|  |
| --- |
| **Università degli Studi di Salerno Corso di Ingegneria del Software** |

***SocialBook  
Documento System Design***



**Partecipanti:**

|  |  |
| --- | --- |
| Nome | Matricola |
| Barbato Alessia | 0512105858 |
| Proietto Angelica | 0512105762 |
| Russo Luca | 0512105840 |
| Russo Salvatore | 0512106098 |

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versione | Descrizione | Autore |
| 22/12/2020 | 1.0 | Stesura introduzione (Paragrafi 1.1, 1.2, 1.3, 1.4) | Barbato Alessia |
| 26/12/2020 | 2.0 | Stesura architettura sistema corrente e proposto (Paragrafi 3.1, 3.2, 3.3, 3.4) | Barbato Alessia |
| 29/12/2020 | 3.0 | Stesura controllo degli accessi e sicurezza, controllo globale del software, condizioni limite | Proietto Angelica |
| 30/12/2020 | 4.0 | Stesura servizi dei sottosistemi, glossario | Russo Luca |

Indice

[1. Introduzione 4](#introduzione)

[1.1 Obiettivi del sistema](#obiettivi) 4

[1.2 Design goals 4](#dg)

[1.3 Acronimi e abbreviazioni 5](#acr_e_abbr)

[1.4 Riferimenti 5](#riferimenti)

[1.5 Panoramica 5](#panoramica)

[2. Architettura del sistema corrente 6](#sis_corrente)

[3. Architettura del sistema proposto 6](#sis_proposto)

3.1 Panoramica 6

[3.2 Decomposizione in sottosistemi](#decomposizione) 7

[3.3 Mapping hardware/software](#mapping) 9

[3.3.1 Deployment](#deployment) 9

[3.4 Gestione dati persistenti](#persistenti) 10

[3.4.1 Schema logico 11](#schema)

[3.4.2 Struttura delle tabelle](#tabelle) 12

[3.5 Controllo degli accessi e sicurezza 16](#sicurezza)

[3.6 Controllo globale del software](#controllo_sw) 18

[3.7 Condizioni limite](#limite) 18

[4. Servizi dei sottosistemi](#servizi) 19

[5. Glossario](#glossario) 21

1. Introduzione
   1. Obiettivi del sistema

L’obiettivo del sistema è quello di offrire a tutti gli appassionati di lettura una piattaforma online sicura che permetta all’utente non solo di acquistare libri (in formato digitale), ma anche di organizzarli come meglio si preferisce, recensirli, e comunicare con gli altri utenti.

L’interfaccia del sistema dovrà essere estremamente intuitiva, in modo da invogliare gli utenti ad interagire tra di loro.

L’obiettivo della piattaforma è quella di fornire un servizio veloce e sempre aggiornato in modo da garantire l’esperienza migliore possibile agli utenti.

* 1. Design Goals

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **DESCRIZIONE** | **CATEGORIA** | **ORIGINE** |
| **DG\_1 Tempo di**  **risposta** | Il sistema garantisce il minor tempo possibile di risposta (massimo 1 sec). | Prestazioni | NFR\_4 |
| **DG\_2 Portabilità** | Il sistema è responsive (visualizza correttamente le pagine adattandosi ai display di dispositivi diversi: pc, tablet e cellulari). | Usabilità  Implementazione  Mantenimento | NFR\_1  NFR\_6 |
| **DG\_3 Usabilità** | Il sistema è strutturato con una interfaccia intuitiva (organizzazione di pulsanti, form e menu chiara e ben precisa). | Usabilità | NFR\_2 |
| **DG\_4 Throughput** | In caso di massimo carico del lavoro, il sistema non deve bloccarsi. | Prestazioni | NFR\_4 |
| **DG\_5 Sicurezza** | Il sistema protegge i dati sensibili degli utenti attraverso la crittografia; l’utente può usufruire esclusivamente delle funzionalità associate alla propria categoria. | Affidabilità  Security | NFR\_7  NFR\_3 |
| **DG\_6 Memoria** | Il sistema fa affidamento sul database relazionale MySQL per rendere persistenti e gestire le informazioni | Implementazione | NFR\_6 |
| **DG\_7 Disponibilità** | Fatta eccezione per il tempo di manutenzione, le funzionalità del sistema sono accessibili in ogni momento. | Affidabilità | NFR\_3 |
| **DG\_8 Robustezza** | L’inserimento di dati non conformi o il mancato riempimento dei campi viene segnalato tramite notifica di errore. | Affidabilità | NFR\_3 |
| **DG\_9 Affidabilità** | Il sistema garantisce una corretta gestione delle funzionalità. | Affidabilità | NFR\_3 |
| **DG\_10 Adattabilità** | Il sistema è flessibile (modulare ed estendibile) e si adatta alle nuove necessità dell’utente. | Implementazione  Mantenimento | NFR\_5  NFR\_6 |

