

A.A. 2021 - 2022

M. CECCARELLI - E. D'ANGELIS - P. ZARRI

LIBRARY SEAT RESERVATION

ANALISI DEI REQUISITI



REQUISITI FUNZIONALI

Possiamo suddividere le funzionalità principali che il sistema deve fornire nelle seguenti categorie:

- Gestione degli utenti;
- Gestione delle biblioteche;
- Gestione delle prenotazioni.

Di seguito viene riportata un'analisi più dettagliata delle funzionalità.

GESTIONE DEGLI UTENTI

Il sistema deve permettere la gestione di risorse di tipo **Utente**:

- Il sistema deve permettere la **registrazione** di un Utente.
- Il sistema deve permettere il **login/logout** di un Utente.

GESTIONE DELLE BIBLIOTECHE

Il sistema deve permettere la gestione di risorse di tipo **Biblioteca**:

- Il sistema deve permettere l'**aggiunta** di una nuova Biblioteca;
- Il sistema deve permettere la **modifica** della capienza di una Biblioteca;
- Il sistema deve permettere la **cancellazione** di una Biblioteca;
- Il sistema deve permettere la **consultazione** delle informazioni di una Biblioteca.

GESTIONE DELLE PRENOTAZIONI

Il sistema deve permettere la gestione di risorse di tipo **Prenotazione**:

- Il sistema deve permettere la **creazione** di una nuova Prenotazione;
- Il sistema deve permettere la **cancellazione** di una Prenotazione.
- Il sistema deve permettere la **consultazione** delle prenotazioni effettuate.

REQUISITI NON FUNZIONALI

IMPLEMENTAZIONE

- Il sistema dovrà avere una **natura distribuita**, ovvero deve essere diviso in tre moduli: un **modulo frontend**, un **gateway** che intermedia le richieste per la gestione della coda e un **modulo backend**;
- Il sistema deve essere realizzato su una **architettura RESTful**;
- Il modulo Front-End dovrà essere implementato utilizzando **Angular (Typescript)**;
- Il modulo proxy e il modulo backend dovranno essere implementati utilizzando **Java Enterprise Edition (JEE)**;
- La logica del modulo **gateway** deve essere indipendente dal contesto applicativo.

REQUISITI DI DOMINIO

Di seguito sono riportati i requisiti di dominio per i tre modelli: **Utente**, **Biblioteca** e **Prenotazione**.

- Una risorsa di tipo **Utente** deve prevedere i seguenti attributi:
 - un identificativo
 - una email
 - un nome
 - un cognome
 - una password
 - una lista di ruoli
- Una risorsa di tipo **Biblioteca** deve prevedere i seguenti attributi:
 - un identificativo
 - un nome
 - un indirizzo
 - una capienza

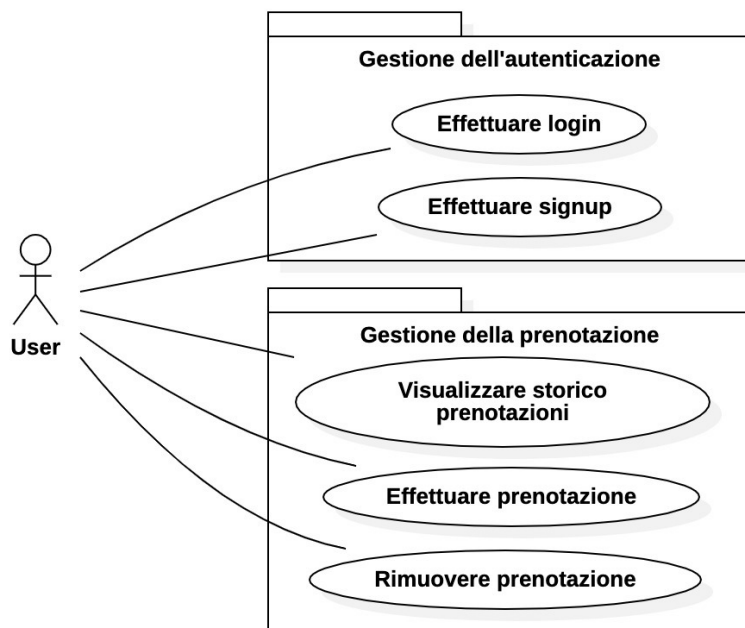
- Una risorsa di tipo **Prenotazione** deve prevedere i seguenti attributi:
 - un identificativo
 - un identificativo dell'utente che ha effettuato la prenotazione
 - un identificativo della biblioteca relativa alla prenotazione
 - una data e una fascia oraria

