

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	19
4.1 Repository Organization	19
4.2 <i>Branching Strategy</i> e organizzazione del lavoro	20
4.3 <i>Dependency</i>	20
4.4 <i>Database</i>	21
4.5 Testing	23
5 FrontEnd	33
6 Deployment	34

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

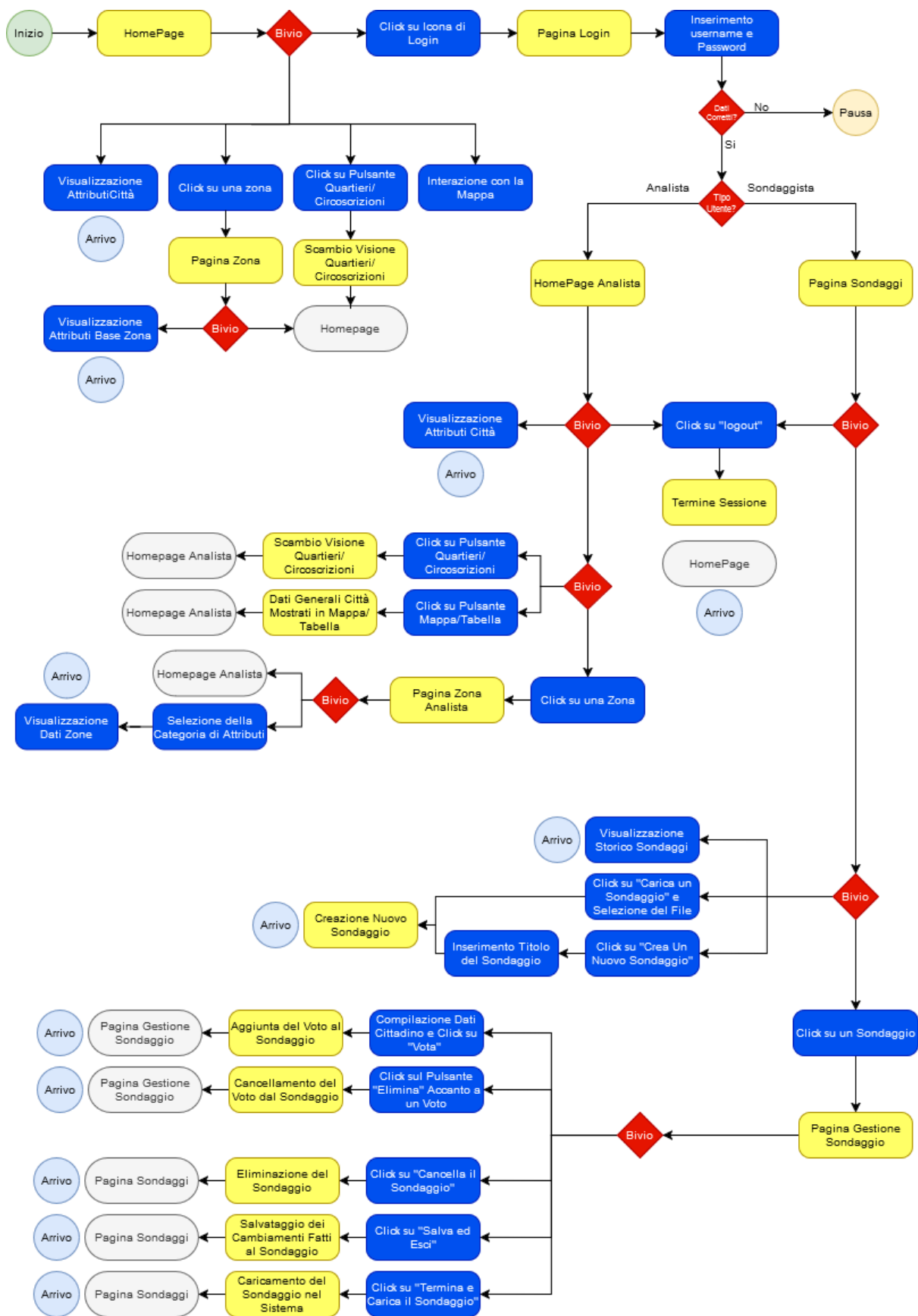


Figura 2.2: User Workflow

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

```

481     schema:
482         type: boolean
483         default: true
484         allowEmptyValue: true
485     get:
486         summary: "Ottieni informazioni di una circoscrizione"
487         responses:
488             200:
489                 description: "Risposta con le informazioni della circoscrizione."
490                 content:
491                     application/json:
492                         schema:
493                             oneOf:
494                                 - $ref: "#/components/schemas/Circoscrizione"
495                                 - $ref: "#/components/schemas/CircoscrizioneNC"
496             404:
497                 description: "Circoscrizione non trovata."
498                 content:
499                     application/json:
500                         schema:
501                             "$ref": "#/components/schemas/Error"
502         tags:
503             - Circoscrizioni
504             - Dati Statici
505     '/generalInfo' :
506         description: "Risorsa che restituisce le informazioni generali della città di Trento."
507         get:
508             summary: "Ottieni informazioni generali"
509             description: "Restituisce le informazioni generali della città di Trento."
510             responses:
511                 200 :
512                     description: "Successful response"
513                     content:
514                         application/json:
515                             schema:
516                                 type: "object"
517                                 properties:
518                                     self:
519                                         type: "string"
520                                         description: "Url della risorsa."
521                                     popolazione :
522                                         type: integer
523                                         description: "Numero di abitanti della città di Trento."
524                                     superficie :
525                                         type: number
526                                         description: "Superficie della città di Trento in km^2."
527                                         format: double
528                                     etaMedia :
529                                         type: number
530                                         description: "Età media degli abitanti della città di Trento."
531                                         format: double
532                                     soddisfazioneMedia:
533                                         type: number
534                                         description: "Soddisfazione media degli abitanti della città di Trento."
535                                         format: double
536                                         minimum: 0
537                                         maximum: 100
538             tags:
539                 - Dati Statici
540     '/session':
541         description: "Risorsa che restituisce le informazioni della sessione dell'utente."
542         get:
543             summary: "Ottieni informazioni della sessione"
544             description: "Restituisce le informazioni della sessione dell'utente, se presente."
545             security:
546                 - BearerAuth: []
547             responses:
548                 200:
549                     description: "Successful response"
550                     content:
551                         application/json:
552                             schema:
553                                 "$ref": "#/components/schemas/User"

