

# Deliverable 1 v0.2

Riccardo Sandru, Ernesto Beltrami, Taufge Songne

November 24, 2025

## Abstract

Add  
abstract

## Contents

<b>1</b>	<b>Il Progetto IoSonoTrento</b>	<b>3</b>
1.1	Il problema che stiamo risolvendo	3
1.2	Obiettivo del progetto	3
1.3	Vantaggi per il comune	3
1.4	Vantaggi per gli utenti	3
1.5	Limitazioni dell'applicazione	4
1.6	Cio' che ci aspettiamo	4
<b>2</b>	<b>Requisiti Funzionali</b>	<b>4</b>
2.1	Requisiti funzionali comuni per operatore e utente	4
2.2	Requisiti funzionali solo per operatore	4
2.3	Requisiti funzionali solo per cittadini	5
<b>3</b>	<b>Requisiti non funzionali</b>	<b>5</b>
3.1	Compatibilità	5
3.2	Sicurezza e privacy	6
3.3	Prestazioni	6
3.4	Manutenibilità ed Estendibilità	7
3.5	Compatibilità	7
<b>4</b>	<b>Use Case</b>	<b>7</b>
4.1	RF1: Gestione Accessi	7
4.2	RF2: Accesso alla Dashboard	8
4.3	RF3: Riepilogo	8
4.4	RF4: Ricerca	9
4.5	RF5: Creazione Votazioni	9
4.6	RF6: Gestione Votazioni	10
4.7	RF7: Creazione Sondaggi	10
4.8	RF8: Gestione Sondaggi	10
4.9	RF9: Moderazione Contenuti	10
4.10	RF10: Gestione Iniziative dei Cittadini	11
4.11	RF11: Votazione Iniziative	11
4.12	RF12: Votazioni e Sondaggi (Invio Univoco)	11
4.13	RF13: Accesso alla dashboard per cittadini attivi	11
<b>5</b>	<b>User Stories</b>	<b>11</b>
5.1	User Story 1: Gestione Accessi (UC RF1)	11
5.2	User Story 2: Ricerca (UC RF4)	12
5.3	User Story 3: Ricerca (UC RF4)	12
5.4	User Story 4: Creazione Votazioni (UC RF5)	13
5.5	User Story 5: Gestione Votazioni (UC RF6)	13
5.6	User Story 6: Gestione Iniziative (UC RF10)	13
5.7	User Story 7: Votazione Iniziative (UC RF11)	14
5.8	User Story 8: Votazioni (UC RF12)	14
5.9	User Story 9: Moderazione Contenuti (UC RF9)	14

5.10 User Story 10: Dashboard attività attive (UC RF13) . . . . .	15
---	----

# 1 Il Progetto IoSonoTrento

## 1.1 Il problema che stiamo risolvendo

Il problema che stiamo affrontando riguarda il basso coinvolgimento dei cittadini nei processi decisionali del Comune. Attualmente, seguire i processi decisionali del comune o informarsi sugli argomenti di discussione amministrativa, risulta spesso poco accessibile e presenta criticità in termini di comodità d'uso. I cittadini di per sé hanno diritto a partecipare nella vita comunale in diversi modi, tramite diversi strumenti come le istanze, le petizioni e proposte di deliberazione e infine le elezioni comunali. La legge 241/1990 è una legge che regola tutto l'operato di un'amministrazione e tratta un tema molto importante come la trasparenza. Un comune che rende accessibile i dati (a seconda del tipo di sensibilità) riceve una maggior fiducia da parte dei cittadini, aumentando il senso di benessere comune. I cittadini quindi in genere percepiscono una lontananza dall'amministrazione, non si rendono conto di quanto il loro ruolo sia importante per il futuro della città e di come le loro scelte possano contribuire allo sviluppo della città. La loro voce si sente lontana. In un'epoca in cui la tecnologia digitale fa parte della vita quotidiana, bisogna cercare di adottare strumenti innovativi per migliorare la comunicazione tra Comune e cittadini.

## 1.2 Obiettivo del progetto

Il progetto ha come obiettivo la realizzazione di un'applicazione web dedicata alla partecipazione civica, accessibile via browser e destinata a due principali categorie di utenti: cittadini e amministrazione comunale. Il sistema sarà progettato per facilitare l'inclusione dei cittadini nei processi decisionali del Comune. Infatti, i cittadini saranno in grado di proporre e votare idee, partecipare a sondaggi rapidi e saranno in grado di vedere i risultati conoscendo anche l'opinione dei loro concittadini. Gli operatori del comune invece potranno realizzare dei sondaggi riguardanti l'ordine del giorno che discutono, le assemblee e le proposte di iniziative che vorrebbero portare avanti. Inoltre potranno osservare in maniera dettagliata il risultato dei sondaggi, usandolo per fare un'analisi più profonda ed accurata ad esempio in base alla struttura demografica. L'app avrà un linguaggio chiaro, semplice e diretto, facilmente comprensibile da tutti. E' necessario che sia conforme alle normative europee e dunque la protezione dei dati dei cittadini sarà una priorità assoluta. I dati sensibili saranno protetti in modo rigoroso e verrà garantito l'anonimato dell'utente che vota. E' importante che ciò venga fatto per proteggere la privacy dell'utente e le sue opinioni personali.