CASI D'USO

Il modello dei casi d'uso rappresenta un'astrazione di alto livello del modello di interazione e definisce il comportamento funzionale che il sistema offre a chi lo utilizza. Ci sono due elementi fondamentali:

- gli **attori**, cioè coloro che interagiscono in maniera diretta con il sistema (persone, dispositivi o altri sistemi);
- i **casi d'uso**, cioè un'astrazione delle interazioni che avvengono tra gli attori e il sistema (funzionalità che il sistema offre a chi lo utilizza).
-

CASI D'USO DELL'UTENTE



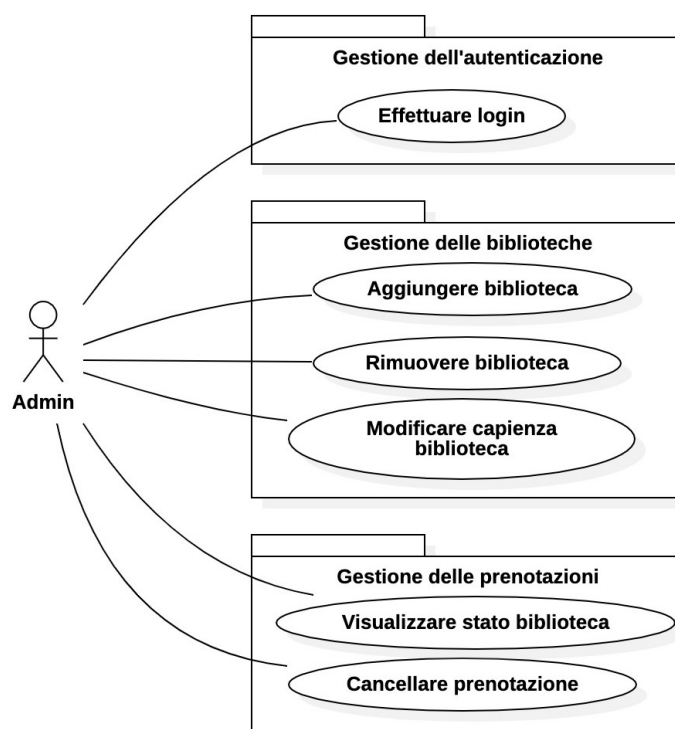
Esempio di **casi d'uso dell'utente** per quanto riguarda la **gestione dell'autenticazione**:

Caso d'uso: Effettuare il login
ID: UC01
Attori: Utente
Sequenza degli Eventi: <ol style="list-style-type: none"> 1. L'utente accede alla pagina di login; 2. L'utente inserisce le proprie credenziali; 3. Il frontend prepara i dati da inviare al backend; 4. Il frontend invia una richiesta al backend; 5. Il backend verifica la correttezza delle credenziali inserite; 6. Il backend invia una risposta al frontend comunicando l'esito dell'operazione; 7. Il frontend utilizza le informazioni ricevute dal backend per autenticare l'utente.
Postcondizioni: <ol style="list-style-type: none"> 1. L'utente è correttamente autenticato nell'applicazione.

