



BeeLive

Deliverable 1

Gruppo 21:
Cipriani Pietro, 226959
Orlando Dennis, 227688
Ziviani Elia, 228172

28 Marzo 2024



Indice

1 Descrizione dell'applicativo	4
2 Obiettivi	5
3 Attori di sistema e Mind map	6
3.1 Attori di sistema Interni	6
3.1.1 Utente anonimo	6
3.1.2 Utente autenticato	6
3.1.3 Dipendente autorizzato del Comune di Trento	6
3.1.4 Enti delegati	6
3.1.5 Amministratore di sistema	6
3.2 Attori di sistema Esterni	6
3.2.1 Enti delegati	6
3.3 Componenti di sistema Interni	6
3.3.1 Applicativo mobile	6
3.3.2 Applicativo desktop	6
3.3.3 Web server	6
3.3.4 Database	6
3.4 Componenti di sistema Esterni	6
3.4.1 Server di autenticazione	6
3.4.2 Server provider	6
3.5 Mind map	7
4 Prototipo del sistema	8
4.1 Mockup applicazione mobile	8
4.1.1 Schermata principale	8
4.1.2 Dettagli variazione selezionata	9
4.1.3 Mockup notifiche degli eventi	10
4.2 Mockup interfaccia desktop	11
5 Requisiti Funzionali	12
5.1 Utente anonimo tramite applicativo mobile	12
5.1.1 Deve poter visualizzare la lista degli eventi	12
5.1.2 Deve poter filtrare gli eventi per categoria	12
5.1.3 Deve poter nascondere eventi specifici che non gli interessano	12
5.1.4 Deve poter visualizzare gli eventi su mappa	12
5.1.5 Deve potersi registrare senza perdita di impostazioni	12
5.1.6 Deve potersi autenticare	12
5.1.7 Deve poter indicare categorie di interesse	12
5.1.8 Deve poter indicare aree di interesse	12
5.1.9 Deve poter indicare percorsi di interesse	12
5.1.10 Deve poter ricevere notifiche riguardo agli eventi	12
5.2 Utente autenticato tramite applicativo mobile	13
5.2.1 Deve avere le impostazioni sincronizzate tra i vari dispositivi	13
5.2.2 Non deve avere altri vantaggi rispetto l'utente non autenticato	13
5.3 Dipendente autorizzato del Comune di Trento tramite applicativo desktop	13
5.3.1 Deve poter definire sottocategorie	13
5.3.2 Deve poter gestire gli eventi di sua competenza	13
5.3.3 Non deve poter gestire gli eventi che non sono di sua competenza	13
5.3.4 Deve poter risalire al dipendente autorizzato che ha gestito un evento specifico	13
5.3.5 Deve poter visionare lo storico della gestione degli eventi	13
5.3.6 Deve poter definire elementi geografici di interesse per facilitare la selezione delle aree interessate all'evento	13



5.3.7	Deve poter sospendere la gestione di un evento e riprenderla successivamente	13
5.3.8	Deve poter caricare elementi geografici provenienti da file esterni	13
5.3.9	Deve poter creare un evento a partire da file di strutture specifiche	14
5.4	Enti delegati tramite API gestionali del Web Server	14
5.4.1	Deve avere esclusività di gestione durante il corso dell'operazione	14
5.5	Amministratore di sistema	14
5.5.1	Deve poter gestire le policies di access control	14
5.5.2	Deve poter definire le aree di competenza delle entità di gestione	14
5.5.3	Deve avere accesso privilegiato al sistema	14
5.5.4	Deve poter definire le entità di gestione come <i>trusted</i> o <i>un-trusted</i>	14
5.6	WebServer	14
5.6.1	Deve garantire l'accesso al servizio ai soli utenti autorizzati	14
5.7	DataBase	14
5.7.1	Deve poter eseguire query spaziali	14
6	Requisiti non funzionali	15
6.0.1	Affidabilità dei dati forniti	15
6.0.2	Usabilità	15
6.0.3	Scalabilità	15
6.0.4	Robustness	15
6.0.5	Prestazioni di notifica	15
6.0.6	Reliability	15
6.0.7	WebServer diviso	15
7	Grafo BPMN	16
8	Diagramma dei casi d'uso	17
9	Resoconto	18



1 Descrizione dell'applicativo

La Municipalità di Trento ha richiesto la valutazione dei problemi cittadini e la proposta di soluzioni attraverso lo sviluppo di un'applicazione web ad-hoc.

