

# System Design Document

## Dante's Library



<i>Cognome</i>	<i>Nome</i>	<i>Matricola</i>
Buongusto	Andrea	0512105272
Salierno	Marco	0512105332

# Indice

<b>1. Introduzione</b>	<b>3</b>
1.1 Obiettivo del sistema	3
1.2 Obiettivi di progettazione	3
1.3 Acronimi e abbreviazioni	4
1.4 Riferimenti	4
1.5 Panoramica	4
<b>2. Architettura del software attuale</b>	<b>5</b>
<b>3. Architettura del software proposto</b>	<b>6</b>
3.1 Panoramica	6
3.2 Scomposizione in sottosistemi	7
3.3 Mappatura hardware/software	8
3.4 Gestione dei dati persistenti	9
3.5 Controllo e sicurezza degli accessi	10
3.6 Flusso di controllo globale	11
3.7 Condizioni limite	11
<b>4. Servizi forniti dai sottosistemi</b>	<b>13</b>
<b>Glossario</b>	<b>14</b>

# 1. Introduzione

## 1.1 Obiettivo del sistema

Dante's Library si prefigge lo scopo di creare un sistema web per biblioteche. Il cliente ha la possibilità di visionare i libri disponibili presso la biblioteca che utilizza il software, prenotarli e richiedere una tessera (o registrarne una già in suo possesso).

Ai bibliotecari invece, la piattaforma consente di avere un'area privata per i vari dipendenti che permette di gestire tutti gli aspetti necessari di una biblioteca. In questo modo, le biblioteche che adotteranno il software, potranno avere un sistema automatizzato e personalizzabile.

## 1.2 Obiettivi di progettazione

- **Criteri di prestazioni**  
I tempi di risposta del sistema non dovranno superare i 5 secondi, tranne in situazioni di carico superiore alla media.
- **Criteri di affidabilità**  
Il sistema deve assicurarsi che, nel caso di prestiti librari effettuati su uno stesso libro, questi avvengano in maniera atomica in modo da non generare anomalie di "Disponibilità".
- **Criteri di costi**  
Non ci devono essere costi aggiuntivi tranne quelli per il mantenimento delle apparecchiature hardware.
- **Criteri di sicurezza**  
Il sistema deve impedire ad utenti non registrati di accedere a funzionalità riservate esclusivamente agli utenti registrati.  
Inoltre, qualsiasi informazione sensibile non dovrà essere memorizzata in chiaro.
- **Criteri di sostenibilità**  
Il Software deve essere progettato in maniera tale da poter funzionare su tutte le piattaforme e browser.

## 1.3 Acronimi e abbreviazioni

DBMS: Database Management System

RAD: Requirements Analysis Document

UI: User Interface

## 1.4 Riferimenti

- [Problem Statement](#)
- [Use Case And Requirements](#)
- [RAD \(Requirements Analysis Document\)](#)

## 1.5 Panoramica

Il seguente documento "System Design" è diviso in sezioni ed ha la seguente composizione:

- **Introduzione**  
In questa sezione vengono descritti i dati introduttivi del progetto tra cui gli obiettivi di progettazione e del sistema. Successivamente vengono descritte le definizioni, gli acronimi e le abbreviazioni usati per facilitare la lettura del documento, infine il paragrafo "Riferimenti" contiene i collegamenti ai documenti precedenti.
- **Sistema Attuale**  
Questa sezione è dedicata a definire di un eventuale sistema attuale prima dello sviluppo.
- **Sistema Corrente**  
Questa sezione contiene le informazioni riguardanti il sistema corrente tra cui: la scomposizione in sottosistemi, la mappatura hardware/software, la gestione dei dati persistenti, controllo e sicurezza degli accessi, il flusso di controllo globale, le condizioni di limite e i servizi forniti dai sottosistemi.
- **Glossario**  
Sezione che contiene una raccolta di vocaboli appartenenti al contesto del documento.

## 2. Architettura del software attuale

Il progetto Dante's Library è da realizzare per intero. Non è presente alcun sistema attualmente da rimpiazzare.

