

Università degli Studi di Salerno
Corso di Ingegneria del Software

BEAT – Booking Events And Tickets

Problem Statement

Versione 1.1



Data: 14/10/2025

Coordinatore del progetto: Prof. Andrea De Lucia

Partecipanti: Carnevale Luigi[0512119029], Di Manso Carmine[0512119521], Clemente Manuel[0512119395]

Scritto da: Carnevale Luigi, Di Manso Carmine, Clemente Manuel

Repository GitHub: [github.com/Luigi-Carnevale/BEAT-Booking Events And Tickets](https://github.com/Luigi-Carnevale/BEAT-Booking-Events-And-Tickets)

Revision History

Data	Versione	Descrizione	Autore
13-10-2025	1.0	Creazione Problem Statement	Carnevale Luigi, Di Manso Carmine, Clemente Manuel
15-10-2025	1.1	Applicate le correzioni suggerite dal professore	Carnevale Luigi, Di Manso Carmine, Clemente Manuel

Indice

Revision History

1	Purpose	1
2	Audience	1
3	Outline	1
3.1	Problem Domain	1
3.2	Scenarios	1
3.2.1	Un organizzatore vuole aggiungere un evento:	1
3.2.2	Utente registrato vuole acquistare un biglietto:	2
3.3	Functional Requirements	2
3.4	Nonfunctional Requirements	3
3.5	Target Environment	3
3.6	Deliverables e Deadlines	4
4	Conclusione	4

1 Purpose

L'obiettivo del progetto BEAT è sviluppare un sistema con interfaccia web per la gestione e prenotazione di eventi, che consenta agli utenti di consultare, prenotare e gestire biglietti per una varietà di eventi (concerti, seminari, corsi, spettacoli). Il sistema intende fornire una piattaforma intuitiva e dinamica che semplifichi sia la gestione amministrativa degli eventi sia l'esperienza dell'utente finale nella consultazione e prenotazione.

2 Audience

Il documento è destinato a:

1. Il team di sviluppo, per definire in modo chiaro i requisiti funzionali e non funzionali del sistema;
2. Il docente e i valutatori del corso di Ingegneria del Software, come riferimento per seguire la progettazione e l'implementazione del progetto;
3. Gli utenti finali (amministratori e partecipanti), ai quali il documento offre una panoramica sintetica sugli obiettivi del sistema e sulle principali funzionalità messe a loro disposizione.

3 Outline

3.1 Problem Domain

La gestione di eventi e biglietti spesso richiede piattaforme multiple e complesse da usare, con scarsa integrazione tra parte organizzativa e parte utente. Il dominio del problema riguarda quindi la digitalizzazione completa del ciclo di vita di un evento, dalla creazione alla prenotazione dei posti, fino al riepilogo delle partecipazioni, con una gestione unificata e accessibile via web.

3.2 Scenarios

3.2.1 Un organizzatore vuole aggiungere un evento:

Vincenzo vuole aggiungere all'elenco degli eventi prenotabili una data di un evento che si terrà prossimamente. Vincenzo tramite un apposito pulsante "login" verrà reindirizzato alla pagina di login, qui selezionerà l'opzione di login per gli organizzatori, inserirà le sue credenziali, email (es. vincenzo123@gmail.com) e password (es. Password@123), una volta inserite potrà cliccare il bottone "login" per accedere al proprio account ed essere reindirizzato sulla pagina della dashboard degli organizzatori. Fra le varie opzioni a sua disposizione (crea evento, modifica evento, elimina evento) Vincenzo selezionerà "crea evento" e gli verrà mostrata a schermo la pagina in cui inserire i dati relativi all'evento. I campi che vanno compilati per l'aggiunta dell'evento sono "titolo dell'evento", "data" (formato gg-mm-aaaa), "orario di inizio" (formato [hh]:[mm]), "protagonista" (chi svolge l'evento, un cantante se si tratta di un concerto, un professore se è un seminario, o chiunque sia) questo campo può essere lasciato vuoto solo se si tratta di una fiera con tanti ospiti e nessun artista "protagonista" ma in questo caso è presente una casella da flaggare

in cui indicare "si tratta di una fiera", va compilato il campo relativo ai "posti disponibili" in cui andrà inserito il limite massimo di prenotazioni accettabili, le "info sull'evento" in cui potrà essere inserita una breve descrizione dell'evento che si terrà. È previsto un campo in cui sarà possibile inserire un'immagine, che potrebbe essere la locandina dell'evento, se presente. Si inserisce poi il "prezzo" che le persone dovranno pagare per avere la loro prenotazione. È prevista una casella da flaggare che permette di impostare l'evento come gratuito, selezionando quindi "ingresso gratuito" il campo in cui inserire il prezzo verrà bloccato e l'evento risulterà "gratis". Una volta inserite tutte le informazioni necessarie, Vincenzo potrà cliccare sul pulsante posto in basso nella pagina "crea evento" per ufficializzare la creazione dell'evento, prima di aver completato gli verrà mostrato un riepilogo delle informazioni inserite, così da poter controllare se ci sono errori in qualche campo, se non ci sono errori allora potrà cliccare sul nuovo pulsante "conferma creazione". L'evento viene aggiunto all'elenco di quelli disponibili e l'organizzatore viene reindirizzato alla sua "dashboard eventi" dove potrà vedere fra tutti gli eventi che ha inserito, nella sezione relativa a quelli "ancora da svolgere" l'evento appena aggiunto con tutte le informazioni relative, inserite precedentemente durante il procedimento di creazione. A questo punto gli utenti potranno prenotare l'evento o acquistare i biglietti.

