoSondaggi"
667     401:
668         description: "Utente non autenticato."
669         content:
670             application/json:
671                 schema:
672                     "$ref": "#/components/schemas/Error"
673     403:
674         description: "Utente loggato ma non ha i permessi per visualizzare i sondaggi."
675         content:
676             application/json:
677                 schema:
678                     "$ref": "#/components/schemas/Error"
679 tags:
680     - Sondaggi
681     - Sondaggista
682 post:
683     summary: "Crea un nuovo sondaggio"
684     description: "Crea un nuovo sondaggio. Solo per utenti con ruolo 'sondaggista'."
685 security:
686     - BearerAuth: []
687 requestBody:
688     required: true
689     content:
690         application/json:
691             schema:
692                 "$ref": "#/components/schemas/AddSondaggio"
693 responses:
694     201:
695         description: "Sondaggio creato con successo."
696         headers:

```

```
697         Location:
698             description: "Url della risorsa creata."
699             schema:
700                 type: string
701     400:
702         description: "Dati del sondaggio non validi."
703         content:
704             application/json:
705                 schema:
706                     "$ref": "#/components/schemas/Error"
707     401:
708         description: "Utente non autenticato."
709         content:
710             application/json:
711                 schema:
712                     "$ref": "#/components/schemas/Error"
713     403:
714         description: "Utente loggato ma non ha i permessi per creare un sondaggio."
715         content:
716             application/json:
717                 schema:
718                     "$ref": "#/components/schemas/Error"
719     tags:
720         - Sondaggi
721         - Sondaggista
722     '/sondaggi/{id}' :
723         description: "Risorsa che restituisce le informazioni di un sondaggio."
724         parameters:
725             - name: id
726               in: path
727               description: "Identificatore univoco del sondaggio."
728               required: true
729               schema:
730                   type: string
731     get:
732         summary: "Ottieni informazioni di un sondaggio"
733         description: "Restituisce le informazioni di un sondaggio, solo per utenti con ruolo 'sondaggista' o
734 ↪ 'amministratore'."
735         security:
736             - BearerAuth: []
737     responses:
738         200:
739             description: "Successful response"
740             content:
741                 application/json:
742                     schema:
743                         "$ref": "#/components/schemas/Sondaggio"
744         400:
745             description: "ID proposto non valido."
746             content:
747                 application/json:
748                     schema:
749                         "$ref": "#/components/schemas/Error"
750         401:
751             description: "Utente non autenticato."
752             content:
753                 application/json:
754                     schema:
755                         "$ref": "#/components/schemas/Error"
756         403:
757             description: "Utente loggato ma non ha i permessi per visualizzare il sondaggio."
758             content:
759                 application/json:
760                     schema:
761                         "$ref": "#/components/schemas/Error"
762         404:
763             description: "Sondaggio non trovato."
764             content:
765                 application/json:
766                     schema:
767                         "$ref": "#/components/schemas/Error"
768     tags:
769         - Sondaggi
```

```

769     - Sondaggista
770 patch:
771     summary: "Modifica un sondaggio"
772     description: "Modifica un sondaggio. Solo per utenti con ruolo 'sondaggista'."
773     security:
774     - BearerAuth: []
775
776     requestBody:
777         required: true
778         content:
779             application/json:
780                 schema:
781                     type: object
782                     properties:
783                         isAperto:
784                             type: boolean
785                             description: "Indica se il sondaggio è aperto (in corso) o chiuso."
786
787     responses:
788         200:
789             description: "Sondaggio modificato con successo."
790             headers:
791                 Location:
792                     description: "Url della risorsa modificata."
793                     schema:
794                         type: string
795
796         400:
797             description: "Dati del sondaggio non validi."
798             content:
799                 application/json:
800                     schema:
801                         "$ref": "#/components/schemas/Error"
802
803         401:
804             description: "Utente non autenticato."
805             content:
806                 application/json:
807                     schema:
808                         "$ref": "#/components/schemas/Error"
809
810         403:
811             description: "Utente loggato ma non ha i permessi per modificare il sondaggio."
812             content:
813                 application/json:
814                     schema:
815                         "$ref": "#/components/schemas/Error"
816
817         404:
818             description: "Sondaggio non trovato."
819             content:
820                 application/json:
821                     schema:
822                         "$ref": "#/components/schemas/Error"
823
824     tags:
825     - Sondaggi
826     - Sondaggista
827
828 delete:
829     summary: "Elimina un sondaggio"
830     description: "Elimina un sondaggio. Solo l'utente che ha creato il sondaggio può eliminarlo o un
831 ↪ amministratore, se il sondaggio è in attesa. Se il sondaggio è approvato o rifiutato, solo
832 ↪ l'amministratore può eliminarlo."
833
834     security:
835     - BearerAuth: []
836
837     responses:
838         204:
839             description: "Sondaggio eliminato con successo."
840
841         400:
842             description: "ID proposto non valido."
843             content:
844                 application/json:
845                     schema:
846                         "$ref": "#/components/schemas/Error"
847
848         401:
849             description: "Utente non autenticato."
850             content:
851                 application/json:
852                     schema:

```

```
840         "$ref": "#/components/schemas/Error"
841     403:
842         description: "Utente loggato ma non ha i permessi per eliminare il sondaggio."
843         content:
844             application/json:
845                 schema:
846                     "$ref": "#/components/schemas/Error"
847     404:
848         description: "Sondaggio non trovato."
849         content:
850             application/json:
851                 schema:
852                     "$ref": "#/components/schemas/Error"
853     tags:
854         - Sondaggi
855         - Sondaggista
856 '/voti' :
857     description: "Risorsa per la gestione dei voti di un sondaggio."
858     parameters:
859         - name: idSondaggio
860           in: query
861           description: "Identificatore univoco del sondaggio."
862           required: true
863           schema:
864               type: string
865     get:
866         summary: "Ottieni lista dei voti di un sondaggio"
867         description: "Restituisce la lista dei voti assegnati al sondaggio."
868         security:
869             - BearerAuth: []
870     responses:
871         200:
872             description: "Successful response"
873             content:
874                 application/json:
875                     schema:
876                         type: array
877                         items:
878                             "$ref": "#/components/schemas/Voto"
879         400:
880             description: "ID proposto non valido."