## 1.3 Vantaggi per il comune

1. **scelte data-driven:** Il comune avrà una risorsa in più, sulla quale orientare le sue decisioni osservando ad esempio a livello demografico quale sia la fascia più interessata ad una determinata proposta.
2. **migliore gestione delle risorse:** Le risorse del comune potranno essere gestite ancora meglio di prima, facendo scelte più precise e mirate. Dunque si tratta di uno strumento che può avere impatto anche a livello economico.
3. **trasparenza sui progetti in corso:** Il comune nel caso in cui prendesse in considerazione una delle proposte avanzate sulla piattaforma, terrà aggiornati gli utenti sull'andamento del progetto.
4. **fiducia istituzionale:** La trasparenza comporta un aumento della fiducia istituzionale dei cittadini verso il Comune. Ciò contribuisce ad un aumento della qualità di vita, aspetto che comporta ad avere una città sana ed ottimista.

## 1.4 Vantaggi per gli utenti

1. **cittadinanza attiva:** I cittadini non rimangono in disparte, ma passano anche all'azione. Nasce un reale coinvolgimento di tutti che si sentono partecipe di un movimento, di una città unita e che guarda verso il futuro.
2. **maggior consapevolezza:** Da un lato, il cittadino conosce in parte cosa succede in città e dall'altra sa che le sue scelte possono contribuire ad un cambiamento.
3. **strumento intuitivo:** L'utente ha a disposizione uno strumento semplice, intuitivo e facile da utilizzare. La grafica della piattaforma è pensata per rendere chiaro quali siano le azioni che si possano compiere, ignorando informazioni superflue.

## 1.5 Limitazioni dell'applicazione

- **campione di voti non assoluto:** I voti o le risposte dei sondaggi non rappresentano l'opinione di tutta la cittadinanza di Trento. Probabilmente non tutti i cittadini di Trento utilizzeranno la piattaforma. Dunque, i dati rappresenteranno soltanto un gruppo di persone.
- **limiti di accessibilità per quanto riguarda persone con fascia alta d'età:** Nonostante il fatto che la piattaforma sia pensata per essere di facile utilizzo, la partecipazione della fascia d'età più anziana potrebbe essere limitata. Ci aspettiamo quindi una concentrazione di utenti in una fascia di età compresa tra i 18 e 60 anni.
- **Limitazione dell'accesso degli utenti:** L'applicazione è pensata per consentire l'accesso esclusivamente agli utenti residenti nel comune di Trento. Per implementare un controllo rigoroso dell'identità e della residenza sarebbe necessario integrare SPID come metodo di autenticazione, poiché permette di verificare in modo sicuro le informazioni personali degli utenti. Tuttavia, per motivi di certificazioni da parte di SPID e CieID, l'attuale versione dell'applicazione utilizzerà l'accesso tramite account Google come metodo sostitutivo, senza garanzia di limitazione geografica. Di conseguenza, la verifica della residenza non è pienamente garantita in questa release. prototipo

## 1.6 Cio' che ci aspettiamo

L'obiettivo finale quindi sarebbe quello di fornire uno strumento che possa avere un impatto sulla vita quotidiana trentina e sul futuro trentino, facendo scelte più accurate.

## 2 Requisiti Funzionali

### 2.1 Requisiti funzionali comuni per operatore e utente

**RF1 Gestione accessi:** La piattaforma deve consentire l'accesso agli utenti. Gli utenti si registrano tramite Google Authentication; al primo accesso, l'ID univoco Google viene memorizzato nel database insieme alle informazioni necessarie per consentire i successivi login. Gli operatori del Comune accedono alla dashboard amministrativa tramite nome utente e password predefiniti nel database.

**RF2 Accesso alla Dashboard:** Tutti gli utenti autenticati devono poter accedere alla propria dashboard:

- Ai cittadini viene mostrata una dashboard con le votazioni attive, i risultati delle votazioni concluse, i sondaggi disponibili e la bacheca delle iniziative cittadine.
- Agli operatori comunali è riservata una dashboard estesa che consente di avviare votazioni e sondaggi, monitorare risultati e gestire i contenuti pubblicati dai cittadini.

individuare  
esattamente  
delle  
funzioni

**RF3 Riepilogo:** Tutti gli utenti devono poter visualizzare un riepilogo dei risultati delle votazioni. I cittadini vedono un riepilogo sintetico e anonimizzato: percentuale e numero assoluto di voti per scelta, mentre gli operatori possono accedere a un livello di dettaglio maggiore per effettuare analisi sulle preferenze espresse.

