UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

LSO 2022/2023

Gruppo MovieHub

Navarra Antonio
N86002897

Giordano Vincenzo
N86003039

Sommario

Descrizione dell'applicazione e del Progetto

In questo progetto, tramite un Client in Java e un Server in C, verrà gestita un'applicazione che ci permetterà la visualizzazione adattiva di contenuti come film e serie TV.

L'applicazione avrà una struttura sistematica in modo tale che si adeguerà in base alle scelte dell'utente.

L'obiettivo è di consentire a qualsiasi utente la registrazione e la navigazione, oltre che la visualizzazione della propria scelta.

Caratteristiche Principali

I clienti possono selezionare scegliere i film in due modalità:

1. Utilizzando la barra di ricerca con la possibilità di aggiungere i filtri;
2. Sfruttando le liste personalizzate:
    a. Consigliati da The Movie Database (TMDB): un elenco di film e serie tv consigliati da TMDB;
    b. Prossime uscite: un elenco di prossime uscite di film, nelle sale cinematografiche, e serie tv sulle diverse piattaforme;

Le informazioni relative ai film sono tutte conservate all'interno del database online TMDB, invece per gli utenti, sono conservate in un database locale, per la precisione PostgreSQL.

Il sistema può essere accessibile tramite applicazione per cellulare o tablet, rendendo l'esperienza utente più comoda e portatile.

Scelte e Struttura del Sistema

In questa guida forniremo una panoramica dettagliata di come l'app è strutturata e di come tutte le tecnologie interagiscono tra di loro al fine di permettere il funzionamento dell'applicativo.

Panoramica dell'App

MovieHub è un'applicazione che ti permetterà di scegliere tra tutti i film usciti al cinema e non solo. L'applicazione è composta da tre parti principali e fondamentali: il server, il database (locale e online) e il client Android. Di seguito è riportata la struttura nel dettaglio.

Struttura del Sistema:

- Server:
  - Linguaggio: C
  - Ospitato su WSL Ubuntu 22.04
- Database:
  - Database Management System: PostgreSQL
  - The Movie Database: TMDB
- Client:
  - Linguaggio: Java
  - Framework: Android Studio

Server

Il server è scritto in linguaggio C ed è ospitato su WSL. Gestisce la comunicazione con il client e il database.

Si occupa di ricevere richieste da parte del client, elaborarle e inviare le corrispettive risposte appropriate. Ad esempio, quando un utente vuole registrarsi il server carica i dati nel database locale invece quando vuole effettuare il login, il server carica i dati dal database online.

Per permettere il funzionamento corretto dell'applicazione, abbiamo effettuato una prima scelta importante, il protocollo di comunicazione da utilizzare. La scelta è ricaduta subito sull'utilizzo del protocollo TCP, al fine di garantire una connessione diretta, sicura e senza perdita di dati, in modo da garantire sempre il corretto funzionamento dell'applicazione.

```c
int serverSocket, clientSocket;
struct sockaddr_in serverAddress, clientAddress;
socklen_t clientAddressLength = sizeof(struct sockaddr_in);

// Creazione del socket del server
serverSocket = socket(AF_INET, SOCK_STREAM, 0);
if (serverSocket == -1) {
    perror("Error creating socket");
    exit(EXIT_FAILURE);
}

// Configurazione dell'indirizzo del server
serverAddress.sin_family = AF_INET;
serverAddress.sin_addr.s_addr = INADDR_ANY;
serverAddress.sin_port = htons(PORT_NUMBER);
```

Figura 1: ***Codice relativo al setup del server***

Nella funzione main, dopo aver correttamente inizializzato il server, vengono effettuate le seguenti operazioni:

- Il server entra in modalità di ascolto utilizzando **listen()** per accettare le connessioni dai clienti.
- In un ciclo, il server accetta nuove connessioni e crea nuovi thread **handleConnection**, per gestire ciascuna connessione in thread separati.
- Il thread principale del server attende continuamente nuove connessioni e avvia thread separati per gestirle.
- Il server viene chiuso in modo pulito utilizzando **close()** quando è necessario terminare l'esecuzione.

```
// Associazione del socket all'indirizzo del server
if (bind(serverSocket, (struct sockaddr *)&serverAddress, sizeof(serverAddress)) < 0) {
    perror("Bind failed");
    exit(EXIT_FAILURE);
}

// Ascolto delle connessioni in arrivo
listen(serverSocket, 3);

printf("Server MovieHub online\n");
printf("Server in ascolto su porta %d...\n\n", PORT_NUMBER);

// Accettazione delle connessioni e gestione dei thread
while ((clientSocket = accept(serverSocket, (struct sockaddr *)&clientAddress, &clientAddressLength))) {
    printf("Connection with client established\n");

    pthread_t thread;
    int *newSocket = malloc(sizeof(int));
    if (!newSocket) {
        perror("Error allocating memory");
        exit(EXIT_FAILURE);
    }
    *newSocket = clientSocket;

    if (pthread_create(&thread, NULL, handleConnection, (void *)newSocket) != 0) {
        perror("Error creating thread");
        free(newSocket);
        exit(EXIT_FAILURE);
    }
}
if (clientSocket < 0) {
    perror("Connection with client failed\n");
    exit(EXIT_FAILURE);
}

return 0;
```

