Dante’s Library

Un progetto per il corso di
Ingegneria del Software

Partecipanti

Andrea Buongusto

0512105272

Marco Salierno

0512105332

Dominio del problema

Il numero di utenti che ha la possibilità di collegarsi ad Internet aumenta anno dopo anno. I vecchi sistemi bibliotecari sono obsoleti e richiedono la presenza fisica dell'utente anche per la sola verifica della disponibilità di un libro. L'utente piuttosto che recarsi in biblioteca (e registrarsi per la tessera) senza avere la certezza dell'effettiva presenza del libro, preferisce acquistarlo online piuttosto che risparmiare prendendolo in prestito dalla biblioteca.

RAD

Requirements Analysis Document

Il nostro scopo

Creare una piattaforma online per biblioteche

Ogni utente avrà la possibilità di visionare i libri disponibili presso la biblioteca, prenotarli e gestire la propria tessera.

Facilitare la gestione di una biblioteca

I dipendenti avranno a disposizione un pannello amministratori per gestire con facilità tutti gli aspetti più importanti della biblioteca.

Interfaccia adattabile a qualunque schermo

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer non est in enim placerat varius.

This is a Subtitle

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer non est in enim placerat varius.

Alcuni Requisiti funzionali...

1. La piattaforma è fornita di un sistema di autenticazione completo e con possibilità di recupero password.
2. La piattaforma consente la prenotazione di libri solo se in possesso di una tessera bibliotecaria.
3. La piattaforma dispone di un pannello gestori contenenti sezioni per: Clienti, Libri, Tessere, Prenotazioni.
4. È possibile assegnare uno o più ruoli ad ogni gestore. In base ai ruoli, sarà possibile o meno svolgere delle operazioni con il pannello.

...e non funzionali

Usabilità

L'interfaccia utente si adatta in base al dispositivo utilizzato.

Affidabilità

È possibile fare prenotazioni solo previa autenticazione.

Prestazioni

Tempo di risposta massimo 5 secondi.

Sostenibilità

Funzionante su qualsiasi sistema lo si voglia installare.

Aggiunta Libro

Da scenario a caso d'uso

Attori: *Paolo (Gestore Libri)*
Paolo, gestore libri della biblioteca comunale di Salerno, deve aggiungere un nuovo libro sul sito online della biblioteca.
Nella sezione libri situata nel pannello di amministrazione, Paolo clicca sul tasto "Aggiungi Libro" e compila il form che appare con i seguenti campi:

- Copertina libro: *new-book.jpg*
- Titolo: *L'officina del meccanico quantistico. Dal gatto di Schröedinger al quantum computing*
- Descrizione: *La fisica quantistica sembra [...]*
- Autori: *Fabio Chiarello*
- Casa Editrice: *Maggioli Editore*
- Genere: *Didattica*
- Quantità: 15

Dopodichè conferma l'inserimento cliccando sul tasto "Aggiungi libro" situato in fondo al form.

Attori	*Gestore Libri*	
Condizioni di entrata	UC1: Autenticazione (*Gestore*)	
Flusso degli eventi	**Gestore Libri**	**Dante's Library**
	1. Il *Gestore Libri* accede alla sezione "Libri" del pannello di amministrazione e seleziona la funzione "Aggiungi libro".	
		2. *Dante's Library* risponde visualizzando un form con i campi: titolo, autori, casa editrice, genere, descrizione numero copie e foto copertina.
	3. Il *Gestore Libri* compila e sottomette il form al sistema.	
		4. *Dante's Library* inserisce nel sistema il nuovo libro.
Condizioni di uscita	Il *Gestore Libri* ha inserito un nuovo libro.	
Eccezioni	4a. E3: Campi non compilati correttamente	