Dopo un processo di ideazione, è emerso che a Trento risulta spesso difficile reperire informazioni riguardanti gli eventi che influenzano la viabilità stradale. Le informazioni disponibili sono principalmente testuali e richiedono un impegno attivo da parte dei cittadini per essere trovate, poiché attualmente manca un sistema di notifica automatica per tali modifiche.

Di conseguenza, l'idea concepita prevede lo sviluppo di un sistema informativo che consenta ai cittadini di accedere rapidamente ed intuitivamente a tutte le variazioni della viabilità, senza la necessità di una ricerca attiva.

Il nucleo concettuale si basa sulla rappresentazione grafica delle zone coinvolte nelle modifiche della viabilità, per facilitare la comprensione delle informazioni.

È inoltre previsto un sistema di notifiche push per informare i cittadini in tempo reale sugli eventi che coinvolgono l'intera città o solo le zone da loro selezionate.

Saranno sviluppati due ambienti distinti: Uno riservato all'amministrazione e uno accessibile ai cittadini. Il primo ambiente consentirà all'amministrazione di inserire e modificare le informazioni sulle variazioni della viabilità attraverso un'interfaccia grafica intuitiva e compatibile con gli standard per la visualizzazione dei dati su mappa.

Il secondo sarà un'applicazione mobile installabile sui dispositivi dei cittadini, che consentirà loro di visualizzare graficamente le zone interessate dai cambiamenti, selezionare le aree di loro interesse e ricevere notifiche push in tempo reale.

Ogni variazione inserita nel sistema dall'amministrazione sarà caratterizzata dalla zona interessata, visualizzabile sulla mappa, da una descrizione che ne spiega la natura e da una stima della durata della variazione, tutte informazioni che i cittadini potranno visualizzare tramite l'applicazione mobile dal loro dispositivo.



2 Obiettivi

Archiviazione eventi



3 Attori di sistema e Mind map

3.1 Attori di sistema Interni

3.1.1 Utente anonimo

Si tratta del cittadino che utilizza l'applicativo. Ha accesso a tutte le funzionalità ma non è disponibile la sincronizzazione tra più dispositivi.

3.1.2 Utente autenticato

3.1.3 Dipendente autorizzato del Comune di Trento

3.1.4 Enti delegati

I dati che arrivano da questi enti vengono ritenuti affidabili.

3.1.5 Amministratore di sistema

3.2 Attori di sistema Esterne

3.2.1 Enti delegati

3.3 Componenti di sistema Interni

3.3.1 Applicativo mobile

3.3.2 Applicativo desktop

3.3.3 Web server

Separato in due parti

Ogniuna con utenti autenticati

3.3.4 Database

3.4 Componenti di sistema Esterne

3.4.1 Server di autenticazione

3.4.2 Server provider



3.5 Mind map

Tramite la mind map è possibile avere una rappresentazione chiara degli attori che interagiscono con il sistema.



4 Prototipo del sistema

4.1 Mockup applicazione mobile

4.1.1 Schermata principale

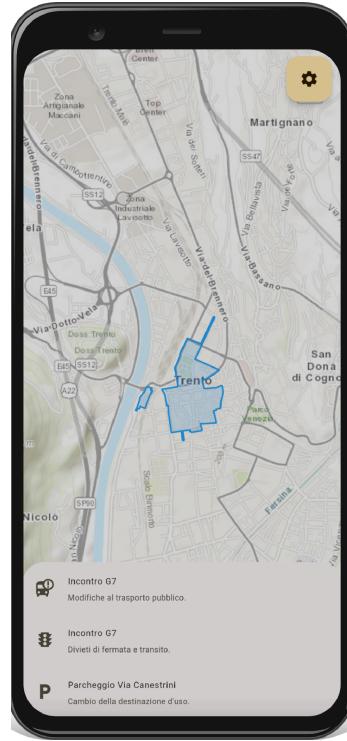


Figure 1: Mockup schermata principale dell'applicazione mobile

E' rappresentata la schermata principale dell'applicazione mobile che il cittadino installerà sul proprio smartphone.

La schermata principale mostra una mappa della città di Trento con le zone interessate dalle variazioni della viabilità.