```



```
554     401:
555         description: "Utente non autenticato."
556         content:
557             application/json:
558                 schema:
559                     "$ref": "#/components/schemas/Error"
560     404:
561         description: "Sessione non trovata."
562         content:
563             application/json:
564                 schema:
565                     "$ref": "#/components/schemas/Error"
566     tags:
567     - Utenti
568     post:
569         summary: "Crea una nuova sessione (aka login)"
570         description: "Crea una nuova sessione per l'utente. Altrimenti conosciuto come login."
571         requestBody:
572             required: true
573             content:
574                 application/json:
575                     schema:
576                         type: object
577                         properties:
578                             email:
579                                 type: string
580                                 description: "Email dell'utente."
581                             password:
582                                 type: string
583                                 description: "Password (codificata) dell'utente."
584         responses:
585             201:
586                 description: "Sessione creata con successo."
587                 content:
588                     application/json:
589                         schema:
590                             "$ref": "#/components/schemas/Session"
591                 headers:
592                     Location:
593                         description: "Url della risorsa creata."
594                         schema:
595                             type: string
596             400:
597                 description: "Dati della sessione non validi."
598                 content:
599                     application/json:
600                         schema:
601                             "$ref": "#/components/schemas/Error"
602             401:
603                 description: "Credenziali non valide."
604                 content:
605                     application/json:
606                         schema:
607                             "$ref": "#/components/schemas/Error"
608     tags:
609     - Utenti
610     delete:
611         summary: "Elimina la sessione (aka logout)"
612         description: "Elimina la sessione dell'utente revocando il token di autenticazione al server,
613 ↪ altrimenti conosciuto come logout."
614         security:
615         - BearerAuth: []
616         responses:
617             204:
618                 description: "Sessione eliminata con successo."
619             401:
620                 description: "Sessione non trovata."
621                 content:
622                     application/json:
623                         schema:
624                             "$ref": "#/components/schemas/Error"
625     tags:
626     - Utenti
```

```
626 '/sondaggi' :
627   description: "Risorsa che restituisce la lista dei sondaggi dell'utente corrente, se amministratore
        ↳ allora restituisce tutti i sondaggi."
628   get:
629     summary: "Ottieni lista dei sondaggi"
630     description: "Restituisce la lista dei sondaggi dell'utente corrente. Se l'utente è amministratore,
        ↳ restituisce tutti i sondaggi."
631     security:
632       - BearerAuth: []
633     parameters:
634       - name: deepData
635         in: query
636         description: "Se impostato a true, restituisce le informazioni complete di ogni sondaggio."
637         required: false
638         schema:
639           type: boolean
640           default: false
641         allowEmptyValue: true
642     responses:
643       200:
644         description: "Successful response"
645         content:
646           application/json:
647             schema:
648               "$ref": "#/components/schemas/ElencoSondaggi"
649       401:
650         description: "Utente non autenticato."
651         content:
652           application/json:
653             schema:
654               "$ref": "#/components/schemas/Error"
655       403:
656         description: "Utente loggato ma non ha i permessi per visualizzare i sondaggi."
657         content:
658           application/json:
659             schema:
660               "$ref": "#/components/schemas/Error"
661     tags:
662       - Sondaggi
663       - Sondaggista
664   post:
665     summary: "Crea un nuovo sondaggio"
666     description: "Crea un nuovo sondaggio. Solo per utenti con ruolo 'sondaggista'."
667     security:
668       - BearerAuth: []
669     requestBody:
670       required: true
671       content:
672         application/json:
673           schema:
674             "$ref": "#/components/schemas/AddSondaggio"
675     responses:
676       201:
677         description: "Sondaggio creato con successo."
678         headers:
679           Location:
680             description: "Url della risorsa creata."
681             schema:
682               type: string
683       400:
684         description: "Dati del sondaggio non validi."
685         content:
686           application/json:
687             schema:
688               "$ref": "#/components/schemas/Error"
689       401:
690         description: "Utente non autenticato."
691         content:
692           application/json:
693             schema:
694               "$ref": "#/components/schemas/Error"
695       403:
696         description: "Utente loggato ma non ha i permessi per creare un sondaggio."
```