## 3. Architettura del software proposto

### 3.1 Panoramica

Questa sezione ha la seguente composizione:

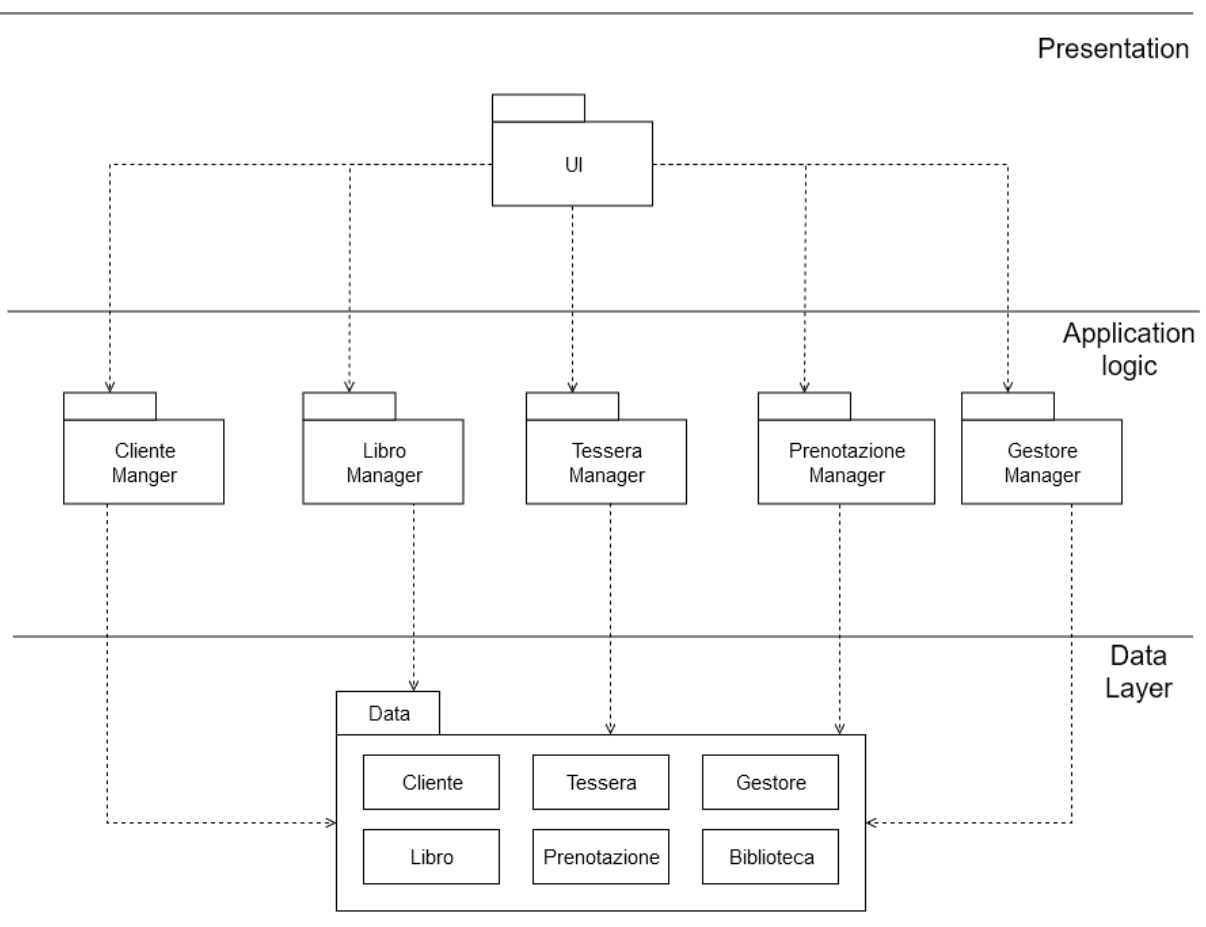
- **Scomposizione in sottosistemi**  
In questa sezione viene illustrato come il sistema viene suddiviso in sottosistemi in base alle funzionalità che esso offre.
- **Mappatura hardware/software**  
In questa sezione vengono descritti i software e la struttura hardware utilizzati per la realizzazione del sistema e come essi interagiscono tra loro per garantirne il corretto funzionamento.
- **Gestione dei dati persistenti**  
In questa sezione vengono identificati gli oggetti persistenti e scelto il tipo di infrastruttura usata per memorizzarli (DBMS, file o altro).
- **Controllo e sicurezza degli accessi**  
In questa sezione vengono definite nel dettaglio le operazioni e le informazioni effettuabili da ogni singolo attore e come questi si autenticano al sistema.
- **Flusso di controllo globale**  
In questa sezione viene definito il flusso di controllo globale, cioè una sequenza di azioni di un sistema che possono includere anche il prendere decisioni su quali operazioni eseguire ed in che ordine. Queste decisioni sono basate su eventi esterni generati da attori o causati dal trascorrere del tempo.
- **Condizioni limite**  
Le condizioni limite del sistema includono lo startup, lo shutdown, l'inizializzazione e la gestione di fallimenti come corruzione di dati, caduta di connessione e altri malfunzionamenti vari. A tale scopo vengono indicati dei casi d'uso che specificano quando un oggetto viene creato e distrutto.