* 1. Acronimi e abbreviazioni

**RAD:** Requirements Analysis Document.

**FR:** Requisiti funzionali.

**NFR:** Requisiti non funzionali.

**SC:** Scenario.

**UC:** Use case.

**HTTPS:** HyperText Transfer Protocol Secure.

**MVC:** Model-View-Controller.

**DG:** Design Goal.

**GUI:** Graphic User Interface.

**DB:** DataBase.

**DBMS:** DataBase Management System.

**JSP:** Java Server Page.

**Utente:** Un utilizzatore della piattaforma che non si è ancora registrato.

**Utente registrato:** Un utilizzatore iscritto alla piattaforma.

**Customer manager:** Amministratore che si occupa della gestione degli utenti.

**System manager:** Amministratore che si occupa della gestione di bug, segnalati da utenti che utilizzano la piattaforma.

**Catalogue manager:** Amministratore che si occupa della gestione del catalogo libri.

**Admin:** Generalizzazione di customer manager,catalogue manager e system manager.

* 1. Riferimenti

● Object-Oriented Software Engineering Using UML, Patterns, and Java, Bernd Bruegge & Allen H. Dutoit.

● SocialBook RAD.

* 1. Panoramica

Il documento si compone di una prima parte in cui vengono introdotti gli obiettivi di design.

Al secondo punto del documento viene presentata l’architettura del sistema

corrente.

Al terzo punto viene presentata l’architettura del sistema proposto e nel dettaglio:

● La decomposizione del sistema in sottosistemi.

● Il mapping hardware/software.

● La gestione dei dati persistenti.

● Il controllo degli accessi e della sicurezza.

● Il controllo del flusso globale del sistema.

● Le condizioni limite.

Al quarto punto vengono presentati i servizi di ogni sottosistema.

Al quinto punto viene fornito il glossario dei termini utilizzati nel documento con le relative definizioni.

1. Architettura del sistema corrente

Attualmente non esiste una piattaforma a tema e-book che offra sia funzionalità di marketing che quelle di un blog. Inoltre, i social attuali afferenti a tale campo sono popolati da cerchie di utenti della stessa zona (per fattori socio-lessicali); dato che limita pericolosamente la qualità della rete di users di cui si potrebbe disporre.

Infine, i siti di questo tipo vengono sponsorizzati da case editrici che forzano la presenza di libri da loro pubblicati affievolendo l’eterogeneità dei prodotti.

1. Architettura del sistema proposto
   1. Panoramica

Il sistema proposto vuole rendere più completa la gamma di funzionalità di una piattaforma associata agli e-book. Per far ciò, offre delle funzionalità fondamentali:

● consente l’acquisto degli e-book e la possibilità di recensirli;

● consente di creare liste di e-book da comprare e leggere;

● consente di seguire altri utenti e/o liste di e-book;

Per la progettazione e per lo sviluppo di SocialBook si è scelta l’architettura MVC (Model-View-Controller) che separa la logica di business da quella di presentazione.



**Model:** Si occupa di gestire i dati persistenti, contiene metodi di lettura e scrittura sul Database.

**View:** Si occupa di far visualizzare i dati all’utente e dell’interazione tra quest’ultimo ed il sistema.

**Controller:** Si occupa della logica di controllo dell’applicazione interagendo con le altre due componenti.

* 1. Decomposizione in sottosistemi

Presentiamo una decomposizione del nostro sistema in 3 layer (o livelli):

● Presentation Layer

● Application Layer

● Data Access Layer

Questi livelli si occupano di gestire funzionalità ed aspetti differenti del sistema.

##### Presentation Layer (View)

Livello che gestisce la comunicazione con le entità esterne al sistema tramite delle componenti che si occupano di rappresentare l’informazione verso i client, consentendo essi di interagire con il sistema.

● *Utente non registrato:* GUI che gestisce l’interfaccia degli utenti che non hanno effettuato la registrazione e/o l’accesso alla piattaforma, i quali sono limitati alla visualizzazione del catalogo dei libri e dei profili degli altri utenti registrati.

● *Utente registrato:* GUI che gestisce l’interfaccia degli utenti che sono registrati e hanno effettuato l’accesso alla piattaforma e permette di visualizzare: il profilo personale (con ordini, ticket e booklist), il carrello, il catalogo dei libri, i preferiti e il profilo di altri utenti.

● *Customer Manager:* GUI che gestisce l’interfaccia dedicata al responsabile della gestione degli utenti e permette di eliminare utenti/commenti degli utenti.

