UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

LSO 2022/2023

Gruppo **MovieHub**

Navarra Antonio
N86002897

Giordano Vincenzo
N86003039

SVILUPPARE UN' INTERFACCIA ADATTIVA PER LA VISUALIZZAZIONE DI CONTENUTI CINEMATOGRAFICI

Sommario

1. **Presentazione del Progetto**
    a. Requisiti del sistema
    b. Principali Caratteristiche
    c. Panoramica dell'App
2. **Le scelte implementative**
    a. Struttura del Sistema
        i. Server
        ii. Database PostgreSQL
        iii. Database TMDB (The Movie DB)
        iv. Client Android
3. **Use Case**
    a. Homepage
    b. Registrazione e login
        i. Impostazioni Accessibilità
    c. Ricerca di un film/serie tv

1. PRESENTAZIONE DEL PROGETTO

Requisiti del sistema

Il cliente ha richiesto un'applicazione per dispositivi Android che includa un'interfaccia adattiva per consentire agli utenti con problemi visivi di navigare in modo confortevole. Il sistema deve essere in grado di gestire l'accesso di più utenti contemporaneamente senza compromettere le prestazioni. Gli utenti devono poter accedere all'applicazione anche tramite i dati biometrici raccolti dal dispositivo. L'applicazione deve utilizzare un database per memorizzare le informazioni richieste.

Principiali Caratteristiche

L'applicazione si compone di tre componenti principali: il Client per dispositivi Android, sviluppato in Java; il Server, in linguaggio C per piattaforme Linux; e due database (**PostgreSQL** e **TMDB**) dedicati alla gestione dei dati.

Panoramica dell'App

Il sistema permette la gestione di una piattaforma di catalogo cinematografico, mirata a fornire una navigazione accessibile a tutti. Gli utenti possono accedere alla piattaforma tramite un'applicazione mobile o un tablet, che offre un'interfaccia adattativa per garantire la migliore esperienza possibile. **MovieHub** ha implementato un sistema tecnologico avanzato che regola automaticamente la gamma di colori e le dimensioni degli oggetti per supportare, ad esempio, coloro che soffrono di daltonismo. Il catalogo di film e serie TV è fornito ed aggiornato costantemente da **The Movie DB**, una fonte affidabile e completa di contenuti multimediali. Gli utenti possono cercare i loro contenuti preferiti utilizzando una barra di ricerca o sfruttando elenchi personalizzati predefiniti, come "consigliati", che offrono una selezione curata di contenuti basata sugli interessi e le preferenze degli utenti.

2. LE SCELTE IMPLEMENTATIVE

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

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 principalmente le connessioni dei client per un servizio di registrazione e accesso utenti. Utilizza un database **PostgreSQ**L per memorizzare le informazioni degli utenti. Il server ascolta su una porta specifica, accetta le connessioni dei client e avvia un **thread** per gestire ciascuna connessione. Durante la comunicazione con i client, gestisce le richieste di registrazione e accesso degli utenti, interagendo con il database di conseguenza. Una volta terminate le connessioni, le risorse vengono liberate correttamente.

```
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: **crea un socket TCP per il server e lo configura per ascoltare su una porta specifica**

Il server entra in modalità di ascolto utilizzando la funzione **listen()** per accettare le connessioni dai client.

- In un ciclo, il server accetta nuove connessioni e crea nuovi thread mediante la funzione **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 sicuro utilizzando la funzione **close()** quando è necessario terminare l'esecuzione.

```c
// 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: ***Connessioni con il client***

La funzione **handleConnection** è il thread che gestisce ogni connessione client. All'interno di quest'ultima vengono effettuate le principali comunicazione, come ad esempio la registrazione e il login dell'utente. Questo thread è responsabile dell'elaborazione delle richieste del client e dell'invio delle response.

```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: ***Scambio messaggi con il Client***

La funzione **sendMessageToClient** ci permette di inviare messaggi al client e nel caso qualcosa vada storto, segnalare l'errore e il suo principio

.

```c
// 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: ***Disconnessione e liberazione delle risorse***

Database PostgreSQL e TMDB (The Movie DB)

Abbiamo optato per l'uso di due database distinti per la gestione dei dati: PostgreSQL è stato scelto per le operazioni di autenticazione degli utenti: il login, la registrazione e la gestione dell'accessibilità selezionata. Mentre per i contenuti cinematografici abbiamo adottato il database online TMDB (The Movie DB) tramite le **API** key fornite dal sito, integrate nel client Android attraverso la libreria **Retrofit**. La scelta di TMDB è stata motivata dalla sua ampia catalogazione e aggiornamenti regolari.

