

Sistema di gestione di una libreria personale

Project Proposal

Prof. Della Pietra

a.a 2023/2024

Introduzione

Questo documento serve come proposta per lo sviluppo di un sistema di gestione di una libreria personale utilizzando Java Spring per il backend e Angular per il frontend. Il sistema consentirà agli utenti di catalogare i propri libri, scrivere e leggere recensioni, e gestire la propria collezione personale.

Obiettivi del Progetto

Questo documento serve come proposta per lo sviluppo di un sistema di gestione di una libreria personale utilizzando Java Spring per il backend e Angular per il frontend. Il sistema consentirà agli utenti di catalogare i propri libri, scrivere e leggere recensioni, e gestire la propria collezione personale.

- Creare un'applicazione web user-friendly per gestire una collezione personale di libri
- Implementare un backend robusto e sicuro utilizzando Java Spring.
- Sviluppare un'interfaccia utente intuitiva con Angular.
- Fornire funzionalità di autenticazione, catalogazione, recensione e ricerca di libri.

Descrizione del Progetto

Il progetto si suddivide in due parti principali: lo sviluppo del backend e del frontend.

Backend

- Sviluppato in Java utilizzando il framework Spring Boot.
- Gestione di autenticazione degli utenti con JWT.
- Implementazione di un'API RESTful per operazioni CRUD sui libri e sulle recensioni.
- Uso di Spring Data JPA per l'interazione con il database.
- Sicurezza e autorizzazione gestite con Spring Security.

Frontend

- Sviluppato con Angular.
- Interfaccia utente responsiva e accessibile.
- Consumo delle API RESTful del backend per visualizzare e gestire i dati.
- Implementazione di funzionalità di autenticazione e registrazione.

Target di mercato

- Lettori e collezionisti di libri che cercano un modo semplice per catalogare e gestire la loro collezione.
- Bibliofili interessati a condividere recensioni e opinioni sui libri.

Tecnologie e strumenti

- **Backend:** Java, Spring Boot, Spring Data JPA, Spring Security, JWT.
- **Frontend:** Angular, HTML, CSS, TypeScript.
- **Database:** MySQL.
- **Sistema di controllo versione:** Git.
- **Strumenti di testing:** JUnit, Mockito, Jasmine, Karma. (Opzionale)

Milestones del Progetto

- **Setup Iniziale e Progettazione** (2 settimane)
- **Sviluppo del Backend** (4 settimane)
- **Sviluppo del Frontend** (4 settimane)
- **Integrazione e Testing** (2 settimane)
- **Revisione e Deployment** (2 settimane)

Rischi

- **Ritardi nello sviluppo:** implementazione di una gestione agile del progetto per adattarsi rapidamente ai cambiamenti
- **Problemi di sicurezza:** adozione di best practices in sicurezza informatica e revisione del codice.
- **Scalabilità:** progettazione di un'architettura flessibile che può essere facilmente scalata in futuro.

Budget stimato

TE LO DIRÒ ALLA FINE ;-)

Documentazione

Backend

Prof. Della Pietra

a.a 2023/2024

Obiettivo

Sviluppare un'API RESTful con Java Spring per gestire una libreria personale. L'API permetterà agli utenti di catalogare i propri libri e gestire la propria collezione.

Strumenti e Tecnologie

- **Java Spring Boot**: per il framework di sviluppo backend.
- **Spring Data JPA**: per l'interazione con il database.
- **Spring Security con JWT**: per l'autenticazione e la sicurezza.
- **Database MySQL**: per la persistenza dei dati.
- **Maven**: per la gestione delle dipendenze.
- **Git**: per il controllo versione.

Fasi di Sviluppo

Fase 1: Configurazione iniziale

Configurazione del Progetto:

- Creare un nuovo progetto Spring Boot utilizzando Maven come sistema di gestione delle dipendenze.
- Aggiungere le dipendenze necessarie nel file pom.xml.

Struttura del progetto:

- Definire pacchetti per modelli, repository, servizi, controller e configurazioni di sicurezza.

Fase 2: Modellazione del database e JPA

Entità:

- Creare classi entità per Libro e Utente.
- Definire attributi, relazioni tra entità e annotazioni JPA.

Repository:

- Creare interfacce repository per ciascuna entità, estendendo JpaRepository.

Fase 3: Autenticazione e Sicurezza

Autenticazione con JWT:

- Implementare il filtraggio delle richieste per generare e verificare i token JWT.
- Creare classi per gestire le richieste di autenticazione.

Configurazione della sicurezza:

- Configurare Spring Security per l'autenticazione e l'autorizzazione.

Fase 4: Sviluppo delle API REST

Controller:

- Creare controller per gestire le richieste HTTP relative a libri e utenti.
- Definire endpoint per operazioni CRUD sui libri e per l'autenticazione.

Servizi:

- Implementare la logica di business nei servizi.
- Gestire le transazioni e la logica di accesso ai dati.

Fase 5: Gestione ruoli utente

Utente:

- Definire diversi ruoli (es. ADMIN, USER).
- Implementare logica per assegnare e verificare i ruoli.

Fase 6: Testing

[Opzionale] Test Unitari e d'integrazione:

- Scrivere test per i servizi e i controller usando JUnit e Mockito.
- Verificare il corretto funzionamento delle API e dell'autenticazione.