La lista sottostante la mappa riporta tutte le variazioni presenti nel sistema che si trovano nel periodo di visualizzazione, con la possibilità di selezionare una variazione per visualizzarne i dettagli.



4.1.2 Dettagli variazione selezionata



Figure 2: Mockup schermata dettagli di un evento dell'applicazione mobile

In questo caso invece è riportata la schermata di visualizzazione dei dettagli di una variazione della viabilità. È stato selezionato il precedente evento "Incontro G7", quindi sono visualizzati tutti i dettagli relativi a tale evento.

Nella fattispecie sono presenti la descrizione dell'evento, la data di inizio e fine, la zona interessata e la mappa con la visualizzazione della zona interessata.

Inoltre è presente un pulsante per tornare alla schermata principale dell'applicazione.



4.1.3 Mockup notifiche degli eventi



Figure 3: Mockup notifiche generate da un'evento dall'applicazione mobile

L'evento preso nell'esempio precedente era impostato per generare delle notifiche push. Quella riportata nell'esempio soprastante rappresenta la notifica di inizio dell'evento "Incontro G7". Interagendo con essa sarà quindi aperta l'applicazione mobile sulla schermata riportata in figura 2, permettendo quindi la consultazione delle modifiche per intero.



4.2 Mockup interfaccia desktop

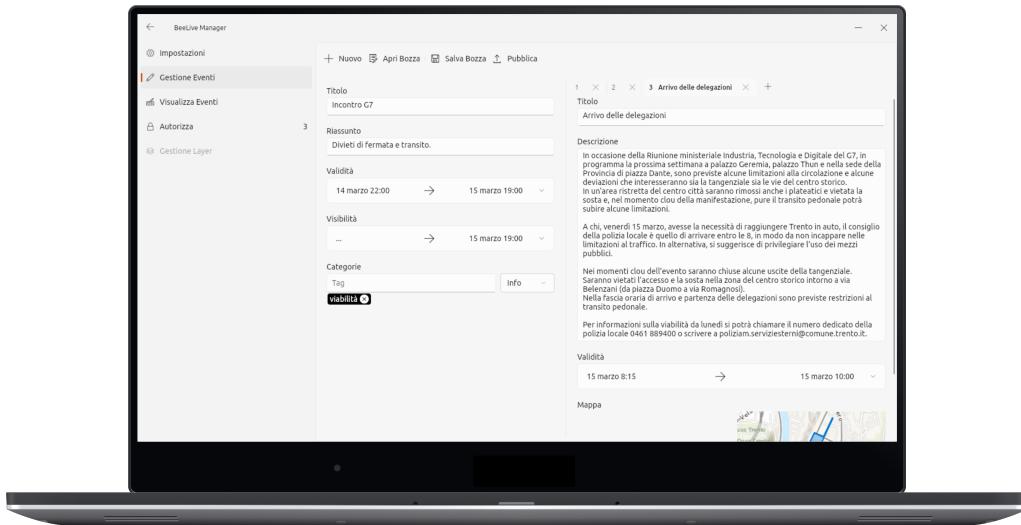


Figure 4: Mockup dell'interfaccia amministrativa

L'immagine rappresenta come dovrebbe essere strutturata l'interfaccia dell'applicativo desktop, utilizzato dagli enti pubblici con potere di amministrazione per la pubblicazione degli eventi in città che ne influenzano la viabilità.

L'interfaccia comprende un menu laterale per la navigazione tra le varie sezioni dell'applicativo e la schermata dove il sottomenù selezionato presenta tutte le sue specifiche funzionalità disponibili.

Il sottomenù selezionato in questo caso è "Gestione Eventi", la sezione in cui è possibile inserire un nuovo evento che modifica la viabilità cittadina. Infatti sono riportati tutti i campi necessari per la creazione di un evento, come il titolo, il riassunto, la data di inizio e fine di validità dell'evento e di visibilità sull'applicativo mobile, e le categorie che caratterizzano l'evento.

L'interfaccia è strutturata in modo che ogni evento abbia la possibilità di essere composto da più sottoeventi. Nella sezione di destra è possibile visualizzare l'inderimento del sottoevento "Arrivo delle delegazioni", dove è possibile inserire il titolo, la descrizione, la data di validità e la zona interessata direttamente su mappa.