**RF4 Ricerca:** L'applicazione deve consentire la ricerca di iniziative e contenuti pubblicati dai cittadini tramite una barra di ricerca, disponibile sia per utenti che per operatori. Potranno essere applicati filtri (ad esempio per argomento, per votazioni, per data,...) per ordinare i risultati rendendo la ricerca più rapida.

### 2.2 Requisiti funzionali solo per operatore

**RFO1 Creazione e gestione delle votazioni:** Il sistema deve consentire agli operatori del Comune di creare, configurare e gestire le votazioni relative agli argomenti discussi in sede comunale. Ogni votazione potrà comprendere uno o più temi, ciascuno corredato da una breve descrizione e da eventuali materiali informativi allegati. Il sistema deve consentire all'operatore di inserire un numero fisso di risposte per ogni tema ed altre opzioni come votazione a risposta singola o multipla. L'operatore deve avere la possibilità di impostare un limite di tempo per la durata della votazione, definendo una data e un'ora di chiusura oltre le quali non sarà più possibile esprimere voti. Inoltre, l'interfaccia dovrà permettere

di modificare o eliminare votazioni già create, garantendo al contempo la tracciabilità delle operazioni effettuate per motivi di trasparenza e controllo.

**RFO2 Creazione e gestione dei sondaggi:** L'applicazione deve permettere agli operatori di creare, pubblicare e gestire mini-sondaggi anonimi rivolti agli utenti registrati. Tali sondaggi possono essere collegati a specifiche votazioni o riguardare tematiche di interesse generale, con l'obiettivo di raccogliere rapidamente opinioni e percezioni della cittadinanza. Ogni sondaggio deve poter essere configurato con un numero variabile di domande a risposta chiusa o multipla, e deve prevedere la possibilità di impostare un periodo di validità (data di apertura e chiusura). Al termine del periodo stabilito, il sistema deve automaticamente chiudere la raccolta delle risposte e rendere disponibili i risultati agli operatori.

**RFO3 Moderazione dei contenuti generati dagli utenti:** Il sistema deve consentire agli operatori di monitorare e moderare le proposte e i contenuti pubblicati dai cittadini sulla piattaforma. In particolare, gli operatori devono poter eliminare o segnalare contenuti che violano le linee guida della community o che contengono linguaggio offensivo, materiale inappropriato o non conforme alle politiche del Comune.

## 2.3 Requisiti funzionali solo per cittadini

**RFC1 Gestione delle iniziative dei cittadini:** I cittadini possono proporre nuove iniziative nella bacheca pubblica e votare quelle esistenti per aumentarne la visibilità. Ogni iniziativa deve riportare il titolo, una descrizione sintetica e il numero di voti ricevuti.

**RFC2 Votazioni:** Il sistema deve garantire che ogni cittadino possa esprimere un solo voto per ciascun tema di votazione e un solo invio di risposte per ogni sondaggio. Le preferenze espresse devono essere registrate in forma anonima e non modificabile, assicurando al contempo la tracciabilità del voto a fini di controllo senza violare la privacy del cittadino.

**RFC3 Visualizzazione votazioni e sondaggi:** Il sistema deve garantire ai cittadini di poter visualizzare sia votazioni che sondaggi attivi nella propria dashboard.

però  
come  
evitiamo  
situazioni  
di  
potere? tipo  
elimino  
qualcosa  
che  
propongono  
pk a me  
moderatore  
non mi  
piace?

## 3 Requisiti non funzionali

I requisiti non funzionali descrivono le caratteristiche qualitative del sistema *IoSonoTrento*, ovvero gli aspetti che determinano la qualità complessiva del servizio e l'esperienza d'uso, indipendentemente dalle funzionalità implementate. Essi contribuiscono a definire la solidità, l'affidabilità, la sicurezza e l'usabilità dell'applicazione, garantendo un servizio pubblico digitale efficiente e accessibile a tutti i cittadini.

(nominativo  
di chi  
vota se  
vuoi poi  
ti spiego  
in che  
senso)

### 3.1 Compatibilità

**RNF1 Interfaccia intuitiva:** l'applicazione deve essere facilmente comprensibile e navigabile anche da utenti con competenze digitali di base. *Criterio di misurazione:* durante test di usabilità con utenti rappresentativi, almeno l'80% dei partecipanti deve riuscire a individuare e utilizzare le funzionalità principali entro **3 minuti** dal primo accesso, senza consultare documentazione o richiedere assistenza. *Motivazione:* garantire un'interfaccia intuitiva favorisce l'adozione del servizio da parte di un'utenza eterogenea, riduce la curva di apprendimento, migliora l'efficienza operativa e aumenta la soddisfazione complessiva dell'utente.

