|  |  |  |
| --- | --- | --- |
| System Design  Document  **Ticket4You** | | |
| **Versione** | **0.4** |
| **Data** | 22/12/2023 |
| **Destinatario** | Esame di ingegneria del software |
| **Presentato da** | Simona Vigorito 0512115473  Tresy Sorrentino 0512114294  Nicolò Sarà 0512115140  Luca Greco 0512106227 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 01/12/2023 | 0.1 | Prima stesura | Simona Vigorito, Tresy Sorrentino, Nicolò Sorà, Luca Greco |
| 06/12/2023 | 0.2 | Definizione design goal, decomposizione in sottosistemi e diagramma architetturale | Tresy Sorrentino, Simona Vigorito, Nicolò Sorà, Luca Greco |
| 20/12/2023 | 0.3 | Aggiunta mapping H/S, gestione dei dati persistenti, controllo degli accessi e sicurezza | Tresy Sorrentino, Simona Vigorito, Nicolò Sorà, Luca Greco |
| 22/12/2023 | 0.4 | Aggiunta controllo globale, condizioni limite, servizi dei sottosistemi e glossario | Tresy Sorrentino, Simona Vigorito, Nicolò Sorà, Luca Greco |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Ruolo nel progetto** | **Acronimo** | **Informazioni del contatto** |
| Simona Vigorito | Team Member | SV | [s.vigorito2@studenti.unisa.it](mailto:s.vigorito2@studenti.unisa.it) |
| Tresy Sorrentino | Team Member | TS | [t.sorrentino15@studenti.unisa.it](mailto:t.sorrentino15@studenti.unisa.it) |
| Nicolò Sorà | Team Member | NS | n.sora@studenti.unisa.it |
| Luca Greco | Team Member | LG | l.greco26@studenti.unisa.it |

**Sommario**

1. Introduzione .............................................................................................................................................. 5

1.1 Scopo del sistema ................................................................................................................................ 5

1.2 Obiettivo di design (Design Goals) ....................................................................................................... 5

1.3. Definizioni, acronimi e abbreviazioni ................................................................................................. 7

1.4 Riferimenti ........................................................................................................................................... 7

1.5 Organizzazione del documento ........................................................................................................... 7

2. Architettura del sistema corrente ............................................................................................................. 8

3. Architettura del Sistema Proposto ............................................................................................................ 8

3.1 Panoramica .......................................................................................................................................... 8

3.2 Decomposizione in sottosistemi .......................................................................................................... 8

3.3 Mapping hardware/software (da sistemare figura) .......................................................................... 10

3.4 Gestione dei dati persistenti.............................................................................................................. 11

Introduzione ............................................................................................................................................ 11

3.5 Controllo degli accessi e sicurezza ..................................................................................................... 13

3.6 Controllo globale del software .......................................................................................................... 15

3.7 Condizioni limite ................................................................................................................................ 15

4. Servizi dei sottosistemi ............................................................................................................................ 17

5. Glossario .................................................................................................................................................. 19

1. Introduzione

**1.1 Scopo del sistema**

Ticket4You, con un approccio innovativo nel settore, si propone di semplificare l'accesso agli eventi dal vivo e aumentare l'entusiasmo degli utenti nel partecipare a esperienze uniche. Il sistema consente agli utenti di registrarsi per seguire i propri eventi preferiti, acquistare comodamente i biglietti online e godersi spettacoli indimenticabili.

Lo scopo di Ticket4You è quello di rendere l’esperienza di acquisto di biglietti un momento semplice e appagante, offrendo agli utenti funzionalità avanzate tra cui:

• ricercare i contenuti informativi relativi ad un evento;

• aggiungere eventi desiderati alla propria lista personale;

•visualizzare dettagli sui biglietti e acquistarli comodamente online.

•ricevere notifiche sugli eventi.

**1.2 Obiettivo di design (Design goals)**

I Design Goal individuati sono stati frazionati nelle seguenti categorie:

• **Performance**: si procede a definire i requisiti di spazio e velocità che il sistema dovrà rispettare.

• **Dependability**: si individua quanto sforzo deve essere speso per minimizzare i crash del sistema e le

loro conseguenze.

• **Maintenance**: determina quanto deve essere difficile modificare il sistema dopo il rilascio.

• **End User**: includono qualità che sono desiderabili dal punto di vista dell’utente, ma che non sono

state coperte dai criteri di performance e dependability.