881             content:
882                 application/json:
883                     schema:
884                         "$ref": "#/components/schemas/Error"
885         401:
886             description: "Utente non autenticato."
887             content:
888                 application/json:
889                     schema:
890                         "$ref": "#/components/schemas/Error"
891     403:
892         description: "Utente loggato ma non ha i permessi per visualizzare i voti."
893         content:
894             application/json:
895                 schema:
896                     "$ref": "#/components/schemas/Error"
897     404:
898         description: "Sondaggio non trovato."
899         content:
900             application/json:
901                 schema:
902                     "$ref": "#/components/schemas/Error"
903     tags:
904         - Sondaggi
905         - Sondaggista
906     post:
907         summary: "Aggiungi un voto"
908         description: "Aggiunge un voto al sondaggio."
909         security:
910             - BearerAuth: []
911         requestBody:
912             required: true
```

```

913     content:
914       application/json:
915         schema:
916           "$ref": "#/components/schemas/AddVoto"
917   responses:
918     201:
919       description: "Voto aggiunto con successo."
920       headers:
921         Location:
922           description: "Url della risorsa creata."
923           schema:
924             type: string
925     400:
926       description: "Dati del voto non validi."
927       content:
928         application/json:
929           schema:
930             "$ref": "#/components/schemas/Error"
931     401:
932       description: "Utente non autenticato."
933       content:
934         application/json:
935           schema:
936             "$ref": "#/components/schemas/Error"
937     403:
938       description: "Utente loggato ma non ha i permessi per aggiungere un voto."
939       content:
940         application/json:
941           schema:
942             "$ref": "#/components/schemas/Error"
943     404:
944       description: "Sondaggio non trovato."
945       content:
946         application/json:
947           schema:
948             "$ref": "#/components/schemas/Error"
949   tags:
950     - Sondaggi
951     - Sondaggista
952   '/voti/{idVoto}':
953     description: "Risorsa per la gestione di un voto di un sondaggio."
954     parameters:
955       - name: idVoto
956         in: path
957         description: "Identificatore univoco del voto."
958         required: true
959         schema:
960           type: string
961     delete:
962       summary: "Elimina un voto"
963       description: "Elimina un voto. Solo l'utente che ha effettuato il voto può eliminarlo"
964       security:
965         - BearerAuth: []
966     responses:
967       204:
968         description: "Voto eliminato con successo."
969       400:
970         description: "ID proposto non valido."
971         content:
972           application/json:
973             schema:
974               "$ref": "#/components/schemas/Error"
975       401:
976         description: "Utente non autenticato."
977         content:
978           application/json:
979             schema:
980               "$ref": "#/components/schemas/Error"
981       403:
982         description: "Utente loggato ma non ha i permessi per eliminare il voto."
983         content:
984           application/json:
985

```



```
986         schema:
987             "$ref": "#/components/schemas/Error"
988     404:
989         description: "Voto non trovato."
990         content:
991             application/json:
992                 schema:
993                     "$ref": "#/components/schemas/Error"
994     tags:
995         - Sondaggi
996         - Sondaggista
```

## 4 Implementazione

L'applicazione è stata sviluppata usando il linguaggio di programmazione "TypeScript" sia per la parte di *frontend* che per quella di *backend*. Per la parte di *backend* è stato sfruttato il *runtime system* "Node.js" e il *framework* "Express.js" per la creazione e gestione del *server*. Per la parte di *frontend* è stato utilizzato il *framework* "Vue.js" con la libreria "PrimeVue" per alcune componenti grafiche. Il *database* utilizzato è "MongoDB" e per la gestione delle dipendenze è stato utilizzato "npm".

Si è inoltre scelto di usare *Vite* come *bundler* per la parte di *frontend* e *Webpack* per la parte di *backend*. Oltre a questo per alcuni stili della libreria *PrimeVue* è stato usato *Tailwind CSS* e come conseguenza è stato usato *PostCSS* per la gestione dei fogli di stile.

La scelta di usare *TypeScript* è scaturita dalla necessità di avere un controllo maggiore sul *type-checking* e per avere una maggiore manutenibilità del codice, è stata creata infatti una vera e propria gerarchia di tipi per la gestione dei dati sia lato *frontend* che lato *backend*.

La scelta del presente *stack* tecnologico è stata fatta in base al materiale fornito dal corso ed conoscenze pregresse di alcuni membri del gruppo.

