­

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| System Design Document  Progetto  SalernArte   |  |  | | --- | --- | | Versione | 0.1 | | Data | 05/05/2022 | | Destinatario | Studenti di Ingegneria del Software 2021/22 | | Presentato da | Martino Lucia [0512105234], Longo Marco [0512105945], Della Pepa Alessia [0512105720] | | Approvato da |  | |

Membri del Team

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Matricola** | **Acronimo** | **Informazioni di contatto** |
| Della Pepa Alessia | 0512105720 | DPA | a.dellapepa5@studenti.unisa.it |
| Longo Marco | 0512105945 | LM | m.longo36@studenti.unisa.it |
| Martino Lucia | 0512105234 | ML | l.martino11@studenti.unisa.it |

RevisionHistory

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versione** | **Descrizione** | **Autori** |
| 08/05/2022 | 0.1 | Prima stesura: aggiunta dello scopo del sistema, design goals e trade-off. | DPA, ML, LM |
| 11/05/2022 | 0.2 | Sistema proposto: decomposizione in sottosistemi, component diagram UML dei sottosistemi e mapping hardware/software | DPA, ML, LM |
| 13/05/2022 | 0.3 | Diagramma architetturale,  gestione dati persistenti | DPA, ML, LM |
| 18/05/2022 | 0.4 | Inserimento sottosistema  Diagramma architetturale,  dizionario dei dati dell’acquisto | LM |
| 18/05/2022 | 0.5 | Controllo globale, revisione della prima parte del documento | DPA, ML, LM |
| 18/05/2022 | 0.6 | Inserimento servizi di sottosistema,  gestione limite e controllo degli accessi e sicurezza | LM |
| 27/05/2022 | 0.7 | Inserimento dei sottosistemi di  registrazione ed autenticazione | ML |

Sommario

[Membri del Team 2](#_Toc102670806)

[RevisionHistory 2](#_Toc102670807)

[1. Introduzione 5](#_Toc102670808)

[1.1. Scopo del Sistema 5](#_Toc102670809)

[1.2. Obiettivi di Design (Design Goals) 5](#_Toc102670810)

[1.3. Definizioni, Acronimi e Abbreviazioni 5](#_Toc102670811)

[1.4. Riferimenti 5](#_Toc102670812)

[1.5. Organizzazione del Documento 5](#_Toc102670813)

[2. Architettura dei Sistema Corrente 5](#_Toc102670814)

[3. Architettura del Sistema Proposto 5](#_Toc102670815)

[3.1. Panoramica della sezione 5](#_Toc102670816)

[3.2. Decomposizione in sottosistemi 5](#_Toc102670817)

[3.3. Mapping Hardware/Software 5](#_Toc102670818)

[3.4. Gestione dei dati persistenti 5](#_Toc102670819)

[3.5. Controllo degli accessi e sicurezza 5](#_Toc102670820)

[3.6. Controllo globale del software 6](#_Toc102670821)

[3.7. Condizioni limite 6](#_Toc102670822)

[4. Servizi dei sottosistemi 6](#_Toc102670823)

[5. Glossario 6](#_Toc102670824)

1. Introduzione

## Scopo del Sistema

SalernArte si propone come obiettivo principale di realizzare un’agenzia online specializzata nella vendita di biglietti riguardanti mostre d’arte ed eventi teatrali e culturali nel salernitano. L’obiettivo è quello di facilitare a tutti i cittadini la ricerca di iniziative culturali salernitane, raggruppando queste ultime in un unico ambiente semplice ed intuitivo, e di ottimizzare il lavoro di organizzatori di eventi che si interfacciano ai cittadini.

Il sistema è gestito da uno o più amministratori e permette l’iscrizione da parte di scolaresche, utenti e organizzatori d’eventi.

La funzionalità principale del sistema consisterà nel permettere agli utenti iscritti alla piattaforma di acquistare biglietti per gli eventi caricati dall’organizzatore e approvati dall’amministratore.

## Obiettivi di Design (Design Goals)

Nella presente sezione si andranno a presentare i Design Goals, ovvero le qualità sulle quali il sistema deve essere focalizzato, formalizzati esplicitamente così che qualsiasi importante decisione di design può essere fatta consistentemente seguendo lo stesso insieme di design goal.

