

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

Scopo del documento	1
1 <i>User Stories</i>	2
2 <i>User Workflow</i>	3
3 <i>Web APIs</i>	5
4 Implementazione	20
4.1 Repository Organization	20
4.2 <i>Branching Strategy</i> e organizzazione del lavoro	21
4.3 <i>Dependency</i>	21
4.4 <i>Database</i>	22
4.5 Testing	24
5 <i>FrontEnd</i>	34
5.1 <i>Home Page</i>	34
5.2 <i>Login</i>	35
5.3 funzionalità Analista	36
5.4 funzionalità Sondaggista	38
6 <i>Deployment</i>	39

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).

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

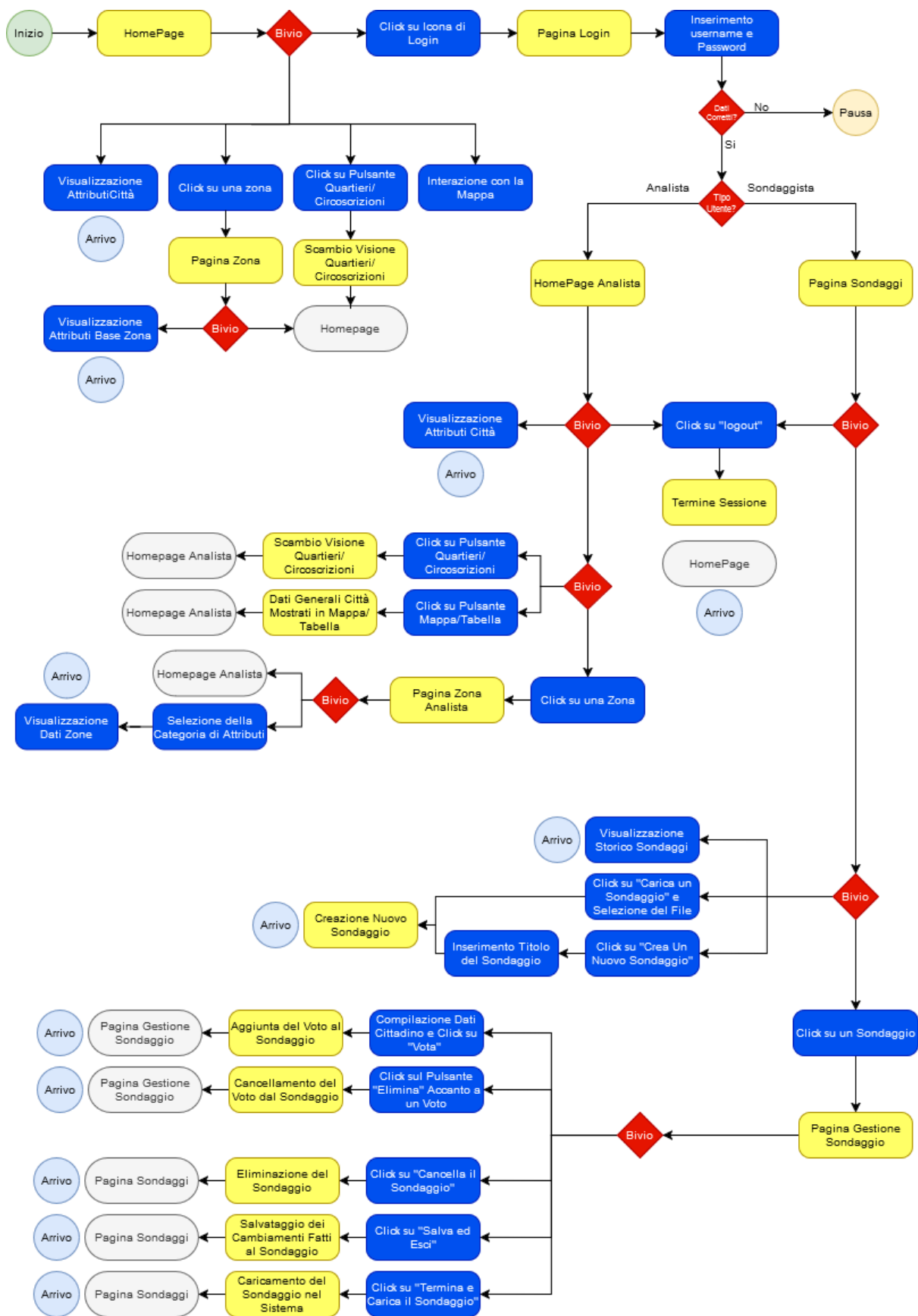


Figura 2.2: User Workflow

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                     "$ref": "#/components/schemas/Error"
427     tags:
428         - Quartieri
429         - Dati Statici
430     '/circoscrizioni' :
431         description: "Risorsa che restituisce la lista delle circoscrizioni di Trento."
432         get:
433             summary: "Ottieni lista delle circoscrizioni"
434             description: "Returns a list of all the districts in the city of Trento."
435             parameters:
436                 - name: deepData
437                   in: query
438                   description: "Se impostato a true, restituisce le informazioni complete di ogni circoscrizione."
439                   required: false
440                   schema:
441                       type: boolean
442                       default: false
443                   allowEmptyValue: true
444                 - name: coordinate
445                   in: query
446                   description: "Se impostato non impostato o impostato a true restituisce le coordinate di ogni
447                               ↳ circoscrizione, altrimenti no."
448                   required: false
449                   schema:
450                       type: boolean
451                       default: true
452                   allowEmptyValue: true
453             responses:
454                 200 :
455                     description: "Successful response"
456                     content:
457                         application/json:
458                             schema:
459                                 oneOf:
460                                     - type: "array"
461                                       items:
462                                           "$ref": "#/components/schemas/CircoscrizioneBase"
463                                     - type: "array"
464                                       items:
465                                           "$ref": "#/components/schemas/Circoscrizione"
466                                     - type: "array"
467                                       items:
468                                           "$ref": "#/components/schemas/CircoscrizioneBaseNC"
469                                     - type: "array"
470                                       items:
471                                           "$ref": "#/components/schemas/CircoscrizioneNC"