### 4.1 Repository Organization

Il codice del progetto, disponibile presso la seguente repository <https://github.com/lucafano04/progettoComune>, è stato organizzato seguendo la seguente struttura:

```

/
├── .github
│   └── /workflows.....Directory per le action di GitHub (Compilazione LATEX e test)
├── .vscode.....Directory per le impostazioni di Visual Studio Code
├── APIdoc
│   └── api.yaml.....Documentazione API
├── /app.....Directory per il codice di backend
│   ├── /db
│   │   ├── /models.....Directory per i modelli del database
│   │   ├── index.ts.....File di inizializzazione del database
│   │   └── schemas.ts.....File per la definizione degli schemi del database
│   ├── /routes.....Directory per le rotte e gli endpoints delle API
│   ├── /tests.....Directory per i test Jest per il backend
│   ├── /utils.....Directory per le utility di backend
│   ├── app.ts.....File di inizializzazione dell'applicazione Express.js
│   └── variables.ts.....File per la definizione delle variabili globali e ambientali
├── /deliverable.....Directory per i deliverable LATEX
│   ├── /D*.....Directory per il deliverable D*, con * numero del deliverable
│   └── /images.....Directory comune per le immagini di tutti i deliverable
├── /src.....Directory per il codice di frontend
│   ├── /assets.....Directory per gli asset dell'applicazione
│   ├── /components.....Directory per le componenti dell'applicazione
│   ├── /utils.....Directory per le utility di frontend
│   ├── App.vue.....Componente radice dell'applicazione
│   ├── index.css.....File per il foglio di stile globale
│   └── main.ts.....File di inizializzazione dell'applicazione Vue.js
├── /types
│   ├── /Circoscrizioni.....Tipi per le circoscrizioni
│   ├── /Dati.....Tipi per i dati comuni a circoscrizioni e quartieri
│   ├── /Quartieri.....Tipi per i quartieri
│   ├── /Sondaggi.....Tipi per i sondaggi
│   ├── /Utenti.....Tipi per gli utenti
│   ├── /Voti.....Tipi per i voti
│   └── index.d.ts.....File per il raggruppamento dei tipi
├── .env.example.....File di esempio per le variabili ambientali
└── .gitignore.....File per la definizione dei file da ignorare

```

---

index.html .....	Pagina HTML di base dell'applicazione
index.ts .....	File di inizializzazione dell'applicazione
package.json .....	File per la definizione delle dipendenze
pitch.pptx .....	Presentazione del progetto
*.config.js ..	File per la configurazione di Webpack, TailWind e PostCSS. Sostituendo * con la configurazione desiderata
tsconfig.*.json .....	File vari per la configurazione di TypeScript. Sostituendo * con la configurazione desiderata
vite.config.ts .....	File per la configurazione di Vite per la parte di <i>frontend</i>

## 4.2 *Branching Strategy* e organizzazione del lavoro

Per la gestione del lavoro si è scelto di utilizzare la piattaforma **GitHub** e di adottare una strategia di *branching* basata su *GitFlow*. In particolare si è deciso di utilizzare i seguenti *branch*:

**main** *Branch* principale, contiene il codice stabile e funzionante;

**frontend** *Branch* per lo sviluppo della parte di *frontend*;

**MongoDB-Backend** *Branch* per lo sviluppo della parte di *backend* e del *database*;

**D\* e modifichD\*** *Branch* per lo sviluppo dei *deliverable* D\* e per le modifiche successive;

**UserStory - UserWorkFlow** *Branch* per la scrittura delle *User Story* il disegno dei *User WorkFlow*.  
Questi erano *branch* temporanei figli del *branch* D2;

**altri** Altri *branch* che sono stati creati per lo sviluppo di *pitch* o prime parti iniziali per il *deliverable* D1;

Come suddivisione dello sviluppo e della scrittura dei *deliverable* si è scelto di assegnare ad ogni membro del gruppo una particolare area di lavoro individuando un responsabile principale per le varie aree di sviluppo e stesura dei documenti, per le slides di *pitch* ognuno ha contribuito in modo equo. Distinguiamo quindi i seguenti ruoli:

**Luca Facchini** Responsabile della parte pratica di *backend*, *frontend*. Responsabile principale per la scrittura del documento D3. Addetto alla prima scrittura di requisiti funzionali e non funzionali per il *deliverable* D1;

**Luca Prigione** Responsabile principale per il documento D1 e co-responsabile per il documento D2. Responsabile inoltre per la parte pratica riguardante la struttura del *database*. Addetto alla stesura delle *UserStories* per il D3;

**Enrico Faa** Co-responsabile per il documento D2 e responsabile di tutti i grafici presenti su tutti i documenti (*UserWorkFlow*, *UseCaseDiagram*, *ClassDiagram*, ...). Addetto alla stesura delle *UserStories* per il D3 ed addetto alla parte di *Testing* per il D3; Addetto inoltre alla prima stesura della descrizione del progetto

## 4.3 *Dependency*

**Dipendenze principali** Sono state utilizzate le seguenti dipendenze principali:

**express** per la creazione e gestione del *server* e delle rotte;

**mongoose** per la gestione del *database*;

**mongodb** per la connessione al *database*;

**jsonwebtoken** per la gestione dei *token* di autenticazione;

### Dipendenze di sviluppo

`concurrently` per l'esecuzione di più comandi in parallelo durante le fasi di sviluppo e *build* del progetto;

`dotenv` per la gestione delle variabili ambientali;

`jest` per la gestione dei test;

`supertest` e `ts-jest` per la gestione dei test *end-to-end*;

`nodemon` per il *hot-reloading* del *server* in fase di sviluppo;

`postcss` per la gestione dei fogli di stile;

`primevue` per la creazione di alcune componenti grafiche;

`tailwindcss` per la disposizione della *responsiveness* delle componenti;

TypeScript per il *type-checking* e la gestione dei tipi;

`ts-node` per l'esecuzione di codice TypeScript direttamente da `Node.js`;

`vite` per la compilazione e il *bundling* del codice;

`vue` per la creazione delle componenti, inoltre i seguenti sotto-moduli sono stati utilizzati:

- `vue-router` per la gestione delle rotte;