## 3.2 Scomposizione in sottosistemi

Per realizzare il sistema Dante's Library viene usata l'architettura Client/Server su tre livelli.

Di seguito vengono illustrati i tre livelli:

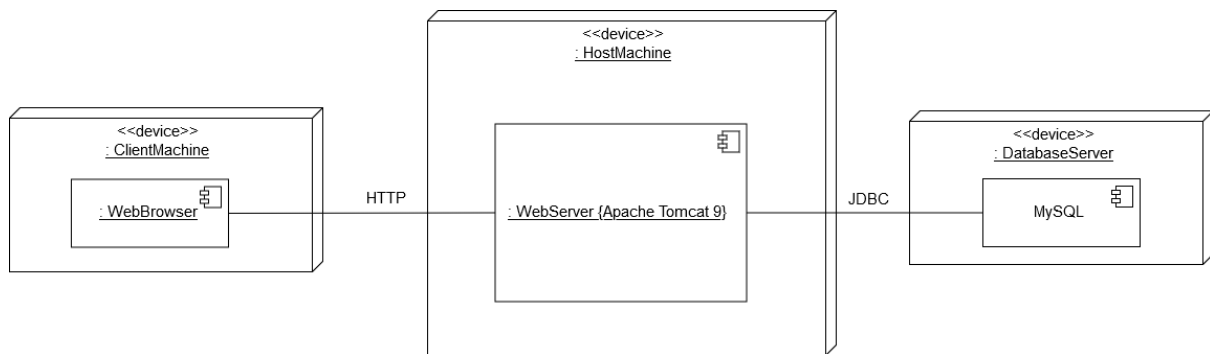
- **Livello Presentation**  
Comprende tutti gli aspetti riguardanti l'interfaccia grafica (UI) del sistema e i meccanismi che permettono all'utente di interagire con essa (oggetti boundary).
- **Livello Application Logic**  
Contiene tutti gli oggetti che si occupano della logica applicativa del sistema nonché dell'interfacciamento con le informazioni persistenti.
- **Data Layer**  
Comprende le entità rilevate in fase di analisi per le quali è necessario memorizzare in maniera persistente i dati.



### 3.3 Mappatura hardware/software

Per il funzionamento di Dante's Library occorrono Web Server e un Database Server. Per quanto riguarda il Web Server la scelta ricade su Apache Tomcat 9, mentre per il Database Server, si è scelto MySQL. La principale motivazione di queste scelte è la compatibilità con molteplici piattaforme (Windows, Mac OS, distribuzioni Linux, ...). Questi due server possono essere eseguiti su uno stesso host o anche su host diversi.

Infine, per poter usufruire dei servizi offerti dal sistema, i dispositivi client si collegheranno al sistema mediante l'uso di un Web Browser qualsiasi.