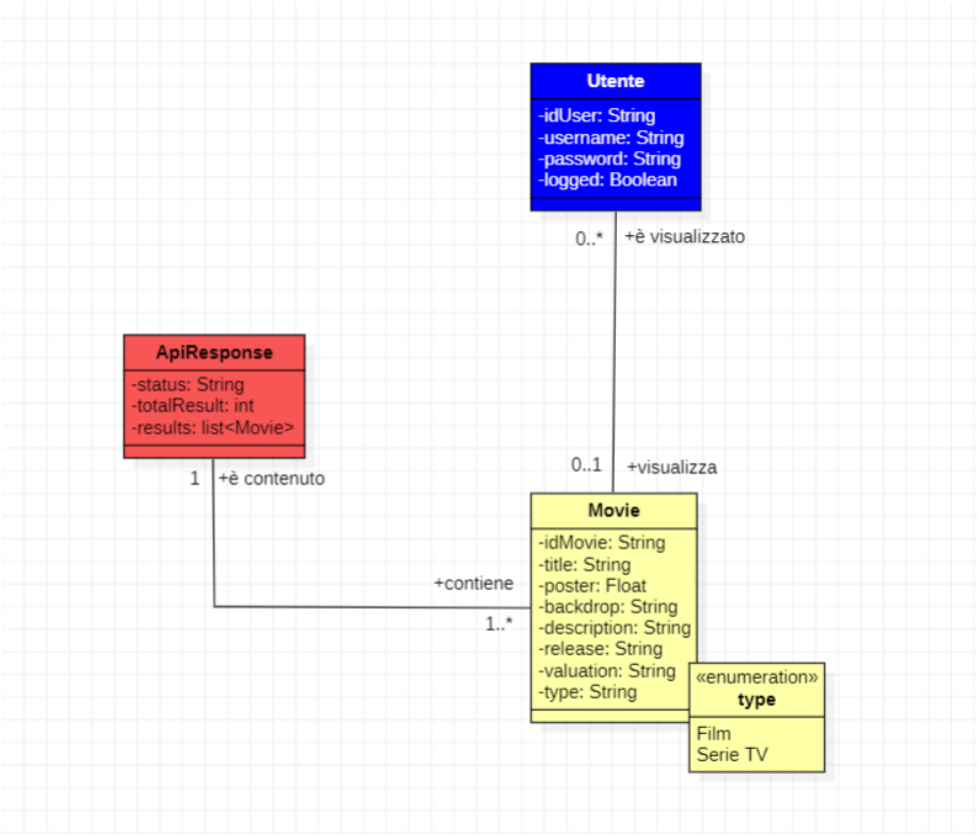

Figura 5: ***Class Diagram***

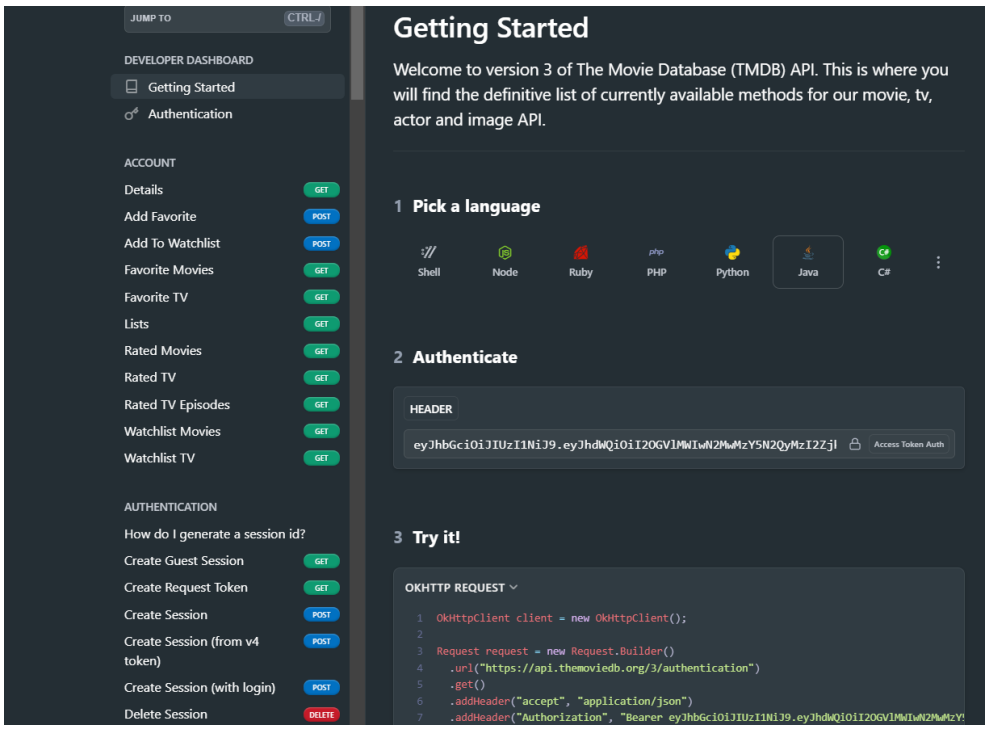

Figura 6: ***TMDB***

Client Android

Il paradigma **Model-View-Controller** è stato adottato per separare la parte grafica dall'interazione con il database e dalla logica dell'applicazione. Ecco una spiegazione più dettagliata:

- Layout (**View**): è la directory che contiene tutte le interfacce grafiche dell'applicazione, insieme a risorse come drawable, values, colors e dimen.