I design goals sono stati suddivisi nelle seguenti categorie:

* Usabilità: include i requisiti relativi alla facilità di utilizzo del sistema
* Prestazioni: include i requisiti prestazionali imposti al sistema
* Affidabilità: include i requisiti relativi all’affidabilità del sistema e delle sue componenti
* Supportabilità: include i requisiti che fanno riferimento alla manutenzione del sistema

Ciascun design goal è descritto da:

* **Rank**: valore di priorità compreso tra 1 e 8 (1 massima e 8 minima).
* **ID Design Goal**: un identificatore univoco e un nome esplicativo.
* **Descrizione:** una descrizione del design goal.
* **Categoria**: la categoria di appartenenza del design goal.
* **RNF di origine**: il requisito non funzionale che lo ha generato.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Categoria** | **ID Design Goal** | **Descrizione** | **Rank** | **RNF di origine** |
| Usabilità | DG\_1  Interfacciabilità | Il sistema deve utilizzare interfacce  semplici grazie all’utilizzo di pulsanti ed etichette, menù chiari e di facile  gestione | 8 | RNF\_U1 |
| DG\_2 Utilità | Il sistema deve permettere di  concludere ogni operazione con un massimo di 10 passaggi. | 3 | RNF\_U2 |
| Prestazioni | DG\_3  Tempo di risposta | Il sistema deve garantire un tempo di risposta non superiore di 5 secondi | 2 | RNF\_P1 |
| Affidabilità | DG\_4  Sicurezza  d’accesso | Il sistema utilizzerà un form di autenticazione che permetterà solo agli utenti  registrati al sistema di poter usufruire delle funzionalità ad essi concesse | 4 | RNF\_A1 |
| DG\_5  Sicurezza dei dati | Il sistema memorizzerà le credenziali d’accesso degli utenti utilizzando la  codifica SHA1 garantendo una  protezione dei dati sensibili | 1 | RNF\_A2 |
| DG\_6  Robustezza | Il sistema controllerà il formato dei dati inseriti nei form così da garantire il corretto funzionamento del sito | 5 | RNF\_A3 |
| Supportabilità | DG\_7  Manutenibilità | Il sistema sarà facilmente manutenibile grazie alla modularità delle funzionalità a livello di codice | 6 | RNF\_S1 |
| DG\_8  Modificabilità | Il sistema presenta una documentazione chiara e un codice ricco di  commenti al fine di rendere facilmente comprensibile ogni parte del sistema in modo da facilitare possibili modifiche future del sistema | 7 | RNF\_S2 |

**Trade-off**

|  |  |
| --- | --- |
| Trade-off | **Descrizione** |
| Tempo di risposta vs sicurezza | Il sistema tenderà a dare maggiore priorità alla sicurezza a discapito del tempo di risposta che potrebbe aumentare fino a 10 secondi |

## Definizioni, Acronimi e Abbreviazioni

|  |  |
| --- | --- |
| **Acronimo/Abbreviazione** | **Definizione** |
| SDD | System Design Document |
| DBMS | Database Management System |
| JDBC | Java Database Connectivity |

## Riferimenti

Di seguito una lista di riferimenti ad altri documenti utili durante la lettura:

## Organizzazione del Documento

Il presente document di SD è formato da 4 sezioni:

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

1. Architettura dei Sistema Corrente

Al momento, non esiste alcun software che condensi l’interezza delle funzionalità di SalernArte in un unico servizio. Il mercato delle possibili alternative a questo software è pertanto incredibilmente frammentato e non esiste una reale architettura a cui è possibile confrontare in maniera ragionevole il sistema.

1. Architettura del Sistema Proposto

## Panoramica della sezione

Il Sistema proposto è basato sullo stile architetturale Three Tier. Questo tipo di architettura è stata scelta poiché si sposa perfettamente per lo sviluppo di Web Application, come previsto dal nostro progetto. L’architettura Three Tier si divide in 3 strati:

* L’interface layer (include tutti i boundary object che interfacciano con l’utente).
* L’application logic layer (include tutti gli oggetti relativi al controllo e alle entità che realizzano l’elaborazione, le regole di verifica e la notifica richieste dall’applicazione).
* Lo storage layer (effettua la memorizzazione, il recupero e l’interrogazione di oggetti persistenti).

Per la parte del Front-end saranno utilizzati HTML5 e CSS3.Per quanto riguarda il Back-end verranno utilizzati Java e Javascript. Per la gestione del database saranno usati MySQL e JDBC.