472             tags:
473                 - Circoscrizioni
474                 - Dati Statici
475     '/circoscrizioni/{id}' :
476         description: "Risorsa che restituisce le informazioni di una circoscrizione di Trento con le
477                     ↳ informazioni di base e i servizi presenti."
478         parameters:
479             - name: id
480               in: path
481               description: "Identificatore univoco della circoscrizione."
482               required: true
483               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   responses:
787     200:
788       description: "Sondaggio modificato con successo."
789       headers:
790         Location:
791           description: "Url della risorsa modificata."
792           schema:
793             type: string
794     400:
795       description: "Dati del sondaggio non validi."
796       content:
797         application/json:
798           schema:
799             "$ref": "#/components/schemas/Error"
800     401:
801       description: "Utente non autenticato."
802       content:
803         application/json:
804           schema:
805             "$ref": "#/components/schemas/Error"
806     403:
807       description: "Utente loggato ma non ha i permessi per modificare il sondaggio."
808       content:
809         application/json:
810           schema:
811             "$ref": "#/components/schemas/Error"
812     404:
813       description: "Sondaggio non trovato."
814       content:
815         application/json:
816           schema:
817             "$ref": "#/components/schemas/Error"
818   tags:
819     - Sondaggi
820     - Sondaggista
821 delete:
822   summary: "Elimina un sondaggio"
823   description: "Elimina un sondaggio. Solo l'utente che ha creato il sondaggio può eliminarlo o un
824   ↳ amministratore, se il sondaggio è in attesa. Se il sondaggio è approvato o rifiutato, solo
825   ↳ l'amministratore può eliminarlo."
826   security:
827     - BearerAuth: []
828   responses:
829     204:
830       description: "Sondaggio eliminato con successo."
831     400:
832       description: "ID proposto non valido."
833       content:
834         application/json:
835           schema:
836             "$ref": "#/components/schemas/Error"
837     401:
838       description: "Utente non autenticato."
839       content:
840         application/json:
841           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.

4.5.1 RF1: Visualizzazione Città

- **Test Case 1**

- **Descrizione:** Apertura dell'applicazione e visualizzazione della homepage come utente non loggato.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** L'applicazione mostra a schermo: sulla sinistra gli attributi demografici e riguardanti la soddisfazione di Trento. Sulla destra la mappa di Trento divisa per quartieri, colorati da rosso a verde in base alla relativa soddisfazione media. In alto a destra l'icona per eseguire il login e il menù drop-down.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 2**