Esempio di **casi d'uso dell'utente** per quanto riguarda la **gestione della prenotazione**:

Caso d'uso: Effettuare una prenotazione
ID: UC02
Attori: Utente
Precondizioni: <ol style="list-style-type: none"> 1. L'utente è autenticato correttamente nell'applicazione.
Sequenza degli Eventi: <ol style="list-style-type: none"> 1. L'utente seleziona una biblioteca; 2. Se il massimo numero di utenti è stato raggiunto, l'utente viene messo in coda; 3. Raggiunto il suo turno, l'utente può accedere alla pagina di prenotazione relativa alla biblioteca scelta; 4. L'utente può vedere la disponibilità della biblioteca per i vari giorni e fasce orarie; 5. L'utente può scegliere un giorno e una fascia oraria; 6. L'utente esegue la prenotazione per la data selezionata; 7. Il frontend prepara i dati da inviare al backend; 8. Il frontend invia una richiesta al backend; 9. Il backend salva la prenotazione dell'utente all'interno del database e invia una risposta al frontend comunicando l'esito; 10. Il frontend utilizza le informazioni ricevute per comunicare visivamente l'esito all'utente.
Postcondizioni: <ol style="list-style-type: none"> 1. L'utente ha effettuato una prenotazione per la biblioteca scelta.

CASI D'USO DELL'ADMIN



Esempio di **casi d'uso dell'admin** per quanto riguarda la **gestione dell'autenticazione**:

Caso d'uso: Effettuare il login
ID: UC03
Attori: Admin
Sequenza degli Eventi: <ol style="list-style-type: none"> 1. L'admin accede alla pagina di login; 2. L'admin inserisce le proprie credenziali; 3. Il frontend prepara i dati da inviare al backend; 4. Il frontend invia una richiesta al backend; 5. Il backend verifica la correttezza delle credenziali inserite; 6. Il backend invia una risposta al frontend comunicando l'esito dell'operazione; 7. Il frontend utilizza le informazioni ricevute dal backend per autenticare l'admin.
Postcondizioni: <ol style="list-style-type: none"> 1. L'admin è correttamente autenticato nell'applicazione.

Esempio di **casi d'uso dell'admin** per quanto riguarda la **gestione delle biblioteche**:

Caso d'uso: Aggiungere una biblioteca	
ID: UC04	
Attori: Admin	
Sequenza degli Eventi: <ol style="list-style-type: none"> 1. L'admin accede alla pagina Home; 2. L'admin può cliccare sul bottone "Aggiungi nuova biblioteca"; 3. L'admin viene reindirizzato alla pagina per compilare i dati relativi alla biblioteca da aggiungere; 4. L'admin inserisce le informazioni necessarie; 5. Il frontend prepara i dati da inviare al backend; 6. Il frontend invia una richiesta al backend; 7. Il backend salva la biblioteca all'interno del database e invia una risposta al frontend comunicando l'esito; 8. Il frontend utilizza l'informazioni ricevute per comunicare visivamente l'esito all'admin. 	
Postcondizioni: <ol style="list-style-type: none"> 1. La biblioteca è correttamente aggiunta all'elenco delle biblioteche. 	

MODELLO DI DOMINIO CONCETTUALE

Basandosi sulle informazioni fornite dai casi d'uso, delineati precedentemente, possiamo definire un modello di dominio:

Nome	Proprietà	Descrizione
Library	<ul style="list-style-type: none"> • un identificativo • un nome • un indirizzo • una capacità 	Oggetto che comprende tutte le informazioni relative ad una biblioteca.
Reservation	<ul style="list-style-type: none"> • un identificativo • un identificativo dell'utente • un identificativo della biblioteca • una data e una fascia oraria 	Oggetto che comprende tutte le informazioni relative ad una prenotazione.
User	<ul style="list-style-type: none"> • un identificativo • una email • un nome • un cognome • una password • una lista di ruoli 	Oggetto che comprende tutte le informazioni relative ad un utente.