● *System Manager:* GUI che gestisce l’interfaccia dedicata al responsabile della gestione del sistema e permette la modifica del codice per la risoluzione di bug o implementazione di nuove features.

● *Catalogue Manager:* GUI che gestisce l’interfaccia dedicata al responsabile del catalogo e permette di aggiungere/eliminare/modificare il prezzo dei libri.

##### Application Layer (Controller)

Livello del sistema che si occupa del processamento dei dati, necessario per produrre i risultati da inoltrare al Presentation Layer.

● *Gestione Utente:* Sottosistema che permette di effettuare la ricerca, contattare il Customer Manager o il System Manager, registrarsi, effettuare il login e il logout, modificare la password, accedere e visualizzare i servizi relativi all’area personale (booklist, carrello, preferiti), seguire altri utenti e altre booklist (con la possibilità di smettere di seguire in un secondo momento), acquistare e recensire libri.

● *Gestione Libri:* Sottosistema che permette la gestione di tutte le funzionalità dedicate al Catalogue Manager, quali inserimento/modifica al prezzo/rimozione dei libri.

● *Gestione Customer:* Sottosistema che permette la gestione delle funzionalità dedicate al Customer Manager, quali visualizzazione di tutti gli utenti registrati la rimozione di commenti/utenti (registrati) dovute a segnalazioni (ticket) da parte di altri utenti.

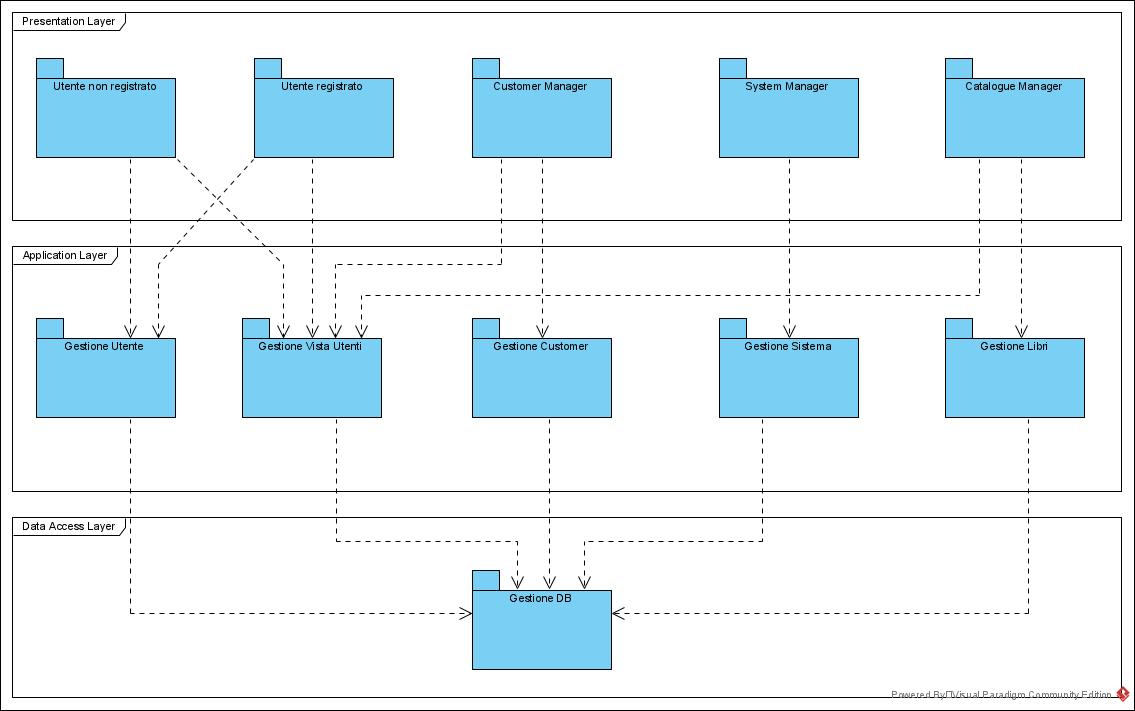
● *Gestione Sistema:* Sottosistema che permette la visualizzazione dei ticket riferiti a bug del sistema, accessibili al System Manager.

● *Gestione Vista Utenti:* Sottosistema che permette la visualizzazione del catalogo, dei vari libri, dei profili degli utenti e, nel caso di utente registrato, degli ordini pregressi e delle liste di follower/seguiti.

##### Data Access Layer (Model)

Livello che gestisce i dati necessari al funzionamento dell’intero sistema, ovvero i dati persistenti.

● Gestione DataBase (DB): Sottosistema che si occupa di immagazzinare e di prelevare i dati persistenti dal nostro DB.