- **Descrizione:** Apertura dell'applicazione e visualizzazione della homepage come utente Analista.
- **Test Data:**
- **Precondizioni:** L'utente è loggato in un account Analista.
- **Dipendenze:**
- **Risultato Atteso:** L'applicazione mostra a schermo: sulla sinistra gli attributi demografici e riguardanti la soddisfazione di Trento. Sulla destra la mappa di Trento divisa per quartieri, colorati da rosso a verde in base alla relativa soddisfazione media. In alto a destra l'immagine di profilo dell'utente, la scritta "Ciao, [nome utente]!", e il menù drop-down.
- **Risultato Riscontrato:**
- **Note:**

4.5.2 RF2: Interazione Con La Mappa

- **Test Case 1**

- **Descrizione:** Spostamento del focus della mappa.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Il centro della mappa si sposta nella direzione in cui viene trascinato.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 2.1**

- **Descrizione:** Zoom in/out della mappa utilizzando la rotella del mouse.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**

- **Risultato Atteso:** La mappa fa zoom in quando la rotella del mouse viene spostata in alto, e fa zoom out quando viene spostata in basso.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 2.2**

- **Descrizione:** Zoom in/out della mappa utilizzando i pulsanti nell'angolo.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** La mappa fa zoom in quando il pulsante con il segno "+" viene premuto, e fa zoom out quando viene premuto il pulsante con il segno "-".
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 3.1**

- **Descrizione:** Selezione di una zona cliccando su di essa nella mappa.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** La mappa si sposta alla sinistra dello schermo e muove il suo focus sulla zona cliccata. Gli attributi della zona cliccata vengono mostrati sulla destra dello schermo.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 3.2**

- **Descrizione:** Selezione di una zona cliccando su di essa sulla mappa quando una zona è già selezionata.
- **Test Data:**
- **Precondizioni:** Una zona è selezionata.
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 3.1, cliccando su una zona diversa da quella cliccata nel test 3.1.
- **Risultato Atteso:** La zona precedentemente selezionata viene deselezionata. La zona cliccata viene selezionata. La mappa rimane sulla sinistra dello schermo.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 3.3**

- **Descrizione:** Deselezione di una zona cliccando su di essa nella mappa.
- **Test Data:**
- **Precondizioni:** Una zona è selezionata
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto i test 3.1 e 3.2, cliccando sulla stessa zona cliccata nel test 3.2.
- **Risultato Atteso:** La mappa torna alla destra dello schermo e muove il suo focus per inquadrare tutta Trento. Gli attributi demografici e riguardanti la soddisfazione di Trento vengono mostrati sulla sinistra. La zona cliccata smette di essere selezionata.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 4.1**

- **Descrizione:** Selezione della divisione per quartieri.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** La mappa mostra Trento divisa per i suoi quartieri.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 4.2**

- **Descrizione:** Selezione della divisione per circoscrizioni.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** La mappa mostra Trento divisa per le sue circoscrizioni.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 5**

- **Descrizione:** Cambio visualizzazione da mappa a tabella.
- **Test Data:**
- **Precondizioni:** L'utente è loggato in un account Analista.
- **Dipendenze:**
- **Risultato Atteso:** La mappa viene sostituita da una tabella che mostra il nome e gli attributi demografici e della soddisfazione delle zone.
- **Risultato Riscontrato:**
- **Note:**

4.5.3 RF6: Login

- **Test Case 1**

- **Descrizione:** Login in un account.
- **Test Data:** [username] di un utente registrato nel sistema, [password] di quell'utente.
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Il login viene effettuato con successo. Una notifica di successo "Login effettuato con successo" appare in alto a destra. Il sistema ritorna automaticamente alla homepage.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 2**

- **Descrizione:** Login in un account con password sbagliata.
- **Test Data:** [username] di un utente registrato nel sistema, [password] diversa da quella di quell'utente.
- **Precondizioni:**
- **Dipendenze:**

- **Risultato Atteso:** Nessun login viene effettuato. Una notifica di errore "Credenziali errate" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 3**

- **Descrizione:** Login in un account che non esiste.
- **Test Data:** [username] non appartenente ad alcun utente registrato nel sistema.
- **Precondizioni:** [username] non appartenente ad alcun utente registrato nel sistema.
- **Dipendenze:**
- **Risultato Atteso:** Nessun login viene effettuato. Una notifica di errore "Credenziali errate" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 4**

- **Descrizione:** Login senza specificare un nome utente.
- **Test Data:** [username] vuoto.
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Nessun login viene effettuato. Una notifica di errore "Credenziali errate" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