Ogni Design Goal è stato organizzato secondo:

• **Rank**: viene specificata la priorità secondo un valore compreso tra 1 (massima) e 16 (minima);

• **ID Design Goal**: identificativo e nome assegnato al Design Goal;

• **Descrizione**: descrizione del Design Goal;

• **Categoria:** categoria del Design Goal;

• **RNF di origine:** il requisito non funzionale d’origine, che ha generato il Design Goal;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rank | ID Design Goal | Descrizione | Categoria | RNF di origine |
| 4 | **DG\_1**  Tempi di risposta | Il sistema dovrà garantire tempi di risposta inferiori a 10 secondi | Performance | **RNF\_PRF\_3** |
| 3 | **DG\_2**  Carico massimo | Il sistema sarà in grado di gestire un volume fissato di traffico e operazioni da parte degli utenti durante periodi di elevata affluenza | Performance | **RNF\_PRF\_2** |
| 7 | **DG\_3**  Manutenibilità | Il sistema dovrà favorire la manutenzione, agevolando la rilevazione di errori nel codice | Maintenance | **RNF\_S\_1** |
| 6 | **DG\_4**  Estendibilità | Il sistema dovrà garantire l’aggiunta di nuove funzionalità | Maintenance | **RNF\_S\_2** |
| 1 | **DG\_5**  Disponibilità | Il sistema dovrà essere disponibile per il 99% del tempo | Dependability | **RNF\_AF\_1** |
| 2 | **DG\_6**  Ripristino rapido | Il sistema dovrà poter riprendere a funzionare in caso di guasti nel minor tempo possibile. | Dependability | **RNF\_AF\_2** |
| 5 | **DG\_7**  Interfaccia intuitiva | Il sistema dovrà garantire un’interfaccia intuitiva, affinché l’utente possa usufruire delle funzionalità della piattaforma | End User | **RNF\_U\_1** |
| 8 | **DG\_8**  Consistenza | Il sistema dovrà fornire un’interfaccia le cui componenti dovranno essere coerenti in tutta l’applicazione | End User | **RNF\_U\_3** |

**Design Trade-off**

**Tempo di rilascio vs Staffing.** L’eventuale aumento delle risorse al progetto non potrebbe incrementare la produttività ed il tempo di rilascio del sistema software. Si preferisce, pertanto, non procedere ad incrementare l’insieme delle risorse, facendo riferimento alle forze a disposizione nell’organico.

**Spazio vs Velocità.** Si predilige definire dei tempi di risposta non superiori ai 10 secondi a discapito della memoria. È stata definita questa scelta per garantire una maggiore soddisfazione alle richieste inoltrate dagli utenti.

**Tempo di rilascio vs Funzionalità.** Essendo i tempi di rilascio stringenti, potranno essere rilasciate meno funzionalità di quelle richieste, ma nei tempi giusti.

**Tempo di rilascio vs Qualità**. Siccome i tempi di rilascio risultano essere ristretti, il software viene rilasciato nei tempi prefissati con dei bug, e in tempi successivi correggerli, ottenendo un feedback dal committente rispettando i tempi prefissati e definiti.

|  |  |
| --- | --- |
| TRADE OFF | |
| Tempo di rilascio | Staffing |
| Velocità | Spazio |
| Tempo di rilascio | Funzionalità |
| Tempo di rilascio | Qualità |

**1.3. Definizioni, acronimi e abbreviazioni**

Nella seguente sezioni si presentano e si riportano le definizioni adoperate nel seguente documento:

• **Sottosistema**: un sottoinsieme dei servizi del dominio applicativo, formato da servizi legati da una

relazione funzionale.

• **Design Goal:** le qualità sulle quali il sistema deve essere focalizzato.

• **Dati Persistenti**: dati che sopravvivono all'esecuzione del programma che li ha creati e che dunque

vengono salvati.

• **Mapping Hardware/Software**: studio della connessione tra parti fisiche e logiche di cui si

compongono il sistema.

• **SDD**: System Design Document.

• **RAD**: Requirements Analysis Document.

**1.4 Riferimenti**

Per la realizzazione della documentazione si sono presi in considerazione:

• Slides del corso presenti sulla Piattaforma E-learning del Corso di Laurea in Informatica

dell’Università degli Studi di Salerno.