Le entità coinvolte nel sistema

Diagramma delle classi

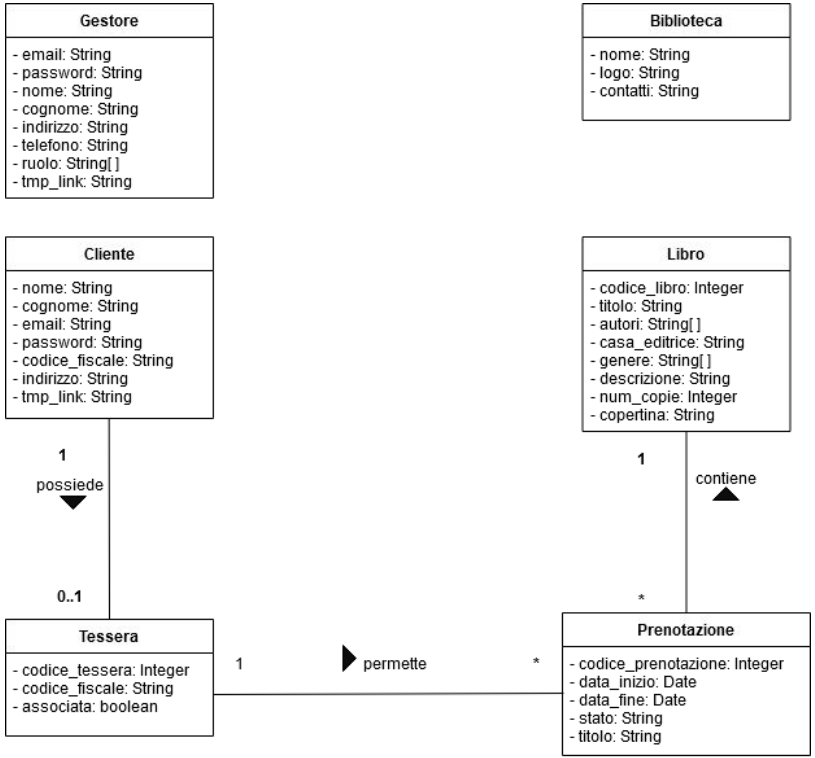


Diagramma di stato

Lo stato della prenotazione di un libro