*Deployment Diagram rappresentante la mappatura*

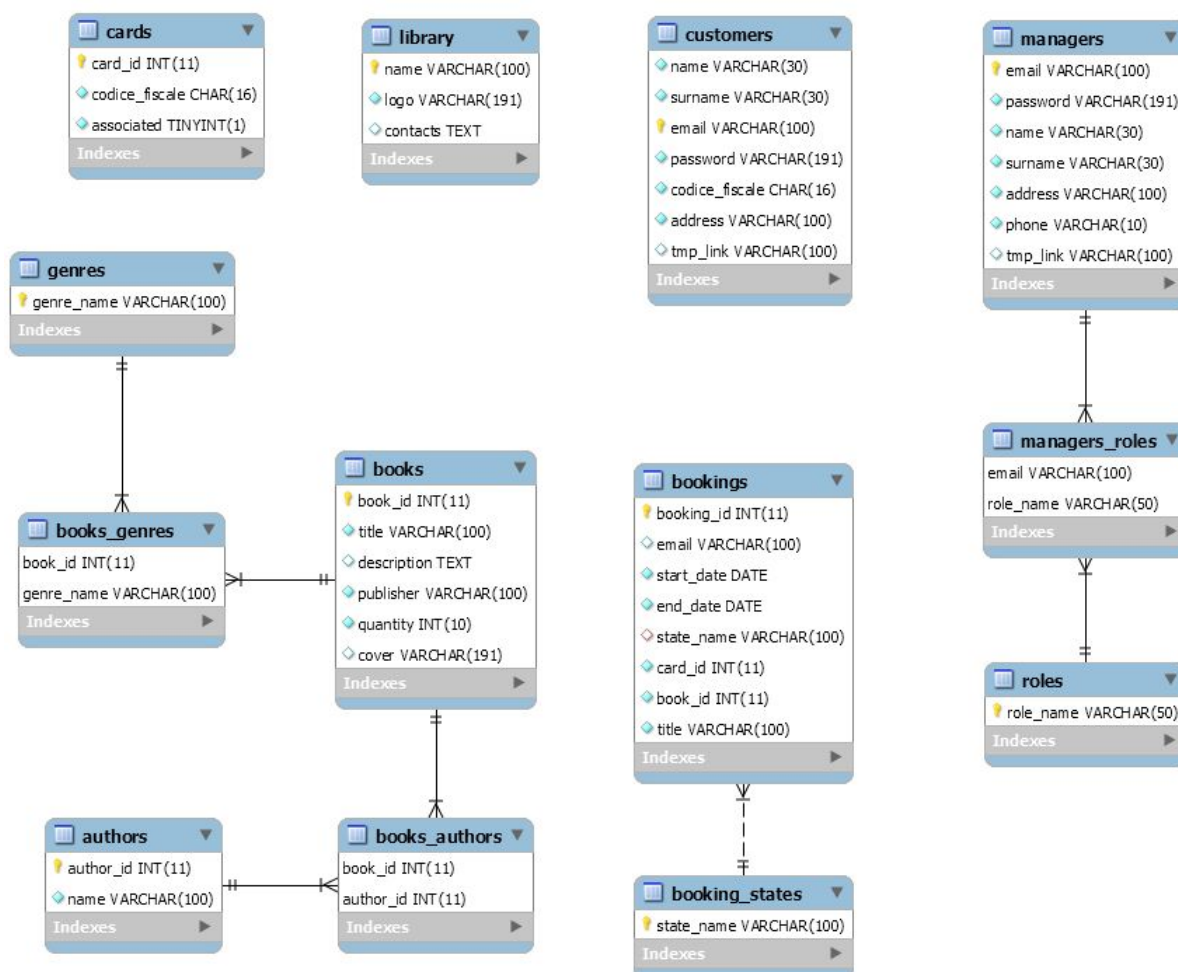


### 3.4 Gestione dei dati persistenti

Dante's Library utilizza un sistema di gestione database di tipo relazionale per la memorizzazione persistente di alcune informazioni quali: i clienti, i libri, le tessere, le prenotazioni effettuate, i gestori, e alcune informazione riguardo la biblioteca stessa. Il DBMS scelto è MySQL. All'interno di esso verranno quindi create tabelle per la memorizzazione di:

- Clienti
- Libri
- Tessere
- Prenotazioni
- Gestori
- Informazioni sulla biblioteca

Il sistema sarà in grado di interfacciarsi al DBMS MySQL grazie all'utilizzo del driver JDBC. (come mostrato nel [punto 3.3](#)).



Schema logico rappresentante la persistenza del sistema.

### 3.5 Controllo e sicurezza degli accessi

Oggetti	Cliente	Tessera	Libro	Prenotazione	Gestore	Biblioteca
Attori						
Cliente	Lettura Modifica		Lettura	Aggiungi Lettura		
Gestore Libri			Aggiungi Lettura Modifica Cancella			
Gestore Prenotazioni				Aggiungi Lettura Modifica Cancella		
Gestore Utenti	Lettura Cancella					
Gestore Tessere		Lettura Aggiungi Cancella				
Gestore Biblioteca	Lettura Cancella	Lettura Aggiungi Cancella	Aggiungi Lettura Modifica Cancella	Aggiungi Lettura Modifica Cancella	Aggiungi Lettura Modifica Cancella	Lettura Modifica

Il software è dotato di un sistema di autenticazione valido per tutti i *Clienti* che verifica la corrispondenza di email e password inseriti in fase di accesso.

L'autenticazione è necessaria per poter effettuare prenotazioni di libri.

Senza di essa, è possibile soltanto svolgere operazioni come:

- Ricerca e Catalogo
- Controllare la disponibilità dei libri
- Visualizzare la pagina dedicata ai contatti della biblioteca

I *Gestori* possono accedere alla propria schermata di login aggiungendo il path `"/admin"` all'URL base del sito (Es: `localhost/admin`).

Le informazioni sensibili quali le password, saranno crittografate prima di essere memorizzate nel database per garantire maggiore sicurezza.