- `vue-tsc` per il *type-checking* di `Vue.js`;
- `vue-leftlet` per la gestione delle mappe;

## 4.4 Database

Il *database* è stato progettato per contenere le seguenti collezioni:

### 4.4.1 Quartiere

La collezione `Quartiere` viene usata per memorizzare i dati relativi ai quartieri di Trento. Ogni documento della collezione contiene i seguenti campi:

- `_id` (`ObjectId`) identificativo univoco del quartiere;
- `nome` (`String`) nome del quartiere;
- `coordinate` (`Number[] []`) coordinate del quartiere;
- `circoscrizione` (`ObjectId`) identificativo della circoscrizione a cui appartiene il quartiere;
- `popolazione` (`Number`) numero di abitanti del quartiere;
- `superficie` (`Number`) superficie del quartiere in  $km^2$ ;
- `serviziTotali` (`Number`) numero totale di servizi presenti nel quartiere;
- `interventiPolizia` (`Number`) numero di interventi della polizia nel quartiere;
- `etaMedia` (`Number`) età media degli abitanti del quartiere;
- `servizi` (`Object`) oggetto contenente i servizi presenti nel quartiere;
  - `areeVerdi` (`Number`) numero di aree verdi presenti nel quartiere;
  - `scuole` (`Number`) numero di scuole presenti nel quartiere;
  - `serviziRistorazione` (`Number`) numero di servizi di ristorazione presenti nel quartiere;
  - `localiNottturni`
- `sicurezza` (`Object`) oggetto contenente i dati relativi alla sicurezza del quartiere;
  - `numeroInterventi` (`Number`) numero di interventi della polizia nel quartiere;
  - `incidenti` (`Number`) numero di incidenti nel quartiere;
  - `tassoCriminalità` (`Number`) tasso di criminalità del quartiere;

#### 4.4.2 Circostrizione

La collezione **Circostrizione** viene usata solamente per la memorizzazione delle coordinate e del nome delle circostrizioni di Trento, questo in quanto i dati relativi alle circostrizioni vengono estrapolati dai dati dei quartieri stessi. Ogni documento della collezione contiene i seguenti campi:

`_id` (`ObjectId`) identificativo univoco della circostrizione;  
`nome` (`String`) nome della circostrizione;  
`coordinate` (`Number[] []`) coordinate della circostrizione;

#### 4.4.3 Sondaggio

La collezione **Sondaggio** viene usata per memorizzare i dati relativi ai sondaggi effettuati dai sondaggisti, in particolare Ogni documento della collezione contiene i seguenti campi:

`_id` (`ObjectId`) identificativo univoco del sondaggio;  
`titolo` (`String`) titolo del sondaggio;  
`dataInizio` (`Date`) data di inizio del sondaggio;  
`isAperto` (`Boolean`) *flag* che indica se il sondaggio è aperto o chiuso;  
`statoApprovazione` (`In attesa|Approvato|Rifiutato`) stato di approvazione del sondaggio;  
`sondaggista` (`ObjectId`) identificativo del sondaggista;

#### 4.4.4 User

La collezione **User** viene usata per memorizzare i dati relativi agli utenti dell'applicazione. Questa collection viene strutturata in questo modo in quanto per lo scopo del progetto non viene implementato il sistema di autenticazione SSO tramite SPID/CIE/TS-CNS. Nel caso si dovesse implementare un sistema di autenticazione tramite SSO pubblico la collezione conterrebbe solo una associazione tra l'utente e i suoi dati personali. Ogni documento della collezione contiene i seguenti campi:

`_id` (`ObjectId`) identificativo univoco dell'utente;  
`nome` (`String`) nome dell'utente;  
`cognome` (`String`) cognome dell'utente;  
`email` (`String`) email dell'utente;  
`password` (`String`) password dell'utente;  
`ruolo` (`Amministratore|Analista|Sondaggista|Circostrizione`) ruolo dell'utente;  
`imageUrl` (`String`) *Hash* dell'immagine profilo dell'utente;

#### 4.4.5 Voto

La collezione **Voto** viene usata per memorizzare i voti dati dai cittadini ai quartieri. Ogni documento della collezione contiene i seguenti campi:

`_id` (`ObjectId`) identificativo univoco del voto;  
`eta` (`Number`) età del votante;  
`voto` (`Number`) voto dato al quartiere;  
`quartiere` (`ObjectId`) identificativo del quartiere votato;  
`dataOra` (`Date`) data e ora del voto;  
`sondaggio` (`ObjectId`) identificativo del sondaggio a cui il voto è associato;

## 4.5 Testing

Per ognuno dei requisiti funzionali implementati nell'applicazione sono stati individuati dei casi di test per assicurare la loro corretta implementazione e, in generale, il corretto funzionamento degli elementi di frontend e backend del sistema, anche in situazioni di casi limite.

N.	Descrizione	Data	Precondizioni	Dipendenze	Risultato Atteso	Risultato Riscontrato	Note
1	Ottenimento lista quartieri base senza coordinate	<i>deepData=false, coordinate=false</i>	-	-	Lista quartieri senza coordinate	Lista quartieri senza coordinate	-
1.1	Ottenimento lista quartieri base con coordinate	<i>deepData=false, coordinate=true</i>	-	-	Lista quartieri con coordinate	Lista quartieri con coordinate	-
1.2	Ottenimento lista quartieri dettagliata senza coordinate	<i>deepData=true, coordinate=false</i>	-	-	Lista quartieri dettagliata senza coordinate	Lista quartieri dettagliata senza coordinate	-
1.3	Ottenimento lista quartieri dettagliata con coordinate	<i>deepData=true, coordinate=true</i>	-	-	Lista quartieri dettagliata con coordinate	Lista quartieri dettagliata con coordinate	-
2	Ottenimento singolo quartiere con coordinate	<i>quartiereId, coordinate=true</i>	Il quartiere deve esistere	-	Quartiere con coordinate	Quartiere con coordinate	-
2.1	Ottenimento singolo quartiere senza coordinate	<i>quartiereId, coordinate=false</i>	Il quartiere deve esistere	-	Quartiere senza coordinate	Quartiere senza coordinate	-
2.2	Ottenimento quartiere non esistente	<i>quartiereId, coordinate=any</i>	Il quartiere non deve esistere	-	Errore 404	Errore 404	-
2.3	Ottenimento quartiere con ID non valido	<i>quartiereId, coordinate=any</i>	Il ID del quartiere deve essere valido	-	Errore 400	Errore 400	-

## 5 *FrontEnd*

Di seguito viene illustrato la parte di *FrontEnd* del progetto. Verranno illustrate le principali funzionalità sviluppate ed alcuni dettagli implementativi.