* 1. Mapping hardware/software

Per il progetto software che si vuole sviluppare, verrà fatto uso di componenti “off-the-shelf”, componenti software disponibili sul mercato per facilitare la creazione del progetto.

Il framework che verrà adottato comprenderà Bootstrap, jQuery, JavaScript ed AJAX.

Bootstrap è un framework open-source che contiene una raccolta di strumenti per facilitare lo sviluppo di interfacce web;

jQuery è una libreria JavaScript che viene utilizzata per accelerare lo sviluppo della logica di business delle interfacce web;

JavaScript è un linguaggio di scripting orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web lato client.

AJAX è una tecnica di sviluppo software che si basa su uno scambio di dati in background fra web browser e server, consentendo così l’aggiornamento dinamico di una pagina web senza esplicito ricaricamento da parte dell’utente.

Sono state inoltre scelti questi tipi di configurazione:

per la parte di Presentation (View), saranno utilizzate le JSP, i linguaggi HTML, CSS e JavaScript;

per la parte di Application (Control), verrà utilizzato il server Apache Tomcat e il linguaggio Java;

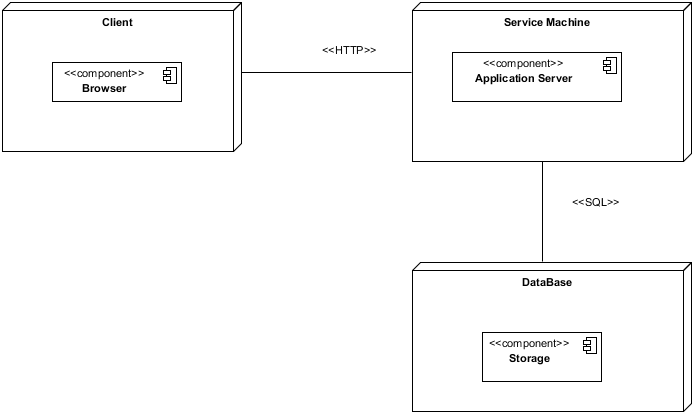
per la parte di Data Access (Model), verranno utilizzati MySQL e JDBC.

Il protocollo di comunicazione sarà HTTP, HTTPS (durante le sessioni sicure) e TCP/IP.

3.3.1 Deployment

Gli utenti potranno interagire con SocialBook tramite un web browser collegandosi

all’indirizzo del sito.