5 Requisiti Funzionali

5.1 Utente anonimo tramite applicativo mobile

5.1.1 Deve poter visualizzare la lista degli eventi

L'utente anche se non autenticato a sistema deve avere la possibilità di visualizzare la lista intera di tutti gli eventi pubblicati.

5.1.2 Deve poter filtrare gli eventi per categoria

Gli eventi sono divisi in categorie, l'utente deve avere la possibilità di esprimere le proprie preferenze sugli eventi da visualizzare in lista definita in 5.1.1.

5.1.3 Deve poter nascondere eventi specifici che non gli interessano

In caso l'utente non dimostri interesse per un evento specifico, deve avere la possibilità di nasconderlo dalla lista degli eventi.

5.1.4 Deve poter visualizzare gli eventi su mappa

L'utente deve essere in grado di visualizzare in modo intuitivo l'area fisica interessata dagli eventi di suo interesse.

5.1.5 Deve potersi registrare senza perdita di impostazioni

Le impostazioni selezionate da un utente anonimo non devono essere perse nel caso in cui questo si registri a sistema.

5.1.6 Deve potersi autenticare

Gli utenti devono avere la possibilità di autenticarsi a sistema per usufruire della sincronizzazione tra dispositivi.

5.1.7 Deve poter indicare categorie di interesse

Deve poter indicare categorie di interesse in maniera indipendente per eventi in *zone di interesse* (Aree e percorsi) e non.

- Granularità maggiori (ad esempio: per zona di interesse e per label di rischio) non sono previste, in quanto **rischiano** di compromettere l'user experience

5.1.8 Deve poter indicare aree di interesse

L'utente deve poter indicare delle zone su mappa per cui dimostra più interesse.

5.1.9 Deve poter indicare percorsi di interesse

L'utente deve poter indicare delle percorsi su mappa per cui dimostra più interesse.

5.1.10 Deve poter ricevere notifiche riguardo agli eventi

L'utente deve poter essere notificato all'insorgere determinati eventi.



5.2 Utente autenticato tramite applicativo mobile

5.2.1 Deve avere le impostazioni sincronizzate tra i vari dispositivi

Siccome l'utente autenticato tramite applicativo mobile eredita le caratteristiche dell'utente anonimo, valgono tutti i suoi requisiti, con l'aggiunta che in questo caso l'utente ha tutte le impostazioni sincronizzate su tutti i dispositivi in cui è autenticato.

5.2.2 Non deve avere altri vantaggi rispetto l'utente non autenticato

L'applicativo deve poter essere utilizzato in maniera completa senza forzare l'autenticazione dell'utente, che ne comporta una barriera d'utilizzo, salvo eventuali problematiche implementative.

5.3 Dipendente autorizzato del Comune di Trento tramite applicativo desktop

5.3.1 Deve poter definire sottocategorie

Le categorie sono organizzate in modo gerarchico per migliorare la granularità di filtraggio.

5.3.2 Deve poter gestire gli eventi di sua competenza

La gestione degli eventi di un dipendente include la creazione, la modifica, l'eliminazione e la pubblicazione (Solo se *trusted*) degli eventi di sua competenza.

5.3.3 Non deve poter gestire gli eventi che non sono di sua competenza

Il dipendente ha accesso solamente alle operazioni riguardanti gli eventi di sua competenza e non può gestire tutti gli altri.

5.3.4 Deve poter risalire al dipendente autorizzato che ha gestito un evento specifico

Deve sempre essere possibile individuare gli utenti autorizzati dal comune che hanno eseguito azioni sugli eventi pubblicati a sistema.

5.3.5 Deve poter visionare lo storico della gestione degli eventi

L'utente autorizzato deve poter visionare lo storico delle interazioni sugli eventi, con la possibilità di visionare chi ha effettuato l'attività e quando.

5.3.6 Deve poter definire elementi geografici di interesse per facilitare la selezione delle aree interessate all'evento

Per facilitare la selezione delle aree interessate all'evento, l'utente autorizzato deve poter definire elementi geografici notevoli.

5.3.7 Deve poter sospendere la gestione di un evento e riprenderla successivamente

Si deve avere la possibilità di sospendere la gestione di un evento, potendola riprendere successivamente senza alcuna perdita di dati.