### 5.1 *Home Page*

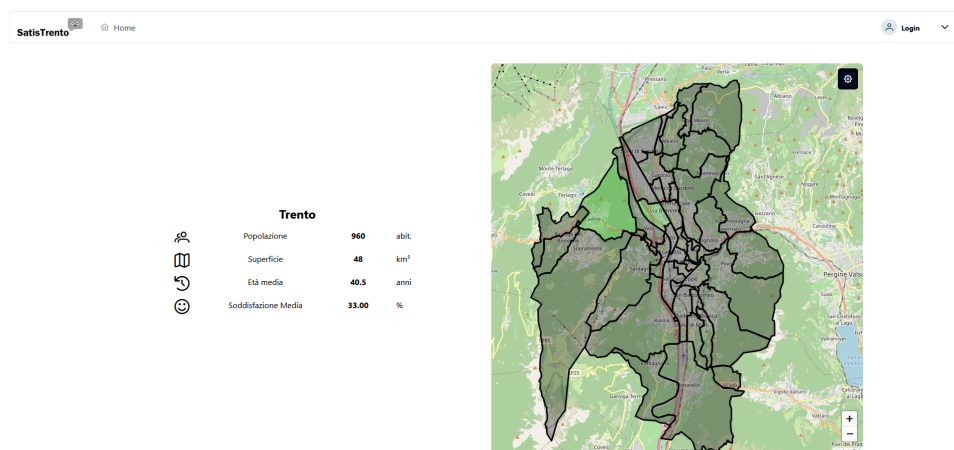


Figura 5.1: *Home Page* per utenti non loggati

Nella Figura 5.1 è possibile vedere la *Home Page* della *web-app*. Come da specifiche descritte nei documenti precedenti questa presenta: i dati generali relativi all'intero comune, la mappa interattiva, tematica sulla base della soddisfazione media e suddivisa per quartieri. Nell'angolo in alto a destra della mappa è presente un pulsante per aprire il menù delle impostazioni per passare da “quartieri” a “circoscrizioni” e viceversa. Inoltre, nell'angolo superiore destro della pagina è presente un pulsante per accedere alla pagina di *login*.

#### Dettaglio menù impostazioni



Figura 5.2: Menù impostazioni

Come precedentemente descritto, il menù delle impostazioni sopra raffigurato, chiudibile premendo la “X”, permette di passare da una visualizzazione dei dati per quartieri ad una per circoscrizioni e viceversa. La figura sopra è relativa a tutte le tipologie di utenti ad eccezione degli utenti con ruolo “analista”.

## Selezione di un “quartiere”/“circonscrizione”

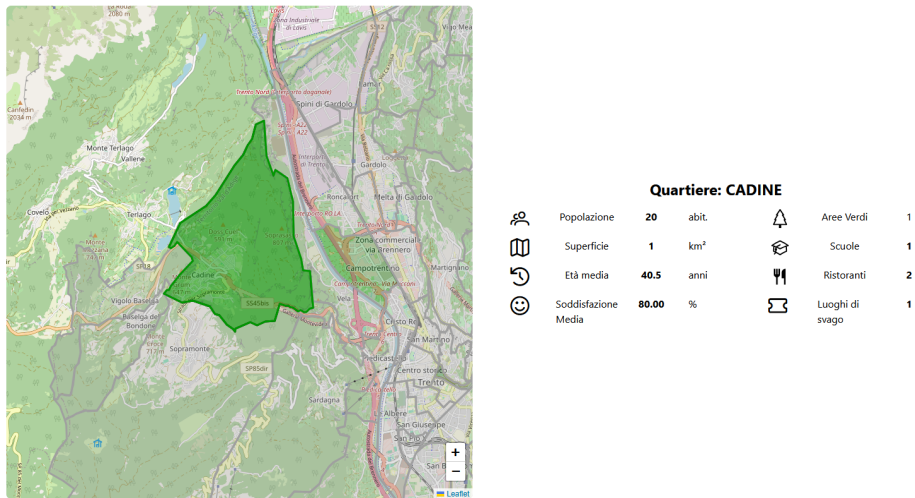


Figura 5.3: Selezione di un “quartiere”/“circonscrizione”

Notiamo dalla Figura 5.3 come la selezione di un “quartiere” o “circonscrizione” avvenga tramite un click sulla mappa. Una volta selezionato un “quartiere” o “circonscrizione” verranno visualizzati degli ulteriori informazioni più specifiche riguardanti il “quartiere” o la “circonscrizione” selezionata. Inoltre il “quartiere” o la “circonscrizione” selezionata verrà evidenziata sulla mappa oscurando gli altri “quartieri” o le altre “circonscrizioni”, inoltre la mappa avrà il suo centro sul centro del quartiere ed avrà uno zoom adeguato per visualizzare il “quartiere” o la “circonscrizione” selezionata nella sua interezza. Da questa schermata è possibile tornare alla visualizzazione generale ri-selezionando il “quartiere” o la “circonscrizione” selezionata oppure è possibile cambiare “quartiere” o “circonscrizione” selezionando un altro “quartiere” o “circonscrizione” dalla mappa.

## 5.2 Login

**Login**

Email

Password

Login

Figura 5.4: Dettaglio pagina di *login*

Nella Figura 5.4 è possibile vedere il dettaglio della pagina di *login*. Da questa pagina tutti gli utenti in possesso di credenziali valide possono accedere.



## Messaggio di errore

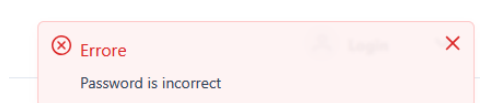


Figura 5.5: Messaggio di errore

Nella Figura 5.5 è possibile vedere come si presenta il messaggio di errore in caso di credenziali errate. Questo viene visualizzato per 5 secondi in caso di credenziali errate o altro errore.

Si noti come il presente “formato” di *feedback* per i messaggi di errori sia stato implementato per tutte le pagine della *web-app* in caso di un qualsiasi errore, viene difatti mostrato il titolo dell’errore con l’azione che non è stata possibile effettuare ed una descrizione più dettagliata del perché non è stato possibile effettuare l’azione richiesta.

## Messaggio di login effettuato

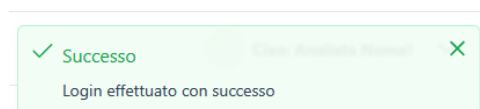


Figura 5.6: Messaggio di login effettuato

Nella Figura 5.6 è possibile vedere come si presenta il messaggio di login effettuato con successo. Questo viene visualizzato per 3 secondi in caso di login effettuato con successo.

Si noti come il presente “formato” di *feedback* sia stato implementato per tutte le pagine della *web-app* in caso di un qualsiasi errore, viene difatti mostrato la scritta “successo”, o altra equivalente, ed una breve descrizione di cosa è stato effettuato con successo.

## 5.3 Funzionalità Analista

Nella seguente sezione verranno illustrate le funzionalità disponibili per gli utenti con ruolo “analista”. Questi utenti hanno accesso a funzionalità specifiche per la loro mansione e non disponibili agli altri utenti.