**RNF2 Accessibilità:** il design dovrà rispettare le linee guida WCAG 2.1 livello AA, garantendo l'accesso anche a persone con disabilità visive o motorie. *Motivazione:* il rispetto delle norme sull'accessibilità è essenziale per assicurare pari opportunità di utilizzo del servizio, in linea con i principi di inclusività e con la normativa vigente (Legge Stanca n. 4/2004). Garantire un'esperienza accessibile significa rendere il sistema realmente pubblico e universale.

**RNF3 Consistenza visiva:** layout e componenti grafici devono risultare coerenti in tutte le sezioni dell'app, utilizzando colori istituzionali e un contrasto adeguato. *Motivazione:* la coerenza visiva rafforza l'identità del progetto, migliora la leggibilità dei contenuti e riduce la possibilità di confusione durante la navigazione, offrendo un'esperienza d'uso più fluida e professionale.

**RNF4 Multilingua:** Il linguaggio principale dell'applicazione sarà l'italiano ma l'applicazione dovrà prevedere la possibilità di estendere l'interfaccia ad altre lingue per includere tutto lo spettro dei cittadini di Trento. *Motivazione:* la presenza di un'interfaccia multilingua favorisce l'inclusione dei cittadini stranieri o non italofoni, promuovendo l'integrazione e la partecipazione civica attiva di tutte le comunità presenti sul territorio.

### 3.2 Sicurezza e privacy

**RNF5 Protezione dei dati personali:** tutti i dati utente devono essere trattati nel rispetto del GDPR (Reg. UE 2016/679). *Motivazione:* la conformità al GDPR è obbligatoria per qualsiasi sistema che gestisca informazioni personali. Garantisce la tutela dei diritti degli utenti e rafforza la fiducia nel servizio pubblico digitale.

**RNF6 Anonimato dei voti:** le preferenze espresse dagli utenti devono essere anonimizzate e non riconducibili direttamente all'identità del cittadino. *Motivazione:* l'anonimato assicura l'imparzialità del processo decisionale e la libertà di espressione, elementi fondamentali in un contesto partecipativo come quello proposto da IoSonoTrento.

**RNF7 Autenticazione sicura:** l'accesso avviene tramite Google OAuth 2.0; in futuro sarà integrabile SPID o CIE per garantire una verifica ufficiale dell'identità. *Motivazione:* un sistema di autenticazione sicuro riduce il rischio di accessi non autorizzati e consente di validare l'identità degli utenti, preservando la sicurezza del sistema e la legittimità delle interazioni.

**RNF8 Gestione sicura delle sessioni:** le sessioni devono scadere automaticamente dopo un periodo di inattività prestabilito, prevenendo accessi non autorizzati. Abbiamo deciso di impostare questo limite a 15 minuti di attività per arrivare a un compromesso tra durata della sessione e sicurezza dei dati. *Motivazione:* la gestione corretta delle sessioni riduce i rischi legati all'uso improprio dei dispositivi condivisi o smarriti, garantendo un livello aggiuntivo di protezione.

**RNF9 Crittografia dei dati:** tutte le comunicazioni client-server devono avvenire tramite protocollo HTTPS con TLS 1.3. *Motivazione:* la crittografia protegge la riservatezza e l'integrità dei dati trasmessi, impedendo intercettazioni e manomissioni da parte di terzi, requisito indispensabile per un'applicazione che gestisce informazioni sensibili.

### 3.3 Prestazioni

**RNF10 Tempi di risposta:** le operazioni principali (login, caricamento dashboard, votazione, consultazione risultati) devono completarsi entro 3 secondi in condizioni normali di rete, per garantire un'esperienza di uso fluida e piacevole. *Motivazione:* tempi di risposta rapidi sono essenziali per mantenere l'attenzione e la soddisfazione dell'utente, evitando frustrazione o abbandono del servizio. In un contesto civico come IoSonoTrento, la reattività aumenta la percezione di affidabilità e professionalità della piattaforma.

**RNF11 Scalabilità:** il sistema deve poter gestire un aumento del numero di utenti registrati e di votazioni attive senza compromettere le prestazioni. Ci si aspetta di dover gestire al massimo un numero di utenti uguale alla popolazione della città di Trento (120.000 persone al giorno 24 Novembre 2025) *Motivazione:* la scalabilità assicura che l'applicazione resti efficiente anche in caso di campagne di partecipazione massiva o durante eventi locali con un forte coinvolgimento cittadino. Permette inoltre una crescita sostenibile del sistema nel tempo.