4.5.4 RF7: Logout

- **Test Case 1**

- **Descrizione:** Uscita da un account all'interno della homepage.
- **Test Data:**
- **Precondizioni:** L'utente è loggato in un account e si trova sulla homepage dell'applicazione.
- **Dipendenze:**
- **Risultato Atteso:** L'utente torna allo stato di utente non loggato. Una notifica di successo "Logout effettuato con successo" appare in alto a destra. Il sistema ricarica automaticamente la homepage.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 2**

- **Descrizione:** Uscita da un account Sondaggista all'interno dell'area gestione sondaggi.
- **Test Data:**
- **Precondizioni:** L'utente è loggato in un account Sondaggista e si trova all'interno dell'area di gestione dei sondaggi.
- **Dipendenze:**
- **Risultato Atteso:** L'utente torna allo stato di utente non loggato. Appare una notifica di successo "Logout effettuato con successo". Il sistema ritorna automaticamente alla homepage.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 3**

- **Descrizione:** Uscita dall'account Sondaggista durante un voto.
- **Test Data:**
- **Precondizioni:** L'utente è loggato in un account Sondaggista. Il sistema si trova nel mezzo della creazione di un voto, dopo che è stato premuto il primo pulsante "Vota", ma prima di premere il secondo pulsante "Vota".
- **Dipendenze:**
- **Risultato Atteso:**
- **Risultato Ricontrato:** Il voto non viene aggiunto al sondaggio. L'utente torna allo stato di utente non loggato. Appare una notifica di successo "Logout effettuato con successo". Il sistema ritorna automaticamente alla homepage.
- **Note:**

4.5.5 RF9: Gestione Sondaggi

- **Test Case 1.1**

- **Descrizione:** Creazione di un nuovo sondaggio
- **Test Data:** [nome sondaggio] non vuoto.
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Viene creato un nuovo sondaggio con il nome inserito. Una notifica di successo "Sondaggio aggiunto con successo" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 1.2**

- **Descrizione:** Creazione di due sondaggi con lo stesso nome.
- **Test Data:** [nome sondaggio] già inserito nel sistema.
- **Precondizioni:** Un sondaggio con lo stesso nome esiste già nel sistema.
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 1.1, inserendo lo stesso [nome sondaggio].
- **Risultato Atteso:** Viene creato un secondo sondaggio con lo stesso nome, ma distinto da quello già esistente. Una notifica di successo "Sondaggio aggiunto con successo" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 1.3**

- **Descrizione:** Creazione di un nuovo sondaggio senza nome.
- **Test Data:** [nome sondaggio] vuoto
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Il sondaggio non viene creato. Una notifica di errore "Nome del sondaggio mancante" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 2**

- **Descrizione:** Continuazione di un sondaggio in corso.

- **Test Data:**
- **Precondizioni:** Esiste almeno un sondaggio in corso.
- **Dipendenze:**
- **Risultato Atteso:** Viene aperta la pagina di gestione del sondaggio selezionato. Sulla sinistra dello schermo vengono mostrati: il numero parziale dei voti nel sondaggio, l'età media parziale dei voti nel sondaggio, e il numero di voti nel sondaggio per ogni quartiere di Trento (i quartieri per i quali non ci sono voti non vengono mostrati). In alto a destra dello schermo viene mostrata l'interfaccia per aggiungere nuovi voti. In basso a destra dello schermo viene mostrata la lista dei voti e i pulsanti "Esci dal sondaggio", "Chiudi e completa la sessione", "Cancella definitivamente".
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 3**

- **Descrizione:** Salvataggio di un sondaggio in corso tramite il pulsante "Esci dal sondaggio".
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Il sondaggio non viene modificato, non viene cancellato, e non viene completato. Il sistema ritorna alla pagina di gestione dei sondaggi.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 4.1**

- **Descrizione:** Eliminazione di un sondaggio in corso tramite il pulsante "Cancella definitivamente".
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Il sistema mostra una finestra "Conferma eliminazione", una volta premuto il pulsante "Elimina" il sondaggio viene cancellato. Una notifica di successo "Sondaggio eliminato con successo" appare in alto a destra. Il sistema ritorna alla pagina di gestione dei sondaggi.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 4.2**

- **Descrizione:** Annullamento dell'eliminazione nella finestra "Conferma eliminazione".
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 4.1 e aver constatato che la finestra "Conferma eliminazione" viene mostrata quando si preme il pulsante "Cancella definitivamente".
- **Risultato Atteso:** Se si preme il pulsante "Annulla" nella finestra "Conferma eliminazione" il sondaggio non viene eliminato. Una notifica di informazione "Operazione annullata" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 5.1**