**1.5 Organizzazione del documento**

Il presente documento di System Design consta di quattro sezioni:

• **Introduzione:** Viene descritto in generale lo scopo del sistema, gli obiettivi di design che il sistema

vuole raggiungere.

• **Architettura software corrente**: Viene descritto lo stato attuale dell’architettura del software già

presente.

• **Architettura software proposta**: Viene descritto come il sistema sarà definito e partizionato in

sottosistemi, il loro mapping Hardware/Software, la gestione dei dati persistenti. Verranno poi

presentate la struttura dei singoli sottosistemi e le boundary conditions riguardanti l’intero sistema.

* **Glossario**: Contiene la lista dei termini usati nel documento con annessa spiegazione.

2. Architettura del sistema corrente

L'architettura del sistema di Ticket4You si distingue per la sua orientazione web-based, differenziandosi dalle principali app mobili come Ticketsms, Ticketswap, Eventbrite, ecc.

Contrariamente a queste piattaforme, Ticket4You offre un'esperienza accessibile attraverso browser web anziché tramite applicazioni native. Le funzionalità principali del sistema includono la ricerca e l'acquisto di biglietti per eventi live.

Le componenti principali sono:

**1. Interfaccia Utente Web:**

- Consente agli utenti di navigare e utilizzare tutte le funzionalità attraverso browser web.

- Fornisce un'interfaccia intuitiva per la ricerca, la visualizzazione degli eventi e l'acquisto dei biglietti.

**2. Motore di Ricerca Eventi:**

- Implementa algoritmi per consentire una ricerca rapida e accurata degli eventi.

- Supporta filtri personalizzati per una ricerca dettagliata.

**3. Sistema di Acquisto Biglietti:**

- Gestisce la transazione sicura e la generazione dei biglietti digitali per gli utenti.

- Integra opzioni di pagamento sicure per i biglietti.

**4. Modulo di Gestione Utenti:**

- Consente la registrazione degli utenti, la gestione dei profili e la personalizzazione delle preferenze.

- Implementa un sistema di notifiche per gli utenti registrati.

L'architettura web-based di Ticket4You mira a offrire un'esperienza utente completa, facilitando la scoperta e l'acquisto di biglietti per eventi live in modo accessibile e coinvolgente.

3. Architettura del Sistema Proposto

**3.1 Panoramica**

Ticket4You si propone di promuovere l'accesso semplificato agli eventi live, mirando a coinvolgere tanto gli appassionati frequenti, quanto coloro che non partecipano regolarmente. L'obiettivo è trasformare l'esperienza degli eventi in un momento di intrattenimento e svago per tutti gli utenti. La piattaforma è progettata per essere inclusiva, adattandosi a diverse fasce d'età e offrendo la possibilità di tracciare l'insieme degli eventi partecipati, in corso o pianificati.

La registrazione su Ticket4You diventa un passo essenziale per gli utenti desiderosi di personalizzare la propria esperienza. Oltre a fornire l'accesso alle funzionalità di acquisto, la registrazione consente agli utenti di salvare i loro eventi preferiti, ricevere suggerimenti personalizzati e partecipare a promozioni esclusive. Registrandosi, gli utenti possono anche beneficiare di un processo di acquisto semplificato e veloce, con la possibilità di tracciare la storia degli acquisti e ricevere notifiche relative agli eventi preferiti.

Il sistema architetturale di Ticket4You segue un modello Three-tier, organizzando i sottosistemi in strati di comunicazione interconnessi. Questa architettura flessibile consente la realizzazione di diverse interfacce per offrire un'esperienza utente coerente e intuitiva, mantenendo al contempo una logica applicativa robusta e scalabile.

**3.2 Decomposizione in sottosistemi**

I sottoinsiemi individuati sono:

• **Registrazione**: Gestisce il processo di registrazione degli utenti, raccogliendo e validando le informazioni necessarie.

• **Login**: Definisce le funzionalità di accesso all'area personale, garantendo la sicurezza e la validità delle credenziali.

• **Area Personale:** Consente la disconnessione dall'area personale, la visualizzazione delle informazioni personali e la modifica dei dati utente.

* **Evento**: Responsabile della visualizzazione delle informazioni dettagliate sugli eventi, ricerca e aggiunta manuale di eventi.

• **Notifica eventi:** Invia notifiche agli utenti registrati riguardo agli eventi preferiti, offerte speciali o eventuali aggiornamenti.