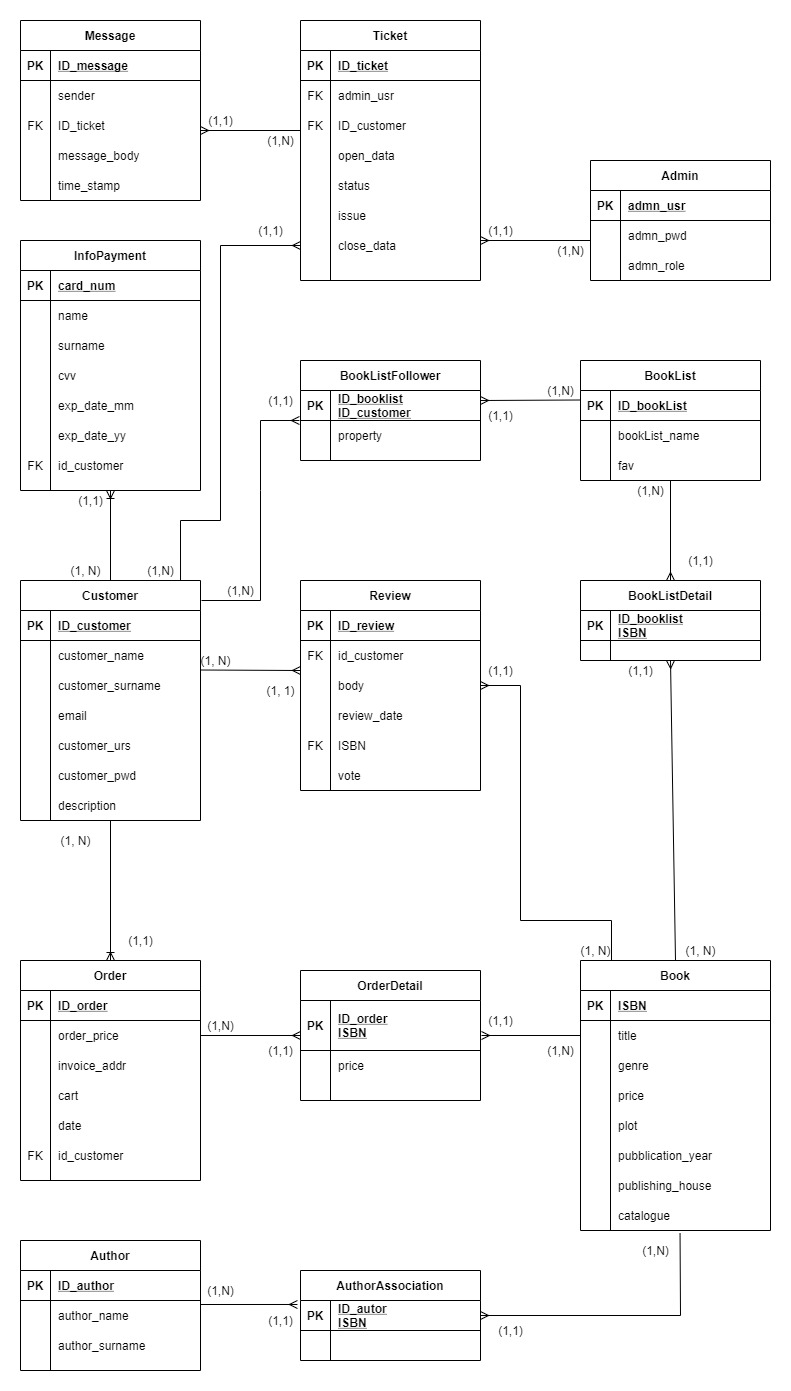


* 1. Gestione dati persistenti

Il sistema sarà installato su un solo computer e utilizzerà un DBMS MySQL installato sullo stesso.

Il sistema sarà diviso in client e server che inizialmente saranno lo stesso PC, ma in caso di futura necessità potranno facilmente essere divisi in quanto i servizi saranno progettati separatamente.

3.4.1 Schema logico



3.4.2 Struttura delle tabelle

##### Tabella customer

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_customer | Integer | PRIMARY KEY |
| customer\_name | String | NOT NULL |
| customer\_surname | String | NOT NULL |
| e-mail | String | NOT NULL |
| customer\_pwd | String | NOT NULL |
| customer\_usr | String | NOT NULL |
| description | String | NOT NULL |

##### Tabella book

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| ISBN | String | PRIMARY KEY |
| title | String | NOT NULL |
| genre | String | NOT NULL |
| price | Decimal | NOT NULL |
| pubblication\_year | Date | NOT NULL |
| publishing\_house | String | NOT\_NULL |
| plot | String | NOT NULL |
| catalogue\* | Bool | NOT NULL |

\*1 = presente nel catalogo, 0 = non presente nel catalogo

##### Tabella author

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| Id\_author | String | PRIMARY KEY |
| author\_surname | String | NOT NULL |
| author\_name | String | NOT NULL |

##### Tabella authorAssociation

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_author | Integer | PRIMARY KEY- FOREIGN KEY AUTHOR(id\_author) |
| ISBN | String | PRIMARY KEY- FOREIGN KEY BOOK(ISBN) |

##### Tabella customerOrder

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_order | Integer | PRIMARY KEY |
| order\_price | Decimal | NOT NULL |
| invoice\_addr | String | NOT NULL |
| cart\* | Bool | NOT NULL |
| date | Date |  |
| id\_customer | Integer | FOREIGN KEY CUSTOMER(id\_customer) |

\* 0 = carrello, 1 = ordine

##### Tabella orderDetail

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_order | Integer | PRIMARY KEY – FOREIGN KEY ORDER(id\_order) |
| ISBN | String | PRIMARY KEY – FOREIGN KEY BOOK(ISBN) |
| price | Decimal | NOT NULL |

##### Tabella infoPayment

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| card\_number | String | PRIMARY KEY |
| name | String | NOT NULL |
| surname | String | NOT NULL |
| exp\_date\_mm | String | NOT NULL |
| exp\_date\_yy | String | NOT NULL |
| cvv | String | NOT NULL |
| id\_customer | String | FOREIGN KEY CUSTOMER(id\_customer) |

##### Tabella booklist

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_booklist | Integer | PRIMARY KEY |
| booklist\_name | String | NOT NULL |
| favorite\* | Boolean | NOT NULL |

\* 0 = booklist, 1 = preferiti

##### Tabella booklistDetail

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_booklist | Integer | PRIMARY KEY – FOREIGN KEY BOOKLIST(id\_booklist) |
| ISBN | String | PRIMARY KEY – FOREIGN KEY BOOK(ISBN) |

##### Tabella booklistFollower

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_booklist | Integer | PRIMARY KEY – FOREIGN KEY BOOKLIST(id\_booklist) |
| id\_customer | Integer | PRIMARY KEY – FOREIGN KEY CUSTOMER(id\_customer) |
| property\* | Boolean | NOT NULL |

\* 0 = proprietario della booklist , 1 = follower della booklist

##### Tabella admin

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| admn\_usr | String | PRIMARY KEY |
| admn\_pwd | String | NOT NULL |
| admn\_role | String | NOT NULL |