Figura 2: ***Codice responsabile della creazione di nuovi thread***

La funzione **handleConnection** è il thread che gestisce ogni connessione client. All'interno di questa funzione, vengono effettuate le operazioni di comunicazione con il client, come ad esempio la registrazione e il login dell'utente, e il database PostgreSQL. Questo thread è responsabile dell'elaborazione delle richieste del client e dell'invio delle risposte appropriate.

```c
// Ricevi un messaggio dal client
while (recv(clientSocket, clientMessage, sizeof(clientMessage), 0) > 0) {
    char *token = strtok(clientMessage, ";");

    if (token != NULL) {
        if (strcmp(token, "Registrazione") == 0) {
            printf("Register new user...\n");

            token = strtok(NULL, ";");
            strcpy(user.username, token);

            token = strtok(NULL, ";");
            strcpy(user.password, token);

            token = strtok(NULL, ";");
            strcpy(user.accessibility, token);

            if (registerUser(user, connection)) {
                printf("Registration was successful\n\n");
                strcpy(message, "OK");
            } else {
                printf("Error during registration\n\n");
                strcpy(message, "Errore");
            }
        }
```

```c
if (strcmp(token, "Login") == 0) {
    printf("Login utente ...\n\n");

    token = strtok(NULL, ";");
    strcpy(user.username, token);

    token = strtok(NULL, ";");
    strcpy(user.password, token);

    if (loginUser(&user, connection)) {
        printf("Login successful\n\n");
        strcpy(message, user.accessibility);
    } else {
        printf("Error during login\n\n");
        strcpy(message, "Errore");
    }
}

sendMessageToClient(message, clientSocket);
```

Figura 3: ***Codice responsabile per la connessione al client***

La funzione **sendMessageToClient** ci permette di inviare un messaggio al client nel caso in cui qualcosa vada storto durante la connessione che avviene all'interno di **handleConnection**.

All'interno di quest'ultima viene effettuata la disconnessione del client e la chiusura della socket con eventuale liberazione delle risorse

```
// Gestione della disconnessione del client
ssize_t recvResult = recv(clientSocket, message, sizeof(message), 0);
if (recvResult == 0) {
    puts("Client disconnected\n\n");
    fflush(stdout);
    close(clientSocket);
} else if (recvResult == (ssize_t)-1) {
    perror("recv failed");
    close(clientSocket);
}

// Chiusura del socket e liberazione delle risorse
close(clientSocket);
PQfinish(connection);
free(socketDesc);
return 0;
```

Figura 4: ***Codice responsabile della disconnessione del client, chiusura della socket e liberazione delle risorse***

Database

Il database in locale è basato su PostgreSQL e memorizza i dati relativi agli utenti. Contiene tabelle per la registrazione e il login da parte dell'utente.

Il database online, The Movie Database (TMDB) ci permette di recuperare i migliori film in streaming, in tv, a noleggio e al cinema. Offre delle librerie (API) che permettono di caricare dinamicamente serie tv, recensioni e tutto quello che ricopre il mondo del cinema.

Client Android

Il client Android è scritto in Java e sviluppato utilizzando Android Studio. Gli utenti possono accedere all'app tramite dispositivi mobili come smartphone e tablet.

Attraverso l'app, gli utenti possono visualizzare la homepage dove ci saranno una serie di film dedicati divisi in due categorie:

1. Consigliati da TMDB
2. Le prossime uscite al cinema

Funzionalità dell’app

Registrazione e login

All'avvio dell'applicazione, gli utenti vengono accolti da una homepage dinamica e ricca di contenuti. La pagina principale è suddivisa in due sezioni distinte:

1. **Film Consigliati da The Movie Database**: Una selezione curata di film consigliati, basata sui gusti e sulle preferenze dell'utente.
2. **Prossime Uscite**: Un elenco delle prossime uscite cinematografiche e televisive, consentendo agli utenti di rimanere aggiornati sulle ultime novità.

