

# **Progetto LSO**

Libreria

Ingenito Roberto - N86004077

Ingenito Simone - N86004063

Sequino Lorenzo - N86004367

# Indice

<b>1</b>	<b>Analisi dei requisiti</b>	<b>2</b>
1.1	Obiettivo del progetto . . . . .	3
1.2	Funzionalità principali . . . . .	3
1.2.1	Registrazione e login utenti . .	3
1.2.2	Gestione del catalogo dei libri .	3
1.2.3	Prestito di libri . . . . .	3
1.2.4	Gestione del carrello e check-out	3
1.2.5	Notifiche agli utenti . . . . .	3
<b>2</b>	<b>Implementazione</b>	<b>4</b>
2.1	Linguaggi di programmazione utilizzati	5
2.2	Struttura del codice . . . . .	5
2.3	Registrazione e login . . . . .	6
2.4	Gestione dei libri e del catalogo . . . .	6
2.5	Prestito e restituzione libri . . . . .	6
2.6	Carrello e check-out . . . . .	6
2.7	Gestione delle notifiche . . . . .	6
<b>3</b>	<b>Database</b>	<b>7</b>
3.1	Schema . . . . .	8
3.2	Descrizione delle tabelle . . . . .	8
3.3	Query principali utilizzate . . . . .	8

## **Analisi dei requisiti**

## **1.1 Obiettivo del progetto**

## **1.2 Funzionalità principali**

### **1.2.1 Registrazione e login utenti**

### **1.2.2 Gestione del catalogo dei libri**

### **1.2.3 Prestito di libri**

### **1.2.4 Gestione del carrello e check-out**

### **1.2.5 Notifiche agli utenti**

# Implementazione

## 2.1 Linguaggi di programmazione utilizzati

C e SQL

## 2.2 Struttura del codice

Il progetto è organizzato in una struttura di cartelle che separa logicamente i vari componenti del progetto. Questa organizzazione aiuta a mantenere il codice pulito, modulare e facile da gestire.

Di seguito, è riportata la struttura gerarchica delle cartelle:

```
project_root
+-- client
|   +-- build
|   +-- include
|   +-- src
+-- config
+-- database
|   +-- include
|   +-- query
|   +-- src
+-- server
    +-- build
    +-- include
    +-- src
```

### client / server

Le cartelle `client` e `server` hanno la stessa struttura:

- **build:** Contiene i file oggetto generati dalla compilazione del codice contenuto in `src`
- **include:** Contiene i file header
- **src:** Contiene il codice sorgente

La separazione tra `include` e `src` permette una chiara distinzione tra le interfacce e le implementazioni, migliorando la manutenibilità del codice.

La cartella `build` isola i file generati dalla compilazione, mantenendo pulito il repository.

### config

Questa cartella possiede dei file di configurazione con implementazioni comuni sia per il *client* che per il *server*. Ad esempio la configurazione del database, le richieste che client e server possono (rispettivamente) fare e ricevere, oppure la configurazione degli indirizzi.

Centralizzare le configurazioni in una cartella dedicata facilita la gestione delle impostazioni del sistema e permette di modificare facilmente i parametri di configurazione senza dover cercare all'interno del codice sorgente.

### database

- **include:** Contiene i file header per la gestione del database.
- **query:** Contiene i file SQL utilizzati per la gestione del database. (Creazione delle tabelle, trigger, funzioni, viste, ...)
- **src:** Contiene il codice sorgente per l'interazione con il database.

La separazione dei file SQL e del codice sorgente relativo al database aiuta a mantenere organizzate le operazioni di gestione del database e facilita eventuali modifiche o aggiornamenti.

## **2.3 Registrazione e login**

Client

Server

## **2.4 Gestione dei libri e del catalogo**

Client

Server

## **2.5 Prestito e restituzione libri**

Client

Server

## **2.6 Carrello e check-out**

Client

Server

## **2.7 Gestione delle notifiche**

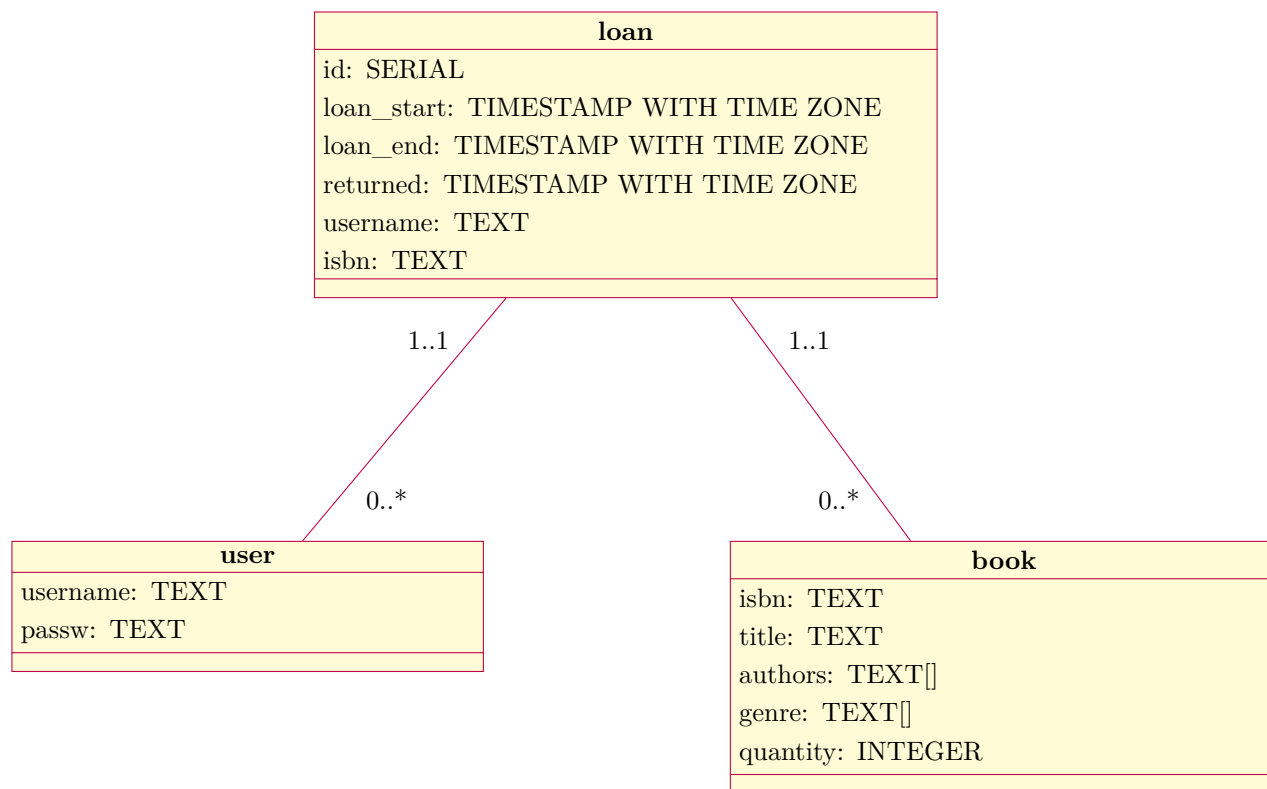
Client

Server

# Database



### 3.1 Schema



### 3.2 Descrizione delle tabelle

### 3.3 Query principali utilizzate