- **Descrizione:** Completamento di un sondaggio in corso tramite il pulsante "Chiudi e completa la sessione".
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Il sistema mostra una finestra "Conferma chiusura", una volta premuto il pulsante "Chiudi" il sondaggio viene completato e aggiunto alla lista "Sessioni completate". Il sistema ritorna alla pagina di gestione dei sondaggi.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 5.2**

- **Descrizione:** Annullamento del completamento nella finestra "Conferma chiusura".
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 5.1 e aver constatato che la finestra "Conferma chiusura" viene mostrata quando si preme il pulsante "Chiudi e completa la sessione".
- **Risultato Atteso:** Se si preme il pulsante "Annulla" nella finestra "Conferma chiusura" il sondaggio non viene completato. Una notifica di informazione "Operazione annullata" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

4.5.6 RF11: Gestione Voti

- **Test Case 1.1**

- **Descrizione:** Aggiunta di un voto.
- **Test Data:** [quartiere] non vuoto, [età] non vuota, [valutazione] selezionata.
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Viene aperta una pagina di selezione della valutazione, una volta selezionata una valutazione e premuto il pulsante "Vota" il voto viene aggiunto alla lista, il numero dei voti nel sondaggio viene aumentato, e l'età media parziale del sondaggio viene aggiornata. Il sistema presenta una schermata "Grazie per aver votato!", premendo il pulsante "Chiudi" si ritorna alla pagina di gestione del sondaggio.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 1.2**

- **Descrizione:** Aggiunta di un voto senza specificare il quartiere.
- **Test Data:** [quartiere] vuoto.
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Non viene aperta la pagina di selezione della valutazione. Una notifica di errore "Compilare tutti i campi" appare in alto a destra.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 1.3**

- **Descrizione:** Aggiunta di un voto senza specificare l'età.
- **Test Data:** [quartiere] non vuoto, [età] vuota.
- **Precondizioni:**
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 1.1.
- **Risultato Atteso:** Il processo di voto procede ugualmente al caso di test 1.1, ma la media parziale del sondaggio rimane uguale.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 1.4**

- **Descrizione:** Annullamento di un voto con il pulsante "Annulla".
- **Test Data:** [quartiere] non vuoto.
- **Precondizioni:**
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 1.1.
- **Risultato Atteso:** Il voto non viene aggiunto al sondaggio. Il sistema ritorna alla pagina di gestione del sondaggio.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 2.1**

- **Descrizione:** Eliminazione di un voto.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** Il voto viene eliminato dalla lista dei voti del sondaggio, il numero di voti nel sondaggio diminuisce, e l'età media parziale del sondaggio viene aggiornata. Una notifica di successo "Voto rimosso con successo" appare in alto a destra.
- **Risultato Riscontrato:**
- **Note:**

- **Test Case 2.2**

- **Descrizione:** Eliminazione di un voto due volte cliccando di nuovo l'icona di eliminazione prima che il voto sparisca dall'interfaccia.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 2.1 e aver constatato che l'eliminazione dei voti funziona in modo corretto.
- **Risultato Atteso:** Al primo click il voto viene eliminato normalmente come nel caso di test 2.1, ad ogni altro click successivo non succede niente.
- **Risultato Riscontrato:**
- **Note:**

4.5.7 RF12: Interazione Con La Tabella

- **Test Case 1.1**

- **Descrizione:** Spostamento del focus della tabella utilizzando la rotella del mouse.
- **Test Data:**
- **Precondizioni:**