• **Titolo:** Responsabile delle funzioni riguardanti le visualizzazioni delle schede informative dei biglietti

* **StorageConnection**: Si interpone tra i sottosistemi ed il sottosistema di Storage.
* **Storage**: Responsabile della gestione dei dati persistenti con un database.
* **Calendario:** Responsabile della raccolta dei titoli di un utente.

**Immagine che contiene testo, diagramma, schermata, Parallelo

Descrizione generata automaticamente**

**Diagramma architetturale**

**Immagine che contiene diagramma, Piano, testo, Disegno tecnico

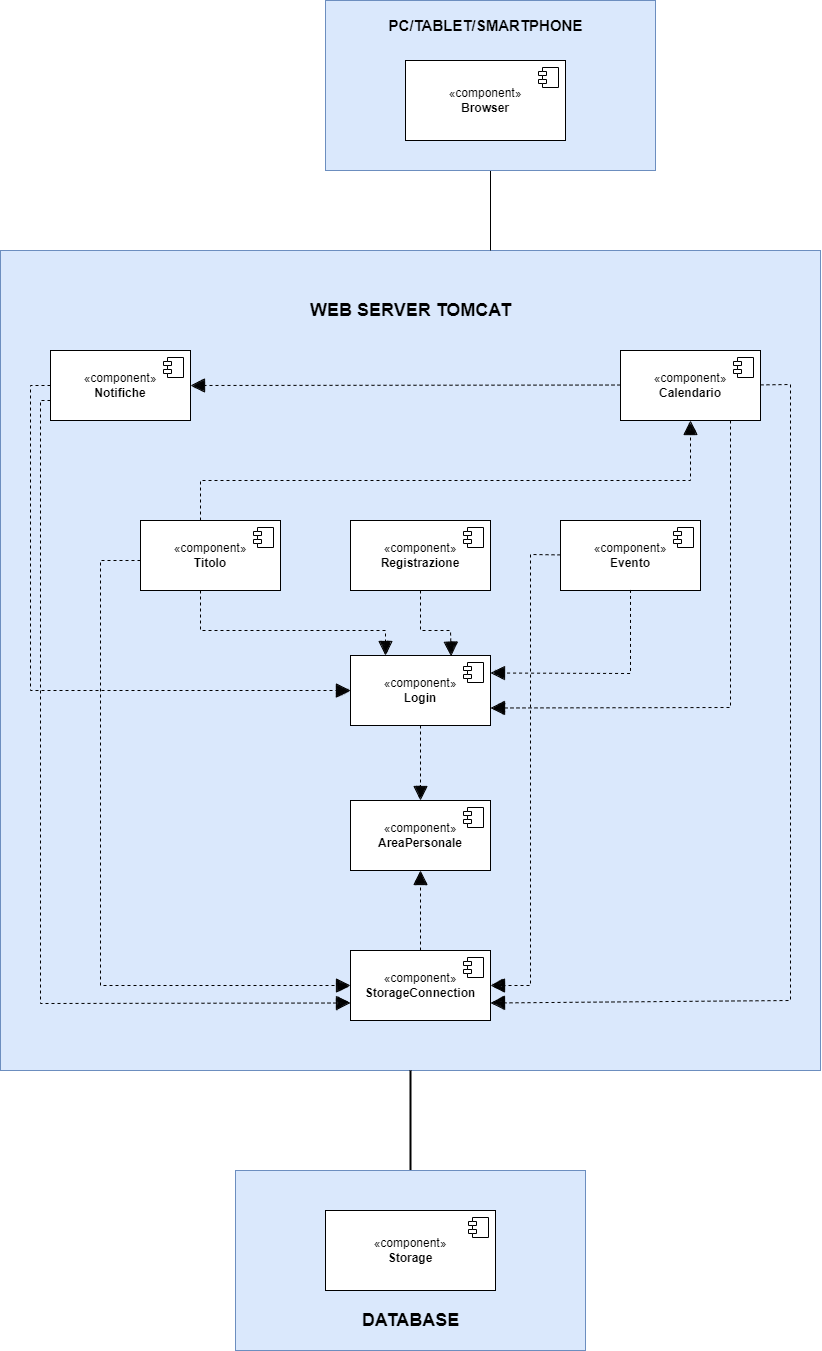
Descrizione generata automaticamente**

**3.3 Mapping hardware/software**

Ticket4You sarà implementato come un'applicazione basata sul web, sfruttando un'architettura client/server. Il client, tramite l'interfaccia fornita dal browser, effettuerà richieste delle operazioni e delle funzionalità attraverso richieste HTTPS. Il server risponderà alle richieste fornendo le funzionalità richieste.