**RNF12 Disponibilità:** l'applicazione deve essere disponibile almeno per il 99% del tempo operativo mensile. *Motivazione:* un'elevata disponibilità è indispensabile per un servizio pubblico digitale, poiché garantisce che i cittadini possano accedere al sistema in qualunque momento. Ridurre i tempi di inattività aumenta l'affidabilità percepita e la fiducia degli utenti.

**RNF13 Backup e recupero dati:** deve essere implementato un sistema di backup giornaliero del database con possibilità di restore in caso di perdita di dati. *Motivazione:* la sicurezza dei dati è un aspetto critico per qualsiasi applicazione pubblica. I backup giornalieri garantiscono la continuità operativa e la possibilità di ripristino rapido in caso di guasti o attacchi informatici.

**RNF14 Gestione degli errori:** in caso di errore o malfunzionamento, il sistema deve informare l'utente con messaggi chiari e non tecnici, evitando la perdita di dati. *Motivazione:* una gestione efficace degli errori contribuisce a migliorare l'esperienza utente e riduce la frustrazione.

Inoltre, messaggi informativi non tecnici favoriscono la comprensione del problema e riducono la necessità di assistenza.

**RNF15 Monitoraggio:** dovranno essere previsti strumenti di logging e monitoraggio del sistema per identificare anomalie e tentativi di accesso non autorizzati. *Motivazione:* il monitoraggio costante del sistema è essenziale per prevenire incidenti di sicurezza, diagnosticare problemi prestazionali e migliorare la manutenzione proattiva. Garantisce un controllo continuo sullo stato di salute dell'infrastruttura.

### 3.4 Manutenibilità ed Estendibilità

**RNF16 Architettura modulare:** il sistema sarà sviluppato seguendo principi di separazione dei componenti (front-end, back-end, database) per facilitare aggiornamenti futuri. *Motivazione:* un'architettura modulare semplifica l'individuazione dei problemi e consente di aggiornare o sostituire parti del sistema senza compromettere l'intero servizio. Favorisce inoltre la collaborazione tra team di sviluppo diversi.

**RNF17 Documentazione:** il codice dovrà essere accompagnato da una documentazione tecnica che ne descriva l'architettura e le API principali. *Motivazione:* la documentazione garantisce la continuità del progetto nel tempo, consentendo a nuovi sviluppatori di comprendere rapidamente la struttura del sistema. È inoltre fondamentale per facilitare manutenzione, audit e aggiornamenti futuri.

**RNF18 Testabilità:** ogni componente deve essere testabile singolarmente (unit testing) e in integrazione (integration testing). *Motivazione:* la testabilità riduce la probabilità di errori in fase di rilascio e migliora la qualità complessiva del software. Un approccio basato sui test consente di identificare rapidamente regressioni o malfunzionamenti durante l'evoluzione del sistema.

### 3.5 Compatibilità

**RNF19 Dispositivi:** la piattaforma dovrà essere pienamente fruibile sia da browser desktop che da dispositivi mobili (responsive design). *Motivazione:* garantire la compatibilità con dispositivi mobili è fondamentale per un servizio rivolto a cittadini che accedono in mobilità. Il responsive design migliora l'usabilità e amplia la platea di utenti potenziali.

**RNF20 Browser supportati:** compatibilità garantita con le versioni recenti di Chrome, Firefox, Safari ed Edge. L'obiettivo è quello di supportare "l'ultima versione stabile meno 1" come standard. Al giorno 24 Novembre 2025 queste versioni sono:

- (a) Chrome: 140+
- (b) Firefox: 144+
- (c) Safari: 26
- (d) Edge: 140+

*Motivazione:* il supporto ai principali browser assicura che tutti gli utenti possano accedere al servizio senza problemi tecnici legati alla compatibilità, riducendo il rischio di esclusione tecnologica.

**RNF21 Integrazione futura:** il sistema dovrà poter essere integrato con servizi esterni dell'amministrazione (es. open data del Comune, SPID). *Motivazione:* la possibilità di integrare il sistema con altri servizi pubblici o di terze parti consente di ampliare le funzionalità future e garantisce interoperabilità con l'ecosistema digitale della pubblica amministrazione.

## 4 Use Case

### 4.1 RF1: Gestione Accessi

**Riassunto:** Descrive come i Cittadini e gli Operatori Comunali accedono alla piattaforma tramite due modalità distinte di autenticazione: SPID/CIE ID per cittadini e credenziali interne per operatori.

