**OnlyTickets**

**Requisiti Funzionali**

**1. Gestione Utenti**

* **Ruoli**:
  + Amministratori (gestione sistema e supervisione)
  + Agenti (gestione dei ticket)
  + Utenti (richiedenti del servizio)
* Registrazione e autenticazione tramite credenziali (email/password).
* Possibilità di modificare il profilo personale.

**2. Creazione e Gestione Ticket**

* Gli utenti possono creare un nuovo ticket con i seguenti campi:
  + Titolo del ticket.
  + Descrizione dettagliata del problema/richiesta.
  + Priorità (bassa, media, alta, urgente) ?? assegnata dall’agente quando smista?
* Gli agenti possono visualizzare, aggiornare lo stato (“Aperto”, "In elaborazione", "Chiuso"), e assegnare un ticket a un altro operatore.
* Notifiche agli utenti quando il ticket cambia stato.

**3. Dashboard e Report**

* **Per gli agenti**:
  + Dashboard che mostra i ticket assegnati, ordinati/ordinabili per priorità e stato.
* **Per gli amministratori**:
  + Visualizzazione di report e statistiche: numero di ticket aperti, chiusi
  + Gestione: gestione agenti, gestione utenti

**4. Ricerca e Filtri**

* Ricerca per parole chiave nei ticket.
* Filtri avanzati:
  + Stato (aperto, chiuso, in sospeso, ecc.).
  + Priorità.
  + Data di creazione.
  + Assegnati a X agente

**5. Integrazione Email (monodirezionale: piattaforma -> utente)**

* Invio automatico di notifiche via email:
  + Quando un ticket viene creato con successo.
  + Quando lo stato cambia.
  + Quando viene aggiunta una risposta.
  + Quando viene risolto

**6. Commenti e Collaborazione**

* Aggiunta di commenti ai ticket (sia da parte degli utenti che degli operatori).
* Possibilità di allegare file ai commenti (es. screenshot, documenti).

**Requisiti Non Funzionali**

**1. Usabilità**

* Interfaccia utente intuitiva, con navigazione semplice per utenti non tecnici.
* Compatibilità mobile.

**2. Prestazioni**

* Risposta rapida delle pagine (tempo di caricamento < X secondi).
* Capacità di gestire fino a 100? utenti simultaneamente senza degrado delle prestazioni.

**3. Sicurezza**

* Crittografia delle comunicazioni (HTTPS).
* Protezione contro attacchi comuni (es. SQL Injection, Cross-Site Scripting).
* Controllo degli accessi basato sui ruoli.
* Backup giornaliero dei dati.

**4. Affidabilità**

* Disponibilità garantita (uptime del 99.9%?).

**5. Scalabilità**

* Progettato per supportare l'espansione (es. aumento del numero di utenti o dipartimenti).

**Requisiti Tecnici**

**1. Frontend**

* ??
* CSS?

**2. Backend**

* Java (Spring Boot).
* API RESTful per la comunicazione con il frontend.

**3. Database**

* Relazionale – quale?

**4. Hosting e Deployment**

* ?
* Sistema di containerizzazione (Docker) per la portabilità. ?

**6. Strumenti di Versionamento e Testing**

* Versionamento del codice con Git - GitHub.