La gestione dei dati persistenti sarà demandata a un sistema di gestione del database (DBMS). Per quanto riguarda il web server, l'incarico sarà affidato a Tomcat, garantendo la gestione efficiente delle richieste e delle comunicazioni con i vari client.

Poiché Ticket4You è progettato come un'applicazione basata sul web, l'accesso sarà possibile da tutti i dispositivi, purché sia disponibile una connessione a Internet per usufruire delle funzionalità offerte dalla piattaforma. La flessibilità di accesso consentirà agli utenti di partecipare agli eventi e gestire i propri biglietti in modo comodo e accessibile da qualsiasi luogo.

****

**3.4 Gestione dei dati persistenti**

**Introduzione**

Per il salvataggio dei dati persistenti del nostro sistema abbiamo deciso di affidarci al database

messo a disposizione da Oracle Corporation.

Il database in questione è MySQL (relational database management system, RDBMS).

**ER: Schema ER del database**

Immagine che contiene diagramma, testo, Piano, Disegno tecnico

Descrizione generata automaticamente

Utente (**ID\_utente**, nome, cognome, password, email)

Organizzatore (**ID\_utente**, numero\_telefono\_or, eventi\_organizzati, indirizzo\_or, piva)

Utente registrato (**ID\_utente**, data\_nascita, indirizzo\_utr, numero\_telefono\_utr)

Amministratore (**ID\_utente**, orario\_inizio\_sessione, orario\_fine\_sessione, numero\_telefono\_am)

Calendario (**ID\_calendario**, data\_calerndario, id\_evento)

Evento (**ID\_evento**, nome\_evento, posti\_disponibili, data\_evento, luogo, ora, tipo)

Titolo (**ID\_titolo**, prezzo\_t , nome\_t, cognome\_t, posto\_prenotato)

Pagamento (**ID\_pagamento**, tipo\_pagamento, stato\_pagamento, somma)

**Dizionario dei dati**

Di seguito verranno mostrati gli attributi e le entità individuate nello schema ER

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome entità Utente** | | | |
| **Descrizione** | | Contiene i dati relativi ad un utente | |
| **Nome campo** | Tipo | Vincolo di chiave | Altri vincoli |
| **ID\_utente** | Int | PRIMARY KEY | NOT NULL |
| **Password** | Varchar(100) |  | NOT NULL |
| **Nome** | Varchar(50) |  | NOT NULL |
| **Cognome** | Varchar(50) |  | NOT NULL |
| **Email** | Varchar(100) |  | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome entità Utente registrato** | | | |
| **Descrizione** | | Contiene i dati relativi ad un utente registrato alla piattaforma | |
| **Nome campo** | Tipo | Vincolo di chiave | Altri vincoli |
| **ID\_utente** | Int | FOREIGN KEY | NOT NULL |
| **Indirizzo\_utr** | Varchar(100) |  | NOT NULL |
| **Numero\_telefono\_utr** | Varchar(20) |  | NOT NULL |
| **Data\_nascita** | Date |  | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome entità Organizzatore** | | | |
| **Descrizione** | | Contiene i dati relativi ad un organizzatore di eventi | |
| **Nome campo** | Tipo | Vincolo di chiave | Altri vincoli |
| **ID\_utente** | Int | FOREIGN KEY | NOT NULL |
| **Indirizzo\_or** | Varchar(200) |  | NOT NULL |
| **piva** | Varchar(50) |  | NOT NULL |
| **Numero\_telefono\_or** | Varchar(20) |  | NOT NULL |
| **Eventi\_organizzati** | Varchar(100) |  | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome entità Amministratore** | | | |
| **Descrizione** | | Contiene i dati relativi ad un amministratore del sistema | |
| **Nome campo** | Tipo | Vincolo di chiave | Altri vincoli |
| **ID\_utente** | Int | FOREIGN KEY | NOT NULL |
| **Orario\_fine\_sessione** | Datetime |  | NOT NULL |
| **Numero\_telefono\_am** | Varchar(20) |  | NOT NULL |
| **Orario\_inizio\_sessione** | Datetime |  | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome entità Evento** | | | |
| **Descrizione** | | Contiene i dati relativi ad un evento | |
| **Nome campo** | Tipo | Vincolo di chiave | Altri vincoli |
| **ID\_evento** | Int | PRIMARY KEY | NOT NULL |
| **Nome\_evento** | Varchar(100) |  | NOT NULL |
| **Prezzo** | Varchar(45) |  | NOT NULL |
| **Posti\_disponibili** | Int |  | NOT NULL |
| **Data\_evento** | Date |  | NOT NULL |
| **Luogo** | Varchar(100) |  | NOT NULL |
| **Ora** | Time |  | NOT NULL |
| **Tipo** | Varchar(200) |  | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome entità Calendario** | | | |
| **Descrizione** | | Contiene i dati relativi al calendario della piattaforma | |
| **Nome campo** | Tipo | Vincolo di chiave | Altri vincoli |
| **ID\_calendario** | Int | PRIMARY KEY | NOT NULL |
| **ID\_evento** | Int | FOREIGN KEY | NOT NULL |
| **Data\_calendario** | Date |  | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome entità Titolo** | | | |
| **Descrizione** | | Contiene i dati relativi ad un titolo di un evento | |
| **Nome campo** | Tipo | Vincolo di chiave | Altri vincoli |
| **ID\_titolo** | Int | PRIMARY KEY | NOT NULL |
| **Prezzo\_t** | Float |  | NOT NULL |
| **Nome\_t** | Varchar(100) |  | NOT NULL |
| **Cognome\_t** | Varchar(100) |  | NOT NULL |
| **Posto\_prenotato** | Int |  | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome entità Pagamento** | | | |
| **Descrizione** | | Contiene i dati relativi al pagamento di un titolo | |
| **Nome campo** | Tipo | Vincolo di chiave | Altri vincoli |
| **ID\_pagamento** | Int | PRIMARY KEY | NOT NULL |
| **Tipo\_pagamento** | Varchar(200) |  | NOT NULL |
| **Somma** | Float |  | NOT NULL |
| **Stato\_pagamento** | Varchar(200) |  | NOT NULL |