## Decomposizione in sottosistemi

I sottosistemi Identificati sono:

* **Registrazione:** si occupa di gestire la registrazione dei vari tipi di utente: UtenteRegistrato, Scolaresca, Organizzatore.
* **Autenticazione:** è responsabile delle funzionalità di Login, Logout, Recupero password, visualizzazione area utente e modifica dell’account utente.
* **Gestione Eventi**: si occupa delle funzionalità riguardanti la richiesta di inserimento di un nuovo evento da parte dell’organizzatore, l’inserimento dell’evento nel sistema da parte dell’amministratore, la visualizzazione dell’evento, la richiesta di modifica dell’evento da parte dell’organizzatore, la conferma della modifica da parte dell’amministratore e la rimozione dell evento da parte dell’organizzatore.
* **Gestione Acquisti:** si occupa delle funzionalità per la gestione del carrello utente (aggiunta biglietti al carrello, modifica quantità biglietti, rimozione biglietti e svuota carrello) e la finalizzazione degli acquisti.
* **Persistenza:** si occupa di gestire la persistenza dei dati con un database.
* **Storage JPA:** si interpone tar i vari sottosistemi e il sottosistema di Persistenza.

Sono mostrate di seguito le dipendenze tra i sottosistemi attraverso un component diagram UML.



Di seguito viene data una vista nel dettaglio di ciascun sottosistema, evidenziando le componenti principali:

* **GUI:** Graphic User Interface, che saranno utilizzate per creare le pagine web che saranno mostrate al cliente.
* **Controller:** si occupa della logica di controllo del sottosistema.
* **Service:** si occupa della logica di business.
* **DAO:** Data Access Object, che si occupa di fornire accesso ai dati persistenti.

**Diagramma Architetturale**



**SOTTOSISTEMI**

**REGISTRAZIONE AUTENTICAZIONE – LUCIA**

**GESTIONE EVENTI ALESSIA**

**Sottosistema Registrazione**

****

**Sottosistema Autenticazione**

****

**Sottosistema Gestione Acquisti**



## Mapping Hardware/Software

In questa sezione è illustrata la mappatura tra Hardware e Software attraverso l’utilizzo di un UML deployment diagram. È possibile osservare che l’app realizzata è basata su una piattaforma hardware costituita da un server che risponde alle richieste HTTP inviate dal client. Il client sarà un qualsiasi dispositivo dotato di connessione ad Internet e di un browser.