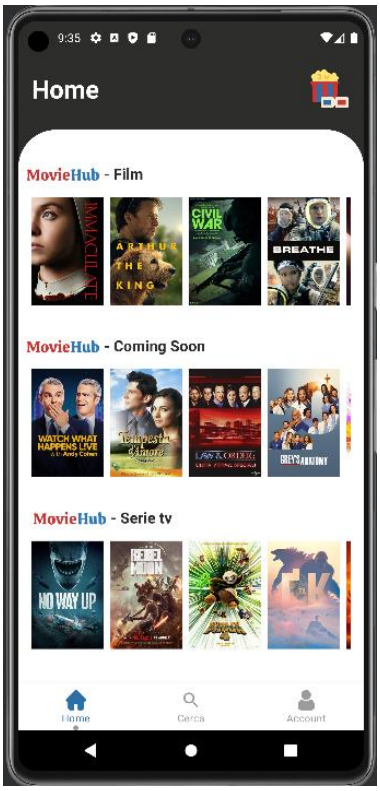

*Figura 5: **Homepage dell'applicazione***

Per offrire un'esperienza personalizzata e garantire la sicurezza dei dati, l'applicazione richiede agli utenti di effettuare il login o la registrazione. Durante il processo di registrazione, gli utenti sono tenuti a fornire un username e una password, con requisiti specifici di sicurezza.

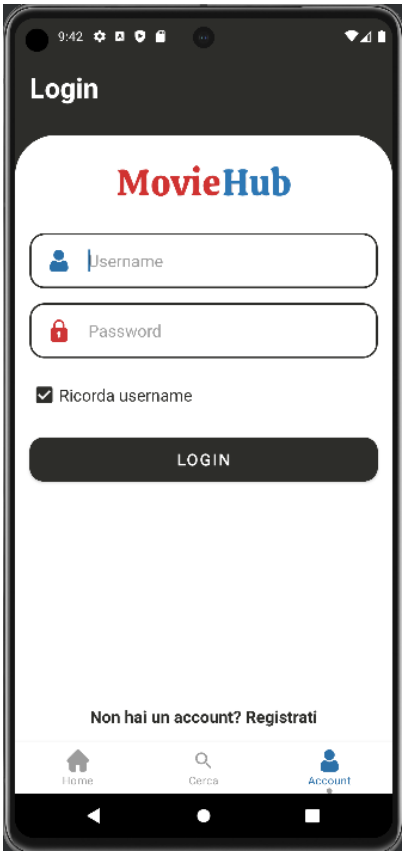

*Figura 6: **Pagina dedicata al login***

Un aspetto fondamentale della nostra applicazione è l'attenzione all'accessibilità visiva. Gli utenti hanno la possibilità di personalizzare l'interfaccia in base alle proprie esigenze visive. Le opzioni di personalizzazione includono:

- **Acromatopsia**: Filtro bianco/nero per gli utenti affetti da acromatopsia.
- **Deuteranopia**: Filtro verde per gli utenti con deuteranopia.
- **Tritanopia**: Filtro blu/giallo per gli utenti con tritanopia.
- **Ipovisione**: Incremento delle dimensioni dell'interfaccia per gli utenti con ipovisione.
- **Acromatopsia + Ipovisione**: Combinazione di filtro bianco/nero e incremento delle dimensioni per gli utenti con acromatopsia e ipovisione.
- **Daltonismo + Ipovisione**: Filtro verde e incremento delle dimensioni per gli utenti con daltonismo e ipovisione.
- **Tritanopia + Ipovisione**: Filtro blu/giallo e incremento delle dimensioni per gli utenti con tritanopia e ipovisione.