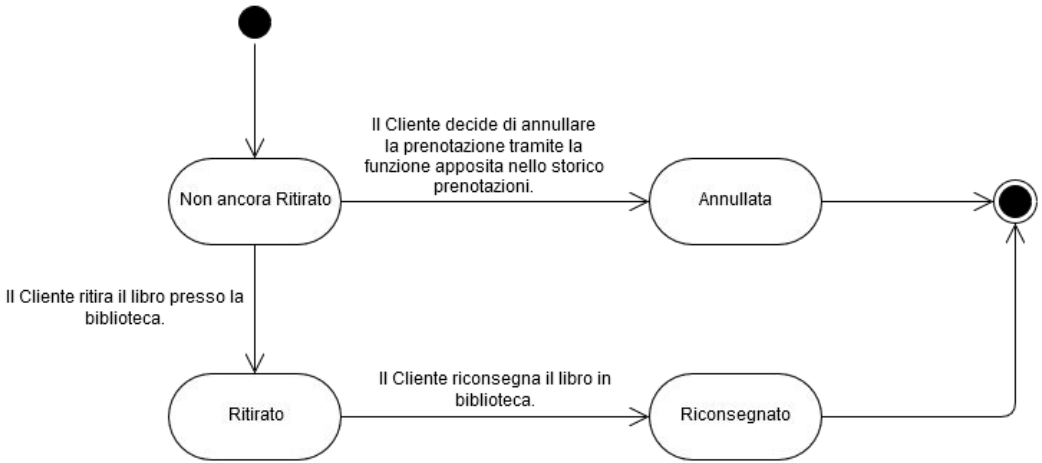


Diagramma di sequenza

L'aggiunta di un libro

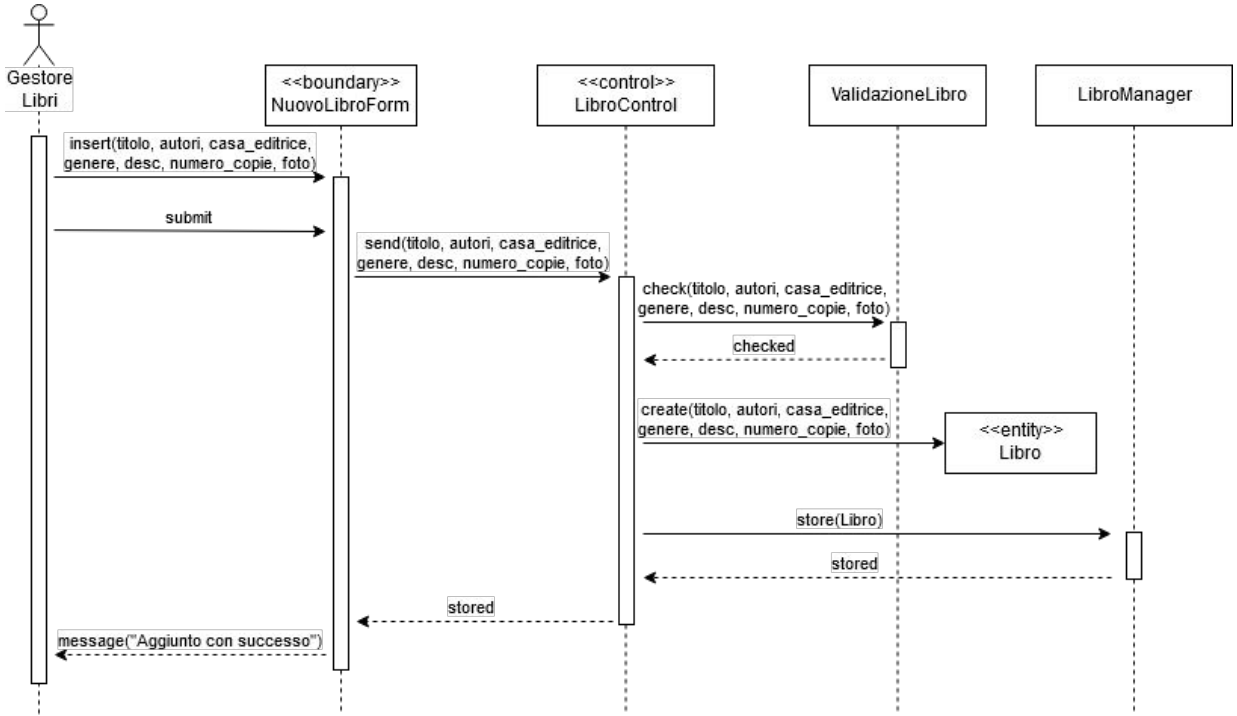
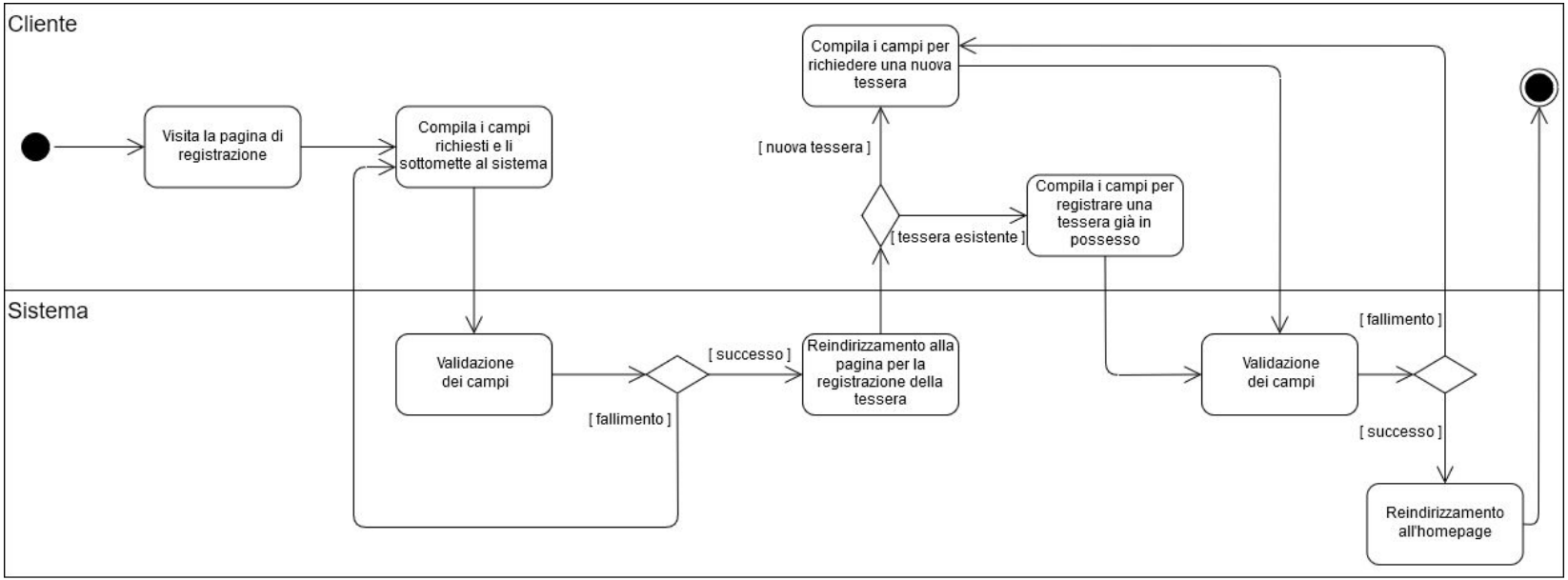