* 1. Gestione dei dati persistenti

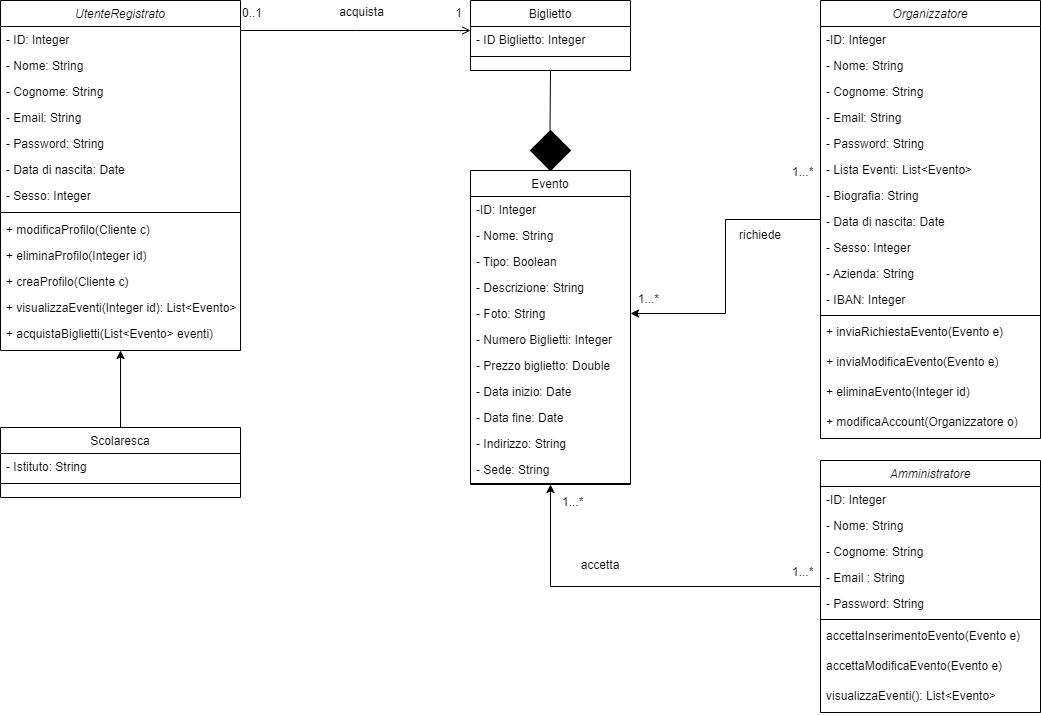
**Introduzione**

Per la gestione del salvataggio dei dati persistenti del sistema si è deciso di utilizzare un database relazionale. Si è scelto questo modello in modo da permettere, per quanto possibile, di rispettare i design goals e potendo quindi contare su:

* **Sicurezza dei dati,** in quanto il sistema memorizzerà le credenziali d’accesso degli utenti utilizzando la codifica SHA1
* **Affidabilità dei dati,** in quanto il DBMS offre dei metodi per salvare copie dei dati e per ripristinare lo stato della base di dati in caso di guasti software e hardware.

**CD\_SDD: Entity Class Diagram ristrutturato**

Di seguito viene riportato il class diagram con alcune modifiche rispetto a quello presente nel RAD.



UTENTE REGISTRATO SCOLARESCA LUCIA

**Dizionario dei dati**

Di seguito viene riportato il dizionario dei dati relativo ad ogni entità del sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome Entità** | **Evento** | | |
| **Descrizione** | Contiene I dati relative all’evento | | |
| **Nome campo** | **Tipo** | **Vincolo di chiave** | **Altri vincoli** |
| **Id** | Integer (32) | PRIMARY KEY | NOT NULL |
| **Nome** | Varchar (50) |  | NOT NULL |
| **Tipo** | Boolean |  | NOT NULL |
| **Descrizione** | Varchar (320) |  | NOT NULL |
| **Foto** | Varchar (100) |  | NOT NULL |
| **Numero Biglietti** | Interger (32) |  | NOT NULL |
| **Prezzo Biglietti** | Double (32) |  | NOT NULL |
| **Data inizio** | Date |  | NOT NULL |
| **Data fine** | Date |  | NOT NULL |
| **Indirizzo** | Varchar (100) |  | NOT NULL |
| **Sede** | Varchar (100) |  | NOT NULL |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome Entità** | **Biglietto** | | |
| **Descrizione** | Contiene I dati relative al biglietto di un evento | | |
| **Nome campo** | **Tipo** | **Vincolo di chiave** | **Altri vincoli** |
| **Id\_biglietto** | Integer (32) | PRIMARY KEY | NOT NULL |
| **Id\_evento** | Integer (32) | FOREIGN KEY | NOT NULL |
|  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome Entità Amministratore Autore: LM** | | | | |
| Descrizione | Contiene i dati relativi all’amministratore del sistema | | |
| Nome Campo | Tipo | Vincolo di chiave | Altri vincoli |
| ID | INTEGER(32) | PRIMARY KEY | NOT NULL |
| Nome | VARCHAR(30) |  | NOT NULL |
| Cognome | VARCHAR(30) |  | NOT NULL |
| Email | VARCHAR(50) |  | NOT NULL |
| Password | VARCHAR(30) |  | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nome Entità Organizzatore Autore: LM** | | | | |
| Descrizione | Contiene i dati relativi all’organizzatore di un evento | | |
| Nome Campo | Tipo | Vincolo di chiave | Altri vincoli |
| ID | INTEGER(32) | PRIMARY KEY | NOT NULL  AUTO INCREMENT |
| Nome | VARCHAR(30) |  | NOT NULL |
| Cognome | VARCHAR(30) |  | NOT NULL |
| Email | VARCHAR(50) |  | NOT NULL |
| Password | VARCHAR(30) |  | NOT NULL |
| Biografia | VARCHAR(100) |  | NOT NULL |
| Data di nascita | DATE |  | NOT NULL |
| Sesso | INTEGER |  | NOT NULL Può assumere  valore:   * 0, se maschio * 1, se femmina * 2, se non   specificato |
| Azienda | VARCHAR(50) |  | NOT NULL |
| IBAN | INTEGER(27) |  | NOT NULL |

* 1. Controllo degli accessi e sicurezza

Di seguito viene mostrata la matrice degli accessi per poter tenere traccia di quali attori possono accedere ai quali dei servizi offerti dal sistema.