- **Dipendenze:**
- **Risultato Atteso:** La tabella scorre in alto quando la rotella del mouse viene spostata in alto, e scorre in basso quando la rotella del mouse viene spostata in basso.
- **Risultato Riscontrato:**
- **Note:**
- **Test Case 1.2**
 - **Descrizione:** Spostamento del focus della tabella utilizzando la barra al lato.
 - **Test Data:**
 - **Precondizioni:**
 - **Dipendenze:**
 - **Risultato Atteso:** La tabella scorre nella direzione in cui viene trascinata la barra al lato.
 - **Risultato Riscontrato:**
 - **Note:**
- **Test Case 2.1**
 - **Descrizione:** Selezione di una zona cliccando sulla sua riga nella tabella.
 - **Test Data:**
 - **Precondizioni:**
 - **Dipendenze:**
 - **Risultato Atteso:** La riga cliccata cambia colore e gli attributi della zona vengono mostrati sulla sinistra dello schermo.
 - **Risultato Riscontrato:**
 - **Note:**
- **Test Case 2.2**
 - **Descrizione:** Selezione di una zona cliccando sulla sua riga nella tabella quando una zona è già selezionata.
 - **Test Data:**
 - **Precondizioni:** Una zona è selezionata.
 - **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 2.1, cliccando su una riga diversa da quella cliccata nel test 2.1.
 - **Risultato Atteso:** La zona precedentemente selezionata viene deselezionata. La riga cliccata cambia colore e gli attributi della zona vengono mostrati sulla destra dello schermo.
 - **Risultato Riscontrato:**
 - **Note:**
- **Test Case 2.3**
 - **Descrizione:** Deselezione di una zona cliccando sulla sua riga nella tabella.
 - **Test Data:**
 - **Precondizioni:** Una zona è selezionata.
 - **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto i test 2.1 e 2.2, cliccando sulla stessa riga cliccata nel test 2.2.
 - **Risultato Atteso:** Gli attributi demografici e riguardanti la soddisfazione di Trento vengono mostrati sulla sinistra. La riga cliccata smette di essere selezionata.
 - **Risultato Riscontrato:**
 - **Note:**
- **Test Case 3.1**

- **Descrizione:** Selezione della divisione per quartieri.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** La tabella mostra gli attributi generali e riguardanti la soddisfazione dei quartieri di Trento.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 3.2**

- **Descrizione:** Selezione della divisione per circoscrizioni.
- **Test Data:**
- **Precondizioni:**
- **Dipendenze:**
- **Risultato Atteso:** La tabella mostra gli attributi generali e riguardanti la soddisfazione delle circoscrizioni di Trento.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 4**

- **Descrizione:** Cambio visualizzazione da tabella a mappa.
- **Test Data:**
- **Precondizioni:** L'utente è loggato in un account Analista.
- **Dipendenze:**
- **Risultato Atteso:**
- **Risultato Ricontrato:** La tabella viene sostituita da una mappa di Trento divisa per zone, colorate da rosso a verde in base alla relativa soddisfazione media.
- **Note:**

4.5.8 RF13: Accesso Completo Agli Attributi

- **Test Case 1.1**

- **Descrizione:** Selezione di una categori di attributi.
- **Test Data:**
- **Precondizioni:** L'utente è loggato in un account Analista. Una zona è selezionata.
- **Dipendenze:**
- **Risultato Atteso:** La scritta del nome della categoria cliccata cambia colore, sulla sinistra dello schermo vengono mostrati gli attributi della zona selezionata appartenenti alla categoria selezionata.
- **Risultato Ricontrato:**
- **Note:**

- **Test Case 1.2**

- **Descrizione:** Selezione di una categoria di attributi già selezionata.
- **Test Data:**
- **Precondizioni:** L'utente è loggato in un account Analista. Una zona è selezionata. Una categoria di attributi è selezionata.
- **Dipendenze:** Questo test deve essere svolto solo dopo aver svolto il test 1.1, cliccando sulla stessa categoria che è stata cliccata nel test 1.1.
- **Risultato Atteso:** La categoria cliccata rimane selezionata.
- **Risultato Ricontrato:**
- **Note:**

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*

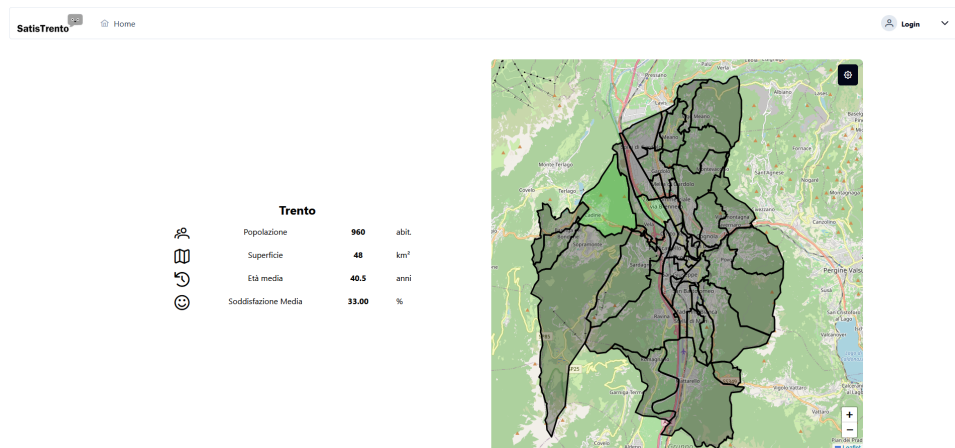


Figura 5.1: *Home Page* per utenti non loggati

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”