3.2.2 Utente registrato vuole acquistare un biglietto:

Mario vuole acquistare un biglietto per il concerto dei JazzBrothers. Mario tramite il pulsante di login viene reindirizzato alla schermata apposita in cui può inserire le proprie credenziali, email (es. mario1234@gmail.com) e password (es. Pssword@456), clicca sul pulsante di login ed accede al proprio account. A questo punto Mario cerca all'interno dell'elenco degli eventi oppure tramite l'apposita barra di ricerca il concerto che gli interessa, procede a selezionare l'evento, vengono quindi mostrate a schermo tutte le informazioni riguardo il concerto dei JazzBrothers (orario di inizio hh:mm, data gg-mm-aaaa, luogo di svolgimento, prezzo del biglietto, posti disponibili). Mario seleziona la data di suo interesse e viene reindirizzato alla schermata di selezione del numero di biglietti o posti da prenotare (massimo 4 per evitare bagarinaggio e consentire un accesso più equo ai biglietti, specialmente per eventi ad alta richiesta), andando avanti verrà reindirizzato alla pagina relativa al pagamento dove potrà inserire i dati relativi al metodo pagamento (paypal o carta di credito) e le informazioni necessarie ad effettuarlo (numero di carta, scadenza, cvv oppure email per paypal), successivamente conferma il pagamento e viene reindirizzato alla schermata relativa ai suoi ordini, nella quale, fra gli eventi che devono ancora svolgersi, vedrà il biglietto appena acquistato, che potrà essere consultato sia tramite interfaccia web (anche su dispositivi mobile), sia tramite un pdf che potrà scaricare grazie ad un apposito pulsante posto nella parte inferiore del riquadro riguardante il biglietto.

3.3 Functional Requirements

Il sistema deve permettere:

1. Un utente non registrato (guest) deve poter consultare l'elenco e i dettagli di eventi disponibili;
2. Gestione eventi (CRUD) da parte degli organizzatori;

3. Visualizzazione pubblica degli eventi disponibili, filtrabili per categoria, data o luogo;
4. Registrazione e login per gli utenti finali;
5. Prenotazione e cancellazione dei biglietti;
6. Riepilogo prenotazioni personale per ogni utente;
7. Gestione del DataBase da parte di un amministratore di catalogo eventi;
8. Gestione delle prenotazioni effettuate da parte di un amministratore degli ordini;
9. Gestione dei ruoli di tutte le figure presenti a cura di un amministratore dei ruoli;

3.4 Nonfunctional Requirements

1. **Implementation:** three-tier (Database, Backend, Frontend);
2. **Affidabilità:** disponibilità minima 99% in ambiente di test locale;
3. **Usabilità:** applicazione web responsiva, accessibile sia da dispositivi desktop che mobile;
4. **Interfaccia:** interfaccia intuitiva, con aggiornamenti dinamici tramite tecniche AJAX;
5. **Sicurezza:** credenziali cifrate e controllo degli accessi (ruoli utente/organizzatore/amministratore di ogni tipologia);
6. **Performance:** buona scalabilità, quindi possibilità di gestire contemporaneamente molti utenti senza perdite significative di prestazioni. Tempo medio di risposta inferiore a 2 secondi per operazioni standard (caricamento di pagine, login, ricerca e prenotazioni).

3.5 Target Environment

Il sistema BEAT verrà sviluppato e testato secondo un'architettura **three-tier**, articolata nei seguenti livelli:

1. **(Data) Database:** **MySQL**, utilizzato per la gestione e la persistenza dei dati relativi a utenti, eventi, prenotazioni e ruoli;
2. **(Application) Backend:** applicazione **Java** basata su **Spring Boot**, responsabile della logica applicativa, della gestione delle API REST e della comunicazione con il database;
3. **(Presentation) Frontend:** interfaccia web realizzata utilizzando **HTML**, **CSS** e **JavaScript**, con l'impiego di **AJAX** per garantire aggiornamenti dinamici e una navigazione fluida.

Il sistema verrà eseguito in ambiente locale mediante **Apache Tomcat** come server di applicazione. Lo sviluppo sarà effettuato in **Eclipse** o **IntelliJ IDEA**, con sistema di versionamento tramite **GitHub** e gestione delle dipendenze mediante **Maven**.

3.6 Deliverables e Deadlines

Deliverable	Scadenza
Problem Statement	14/10/2025
Requistiti e casi d'uso	28/10/2025
Requirement Analysis Document	11/11/2025
System Design Document	25/11/2025
Implementazione progetto	15/12/2025
Object Design Document	16/12/2025
Piano di test di sistema e speifica dei casi di test	16/12/2025
Project design finale	20/12/2025
Consegna progetto completato	13/01/2026

4 Conclusione

BEAT mira a fornire una piattaforma web completa, sicura e user-friendly per la gestione e prenotazione di eventi, realizzata con tecnologie moderne.