**3.5 Controllo degli accessi e sicurezza**

Nella seguente sezione si riporta la matrice degli accessi per poter tracciare degli attori, che possono accedere ai servizi offerti dal sistema.

|  |  |  |
| --- | --- | --- |
| **Attori Oggetti** | **Amministratore** | **Utente non registrato** |
| **Registrazione** |  | RegistrazioneUtenteNonRegistrato |
| **Login** | Login |  |
| **AreaPersonale** | Logout  VisualizzaAreaPersonale  ModificaAreaPersonale  CancellazioneAccount |  |
| **Evento** | CreaEvento  VisualizzaEvento  ModificaEvento  EliminaEvento  AggiungiEvento  RimuoviEvento |  |
| **Notifica eventi** | CreaNotifiche |  |
| **Titolo** | CreaTitolo  InserisciTitolo  ModificaTitolo  VisualizzaTitolo  EliminaTitolo |  |
| **Calendario** | VisualizzaCalendario  ModificaCalendario |  |

|  |  |  |
| --- | --- | --- |
| **Attori o Oggetti** | | **Utente registrato** |
| **Registrazione** |  |  |
| **Login** | Login |  |
| **AreaPersonale** | Logout  VisualizzaAreaPersonale  ModificaAreaPersonale  CancellazioneAccount |  |
| **Evento** | VisualizzaEvento  AggiungiEventiPreferiti |  |
| **Notifiche evento** | VisualizzaNotifiche | |
| **Titolo** | VisualizzaTitolo  AcquistaTitolo  RimuoviTitolo | |
| **Calendario** | VisualizzaCalendario | |

**3.6 Controllo Globale del Software**

Ticket4You è un servizio altamente dinamico e interattivo, dove gli utenti interagiscono con la piattaforma attraverso interfacce grafiche intuitive. A seconda delle funzionalità utilizzate, il controllo e la gestione degli eventi sono centralizzati nel dispatcher, il quale si occupa di coordinare le operazioni dei diversi sottosistemi.

Il modello di controllo del flusso adottato in Ticket4You è orientato agli eventi, poiché la piattaforma è implementata come un'applicazione web. Questo significa che le azioni degli utenti e gli eventi scatenanti sono gestiti in modo reattivo, permettendo una risposta immediata alle interazioni dell'utente. La struttura event-driven contribuisce a fornire un'esperienza utente fluida e altamente responsiva durante la navigazione e l'interazione con la piattaforma.