|  |  |  |  |
| --- | --- | --- | --- |
| Attori  Oggetti | **Ospite** | **Utente Registrato** | **Scolaresca** |
| Registrazione |  |  |  |
| Autenticazione |  |  |  |
| GestioneEventi |  |  |  |
| GestioneAcquisti |  | Visualizza carrello  Acquista biglietti  Aggiungi al carrello  Rimuovi dal carrello  Svuota carrello  Visualizza acquisti effettuati  Modifica Carrello | Visualizza carrello  Acquista biglietti  Aggiungi al carrello  Rimuovi dal carrello  Svuota carrello  Visualizza acquisti effettuati  Modifica Carrello |

|  |  |  |
| --- | --- | --- |
| Attori  Oggetti | **Amministratore** | **Organizzatore** |
| Registrazione |  |  |
| Autenticazione |  |  |
| GestioneEventi |  |  |
| GestioneAcquisti |  |  |

* 1. Controllo globale del software

Il flusso del sistema **SalernArte** fornisce una funzionalità che richiede una continua interazione da parte dell'utente, ragione per cui, abbiamo scelto un controllo del flusso globale esplicito centralizzato di tipo event-driven, ovvero guidato dagli eventi.

Le azioni compiute dall’utente scateneranno un evento che viene gestito dall’handler associato, che a sua volta indirizza il controllo del flusso al sottosistema che si occupa della logica di controllo, e permette al gestore del controllo di rivolgersi ai servizi della logica applicativa.

* 1. Condizioni limite

In questa sezione sono illustrati gli scenari e i casi d’uso concernenti le condizioni limite di: avvio del sistema, spegnimento del sistema, fallimento del sistema ed errore di accesso ai dati persistenti.

AVVIO SISTEMA LUCIA

SPEGNIMENTO E FALLIMENTO ALESSIA

**Errore di accesso ai dati persistenti**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Identificativo | | UCBC\_4 – Errore di Accesso ai Dati Persistenti | Data | 18/05/2022 |
| Versione | 1.0 |
| Autori | Longo Marco |
| Descrizione | | Lo Use Case descrive il comportamento del sistema nel caso in cui ci fosse un errore nell’accesso dei dati | | |
| Attore principale | | Amministratore | | |
| Attori secondari | | NA | | |
| Entry condition | | Il Sistema non può accedere ai dati persistenti OR i dati  risultano corrotti | | |
| Exit condition  On success | | Il Sistema riprende il normale funzionamento | | |
| Exit condition  On failure | | Il Sistema non riprende il normale funzionamento | | |
| Flusso di eventi principale | | | | |
| 1 | Sistema | Notifica l’amministratore dell’impossibilità di accedere ai  dati persistenti | | |
| 2 | Sistema | Cessa di processare eventuali richieste dall’esterno e  risponde a tutte le richieste con un messaggio di errore. | | |
| 3 | Amministratore | Include UCBC\_2 | | |
| 4 | Amministratore | Ripristina l’accessibilità o la sanità dei dati persistenti. | | |
| 5 | Amministratore | Include UCBC\_1 | | |

SERVIZI DI SOTTOSISTEMI:

REGISTRAZIONE AUTENTICAZIONE LUCIA

GESTIONE EVENTI ALESSIA

1. Servizi dei sottosistemi

**Sottosistema Gestione Acquisti**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Servizio** | | **Descrizione** | | **Interfaccia** | |
| Visualizza carrello | Permette di visualizzare i  prodotti presenti nel carrello | | GestioneAcquistiService | |
| Aggiungi al carrello | Permette di aggiungere al  carrello un evento | | GestioneAcquistiService | |
| Acquista biglietti | Permette di acquistare i biglietti nel carrello | | GestioneAcquistiService | |
| Rimuovi dal carrello | Permette di rimuovere dal  carrello un biglietto specifico | | GestioneAcquistiService | |
| Svuota carrello | Permette di rimuovere tutti i biglietti presenti nel carrello | | GestioneAcquistiService | |
| Visualizza acquisti effettuati | Permette di visualizzare nella propria area utente gli acquisti precedentemente effettuati | | GestioneAcquistiService | |
| Modifica carrello | Permette di modificare la quantità di ogni biglietto  presente nel carrello | | GestioneAcquistiService | |

1. Glossario