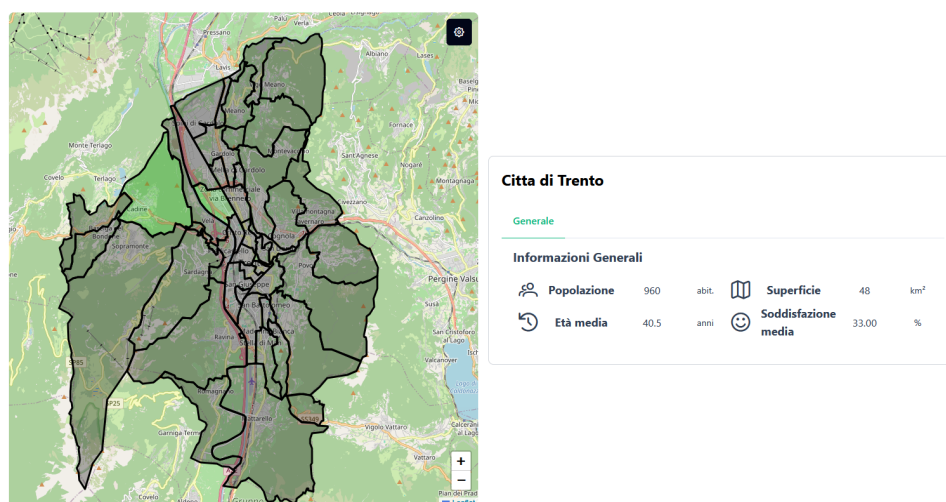


Figura 5.7: Home Page per utenti con ruolo “analista”

Possiamo notare dalla Figura 5.7 come l'interfaccia grafica si sia differenziata rispetto alla Figura 5.1. Possiamo notare come le informazioni generali del comune siano comunque visibili ma sono raggruppate, non cambia molto infatti dalla *homepage* delle altre tipologie di utenti.

### Selezione di un “quartiere”/“circonscrizione” - analista

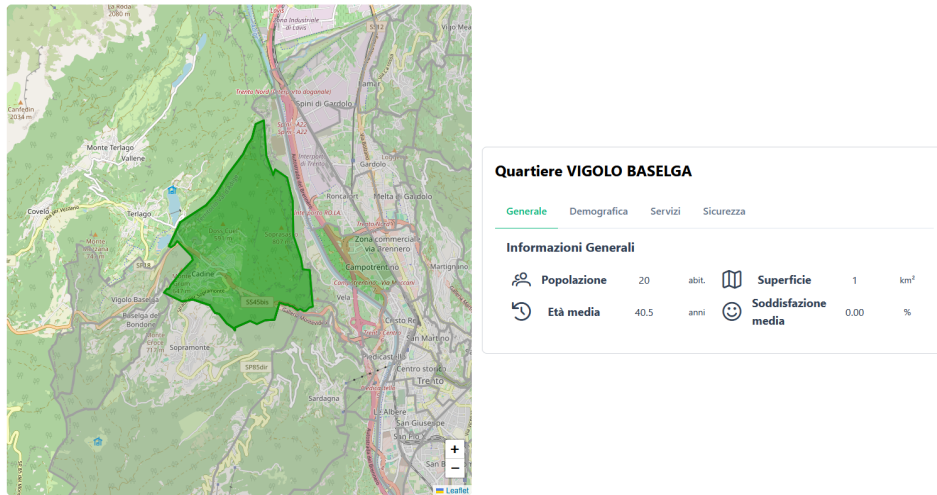


Figura 5.8: Selezione di un “quartiere”/“circonscrizione” per utenti con ruolo “analista”

Notiamo dalla Figura 5.8 come la selezione di un “quartiere” o “circonscrizione” avvenga tramite un click sulla mappa come nel caso degli altri utenti. Una volta selezionato un “quartiere” o “circonscrizione” verranno visualizzati oltre alle informazioni generali del “quartiere” o della “circonscrizione” anche altri indicatori accessibili tramite le diverse pagine del menù posizionato sulla destra della mappa.

Si possa notare come questa visualizzazione sia una estensione della visualizzazione per gli altri utenti, dunque tutte le funzionalità e le azioni che il sistema compie alla selezione/deselezione sono le stesse.

### Dettaglio menù impostazioni

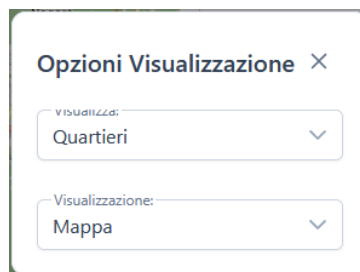


Figura 5.9: Menù impostazioni per utenti con ruolo “analista”

Il menù delle impostazioni per gli utenti con ruolo “analista” estende le funzionalità del menù per tutti gli altri utenti della figura 5.2. In particolare, oltre alla possibilità di passare da una visualizzazione dei dati per quartieri ad una per circoscrizioni e viceversa, è possibile passare dalla visualizzazione di questi da una visualizzazione in “mappa” ad una in “tabella”. Questo permette agli utenti con ruolo “analista” di avere una visione più dettagliata dei dati.

## Visualizzazione dati in “tabella”

Elenco quartieri

Nome	Popolazione	Superficie	Età media	Soddisfazione media
SAN LAZZARO	20	1	40.5	0.00%
SOPRAMONTE	20	1	40.5	0.00%
MATTARELLO	20	1	40.5	0.00%
LASTE	20	1	40.5	0.00%
ROMAGNANO	20	1	40.5	0.00%
CANOVA	20	1	40.5	0.00%
POVO	20	1	40.5	0.00%
GARDOLO	20	1	40.5	0.00%
RONCAFORT	20	1	40.5	0.00%
CAMPO NOMADI RAVINA	20	1	40.5	0.00%
SPINI-GHIAIE	20	1	40.5	0.00%
OLTRECASTELLO	20	1	40.5	0.00%
CADINE	20	1	40.5	80.00%

Città di Trento

Generale

Informazioni Generali

Popolazione

960 abit.