**3.7 Condizioni limite**

Si presentano le boundary condition inerenti all’avvio del sistema, spegnimento del sistema, fallimento del sistema ed errore di accesso ai dati persistenti.

**Avvio del sistema**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Identificativo** | | ***UCBC\_1: Avvio del sistema*** | | *Data* | | *19/12/2023* |
| *Vers.* | *0.00.001* | |
| *Autore* | *Tresy Sorrentino* | |
| **Descrizione** | | Lo UC consente l’avvio del sistema. | | | | |
| **Attore Principale** | | Amministratore | | | | |
| **Attori secondari** | | NA | | | | |
| **Entry Condition** | | L’amministratore accede al Server. | | | | |
| **Exit condition On success** | | Il sistema viene avviato correttamente. | | | | |
|  | |  | | | | |
| **Exit condition**  **On failure** | | Il sistema non viene avviato correttamente. | | | | |
| **Flusso di Eventi Principali** | | | | | | |
| 1 | | Amministratore: | L’amministratore avvia il sistema tramite il comando apposito. | | | |
| 2 | | Sistema: | Il sistema risponde alla richiesta, rendendo disponibili tutte le funzionalità, dopo aver controllato le componenti necessarie | | | |
| **I Flusso di eventi Alternativo:** le componenti necessarie non funzionano correttamente. | | | | | | |
| **2.a1** | **Sistema** | Il sistema comunica all’amministratore che non è possibile avviare il sistema per via di problematiche riguardanti le componenti necessarie. | | | | |
| **2.a2** | **Amministratore** | L’amministratore prosegue con la correzione delle problematiche segnalate. | | | | |
| **2.a3** | **Amministratore** | L’amministratore procede ad avviare il sistema. | | | | |
| **Flusso di eventi di errore:** il sistema non riesce a far funzionare le componenti necessarie. | | | | | | |
| **2** | **Sistema** | Il sistema notifica all’amministratore con una notifica di errore. | | | | |

**Spegnimento del sistema**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identificativo** | ***UCBC\_2: Spegnimento del sistema*** | | *Data* | | *19/12/2023* |
| *Vers.* | *0.00.001* | |
| *Autore* | *Simona Vigorito* | |
| **Descrizione** | Lo UC consente lo spegnimento del sistema. | | | | |
| **Attore Principale** | Amministratore | | | | |
| **Attori secondari** | NA | | | | |
| **Entry Condition** | L’amministratore effettua il login al Server  AND  Il sistema è stato avviato correttamente  AND  Il sistema non è stato ancora spento. | | | | |
| **Exit condition On success** | Il sistema viene spento correttamente. | | | | |
|  |  | | | | |
| **Exit condition**  **On failure** | Il sistema non viene spento correttamente. | | | | |
| **Flusso di Eventi Principali** | | | | | |
| 1 | Amministratore: | L’amministratore esegue il comando di spegnimento del sistema. | | | |
| 2 | Sistema: | Il sistema verifica che non ci siano connessioni in corso, procedendo allo spegnimento del sistema. | | | |

**Fallimento del sistema**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identificativo** | ***UCBC\_3: Fallimento del sistema*** | | *Data* | | *19/12/2023* |
| *Vers.* | *0.00.001* | |
| *Autore* | *Nicolò Sorà* | |
| **Descrizione** | Lo UC definisce il comportamento del sistema quando si verifica un fallimento. | | | | |
| **Attore Principale** | Amministratore | | | | |
| **Attori secondari** | NA | | | | |
| **Entry Condition** | Il sistema si interrompe in modo improvviso. | | | | |
| **Exit condition On success** | Il sistema viene riavviato correttamente. | | | | |
|  |  | | | | |
| **Exit condition**  **On failure** | Il sistema non viene riavviato correttamente. | | | | |
| **Flusso di Eventi Principali** | | | | | |
| 1 | Amministratore: | Include l’avvio del sistema (UCBC\_1). | | | |