##### Tabella review

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_review | Integer | PRIMARY KEY |
| id\_customer | Integer | NOT NULL- FOREIGN KEY CUSTOMER(id\_customer) |
| ISBN | String | NOT NULL – FOREIGN KEY BOOK(ISBN) |
| review\_date | Date | NOT NULL |
| body | String |  |
| vote | Integer |  |

##### Tabella ticket

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_ticket | Integer | PRIMARY KEY |
| id\_customer | Integer | NOT NULL- FOREIGN KEY CUSTOMER(id\_customer) |
| admn\_urs | String | NOT NULL – FOREIGN KEY ADMIN(admn\_urs) |
| open\_date | Date | NOT NULL |
| issue | String | NOT NULL |
| close\_date | Date | NOT NULL |
| status | String | NOT NULL |

##### Tabella message

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_message | Integer | PRIMARY KEY |
| sender\* | Integer | NOT NULL |
| id\_ticket | Integer | NOT NULL – FOREIGN KEY TICKET(id\_ticket) |
| time\_stamp | TimeStamp | NOT NULL |
| message\_body | String | NOT NULL |

\* 0= customer 1 = admin

##### Tabella customerPhoto

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| id\_customer | Integer | PRIMARY KEY |
| customer\_photo | MediumBlob |  |

##### Tabella bookPhoto

|  |  |  |
| --- | --- | --- |
| **Attributo** | **Tipo** | **Vincoli** |
| ISBN | String | PRIMARY KEY |
| book\_photo | MediumBlob |  |

* 1. Controllo degli accessi e sicurezza

Il sistema può essere utilizzato da qualsiasi utente con una connessione internet e la possibilità di utilizzare un web browser, per il compimento di un insieme ristretto di funzionalità che va ad allargarsi quando l'utente si registra alla piattaforma.

La piattaforma predispone dei controlli per l'iscrizione al sistema, dando ad ogni utente la possibilità di registrarsi utilizzando una coppia di credenziali (username e password, dove la password deve necessariamente seguire un formato specifico, cioè deve contenere una lettera maiuscola, una lettera minuscola, un numero e un simbolo speciale). L’accesso alla piattaforma per gli utenti è garantito dall’utilizzo di queste credenziali.

La sicurezza sui dati sensibili è garantita dall’accesso controllato, in quanto soltanto il customer manager può avere accesso ai dati relativi agli utenti.

Tutte le funzionalità del sistema sono descritte nella matrice degli accessi sottostante.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***Attori*** | *Gestione Utente* | *Gestione Vista Utenti* | *Gestione Customer* | *Gestione Sistema* | *Gestione Libri* |
| *Utente* | <<Registrazione>>  <<Ricerca>>  <<Contattare admin>> | <<Visualizzazione catalogo>>  <<Visualizzazione libro>>  <<Visualizzazione profili utenti>> |  |  |  |
| *Utente registrato* | <<Contattare admin>>  <<Ricerca>>  <<Autenticazione>>  <<Visualizzazione area personale e servizi>>  <<Recensione libro>>  <<Elimina recensione>>  <<Inserimento libro nel carrello>>  <<Rimozione libro dal carrello>>  <<Inserimento libro nei preferiti>>  <<Rimozione libro dai preferiti>>  <<Creazione booklist>>  <<Modifica booklist>>  <<Eliminazione booklist>>  <<Acquisto>>  <<Segui utente >>  <<Smetti di seguire utente>>  <<Segui booklist>>  <<Smetti di seguire booklist>>  <<Modifica password>>  <<Logout>> | <<Visualizzazione catalogo>>  <<Visualizzazione libro>>  <<Visualizzazione profili utenti>>  <<Visualizzazione ordini>>  <<Visualizzazione seguiti/seguaci>> |  |  |  |
| *Customer Manager* |  | <<Visualizzazione catalogo>>  <<Visualizzazione libro>>  <<Visualizzazione profili utenti>> | <<Visualizzazione utenti>>  <<Rimozione utente>>  <<Rimozione commento>> | <<Visualizzazione ticket>> |  |
| *Catalogue Manager* |  | <<Visualizzazione catalogo>>  <<Visualizzazione libro>>  <<Visualizzazione profili utenti>> |  |  | <<Inserimento libro>>  <<Rimozione libro>>  <<Modifica al prezzo libro>> |
| *System Manager* |  | <<Visualizzazione catalogo>>  <<Visualizzazione libro>>  <<Visualizzazione profili utenti>> |  | <<Visualizzazione ticket>> |  |

* 1. Controllo globale del software

Il controllo del flusso del software è regolato da classi Java che fungono da ricevitori di eventi e che rispondono alle attivazioni di client.