Diagramma di attività

Registrazione di un nuovo cliente

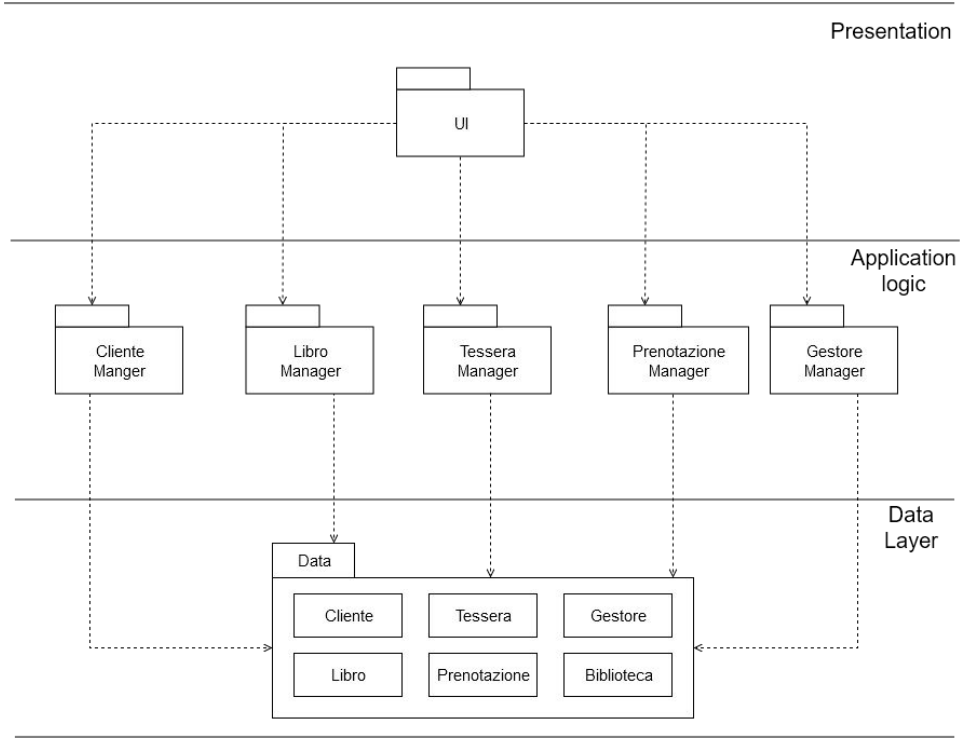


SDD

System Design Document

Scomposizione in sottosistemi

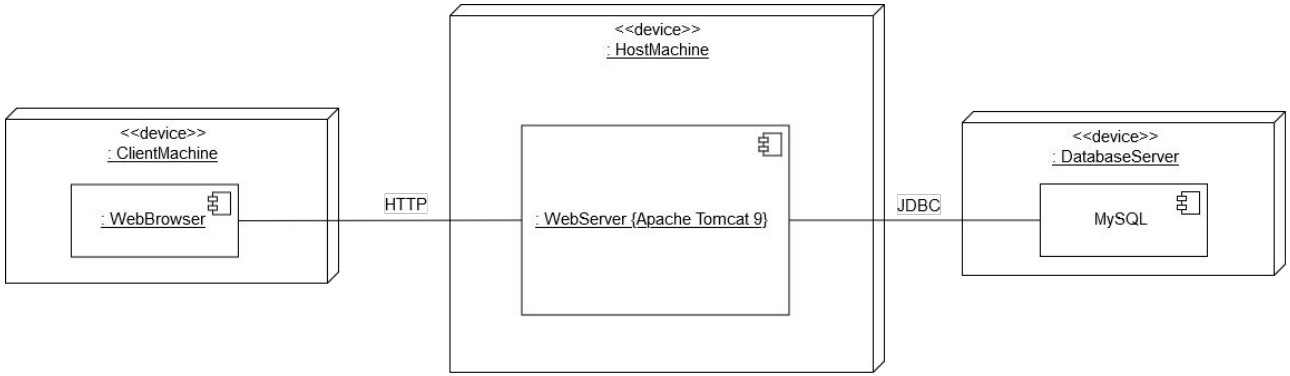
Architettura three-tier



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.