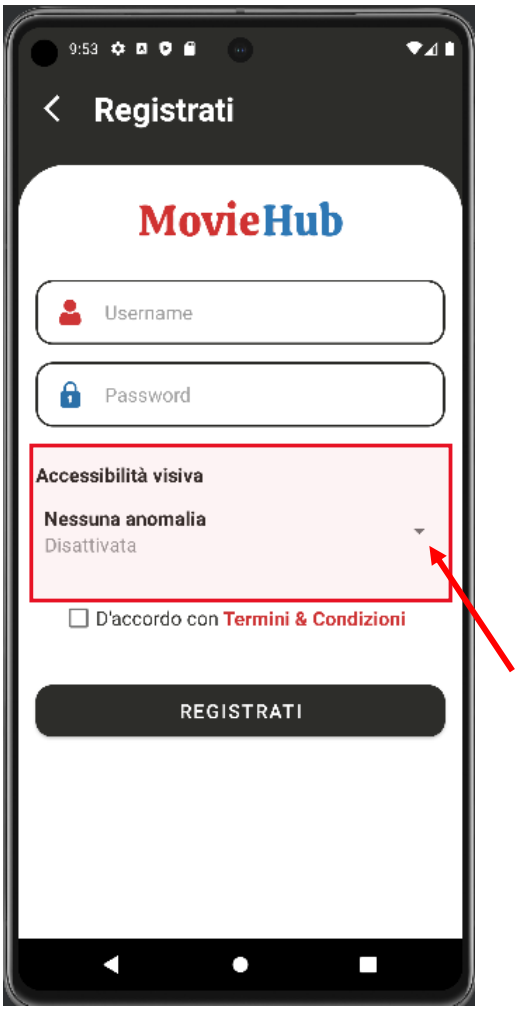

Figura 7: ***Pagina dedicata alla registrazione dell'utente con la scelta dell'accessibilità visiva***

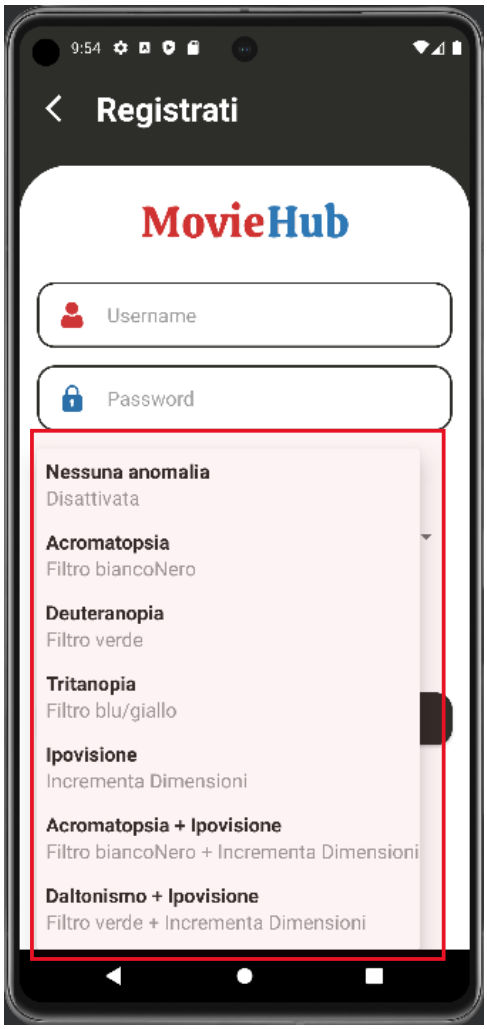

Figura 8: ***Scelta dell'accessibilità***

Benefici dell'Applicazione:

- **Personalizzazione**: L'applicazione offre un'esperienza altamente personalizzata, adattandosi alle preferenze individuali di ogni utente.
- **Accessibilità**: L'attenzione all'accessibilità visiva garantisce un'esperienza inclusiva per gli utenti con esigenze specifiche.

- **Sicurezza**: Il processo di autenticazione garantisce la sicurezza dei dati degli utenti, proteggendo le informazioni personali e sensibili.

Ricerca di un film/serie tv

L'applicazione offre agli utenti la possibilità di cercare film e serie TV attraverso una comoda e intuitiva funzione di ricerca. Questa funzionalità è accessibile tramite un'apposita sezione dedicata, dove gli utenti possono inserire il titolo desiderato all'interno della label per la ricerca.

Modalità di utilizzo:

1. **Accesso alla sezione di ricerca**: Gli utenti possono accedere alla sezione di ricerca dalla barra di navigazione principale dell'applicazione. La presenza di un'icona intuitiva, come una lente d'ingrandimento o un'icona di ricerca, guida gli utenti verso questa funzionalità.
2. **Inserimento del nome del film o della serie tv**: All'interno della sezione di ricerca, gli utenti trovano un campo di inserimento testuale, designato con una label chiara e visibile. Qui possono digitare il nome del film o della serie tv che desiderano cercare.
3. **Interazione con i risultati della ricerca**: Dopo aver inserito il nome del contenuto desiderato, gli utenti possono avviare la ricerca tramite l'apposito pulsante o semplicemente premendo "Invio" sulla tastiera virtuale del dispositivo. L'applicazione avvia quindi il processo di ricerca e visualizza i risultati pertinenti in tempo reale.
4. **Visualizzazione dei risultati**: I risultati della ricerca vengono presentati in modo chiaro e organizzato, consentendo agli utenti di esaminare rapidamente le opzioni disponibili. Ogni risultato è accompagnato da informazioni essenziali, come il titolo del film o della serie TV, una piccola descrizione e un link per visualizzare le informazioni in modo più dettagliato.