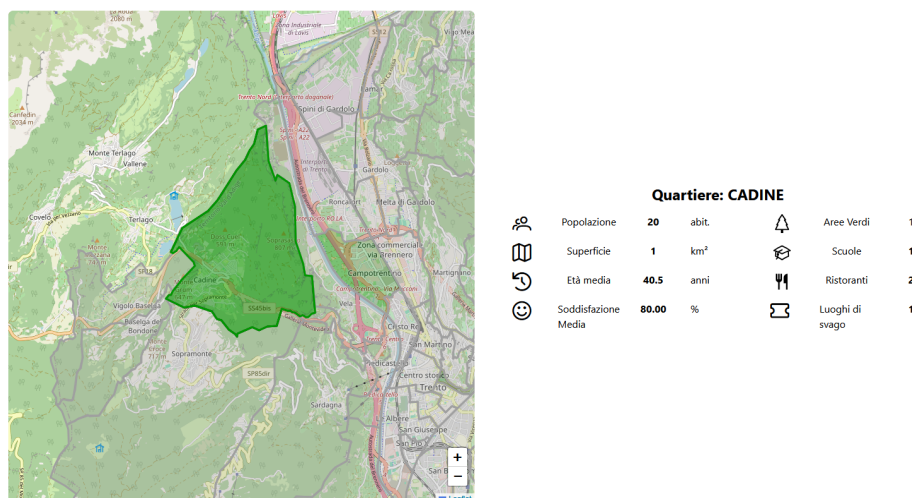


Figura 5.3: Selezione di un “quartiere”/“circonscrizione”

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 oscurano gli altri “quartieri” o le altre “circonsrizioni”, 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

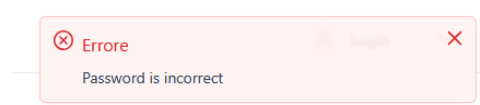


Figura 5.5: Messaggio di errore

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

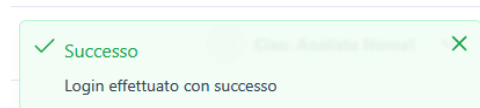


Figura 5.6: Messaggio di login effettuato

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.

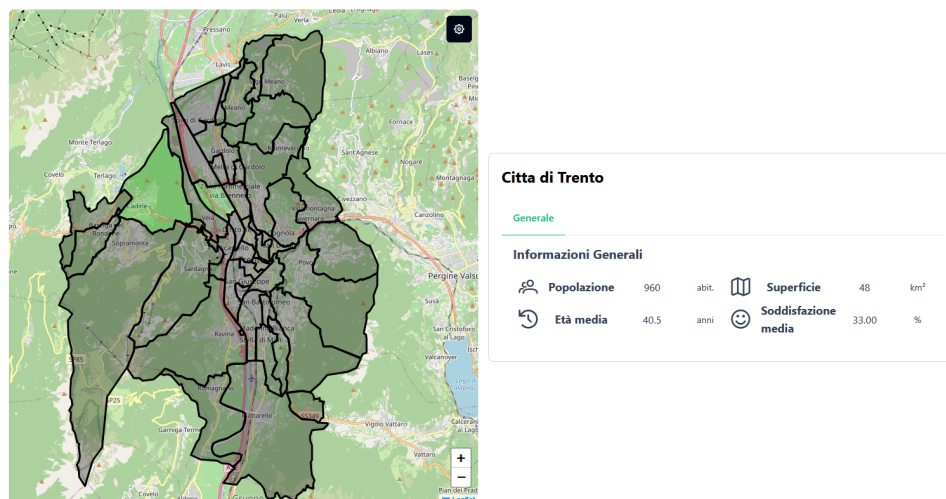


Figura 5.7: Home Page per utenti con ruolo “analista”