Le richieste vengono generate da un client e la classe preposta a gestire quel determinato evento associato alla richiesta, prendendo talvolta degli input, si preoccupa di inizializzare le richieste e di inoltrarle alle classi per lo svolgimento dell’operazione. Una volta ottenuto il risultato dell’opera zione, la classe gestore si preoccupa di inoltrarlo al client che aveva generato la richiesta.

Il sistema software è gestito con l’uso di Servlet e JSP.

Il server centrale attende le richieste di un client (web browser) e una volta ricevuta una richiesta, la processa e la smista alla Servlet deputata.

* 1. Condizioni limite

##### SERVER:

● Fase di Startup:

La fase di avvio del server Apache Tomcat dura circa trenta secondi, dopodiché viene avviato il server dove è contenuto il DBMS MySQL. La connessione al DBMS verrò effettuata tramite il driver JDBC.

Successivamente viene avviata la JVM sul server e vengono letti i dati di configurazione del database.

Infine, vengono caricati tutti i dati in memoria volatile e il sistema è pronto all’uso.

● Fase di Terminazione:

Il server può essere spento in caso di guasto o in caso di necessità di manutenzione.

Terminata l’attività di manutenzione o di riparazione del guasto il processo server viene avviato nuovamente.

● Fallimento:

Il server può fallire in caso di interruzione inaspettata dell’alimentazione e/o di guasti hardware.

Il server può andare in crash quando viene sollevata un’eccezione che non è stata gestita.

##### CLIENT:

● Scenario “Startup”:

Andrea è un tecnico che intende avviare la piattaforma SocialBook.

Una volta avviato il web server Apache Tomcat, configura il DBMS MySQL per essere raggiunto dal medesimo web server.

A questo punto il web server carica su Tomcat l’eseguibile della piattaforma SocialBook.

Dopo circa trenta secondi dall’avvio, la piattaforma è operativa e raggiungibile dagli utenti.

● Fase di Terminazione:

All’utente basterà chiudere la pagina del sistema per interrompere la comunicazione, interrompendo così la sessione.

● Scenario “Fallimento”:

Il server mostra al client una schermata di errore. Per evitare la perdita di dati, viene comunicato al DBMS di effettuare una forzata scrittura su memoria fisica delle informazioni ancora presenti in memoria volatile. Il DBMS mantiene dei registri di log dove sono annotate tutte le operazioni sul DB.