Pagina Dettagliata del Film:

Una volta cliccato sul link "**Mostra Tutto**", l'utente viene reindirizzato a una nuova pagina che offre una visione dettagliata del film selezionato. Questa pagina fornisce una panoramica completa delle caratteristiche principali del film, inclusi dettagli come la trama, il cast, il genere, la durata e la data di rilascio.

1. **Descrizione dettagliata del film**: La pagina dettagliata del film presenta una descrizione approfondita della trama e dei temi principali trattati nel film. Questo permette agli utenti di ottenere una comprensione più completa del contenuto prima di decidere se guardarlo o meno.
2. **Data di rilascio**: Accanto alla descrizione del film, viene fornita la data di rilascio ufficiale. Questo permette agli utenti di tenere traccia delle nuove uscite e di pianificare di conseguenza la loro visione.

Figura 9: ***Campo di ricerca film o serie tv***

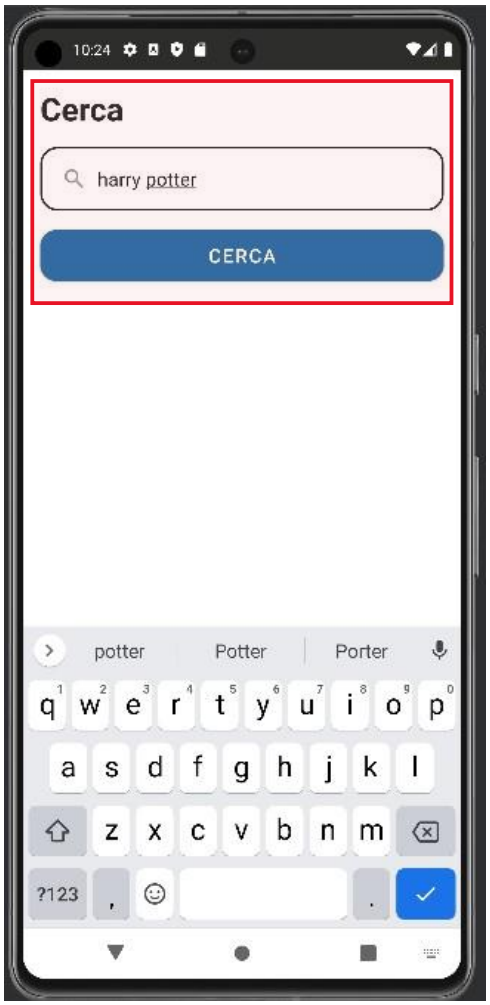

Figura 10: ***Inserimento film o serie tv***

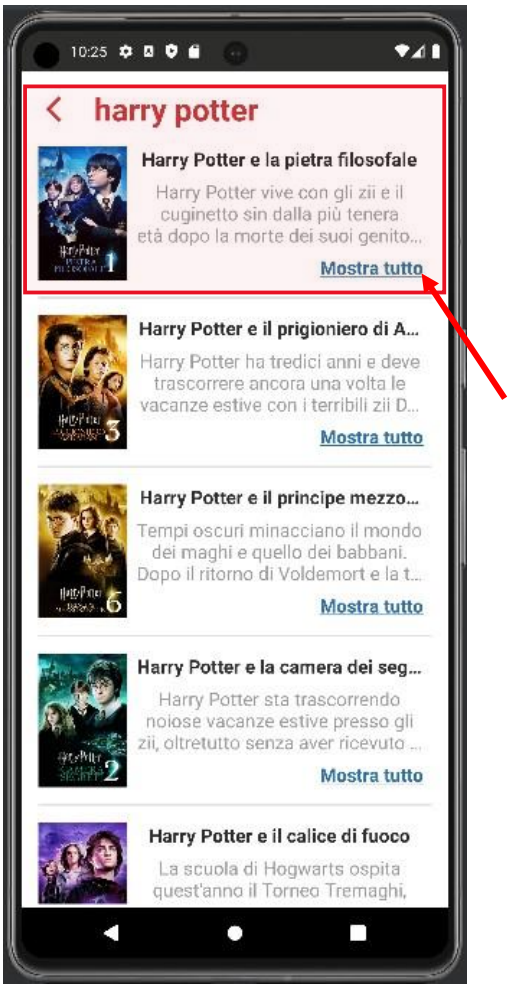

Figura 11: ***Interazione con i risultati della ricerca***

Figura 12: ***Descrizione dettagliata del film***