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

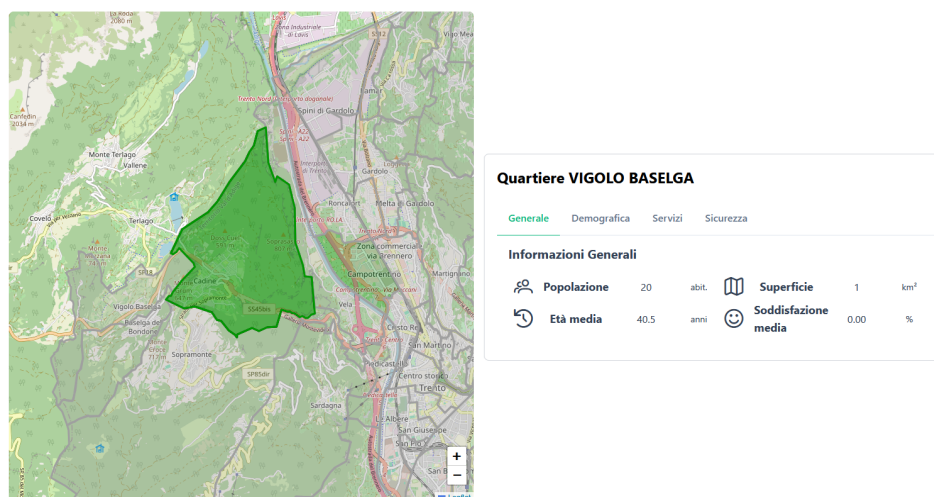


Figura 5.8: Selezione di un “quartiere”/“circonscrizione” per utenti con ruolo “analista”

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”

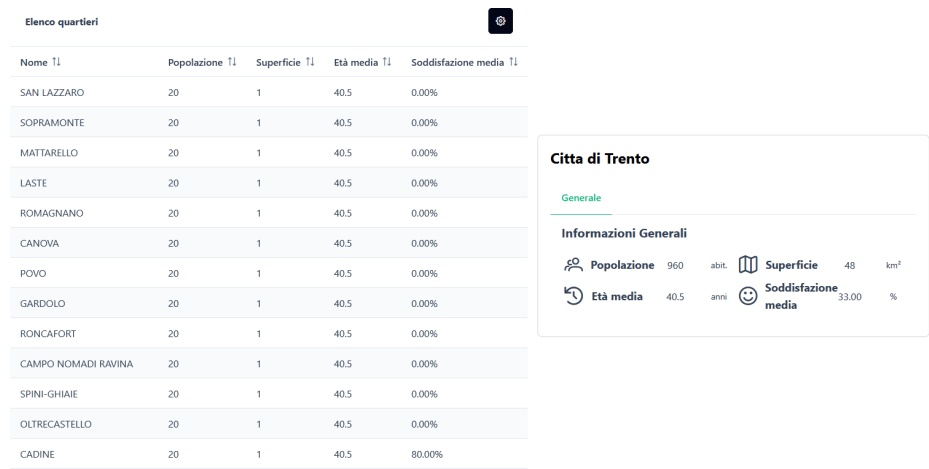


Figura 5.10: Visualizzazione dati in “tabella”

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

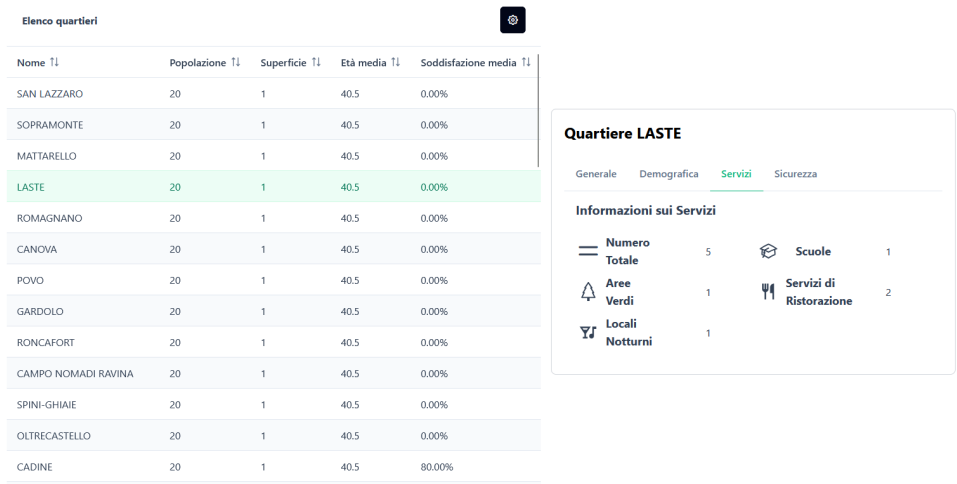


Figura 5.11: “Quartiere”/“circoscrizione” selezionata in visualizzazione tabella

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

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.