Mappatura Hardware/Software



- Per il funzionamento di Dante's Library occorre un 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.

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. Il sistema sarà in grado di interfacciarsi al DBMS MySQL grazie all'utilizzo del driver JDBC.

MySQL®

Controllo e sicurezza degli accessi

Oggetti / Attori	Cliente	Tessera	Libro	Prenotazione	Gestore	Biblioteca
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.

Servizi forniti dai sottosistemi

Cliente Manager

Registrazione, Login, Password dimenticata, Modifica dati Cliente, Ricerca Cliente.

Libro Manager

Ricerca Libro, Aggiunta Libro, Modifica Libro, Cancellazione Libro, Aggiunta Genere, Cancellazione Genere, Lista Generi.

Tessera Manager

Aggiunta Tessera, Richiesta Tessera, Registrazione Tessera, Cancellazione Tessera, Ricerca Tessera.

Prenotazione Manager

Annulla Prenotazione, Ricerca Prenotazione, Aggiungi Prenotazione, Modifica Stato Prenotazione, Cancellazione Prenotazione.

Gestore Manager

Login, Aggiungi Gestore, Modifica Gestore, Cancellazione Gestore, Modifica Dati Biblioteca, Ricerca Gestore.

ODD

Object Design Document

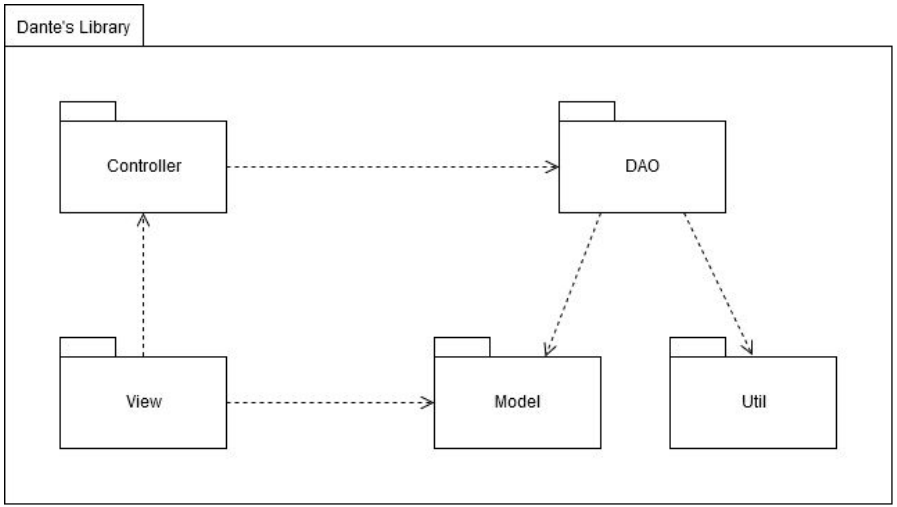
Compromessi di progettazione

Comprensibilità vs Tempo
Il codice dev'essere comprensibile e rispettare le convenzioni concordate, commentando le porzioni di codice che risultano essere meno chiare. In questo modo i futuri sviluppatori esterni al progetto potranno comunque comprendere quanto implementato finora.

Usabilità vs Funzionalità
Il sistema è progettato per essere semplice ed user-friendly, rinunciando all'aggiunta di funzionalità più complesse e meno necessarie rispetto a quelle già presenti.

Robustezza vs Costi
Essendo il sistema una piattaforma online, e che potrebbe potenzialmente essere utilizzata da una larga utenza, si preferisce una maggiore affidabilità rispetto ad un sistema economico e meno sicuro.

Panoramica del sistema



Il sistema comprende i package “Model”, “Controller”, “View”, “DAO” e “Util”.

- Model: comprende le classi Bean, ossia semplici classi Java utilizzate per contenere delle informazioni.
- Controller: comprende Servlet Java per gestire richieste, elaborare informazioni e fornire risposte.
- View: comprende le pagine JSP che realizzano il livello di presentazione, ossia ciò con cui l’utente può interagire.
- DAO: contiene le classi Java responsabili dell’interfacciamento con il Data Layer.
- Util: contiene una classe che specifica le informazioni per la connessione al Database, una classe contenente i parametri di configurazione per l’invio di email, una classe con dei metodi per il controllo del formato degli input e una classe che offre metodi per la cifratura di stringhe (algoritmo di hashing BCrypt).