Superficie

48 km<sup>2</sup>

Età media

40.5 anni

Soddisfazione media

33.00 %

Figura 5.10: Visualizzazione dati in “tabella”

Nella Figura 5.10 è possibile vedere la visualizzazione dei dati in “tabella” per gli utenti con ruolo “analista”. Questa visualizzazione permette di avere una visione, ordinabile per le colonne presenti, dei dati relativi ai “quartieri” o alle “circoscrizioni”. Da questa visualizzazione è possibile selezionare un “quartiere” o una “circoscrizione” premendo sulla riga corrispondente.

## “Quartiere”/“circoscrizione” selezionata in visualizzazione tabella

Elenco quartieri

Nome	Popolazione	Superficie	Età media	Soddisfazione media
SAN LAZZARO	20	1	40.5	0.00%
SOPRAMONTE	20	1	40.5	0.00%
MATTARELLO	20	1	40.5	0.00%
LASTE	20	1	40.5	0.00%
ROMAGNANO	20	1	40.5	0.00%
CANOVA	20	1	40.5	0.00%
POVO	20	1	40.5	0.00%
GARDOLO	20	1	40.5	0.00%
RONCAFORT	20	1	40.5	0.00%
CAMPO NOMADI RAVINA	20	1	40.5	0.00%
SPINI-GHIAIE	20	1	40.5	0.00%
OLTRECASTELLO	20	1	40.5	0.00%
CADINE	20	1	40.5	80.00%

Quartiere LASTE

Generale Demografica Servizi Sicurezza

Informazioni sui Servizi

Numero Totale

5

Aree Verdi

1

Locali Notturni

1

Scuole

1

Servizi di Ristorazione

2

Figura 5.11: “Quartiere”/“circoscrizione” selezionata in visualizzazione tabella

La figura 5.11 mostra come si presenta la visualizzazione di un “quartiere” o una “circoscrizione” selezionata in visualizzazione tabella. Questa situazione è una unione della selezione nel caso della figura 5.8 e della visualizzazione in tabella della figura 5.10. Da questa visualizzazione è possibile tornare alla visualizzazione dei dati della città selezionando il “quartiere” o la “circoscrizione” selezionata oppure è possibile cambiare “quartiere” o “circoscrizione” selezionando un altro “quartiere” o “circoscrizione” dalla tabella, in questo caso la riga selezionata verrà evidenziata ed i dati relativi al “quartiere” o alla “circoscrizione” selezionata verranno visualizzati.

## 5.4 Funzionalità Sondaggista

Si illustrano in questa sezione tutte le funzionalità disponibili per gli utenti con ruolo “sondaggista”. Questi utenti hanno accesso a funzionalità specifiche per effettuare sondaggi, gestirli ed inviarli per l’approvazione. La visualizzazione della *Home Page* per gli utenti con ruolo “sondaggista” è la stessa di quella degli altri utenti, come mostrato nella figura 5.1.

### Gestione sondaggi

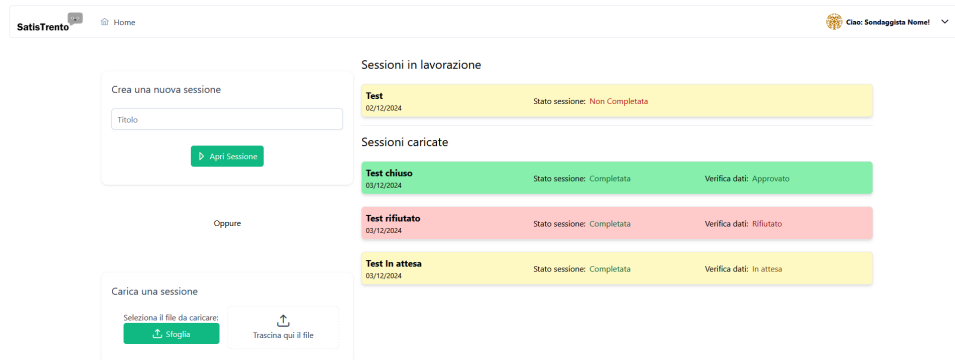


Figura 5.12: Pagina di gestione dei sondaggi

Nella figura 5.12 è possibile vedere la pagina di gestione dei sondaggi per gli utenti con ruolo “sondaggista”. Da questa pagina è possibile iniziare un nuovo sondaggio tramite il modulo con pulsante nella parte superiore-sinistra della pagina, caricare un sondaggio da un file *.csv* tramite il pulsante nella parte inferiore-sinistra della pagina<sup>1</sup> ed infine è possibile vedere l’elenco dei sondaggi in corso e quelli inviati per l’approvazione. Possiamo distinguere come lo stato del sondaggio, oltre che riportato nella riga stessa del sondaggio, sia rappresentato da un colore diverso per ogni stato. In particolare, il colore verde indica che il sondaggio è stato approvato, il colore giallo indica che il sondaggio è in attesa di approvazione o se presente sotto la voce “Sessioni in lavorazione” indica che il sondaggio è in corso, il colore rosso indica che il sondaggio è stato rifiutato.

I sondaggi “in lavorazione” ovvero quelli in corso, sono visualizzati in una sezione separata rispetto a quelli inviati per l’approvazione. Questo in quanto i sondaggi in corso sono quelli che possono essere modificati e completati, mentre quelli inviati per l’approvazione non possono più essere modificati dai sondaggisti. È presente per questi sondaggi un pulsante per visualizzare e riprendere il sondaggio, in modo da poterlo completare e inviare per l’approvazione.

<sup>1</sup>Funzionalità non implementata

## Pagina gestione sondaggio

The screenshot shows the SatisTrento survey management interface. It features a top navigation bar with the SatisTrento logo and a user profile dropdown. The main content area is divided into four panels:

- Statistiche Parziali:** Displays partial statistics including 'Numero parziale voti: 2', 'Età media parziale: 24', and a table of votes by district.
- Aggiungi Voto:** A form for adding a new vote, with dropdown menus for 'Quartiere' and 'Età', and a green 'Vota' button.
- Voti Precedenti:** A list of previously added votes, showing identifiers and timestamps, with delete icons.