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

```
769         type: string
770     400:
771         description: "Dati del sondaggio non validi."
772         content:
773             application/json:
774                 schema:
775                     "$ref": "#/components/schemas/Error"
776     401:
777         description: "Utente non autenticato."
778         content:
779             application/json:
780                 schema:
781                     "$ref": "#/components/schemas/Error"
782     403:
783         description: "Utente loggato ma non ha i permessi per modificare il sondaggio."
784         content:
785             application/json:
786                 schema:
787                     "$ref": "#/components/schemas/Error"
788     404:
789         description: "Sondaggio non trovato."
790         content:
791             application/json:
792                 schema:
793                     "$ref": "#/components/schemas/Error"
794     tags:
795     - Sondaggi
796     - Sondaggista
797 delete:
798     summary: "Elimina un sondaggio"
799     description: "Elimina un sondaggio. Solo l'utente che ha creato il sondaggio può eliminarlo o un
800     ↳ amministratore, se il sondaggio è in attesa. Se il sondaggio è approvato o rifiutato, solo
801     ↳ l'amministratore può eliminarlo."
802     security:
803     - BearerAuth: []
804     responses:
805     204:
806         description: "Sondaggio eliminato con successo."
807     401:
808         description: "Utente non autenticato."
809         content:
810             application/json:
811                 schema:
812                     "$ref": "#/components/schemas/Error"
813     403:
814         description: "Utente loggato ma non ha i permessi per eliminare il sondaggio."
815         content:
816             application/json:
817                 schema:
818                     "$ref": "#/components/schemas/Error"
819     404:
820         description: "Sondaggio non trovato."
821         content:
822             application/json:
823                 schema:
824                     "$ref": "#/components/schemas/Error"
825     tags:
826     - Sondaggi
827     - Sondaggista
828 '/voti' :
829     description: "Risorsa per la gestione dei voti di un sondaggio."
830     parameters:
831     - name: idSondaggio
832       in: query
833       description: "Identificatore univoco del sondaggio."
834       required: true
835       schema:
836         type: string
837     get:
838     summary: "Ottieni lista dei voti di un sondaggio"
839     description: "Restituisce la lista dei voti assegnati al sondaggio."
840     security:
841     - BearerAuth: []
```

```
840     responses:
841       200:
842         description: "Successful response"
843         content:
844           application/json:
845             schema:
846               type: array
847               items:
848                 "$ref": "#/components/schemas/Voto"
849       401:
850         description: "Utente non autenticato."
851         content:
852           application/json:
853             schema:
854               "$ref": "#/components/schemas/Error"
855       403:
856         description: "Utente loggato ma non ha i permessi per visualizzare i voti."
857         content:
858           application/json:
859             schema:
860               "$ref": "#/components/schemas/Error"
861       404:
862         description: "Sondaggio non trovato."
863         content:
864           application/json:
865             schema:
866               "$ref": "#/components/schemas/Error"
867     tags:
868     - Sondaggi
869     - Sondaggista
870   post:
871     summary: "Aggiungi un voto"
872     description: "Aggiunge un voto al sondaggio."
873     security:
874     - BearerAuth: []
875     requestBody:
876       required: true
877       content:
878         application/json:
879           schema:
880             "$ref": "#/components/schemas/AddVoto"
881     responses:
882       201:
883         description: "Voto aggiunto con successo."
884         headers:
885           Location:
886             description: "Url della risorsa creata."
887             schema:
888               type: string
889       400:
890         description: "Dati del voto non validi."
891         content:
892           application/json:
893             schema:
894               "$ref": "#/components/schemas/Error"
895       401:
896         description: "Utente non autenticato."
897         content:
898           application/json:
899             schema:
900               "$ref": "#/components/schemas/Error"
901       403:
902         description: "Utente loggato ma non ha i permessi per aggiungere un voto."
903         content:
904           application/json:
905             schema:
906               "$ref": "#/components/schemas/Error"
907       404:
908         description: "Sondaggio non trovato."
909         content:
910           application/json:
911             schema:
912               "$ref": "#/components/schemas/Error"
```

```
913     tags:
914     - Sondaggi
915     - Sondaggista
916   '/voti/{idVoto}' :
917     description: "Risorsa per la gestione di un voto di un sondaggio."
918     parameters:
919     - name: idVoto
920       in: path
921       description: "Identificatore univoco del voto."
922       required: true
923       schema:
924         type: string
925     delete:
926       summary: "Elimina un voto"
927       description: "Elimina un voto. Solo l'utente che ha effettuato il voto può eliminarlo"
928       security:
929       - BearerAuth: []
930
931     responses:
932       204:
933         description: "Voto eliminato con successo."
934       401:
935         description: "Utente non autenticato."
936         content:
937           application/json:
938             schema:
939               "$ref": "#/components/schemas/Error"
940       403:
941         description: "Utente loggato ma non ha i permessi per eliminare il voto."
942         content:
943           application/json:
944             schema:
945               "$ref": "#/components/schemas/Error"
946       404:
947         description: "Voto non trovato."
948         content:
949           application/json:
950             schema:
951               "$ref": "#/components/schemas/Error"
952     tags:
953     - Sondaggi
954     - 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
│   ├── /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

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;

`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;

`localiNotturni`

`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

Testo di introduzione della parte di testing...

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 Riscontrato:**

- **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 Riscontrato:**
- **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 Riscontrato:**
 - **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 Riscontrato:**
 - **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 Riscontrato:**
 - **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 Ricontrato:**
- **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 Ricontrato:**
 - **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 Ricontrato:**
 - **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 Ricontrato:**
 - **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 categoria 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

6 Deployment