Interfacce delle classi

Un esempio

Nome classe	BookingServlet.java
Descrizione	Classe che riceve richieste GET e POST riguardanti le Prenotazioni e che produce output da inviare come risposta.
Attributi e Metodi	# doPost(request : HttpServletRequest, response : HttpServletResponse) : void # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
Precondizioni	**context** BookingServlet::doPost(request : HttpServletRequest, response : HttpServletResponse) **Pre** : CustomersBean customer = session.getAttribute("customer") <> null **Pre:** IF request.getParameter("cancel_booking") <> null THEN
	booking_id : request.getParameter("booking_id") <> null BookingsDAO.getBookingById(booking_id).getState_name() <> "Annullata" ELSE request.getParameter("book_id") <> null and request.getParameter("start_date") <> null and request.getParameter("end_date") <> null and CardsDAO.getCardByCodice_fiscale(customer.getCodice_fiscale()).isAssociated() = true
Postcondizioni	**context** BookingServlet::doPost(request : HttpServletRequest, response : HttpServletResponse) **Post** : IF @pre request.getParameter("cancel_booking") <> null THEN BookingsDAO.updateBooking(booking_id, "Annullata") <> 0 ELSE BookingsDAO.newBooking(email, start_date, end_date, card_id, book_id) <> 0
Invarianti	

Testing

Caratteristiche testate

Utente

- Login (Cliente, Gestore)

Cliente

- Registrazione di un account
- Richiesta di una Tessera
- Registrazione di una Tessera
- Prenotazione di un Libro
- Modifica dati account

Gestore

- Ricerca di un Cliente
- Cancellazione di un Cliente
- Ricerca di un Libro
- Aggiunta di un Libro
- Modifica di un Libro
- Cancellazione di un Libro
- Aggiunta di un Genere
- Cancellazione di un Genere
- Ricerca di una Tessera
- Aggiunta di una Tessera
- Cancellazione di una Tessera
- Ricerca Prenotazione
- Aggiunta di una Prenotazione
- Modifica Stato Prenotazione
- Cancellazione di una Prenotazione
- Ricerca di un Gestore
- Aggiunta di un Gestore
- Modifica di un Gestore
- Cancellazione di un Gestore
- Modifica Dati Biblioteca

Materiale per il testing

- Un qualsiasi computer sul quale sono stati installati una qualsiasi IDE per Java EE.
- Una qualsiasi versione di JUnit per il testing di alcune funzionalità
- Un browser e il plug-in Selenium IDE.

Specifica di un caso di test

Login di un utente

Parametro	Email
Esistente [EE]	1. È presente all'interno del database. [property EEOK] 2. Non è presente all'interno del database. [error]

Parametro	Password
Associata [AP]	1. È associata all'email. [if EEOK] [property APOK] 2. Non è associata all'email [if EEOK] [error]

ID Test Case	Combinazione	Esito
TC_1_1	EE1, AP1	Successo
TC_1_2	EE2	Errore
TC_1_3	EE1, AP2	Errore

Esecuzione del test

Combinazione 1: Email presente e password associata

Nome	TC_1_1
Output atteso	Reindirizza all' homepage.
Output ottenuto	Reindirizza all' homepage.
Risultato	Corretto

1	*open*	http://localhost:8080/webapp/	
2	*set window size*	1579x1003	
3	*click*	css=#menu > a:nth-child(2)	
4	*click*	id=sign-form	
5	*click*	id=email	
6	*type*	id=email	a@a.it
7	*click*	id=password	
8	*type*	id=password	123456
9	*click*	css=button:nth-child(8)	
10	*click*	css=#menu > a:nth-child(3)	
11	*click*	css=.dropdown-btn:nth-child(2)	
12	*click*	css=.profile-container	
13	*click*	css=.dropdown-btn:nth-child(4)	
14	*click*	css=.dropdown-btn:nth-child(6)	
15	*click*	css=#menu > a:nth-child(1)	

Esecuzione del test

Combinazione 2: Email non esistente

Nome	TC_1_2
Output atteso	“Indirizzo e-mail o password non validi.”
Output ottenuto	“Indirizzo e-mail o password non validi.”
Risultato	Corretto