**Errori di Accesso ai Dati Persistenti**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Identificativo** | ***UCBC\_4: Errori di accesso ai dati persistenti*** | | *Data* | | *19/12/2023* |
| *Vers.* | *0.00.001* | |
| *Autore* | *Luca Greco* | |
| **Descrizione** | Lo UC definisce il comportamento del sistema quando i dati persistenti risultano corrotti o non si riesce ad accedervi. | | | | |
| **Attore Principale** | Amministratore | | | | |
| **Attori secondari** | NA | | | | |
| **Entry Condition** | Il sistema non riesce ad accedere ai dati persistenti  OR  I dati persistenti sono corrotti. | | | | |
| **Exit condition On success** | Il sistema riesce a funzionare correttamente. | | | | |
|  |  | | | | |
| **Exit condition**  **On failure** | Il sistema non riesce a funzionare correttamente. | | | | |
| **Flusso di Eventi Principali** | | | | | |
| 1 | Sistema | Comunica all’amministratore di non poter accedere ai dati persistenti. | | | |
| 2 | Sistema | Smette di eseguire le richieste e termina con un errore. | | | |
| 3 | Amministratore | Include lo spegnimento del sistema (UCBC\_2). | | | |
| 4 | Sistema | Riprende la corretta esecuzione delle attività. | | | |
| **I Flusso di eventi Alternativo:** le funzionalità non vengono ripristinate al riavvio. | | | | | |
| 3 | Amministratore | Si fa carico della risoluzione del problema. | | | |

4. Servizi dei sottoinsiemi

*Sottosistema Registrazione*

|  |  |  |
| --- | --- | --- |
| **Servizio** | **Descrizione** | **Interfaccia** |
| **Registrazione utente** | La seguente funzionalità permette la registrazione dell’utente non registrato alla piattaforma. | RegistrazioneService |

*Sottosistema Login*

|  |  |  |
| --- | --- | --- |
| **Servizio** | **Descrizione** | **Interfaccia** |
| **Login Utente registrato** | La seguente funzionalità permette all’utente registrato di effettuare l’accesso alla piattaforma. | LoginService |

*Sottosistema AreaPersonale*

|  |  |  |
| --- | --- | --- |
| **Servizio** | **Descrizione** | **Interfaccia** |
| **Logout Utente registrato** | La seguente funzionalità permette all’utente registrato di disconnettersi dal suo account. | AreaPersonaleService |
| **Visualizzazione dati personali** | La seguente funzionalità permette all’utente registrato di visualizzare i dati personali inseriti nel sistema. | AreaPersonaleService |
| **Modifica dati personali** | La seguente funzionalità permette all’utente registrato di modificare i dati personali inseriti nel sistema. | AreaPersonaleService |
| **Eliminazione account** | La seguente funzionalità permette all’utente registrato di eliminare l’account personale dal sistema. | AreaPersonaleService |

*Sottosistema Evento*

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Visualizzazione scheda informativa | La seguente funzionalità permette all’utente di visualizzare la scheda informativa relativa all’evento selezionato | EventoService |
| Ricerca evento | La seguente funzionalità permette all’utente di ricercare un evento | EventoService |
| Modifica evento | La seguente funzionalità permette all’amministratore/organizzatore di modificare le informazioni dell’evento | EventoService |
| Aggiunta posti | La seguente funzionalità permette all’amministratore/organizzatore di aggiungere posti per l’evento | EventoService |
| Rimozione evento | La seguente funzionalità permette all’amministratore/organizzatore di rimuovere l’evento dal sistema | EventoService |

*Sottosistema Notifica Evento*

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Invio notifiche | La seguente funzionalità si occupa di inviare notifiche agli utenti | NotificaEventoService |

*Sottosistema Titolo*

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Visualizzazione scheda informativa | La seguente funzionalità permette all’utente di visualizzare la scheda informativa relativa al titolo selezionato | TitoloService |
| Acquista titolo | La seguente funzionalità permette all’utente di acquistare il titolo desiderato | TitoloService |
| Rimozione titolo | La seguente funzionalità permette all’utente di rimuovere il titolo selezionato | TitoloService |

*Sottosistema Calendario*

|  |  |  |
| --- | --- | --- |
| Servizio | Descrizione | Interfaccia |
| Visualizzazione calendario | La seguente funzionalità permette all’utente di visualizzare il calendario degli eventi | CalendarioService |

5. Glossario

|  |  |
| --- | --- |
| **Termine** | **Definizione** |
| Piattaforma | È la base software o hardware sulla quale è sviluppata o in esecuzione l’applicazione. |
| Account | Rappresentazione dell’utente in formato digitale. |
| Biglietto | Rappresentazione in forma digitale di un biglietto, dove viene definita dettagliatamente la relativa scheda informativa. |
| Ticket4You | Nome della web app realizzata. |