1. Servizi dei sottosistemi

|  |  |
| --- | --- |
| **Sottosistema** | **Gestione Utente** |
| *Descrizione* | Permette all’utente, sia registrato che non, di effettuare varie operazioni. |
| **Servizi offerti** | |
| *Servizio* | *Descrizione* |
| Registrazione | Consente la registrazione di un qualsiasi utente che non sia già registrato. |
| Ricerca | Consente la ricerca di un libro per titolo/genere/autore. |
| Autenticazione | Consente l’autenticazione di un qualsiasi utente già registrato. |
| Logout | Consente il logout di un qualsiasi utente che abbia effettuato l’accesso. |
| Visualizzazione area personale e servizi | Consente la visualizzazione dell’area personale di un qualsiasi utente registrato (carrello, preferiti, booklist) |
| Modifica password | Consente a un utente registrato di modificare la propria password. |
| Recensione libro | Consente a un utente registrato di aggiungere una recensione a un qualsiasi libro. |
| Elimina Recensione | Consente a un utente registrato di eliminare una propria recensione inserita precedentemente. |
| Inserimento libro nel carrello | Consente a un utente registrato di aggiungere qualsiasi libro al carrello. |
| Rimozione libro dal carrello | Consente a un utente registrato di rimuovere un qualsiasi libro aggiunto precedentemente al carrello. |
| Contattare admin | Consente di inviare un ticket a un admin (System Manager o Customer Manager) per segnalare un bug oppure un utente/commento. |
| Inserimento libro nei preferiti | Consente a un utente registrato di inserire un qualsiasi libro ai preferiti. |
| Rimozione libro dai preferiti | Consente a un utente registrato di rimuovere un qualsiasi libro aggiunto precedentemente ai preferiti. |
| Creazione booklist | Consente a un utente registrato di creare una booklist. |
| Modifica booklist | Consente a un utente registrato di modificare una qualsiasi delle proprie booklist. |
| Elimina booklist | Consente a un utente registrato di eliminare una qualsiasi delle proprie booklist. |
| Acquisto | Consente a un utente registrato di acquistare i libri che ha aggiunto al carrello. |
| Segui utente | Consente a un utente registrato di seguire un qualsiasi altro utente registrato (e quindi tutte le sue booklist). |
| Smetti di seguire utente | Consente a un utente registrato di smettere di seguire un utente che aveva seguito precedentemente. |
| Segui booklist | Consente a un utente registrato di seguire una qualsiasi booklist. |
| Smetti di seguire booklist | Consente a un utente registrato di smettere di seguire una booklist che aveva seguito precedentemente. |
| **Sottosistema** | **Gestione Vista Utenti** |
| *Descrizione* | Permette all’utente la visualizzazione del catalogo, della pagina di un qualsiasi libro, dei profili degli utenti e in caso l’utente è registrato, anche degli ordini pregressi e della lista dei seguiti/seguaci. |
| **Servizi offerti** | |
| *Servizio* | *Descrizione* |
| Visualizzazione catalogo | Consente a un utente di visualizzare il catalogo dei libri. |
| Visualizzazione libro | Consente a un utente di visualizzare la pagina di un libro qualsiasi.. |
| Visualizzazione profili utenti | Consente a un utente di visualizzare il profilo di un qualsiasi utente registrato. |
| Visualizzazione ordini | Consente a un utente registrato di visualizzare tutti i suoi ordini pregressi. |
| Visualizzazione seguiti/seguaci | Consente a un utente registrato di visualizzare la lista dei propri seguiti/seguaci. |
| **Sottosistema** | **Gestione Customer** |
| *Descrizione* | Permette al Customer Manager di visualizzare tutti gli utenti registrati e i vari ticket e di rimuovere un utente/commento segnalato. |
| **Servizi offerti** | |
| *Servizio* | *Descrizione* |
| Visualizzazione utenti | Consente al Customer Manager di visualizzare la lista degli utenti registrati e i loro dati. |
| Rimozione utente | Consente al Customer Manager di rimuovere un utente registrato che ha ricevuto tante segnalazioni. |
| Rimozione commento | Consente al Customer Manager di rimuovere un commento che ha ricevuto tante segnalazioni. |
| **Sottosistema** | **Gestione Sistema** |
| *Descrizione* | Permette al System Manager di visualizzare i ticket mandati dagli utenti che riguardano segnalazioni di bug. |
| **Servizi offerti** | |
| *Servizio* | *Descrizione* |
| Visualizzazione ticket | Consente al System Manager di visualizzare i ticket e risolverne i problemi. |
| **Sottosistema** | **Gestione Libri** |
| *Descrizione* | Permette al Catalogue Manager di gestire il catalogo dei libri. |
| **Servizi offerti** | |
| *Servizio* | *Descrizione* |
| Inserimento libro | Consente al Catalogue Manager di inserire un nuovo libro nel catalogo. |
| Modifica al prezzo libro | Consente al Cataogue Manager di modificare il prezzo di un libro presente nel catalogo. |
| Rimozione libro | Consente al Catalogue Manager di rimuovere un libro dal catalogo. |

1. Glossario

**SDD (System Design Document):** Documento formalizzato alla definizione di obiettivi di progettazione del sistema, decomposizione del sistema in sottosistemi più piccoli e scelta di architettura software più adatta al sistema.

**RAD (Requirement Analysis Document):** documento contenente informazioni inerenti al sistema da realizzare raccolte durante la fase di Requirement Analysis e Requirement Elicitation.

**Server:** componente che gestisce traffico di informazioni e fornisce servizi e risorse attraverso la rete.

**Client:** componente che accede a servizi e risorse del server.

**Web Browser:** applicazione software installata sul client che permette di visualizzare e navigare le risorse del web.

**DBMS:** sistema software per la creazione, manipolazione e interrogazione efficiente di database.

**HTTP:** protocollo di trasferimento di ipertesti che consente a due macchine, client e server, di interagire attraverso un meccanismo di richiesta/risposta. Il client inoltra una richiesta al server, che verrà soddisfatta con la risposta di quest’ultimo.

**HTTPS:** protocollo per la comunicazione sicura attraverso una rete di computer, utilizzato su Internet all’interno di una connessione criptata.

**Servlet:** oggetti Java all'interno del server web che permettono di creare web applications in combinazione con JSP.

**JSP:** tecnologia di programmazione web utilizzata per fornire contenuti dinamici.

**Piattaforma:** Definisce l’insieme delle funzionalità fornite dal sistema attraverso l’applicazione web.

**Utente:** Un utilizzatore della piattaforma che non si è ancora registrato.

**Utente registrato:** Un utilizzatore iscritto alla piattaforma.

**Customer manager:** Amministratore che si occupa della gestione degli utenti.

**System manager:** Amministratore che si occupa della gestione di bug, segnalati da utenti che utilizzano la piattaforma.

**Catalogue manager:** Amministratore che si occupa della gestione del catalogo libri.

**Admin:** Generalizzazione di customer manager, catalogue manager e system manager.