### **Flusso A: Accesso Cittadino (SPID/CIE ID)**

1. Il Cittadino visualizza la schermata di login e seleziona “*Accedi con SPID/CIE ID*”.
2. Il sistema avvia il protocollo SAML reindirizzando l’utente all’Identity Provider (IdP).
3. L’IdP restituisce una risposta SAML firmata digitalmente.
4. Il sistema verifica firma, validità e consistenza dell’asserzione.
5. Il sistema estrae ID univoco e attributi dell’utente.
6. Il sistema consulta il database usando l’ID univoco.
7. Se l’ID è presente, l’accesso è autorizzato; altrimenti viene attivata l’Estensione 1.

#### **Estensione 1 — Registrazione ID Univoco**

*Condizione:* primo accesso del cittadino.

*Azione:* l’ID univoco viene memorizzato nel database e l’utente accede alla dashboard.

#### **Eccezioni**

- Autenticazione SPID/CIE fallita → messaggio di errore e accesso negato.

### **Flusso B: Accesso Operatore**

1. L’Operatore inserisce nome utente e password nel form di login.
2. Preme “*Accedi*”.
3. Il sistema verifica le credenziali nel database.
4. Se valide, l’Operatore accede alla dashboard amministrativa.

#### **Eccezioni**

- Credenziali errate → messaggio di errore e nuovo tentativo.

## **4.2 RF2: Accesso alla Dashboard**

**Riassunto:** Descrive come gli utenti autenticati accedono alla propria dashboard con funzionalità differenti a seconda del ruolo.

### **Flusso A: Dashboard Cittadino**

1. Il cittadino autenticato viene reindirizzato alla dashboard.
2. Il sistema verifica il ruolo.
3. Il sistema mostra votazioni attive, risultati conclusi, sondaggi disponibili e iniziative cittadine.

### **Flusso B: Dashboard Operatore**

1. L’Operatore autenticato viene reindirizzato alla dashboard.
2. Il sistema verifica il ruolo.
3. Il sistema mostra tutte le funzionalità del cittadino.
4. Il sistema aggiunge funzionalità amministrative: pubblicazione votazioni/sondaggi, gestione contenuti, monitoraggio risultati.

## **4.3 RF3: Riepilogo**

**Riassunto:** Permette la visualizzazione dei risultati delle votazioni concluse con diversi livelli di dettaglio.



### Flusso A: Riepilogo Sintetico

1. L'utente richiede i risultati di una votazione conclusa.
2. Il sistema recupera i dati dal database.
3. Il sistema calcola percentuali e conteggi assoluti.
4. Il sistema presenta un riepilogo sintetico e anonimizzato.

### Flusso B: Analisi Demografica (Operatore)

1. Include tutti i passi del caso base.
2. L'Operatore applica eventuali filtri demografici.
3. Il sistema elabora e presenta risultati aggregati per categoria.

#### Estensione: Applicazione filtri

L'utente può applicare filtri demografici o tematici per analizzare i dati aggregati.

## 4.4 RF4: Ricerca

**Riassunto:** Permette la ricerca di iniziative e contenuti tramite parole chiave, con filtri e ordinamenti opzionali.

1. L'utente inserisce parole chiave nella barra di ricerca.
2. Avvia la ricerca.
3. Il sistema interroga il database.
4. Il sistema mostra i risultati in ordine predefinito.

#### Estensioni

- **Filtri:** per argomento, data, tipologia.
- **Ordinamento:** per data crescente/decrescente o popolarità.

**Eccezione:** Nessun risultato trovato.

## 4.5 RF5: Creazione Votazioni

**Riassunto:** L'Operatore crea una nuova votazione inserendo tutti i campi obbligatori.

1. L'Operatore seleziona *Crea nuova votazione*.
2. Il sistema richiede:
  - Titolo
  - Breve descrizione
  - Opzioni di risposta
  - Tipologia (singola/multipla)
  - Data apertura e chiusura
3. Il sistema valida i dati.
4. Salva la votazione in stato *bozza*.

**Eccezione:** Campi mancanti o illogici.

## 4.6 RF6: Gestione Votazioni

**Riassunto:** L'Operatore può modificare, pubblicare, eliminare e archiviare votazioni.

1. L'Operatore accede a *Gestione Votazioni*.
2. Il sistema mostra tutte le votazioni con stato.
3. L'Operatore seleziona una votazione.
4. Il sistema richiede conferma per l'azione scelta.

### Estensioni

- Modifica (solo se in bozza)
- Eliminazione (solo bozza)
- Pubblicazione → stato “attiva”
- Archiviazione → stato “archiviata”

**Transizione automatica:** alla scadenza → “conclusa”.

## 4.7 RF7: Creazione Sondaggi

**Riassunto:** L'Operatore crea un sondaggio anonimo in stato iniziale “bozza”.

1. L'Operatore seleziona *Crea nuovo sondaggio*.
2. Inserisce:
  - Titolo e descrizione
  - Domande chiuse/multiple
  - Data e ora di apertura/chiusura
3. Il sistema valida e salva in bozza.