- Impostazioni Sessione:** A panel with three buttons: 'Esci dal sondaggio' (blue), 'Chiudi e completa la sessione' (green), and 'Cancella definitivamente' (red).

Quartiere	Numero Voti
CANOVA	1
BOLGHERA	1

Figura 5.13: Pagina di gestione di un sondaggio

La pagina mostrata dalla figura 5.13 si può dividere in quattro aree, sulla parte sinistra è possibile individuare le “Statistiche parziali” ovvero: il numero di voti raccolti fino ad ora, l’età media dei votanti, ed una tabella contenente il numero di voti per ogni quartiere di provenienza dei votanti. Nel riquadro che occupa la parte superiore centrale e superiore destra è presente il pre-modulo per l’inserimento di un voto all’interno del sondaggio, questi si compongono di due selettori del tipo “menù a discesa”, uno per il quartiere di appartenenza (obbligatorio impostarne il valore) ed uno per l’età del votante (opzionale). Nella parte centrale inferiore è presente l’elenco dei voti già inseriti, per motivi di *privacy* si è scelto di non mostrare il voto, ma solo un identificativo ed l’orario di inserimento, questo in quanto la possibilità di eliminare un voto è fornita solo per la correzione di errori. Infine nella parte inferiore destra sono presenti i controlli per uscire dal sondaggio e tornare all’elenco dei sondaggi, per inviare il sondaggio per l’approvazione e per eliminare il sondaggio.

## Dettaglio invio sondaggio/eliminazione sondaggio

The screenshot shows a confirmation dialog box with the title "Conferma chiusura". It contains a warning icon and the text "Sei sicuro di voler chiudere il sondaggio?". At the bottom, there are two buttons: "Annulla" (blue) and "Chiudi" (green).

Figura 5.14: Dettaglio invio sondaggio

Nella figura 5.14 è possibile vedere il dettaglio della finestra di conferma per l’invio del sondaggio per l’approvazione. Questa finestra di conferma è presente anche per l’eliminazione del sondaggio, col pulsante di eliminazione colorato di rosso ed altri testi, ma con la stessa struttura. Questa finestra di conferma è presente per evitare eliminazioni accidentali di sondaggi o invii accidentali per l’approvazione.

## Processo di voto

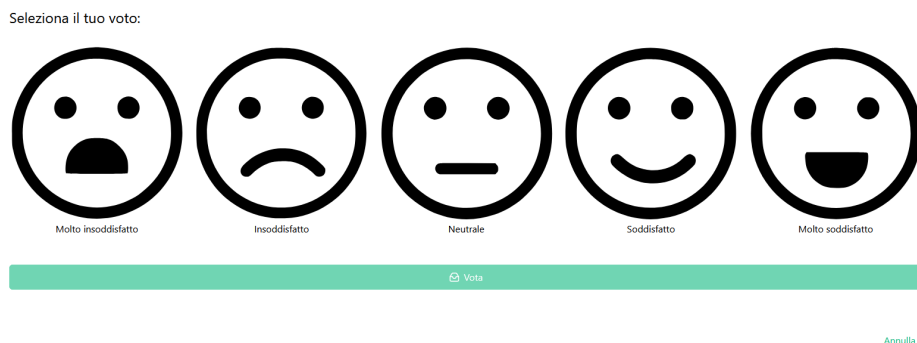


Figura 5.15: Dettaglio pagina di voto

Dalla presente figura 5.15 si può notare come si presenta al votante l'interfaccia di voto dopo che il sondaggista ha compilato "quartiere" ed opzionalmente "età" del votante e premuto il tasto "vota".



Figura 5.16: Dettaglio pagina di voto con voto selezionato

Dalla figura 5.16 si può notare come si presenta al votante l'interfaccia di voto dopo che il votante ha selezionato il voto. Il voto è selezionabile tramite un click su uno dei pulsanti con il voto desiderato. Una volta selezionato il voto il pulsante selezionato verrà evidenziato con un colore diverso dagli altri pulsanti. Una volta selezionato il voto è possibile confermare il voto premendo il pulsante "vota" oppure se sono stati commessi degli errori nella pre-compilazione del voto è possibile annullare il voto premendo il pulsante "annulla".

Grazie per aver votato!

La tua preferenza è stata registrata, restituisci il dispositivo all'operatore

Chiudi

Figura 5.17: Dettaglio pagina di voto post-voto

Una volta che l'utente ha confermato il voto, verrà visualizzata la pagina di conferma del voto come mostrato nella figura 5.17. A questo punto il dispositivo dovrebbe essere restituito al sondaggista il quale tornerà tramite il pulsante "chiudi" alla pagina di gestione del sondaggio.

## 6 *Deployment*

**Live-Demo** I *deployment* sono stati effettuati su un unico *cluster* di `render.com` tramite il quale si gestisce anche l'automazione di CD. L'url per accedere all'applicazione è `https://satistrento.onrender.com/`. Per accedere alle aree riservate è necessario utilizzare delle credenziali:

**Utente Sondaggista** `sondaggista@test.com` - password

**Utente Analista** `analista@test.com` - password

Altre tipologie di utenti precedentemente definite dai requisiti funzionali non sono state implementate.

**CI/CD** Come precedentemente descritto il *deployment* è stato automatizzato tramite **GitHub Actions** e `render.com`, questo ad ogni commit sul branch `main` effettua il *deployment* in modo automatico della nuova versione dell'applicazione.

Per la parte di CI sono stati implementati dei *tests* tramite **Jest** e **supertest** per verificare il corretto funzionamento delle *API* e delle funzionalità dell'applicazione, queste sono state automatizzate dal file `.github/workflows/jestTesting.yml` presente nella repository del progetto, inoltre in quanto si è usato **TypeScript** è stato scelto di verificare anche la correttezza del codice tramite la **GitHub Action** definita nel file `.github/workflows/buildTS.yaml`.

**Help** Per problemi sulla *live-demo* contattare Luca Facchini all'indirizzo `luca.facchini-1@studenti.unitn.it` in quanto è il solo abilitato ad accedere alla dashboard di `render.com` e quindi a poter risolvere eventuali problemi.