Running 'TC_1_2'

1. open on http://localhost:8080/webapp/ OK
2. setWindowSize on 1579x1003 OK
3. click on css=#menu > a:nth-child(2) OK
4. click on id=email OK
5. type on id=email with value nonesiste@gmail.com OK
6. type on id=password with value 123456 OK
7. click on css=button:nth-child(8) OK

'TC_1_2' completed successfully

Esecuzione del test

Combinazione 3: Password errata

Nome	TC_1_3
Output atteso	“Indirizzo e-mail o password non validi.”
Output ottenuto	“Indirizzo e-mail o password non validi.”
Risultato	Corretto

Running 'TC_1_3'

1. open on http://localhost:8080/webapp/ OK
2. setWindowSize on 1579x1003 OK
3. click on css=#menu > a:nth-child(2) OK
4. click on id=email OK
5. type on id=email with value a@a.it OK
6. click on id=password OK
7. type on id=password with value abc OK
8. click on css=button:nth-child(8) OK

'TC_1_3' completed successfully

Risultato finale

Visualizzazione Desktop

Dante's Library

ANTONIO

I Promessi Sposi

Romanzo di un amore contrastato nell'Italia del Seicento, "I Promessi Sposi" sono anche il sillabario della nostra modernità: mettono alla prova i valori del cattolicesimo, l'etica borghese, gli ideali risorgimentali, e per le tecniche narrative che adoperano e la lingua viva che inventano segnano un nuovo inizio per la letteratura italiana. Questa edizione, diretta da Francesco de Cristofaro e realizzata da un'equipe multidisciplinare di studiosi, offre una aggiornata panoramica sul capolavoro manzoniano, spaziando dagli aspetti lessicali e interpretativi a quelli linguistici e stilistici. Il volume include la "Storia della Colonna infame" e propone, per la prima volta, una sistematica analisi delle illustrazioni di Gonin & Co. che scandiscono, secondo precisa regia manzoniana, l'edizione definitiva del romanzo.

Genere: Romanzo Storico

Autori: Alessandro Manzoni

Editore: BUR Biblioteca

Disponibile

Data inizio prestito

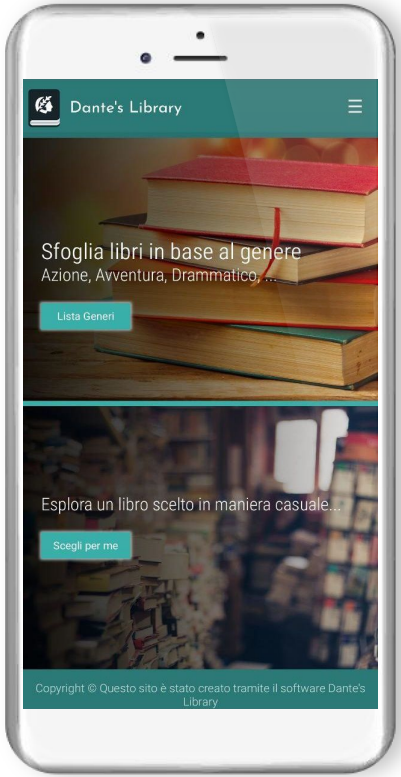
23 Gennaio 2020

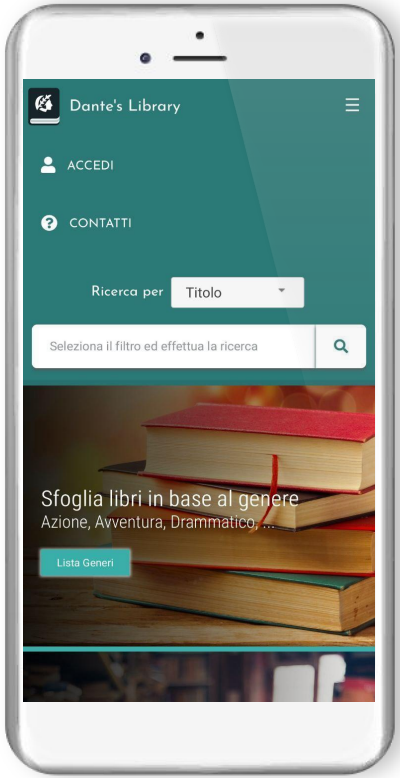
Data fine prestito

Marzo 2020

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Visualizzazione Mobile





Fine