**Eccezioni:** Campi mancanti o inconsistenza date.

## 4.8 RF8: Gestione Sondaggi

1. L'Operatore accede alla sezione dedicata.
2. Il sistema mostra sondaggi e stato.
3. L'Operatore seleziona l'azione.

### Estensioni

- Pubblicazione (da bozza)
- Chiusura manuale (da attivo)
- Analisi risultati (concluso/archiviato)
- Archiviazione (da concluso)

**Transizione automatica:** alla scadenza → “concluso”.

## 4.9 RF9: Moderazione Contenuti

**Riassunto:** L'Operatore può rivedere, approvare o eliminare contenuti generati dai cittadini.

1. L'Operatore accede alla sezione Moderazione.
2. Il sistema mostra i contenuti in revisione.
3. L'Operatore visualizza il contenuto e valuta.

**Estensione:** Eliminazione contenuto su conferma.

#### 4.10 RF10: Gestione Iniziative dei Cittadini

1. Il cittadino seleziona *Proponi iniziativa*.
2. Inserisce titolo e descrizione.
3. Il sistema associa automaticamente nome utente.
4. Il cittadino conferma e il sistema pubblica l'iniziativa.

**Eccezione:** Campi obbligatori mancanti.

#### 4.11 RF11: Votazione Iniziative

1. Il cittadino visualizza la bacheca iniziative.
2. Seleziona *Vota*.
3. Il sistema verifica che non abbia già votato.
4. Se valido, incrementa il contatore e blocca ulteriori voti.

**Eccezione:** Voto duplicato.

#### 4.12 RF12: Votazioni e Sondaggi (Invio Univoco)

1. Il cittadino invia il proprio voto/risposta.
2. Il sistema verifica se esiste già un invio precedente.
3. Se valido:
  - Anonimizza il voto
  - Registra hash di tracciabilità
  - Impedisce ulteriori invii

**Eccezione:** Invio duplicato.

#### 4.13 RF13: Accesso alla dashboard per cittadini attivi

1. Il cittadino accede alla sua dashboard (RF2) o alla sezione dedicata alle attività in corso.
2. Il sistema recupera votazioni e sondaggi.
3. Il sistema filtra per stato **attivo**.
4. Il sistema presenta al cittadino l'elenco degli elementi attivi, separati per tipologia, con i dettagli essenziali per l'interazione (es: titolo, periodo di validità).

**Eccezione:** Nessun elemento attivo: se il sistema non trova né votazioni né sondaggi **attivi**, mostra un messaggio informativo (es: "Al momento non ci sono attività in corso").

## 5 User Stories

### 5.1 User Story 1: Gestione Accessi (UC RF1)

**Titolo:** Scelta metodo di login (Google o credenziali locali)

**Come** cittadino o operatore, **voglio** poter scegliere se accedere tramite credenziali locali o autenticazione Google, **in modo da** selezionare il metodo che preferisco.

**Criteri di accettazione:**

- Entrambi gli utenti possono scegliere tra credenziali locali e Google.
- In caso di successo, il sistema reindirizza alla dashboard corretta in base al ruolo.
- In caso di errore, viene mostrato un messaggio informativo adeguato.

**Tasks:**

1. Integrare entrambe le opzioni di login nella UI.
2. Implementare la logica di selezione del metodo di autenticazione.
3. Gestire il routing verso la dashboard corretta.
4. Implementare messaggi di errore specifici per ogni metodo.
5. Testare entrambe le modalità.

## 5.2 User Story 2: Ricerca (UC RF4)

**Titolo:** Ricerca iniziative tramite parole chiave

**Come** cittadino, **voglio** cercare iniziative tramite parole chiave, **in modo da** trovare rapidamente contenuti di mio interesse.

**Criteri di accettazione:**

- L'utente può inserire una o più parole chiave.
- Il sistema mostra i risultati ordinati per pertinenza.
- Ogni risultato mostra titolo, autore, categoria e data.
- Se non ci sono risultati, viene mostrato un messaggio dedicato.

**Tasks:**

1. Implementare la barra di ricerca nella dashboard.
2. Sviluppare la logica di ricerca nel backend.
3. Visualizzare i risultati con i metadati richiesti.
4. Gestire la mancanza di risultati.
5. Testare la ricerca su keyword differenti.

## 5.3 User Story 3: Ricerca (UC RF4)

**Titolo:** Applicazione filtri alla ricerca

**Come** utente della piattaforma, **voglio** applicare filtri sulla ricerca (argomento, data, popolarità), **in modo da** affinare i risultati.

**Criteri di accettazione:**

- L'utente può selezionare uno o più filtri.
- I risultati riflettono i filtri scelti.
- I filtri possono essere rimossi singolarmente.

