

Lista dei Requisiti

Requisiti Funzionali:

Requisiti Utente:

- Il sistema deve permettere di visualizzare tutti i comuni iscritti a Municipath
- Il sistema deve permettere di visualizzare tutti i punti di ogni comune iscritto a Municipath
- I gestori della piattaforma devono poter creare ed eliminare comuni.
- I gestori della piattaforma devono poter dare l'autorizzazione di curatore ad un utente registrato.
- Un utente dovrebbe poter cercare comuni, utenti registrati e contenuti tramite parole chiave.
- Un utente registrato deve poter segnalare un punto, se lo ritiene inappropriato.
- Un curatore deve poter cambiare l'autorizzazione rispetto al proprio comune di un utente registrato tra: turista, contribuente non autorizzato, contribuente autorizzato e moderatore.
- Un curatore potrebbe scambiare la sua autorizzazione rispetto al comune in cui è curatore ad un utente con l'autorizzazione di moderatore presso lo stesso comune.
- Un curatore o un moderatore dovrebbe poter eliminare o modificare un punto del comune in cui è autorizzato.
- Un curatore o un moderatore deve poter leggere le segnalazioni di un comune in cui è autorizzato.
- Un contribuente non autorizzato deve poter proporre un punto ad un comune in cui ha l'autorizzazione di turista, che va in stato di pending.
- Un curatore o un moderatore deve poter vedere tutti i punti e tutte le modifiche in stato di pending del comune.
- Un curatore o un moderatore potrebbe modificare un punto o una modifica in stato di pending del proprio comune.
- Un curatore o un moderatore deve poter eliminare un punto o una modifica in stato di pending del proprio comune.

- Un curatore o un moderatore deve poter cambiare lo stato di un punto o una modifica da pending a pubblicato nel proprio comune.
- Un contribuente autorizzato, moderatore o curatore deve poter pubblicare un punto senza passare per lo stato di pending nei comuni in cui possiede questa autorizzazione.
- Un contribuente non autorizzato potrebbe ricevere una notifica sull'esito del giudizio sul suo contenuto in pending.
- Un utente deve poter eliminare tutti i punti di cui è autore.
- Un utente può modificare un punto nello stato di pubblicato di cui è autore, ma se il post appartiene ad un comune dove l'utente ha l'autorizzazione di contributor non autorizzato le modifiche vanno in stato di pending.
- Un turista autenticato potrebbe poter aggiungere immagini ad un post come contributo. Quest'ultime andranno in stato di pending.
- Un utente registrato deve poter richiedere ad un comune di ottenere l'autorizzazione di contribuente.
- Un curatore deve poter vedere le richieste degli aspiranti contribuenti del proprio comune.
- Un utente non potrà proporre contest presso un comune in cui ha l'autorizzazione di contribuente non autorizzato.
- Un post deve mostrare un titolo composto da almeno tre caratteri.
- Un contest deve permettere ad ogni utente registrato di mandare contenuti multimediali all'autore del post.
- L'autore del contest deve poter vedere i contributi del contest.
- Il curatore e i moderatori del comune dovrebbero poter vedere i contributi dei contest nel proprio comune.
- L'autore del contest o un curatore o moderatore del comune del contest potrebbero poter salvare o eliminare i contenuti multimediali ricevuti da un contest.
- Un utente deve registrarsi per accedere al proprio profilo.
- Ogni profilo potrebbe contenere un'immagine di profilo.
- Un utente registrato deve poter seguire un altro utente registrato o un comune, diventando un suo follower.
- Un utente registrato deve poter smettere di seguire qualcuno che prima seguiva.
- Un utente non può vedere chi lo segue.

- L'autore di un evento deve poter notificare a tutti i suoi follower il tempo rimanente per l'inizio di un evento.
- Il comune deve poter notificare tutti i suoi follower riguardo ad un evento che si trova al suo interno.

Requisiti Sistema:

- Il sistema deve poter rendere persistenti le informazioni multimediali.
- Il sistema deve registrare le visualizzazioni dei comuni e dei punti
- Ogni contenuto deve essere geolocalizzato in un'unica posizione.
- Ogni posizione deve poter contenere più post.
- Ogni punto deve appartenere ad un comune e uno soltanto.
- Un punto deve essere localizzato o comunale.
- Ogni punto localizzato dovrebbe essere nei territori limitrofi del comune di appartenenza.
- Ogni utente deve avere l'autorizzazione di turista in ogni comune dopo la convalida dell'account.
- Ogni comune deve avere un utente registrato con l'autorizzazione da curatore
- Un comune non avrà più di un curatore
- Un utente non sarà curatore di più di un comune
- Un curatore deve poter vedere le visualizzazioni del comune e dei punti del comune.
- Il sistema deve memorizzare nel database ogni punto in stato di pending o pubblicato.
- Un post scritto da un moderatore o il curatore del comune del post deve presentare come autore il comune stesso.
- Un post che presenta come autore un comune potrebbe ricordare chi è il vero autore.
- Un insieme deve essere un'esperienza o un itinerario.
- Un singolo deve essere un post o un contest.
- Un post dovrebbe mostrare i pulsanti per condividerlo negli altri social.
- Un post dovrebbe mostrare tutti gli insiemi che lo contengono.
- Un post di tipo evento deve mostrare il momento di inizio e di fine.
- Un post di tipo evento deve mostrare il meteo del momento di inizio.

- Un momento di inizio deve essere precedente di almeno un'ora a quello di fine.
- Una modifica di un post deve poter permettere di posticipare il momento di inizio.
- Una modifica di un post deve poter permettere di anticipare il momento di inizio solo se il vecchio momento o quello nuovo non si trovano entro 15 giorni dalla data attuale.
- Il sistema deve rendere i contributi ai contest persistenti.
- Un insieme deve contenere almeno due post.
- Il sistema deve permettere di notificare lo stesso evento una volta ogni due ore.
- Il sistema deve ordinare i post nello stesso punto mettendo prima quelli a tempo.
- Il sistema dovrebbe ordinare i post dello stesso tipo mettendo prima quelli con più autorizzazione.
- A parità di tipo e autorizzazione devono essere ordinati cronologicamente.
- Il sistema dovrebbe reperire le informazioni meteo in base al giorno e al comune.
- Tutti i punti del comune dello stesso giorno dovrebbero condividere le stesse informazioni meteo.
- Tutti i punti di tipo Event dovrebbero avere un meteo basato sul loro momento di inizio.

Requisiti Qualitativi:

- L'applicativo deve essere *riusabile* in altre piattaforme e deve essere possibile interfacciarsi con esso tramite chiamate REST.
- Il software deve garantire *efficienza*, quindi una chiamata REST deve dare risposta entro 5 secondi.
- Il codice deve risultare *comprensibile* e *manutenibile*: deve essere facile evincere la logica di business e quindi facile da riparare e adattare.

Vincoli:

- Il lato backend del progetto dovrà essere implementato col linguaggio java e il framework Spring Boot.
- Il lato frontend del progetto dovrà essere un sito internet di valorizzazione territoriale.
- Il lato frontend e il lato backend dovranno comunicare tramite chiamate rest.