5.3.8 Deve poter caricare elementi geografici provenienti da file esterni

L'utente, in fase di pubblicazione di un nuovo evento, deve avere la possibilità di indicare la zona d'interesse tramite il caricamento di un file contenente tutte le sue informazioni geografiche.



5.3.9 Deve poter creare un evento a partire da file di strutture specifiche

I dipartimenti devono poter continuare ad utilizzare i loro file attuali che determinano e caratterizzano gli eventi per la loro gestione sul sistema.

5.4 Enti delegati tramite API gestionali del Web Server

Questo attore estende eredita le capacità di "Dipendente autorizzato del Comune di Trento", integrandone di caratteristiche.

5.4.1 Deve avere esclusività di gestione durante il corso dell'operazione

5.5 Amministratore di sistema

5.5.1 Deve poter gestire le policies di access control

Essendo l'amministratore di sistema, ha il compito di gestire tutte le policies di access control del sistema.

5.5.2 Deve poter definire le aree di competenza delle entità di gestione

E' compito suo definire tutte le aree di competenza di Dipendenti autorizzati del comune e Enti delegati.

5.5.3 Deve avere accesso privilegiato al sistema

Deve avere libertà nell'apportare modifiche specifiche alle componenti (Come il salvataggio dei dati)

5.5.4 Deve poter definire le entità di gestione come *trusted* o *un-trusted*

E' necessaria questa distinzione, solamente gli utenti *trusted* possono accedere ai servizi offerti dal sistema.

5.6 WebServer

5.6.1 Deve garantire l'accesso al servizio ai soli utenti autorizzati

Solamente gli utenti autorizzati (Anche i non autenticati, solo per alcuni servizi) possono accedere al servizio.

5.7 DataBase

5.7.1 Deve poter eseguire query spaziali

Sul DataBase sono salvati anche dati caratterizzati da coordinate. Deve essere possibile eseguire operazioni di ricerca su di esse.



6 Requisiti non funzionali

6.0.1 Affidabilità dei dati forniti

Tutti i dati che provengono dagli enti delegati devono essere considerati affidabili dal resto del sistema.

6.0.2 Usabilità

Per quanto riguarda l'applicazione mobile, i cittadini devono essere in grado di utilizzarla dal primo momento in cui è installata sul dispositivo.

Per invece l'applicativo desktop, precedentemente al primo utilizzo è necessaria al più una specificazione delle funzionalità per il loro completo utilizzo.

6.0.3 Scalabilità

Il requirement di scalabilità riguarda solamente l'applicazione mobile, perché deve essere in grado di gestire un elevato numero di utenti, che rispecchi il numero degli abitanti di Trento.

L'applicativo desktop invece non ha bisogno di scalabilità, in quanto è utilizzato solo da un numero limitato di utenti.

6.0.4 Robustness

Un guasto al webserver di visualizzazione non deve compromettere il funzionamento del webserver di gestione, e viceversa.

6.0.5 Prestazioni di notifica

La notifica di un nuovo evento deve raggiungere tutti i dispositivi mobili, connessi ad internet e con l'applicazione installata, entro un tempo di 10 minuti

6.0.6 Reliability

Il sistema non deve essere soggetto a errori imprevisti di alcun tipo, ed eventuali bug non devono compromettere eventuali dati sensibili presenti sul server.

Il sistema gestionale usato dagli utenti autorizzati non deve essere soggetto a vulnerabilità che ne compromettono la sicurezza.

6.0.7 WebServer diviso

Ci sono due webserver, uno inherente il gestionale desktop e uno per gestire l'applicazione mobile.



7 Grafo BPMN



8 Diagramma dei casi d'uso

- It represents a functional requirement
- It makes clear what you expect from a system ("what?")
- It hides the behaviour of the system ("how?")
- It is a sequence of actions (with variants) that produce a result observable by an actor

Nome caso d'uso	*Nome*
ID	Testo
Descrizione	Testo
Attori primari	Testo
Attori secondari	Testo
Precondizioni	Testo
Flusso principale	Testo
Post-condizioni	Testo
Flussi alternativi	Testo

Table 1: Tabella riassuntiva



9 Resoconto

Segue la tabella dei riferimenti, utile alla comprensione generale delle diverse sezioni riguardanti lo specifico obiettivo.

Obiettivo	Attore	User Story	Use Case diagram	Requisiti funzionali	Mockup
1	2	3	4	5	6

Table 2: Tabella riassuntiva