

# SISTEMA BIBLIOTECARIO

Web application per la gestione di risorse e prenotazioni

## Descrizione generale

La web application *Sistema Bibliotecario* è pensata come tipica applicazione utilizzabile all'interno di un ateneo universitario. Questa versione base è rivolta agli studenti che intendono prenotare dei libri messi a disposizione dalla biblioteca di ateneo. Sia l'utente loggato che il guest possono visualizzare la lista dei libri, ordinati per materia (subject). Il guest deve effettuare la registrazione nel caso in cui volesse prenotarne uno o più. La registrazione dell'utente ospite dovrebbe dare esito positivo solo se quest'ultimo frequenta realmente un corso di laurea presso l'università (questa verifica è soltanto simulata). L'utente loggato avrebbe pieno accesso al sito con la possibilità di prenotare gratuitamente un libro, inserendo la data in cui ritirarlo (initialDate) e la data in cui consegnarlo (finalDate) e avrebbe inoltre la possibilità di cancellare una prenotazione già effettuata. Una feature utile che intende offrire il sistema è quella di una bacheca contenente recensioni sui libri che hanno scritto gli studenti registrati, fornendo così un feedback agli altri colleghi.

## Caratteristiche architetturali

Tecnologie utilizzate:

1. HTML e CSS per la struttura logica e la rappresentazione delle pagine web;
2. JavaScript come linguaggio di scripting lato client;
3. La libreria JavaScript JQUERY con le funzionalità AJAX;
4. Classi Java Servlet e JavaServer Pages (JSP) per la generazione di pagine web dinamiche in risposta a richieste http del client e per il processo di validazione ed elaborazione dei dati ricevuti;
5. Apache Tomcat (versione 8.5) come web server;
6. L'API Java JDBC per accedere ai dati memorizzati nel database relazionale;
7. MySQL come gestore di basi di dati relazionale (RDBMS).

## Requisiti funzionali

La web application dovrà:

1. Presentare in primis un'interfaccia per il login / registrazione / logout dell'utente;
2. Permettere la visualizzazione sia all'utente loggato che guest della lista dei libri posseduti dalla biblioteca (suddivisi per materia: informatica, fisica, etc.);
3. Permettere all'utente loggato di prenotare uno o più libri, di visualizzare le prenotazioni che ha effettuato ed in caso di cancellarle.

Funzionalità lato client:

1. Login / registrazione / logout;
2. Validazione dei dati inseriti dall'utente;
3. Visualizzazione dei libri in possesso della biblioteca;
4. Visualizzazione delle recensioni degli utenti (dalla più recente);
5. Visualizzazione prenotazioni;
6. Aggiornamento delle informazioni personali da parte di ciascun utente;
7. ... (vedi NOTA 2)

Funzionalità lato server:

1. Aggiunta degli utenti, delle prenotazioni e delle recensioni al database;
2. Recupero delle informazioni relative agli utenti, alle prenotazioni, ai libri e alle recensioni dal database;
3. Gestione dell'integrità dei dati e della sicurezza del sistema (un utente non può cancellare una prenotazione che ha effettuato un altro utente, oppure due utenti non possono prenotare lo stesso libro nello stesso arco di tempo e simili);
4. ... (vedi NOTA 2)

## Schema relazionale

**USER** (username, password, email, firstname, lastname, age, university, ...)

**BOOK** (codB, title, author, subject, description, ...)

**RESERVATION** (idR, refU, refB, initialDate, finalDate)

FK: refU **REFERENCE** USER

FK: refB **REFERENCE** BOOK

**REVIEW** (idReview, refUser, refBook, dateReview, titleReview, text)

FK: refUser **REFERENCE** USER

FK: refBook **REFERENCE** BOOK

Grazie ai due vincoli di chiave esterna presenti nelle tabelle RESERVATION e REVIEW, l'integrità dei dati viene preservata. Una prenotazione o una recensione devono necessariamente riferirsi a utenti registrati e a libri posseduti dalla biblioteca.

Un determinato libro (codB) può essere prenotato durante un arco di tempo (da initialDate a finalDate) soltanto da un utente (si è assunto che la biblioteca abbia solo una copia di ciascun libro, come succede di solito). Se un altro utente volesse prenotare il libro in questione, dovrebbe aspettare la data di restituzione (finalDate). Questo controllo viene effettuato grazie al codice lato server che interroga il database e agisce di conseguenza.

### NOTA 1:

L'applicazione dovrà far uso della tecnica di web design *Responsive*, in modo che le pagine web adattino automaticamente il layout in base al dispositivo che le visualizza (PC, tablet o smartphone).

### NOTA 2:

Le sezioni 'Funzionalità lato client' e 'Funzionalità lato server' di questo documento possono essere soggette ad aggiunte durante lo sviluppo della web application.