## 3.6 Flusso di controllo globale

Considerata la natura web-based dell'applicazione Dante's Library, il flusso di controllo viene affidato interamente al container Tomcat (il quale è anche in grado di gestire le problematiche attinenti alla concorrenza). Esso sarà sempre in esecuzione in attesa di richieste da parte degli utenti. Le richieste possono essere effettuate da un Web Browser qualsiasi. Ad ogni richiesta ricevuta il container sarà in grado di gestirla in maniera concorrente creando un nuovo thread apposito.

Gli oggetti Control individuati in fase di analisi rimarranno (fare riferimento al [RAD](#)) in attesa di eventi innescati dagli utilizzatori per ricevere dati (event-driven control), generare contenuti in base ai dati forniti e mostrare infine i risultati.

## 3.7 Condizioni limite

Per la messa in servizio del sistema, occorre che ci sia un Amministratore. Egli si occupa di svolgere tutte le procedure necessarie per assicurare il corretto funzionamento del sistema.

In particolare, dovrà occuparsi dei due componenti principali: il **Web Server** e il **Database Server**.

### Web Server

- *Inizializzazione*: l'Amministratore avvia il Web Server Tomcat mettendolo in ascolto su una porta designata e fa il deploy dell'applicazione (sito web).
- *Arresto*: l'Amministratore arresta il Web Server Tomcat, rendendo inaccessibile l'applicazione da parte di qualsiasi utente.
- *Gestione Eccezioni*: In caso di un errore software (bug), fallimento hardware o qualsiasi altro evento imprevedibile (es. sbalzi di tensione), il Web Server risponderà all'utente con un codice di stato HTTP a seconda del tipo di errore (es. 404, 500).

### Database Server

- *Inizializzazione*: l'Amministratore avvia il server del DBMS MySQL mettendolo in ascolto su una porta designata.
- *Arresto*: l'Amministratore arresta il server MySQL.
- *Gestione Eccezioni*: In caso di un errore software (bug), fallimento hardware o qualsiasi altro evento imprevedibile (es. sbalzi di tensione), l'utente può ancora visitare il sito web, tuttavia la maggior parte delle funzionalità non sarà accessibile (es. autenticazione, prenotazione, ricerca, ...). L'utente sarà notificato del malfunzionamento con un messaggio di errore.

### Oggetti Persistenti

Di seguito vengono riportati gli oggetti persistenti individuati in fase di analisi. Per ognuno di questi vengono specificati i casi d'uso in cui vengono creati e distrutti.

- **Clienti**  
Creato: UC5: Registrazione  
Distrutto: UC13: CancellazioneCliente
- **Libri**  
Creato: UC14: AggiuntaLibro  
Distrutto: UC16: CancellazioneLibro
- **Tessere**  
Creato: UC5.1:RichiestaTessera, UC20: AggiuntaTessera  
Distrutto: UC21: CancellazioneTessera
- **Prenotazioni**  
Creato: UC8: PrenotazioneLibro, UC23: AggiuntaPrenotazione  
Distrutto: UC10: Annulla prenotazione, UC25: CancellazionePrenotazione
- **Gestori**  
Creato: UC27: AggiuntaGestore  
Distrutto: UC29: CancellazioneGestore

## 4. Servizi forniti dai sottosistemi

- **ClienteManager**  
Registrazione, Login, Password dimenticata, Modifica dati Cliente, Ricerca Cliente.
- **LibroManager**  
Ricerca Libro, Aggiunta Libro, Modifica Libro, Cancellazione Libro, Aggiunta Genere, Cancellazione Genere, Lista Generi.
- **TesseraManager**  
Aggiunta Tessera, Richiesta Tessera, Registrazione Tessera, Cancellazione Tessera, Ricerca Tessera.
- **PrenotazioneManager**  
Annulla Prenotazione, Ricerca Prenotazione, Aggiungi Prenotazione, Modifica Stato Prenotazione, Cancellazione Prenotazione.
- **GestoreManager**  
Login, Aggiungi Gestore, Modifica Gestore, Cancellazione Gestore, Modifica Dati Biblioteca, Ricerca Gestore.

# Glossario

Container: Un Web container è il componente di un server Web che interagisce con le servlet Java. Un Web container è responsabile della gestione del ciclo di vita delle servlet, del mapping di un URL a una determinata servlet e della garanzia che il richiedente URL disponga dei diritti di accesso corretti.

Device, host: dispositivo fisico.

Deploy: è l'installazione e la messa in servizio di un'applicazione software. Dopo l'operazione di deployment, l'applicazione è pronta a fornire i suoi servizi.

MySQL: è un Relational database management system composto da un client a riga di comando e un server.