**Tasks:**

1. Aggiungere un pannello filtri.
2. Implementare la logica di filtraggio nel backend.
3. Aggiornare dinamicamente la vista dei risultati.
4. Testare varie combinazioni di filtri.

## 5.4 User Story 4: Creazione Votazioni (UC RF5)

**Titolo:** Creare una nuova votazione

**Come** operatore comunale, **voglio** creare una nuova votazione compilando un form guidato, **in modo da** pubblicare contenuti strutturati e coerenti.

**Criteri di accettazione:**

- Il form richiede titolo, descrizione, opzioni e durata.
- Le date devono essere validate.
- Al salvataggio, lo stato iniziale è “bozza”.

**Tasks:**

1. Creare un form per i campi richiesti.
2. Implementare la validazione lato client e server.
3. Salvare la votazione nel database.
4. Testare vari scenari (campi mancanti, date non valide...).

## 5.5 User Story 5: Gestione Votazioni (UC RF6)

**Titolo:** Pubblicare una votazione

**Come** operatore, **voglio** pubblicare una votazione in bozza, **in modo da** renderla visibile ai cittadini.

**Criteri di accettazione:**

- La votazione può essere pubblicata solo se è in stato “bozza”.
- Il sistema chiede una conferma esplicita.
- Lo stato cambia in “attiva”.

**Tasks:**

1. Implementare il pulsante “Pubblica”.
2. Gestire la transizione dello stato.
3. Aggiornare la dashboard di utenti e operatori.
4. Testare la procedura.

## 5.6 User Story 6: Gestione Iniziative (UC RF10)

**Titolo:** Proporre una nuova iniziativa

**Come** cittadino, **voglio** poter proporre una nuova iniziativa, **in modo da** condividere idee con la comunità.

**Criteri di accettazione:**

- Il form richiede titolo e descrizione.
- Il sistema associa automaticamente il nome utente.
- L’iniziativa appare subito in bacheca.

**Tasks:**

1. Creare il form per la proposta.
2. Validare i campi.
3. Salvare nel database e aggiornare la bacheca.
4. Testare i vari scenari.

## 5.7 User Story 7: Votazione Iniziative (UC RF11)

**Titolo:** Votare un'iniziativa

**Come** cittadino, **voglio** poter votare un'iniziativa una sola volta, **in modo da** contribuire alla sua visibilità.

**Criteri di accettazione:**

- L'utente può votare una singola iniziativa una sola volta.
- Il contatore viene incrementato.
- Il pulsante di voto viene disabilitato.

**Tasks:**

1. Implementare il pulsante di voto.
2. Implementare il controllo di voto duplicato.
3. Aggiornare il contatore in tempo reale.
4. Testare casi validi e duplicati.

## 5.8 User Story 8: Votazioni (UC RF12)

**Titolo:** Inviare un voto anonimo

**Come** cittadino, **voglio** inviare il mio voto in forma anonima, **in modo da** partecipare senza rivelare la mia identità.

**Criteri di accettazione:**

- Il sistema separa l'identità dal voto.
- Viene generato un hash di tracciabilità.
- L'utente non può votare una seconda volta.

**Tasks:**

1. Implementare la logica di anonimizzazione.
2. Registrare hash e voto in modo immutabile.
3. Bloccare ulteriori invii.
4. Testare la sicurezza del processo.

## 5.9 User Story 9: Moderazione Contenuti (UC RF9)

**Titolo:** Moderare contenuti segnalati

**Come** operatore comunale, **voglio** visualizzare e moderare contenuti segnalati, **in modo da** mantenere la piattaforma sicura e rispettosa.

**Criteri di accettazione:**

- La dashboard mostra tutti i contenuti in revisione.
- L'operatore può approvare o eliminare contenuti.
- Le azioni richiedono conferma esplicita.

**Tasks:**

1. Mostrare elenco contenuti segnalati.
2. Implementare approvazione ed eliminazione.
3. Gestire conferme e messaggi.
4. Testare la moderazione.

## 5.10 User Story 10: Dashboard attività attive (UC RF13)

**Titolo:** Visualizzare votazioni e sondaggi attivi

**Come** cittadino, **voglio** visualizzare tutte le votazioni e i sondaggi attivi, **in modo da** sapere a quali attività posso partecipare.

**Criteri di accettazione:**

- Il sistema mostra solo elementi con stato “attivo”.
- I risultati sono divisi per tipologia (votazioni, sondaggi).
- Ogni elemento mostra titolo e periodo di validità.

**Tasks:**

1. Recuperare elementi attivi dal database.
2. Filtrarli per stato.
3. Visualizzarli ordinati per data di apertura.
4. Testare vari scenari inclusi casi vuoti.

**Eccezione:** Nessuna attività attiva.

## References