



TogetherTicket

Requisiti Non Funzionali

1. Requisiti di sistema

- Il sistema deve essere eseguibile su qualsiasi dispositivo dotato di JVM.
- L'applicazione è sviluppata in Java, con interfaccia utente realizzata tramite Swing.
- La persistenza dei dati avviene tramite database MySQL, con interazione attraverso JDBC.

1.1 Requisiti di qualità

- Il sistema deve garantire un tempo di risposta inferiore ai 3 secondi per operazioni standard.
- La struttura del database e del codice deve permettere scalabilità orizzontale in caso di futuro passaggio a un'architettura client-server.

- L'interfaccia utente deve essere intuitiva, chiara e coerente, anche per utenti non esperti.
- I dati devono essere salvati e recuperati in modo efficiente dal database, minimizzando le query ridondanti e ottimizzando l'accesso.

1.2 Requisiti organizzativi

- La documentazione del progetto (manuale utente, requisiti, report) deve essere redatta in lingua italiana.
- Il codice può contenere termini tecnici in inglese (es. nomi di classi, metodi, variabili).
- Il progetto deve seguire una struttura modulare nei package (`controller`, `view`, `model`, `dao`, `util`, `database`) per una chiara separazione delle responsabilità.

1.3 Requisiti legali e normativi

- Al momento della registrazione, l'utente deve accettare il trattamento dei dati personali, in conformità con il Regolamento UE 2016/679 (GDPR).
- I dati utente devono essere conservati in modo sicuro e non divulgati a terzi.
- Attualmente il sistema non prevede transazioni economiche reali; tuttavia, ogni simulazione di acquisto deve riflettere standard minimi di affidabilità e coerenza.

1.4 Requisiti di usabilità

- Il sistema deve essere utilizzabile da desktop, su sistemi Windows, macOS e Linux.
- L'utente deve ricevere messaggi chiari e leggibili in caso di operazioni riuscite o errori (es. login fallito, campo obbligatorio mancante).
- Tutti i flussi utente (registrazione, login, creazione gruppo, unione, scrittura in bacheca) devono essere completabili in pochi passaggi.

1.5 Requisiti di manutenibilità

- - Il codice deve essere commentato, organizzato in classi separate per responsabilità e facilmente estendibile.
- - Ogni nuova funzionalità deve poter essere integrata senza modificare le classi esistenti, seguendo il principio di open/closed.
- - Il sistema deve supportare aggiornamenti incrementali senza compromettere i dati salvati nel database.