- Modello (**Model**): rappresenta la struttura dei dati dell'applicazione e risponde alle richieste delle View per informazioni sui dati stessi. Gestisce la logica dell'applicazione e lo stato dei dati.

- Controllore (**Controller**): funge da intermediario tra il Model e la View. Il suo compito principale è gestire le interazioni dell'utente e aggiornare le View di conseguenza. Interpreta le azioni dell'utente e interagisce con il Model per eseguire le operazioni richieste.

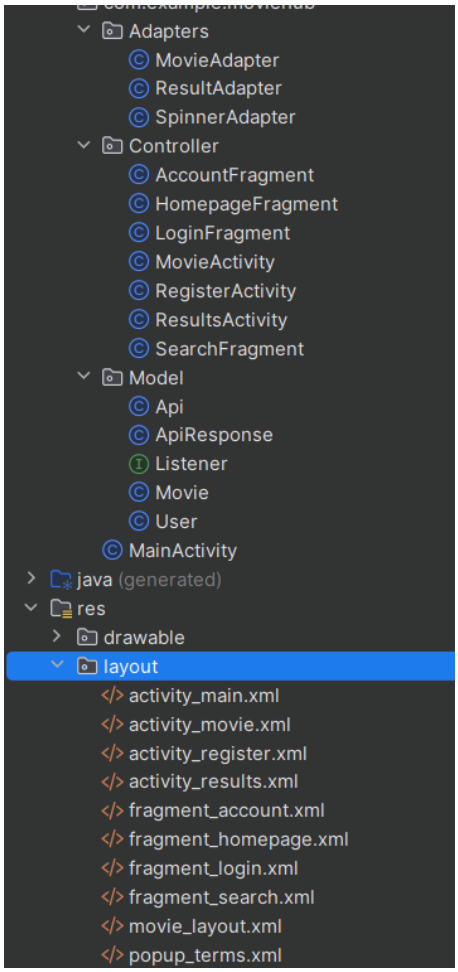

Figura 7: ***File del progetto***

Utilizzando il paradigma **MVC**, l'applicazione è suddivisa in tre componenti distinte che consentono una migliore organizzazione del codice, facilitando la manutenzione e il testing dell'applicazione. La separazione dei ruoli tra Model, View e Controller favorisce una maggiore **modularità** e **scalabilità** del progetto.

3. USE CASE MOVIEHUB

Homepage

All'avvio dell'applicazione, gli utenti vengono accolti da una homepage dinamica e ricca di contenuti. La pagina principale visualizza tre liste raccomandate da TMDB:

1. **Film Consigliati**: Una selezione accurata di film consigliati, basata sui gusti e sulle preferenze dell'utente.
2. **Prossime Uscite**: Un elenco di tutte le nuove uscite cinematografiche e televisive, che consente agli utenti di restare aggiornati sulle novità.
3. **Serie TV Consigliate**: Una vasta selezione delle Serie tv più in voga del momento.

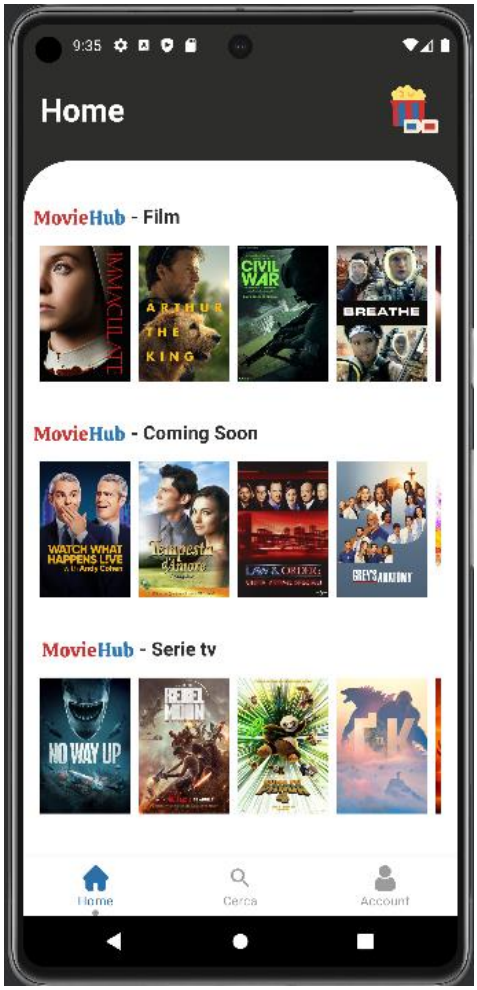

Figura 8: ***Homepage dell'applicazione***

Login e Registrazione

Uno tra gli aspetti fondamentali 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 che utilizza i soli bianco e nero per gli utenti affetti da acromatopsia.
- **Deuteranopia**: Filtro che modifica il verde per gli utenti con deuteranopia.
- **Tritanopia**: Filtro che modifica il blu per gli utenti con caratterizzati dall' assenza della visione del blu.
- **Ipovisione**: Incremento delle dimensioni dei widget per gli utenti con insufficiente capacità discriminativa.
- **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.

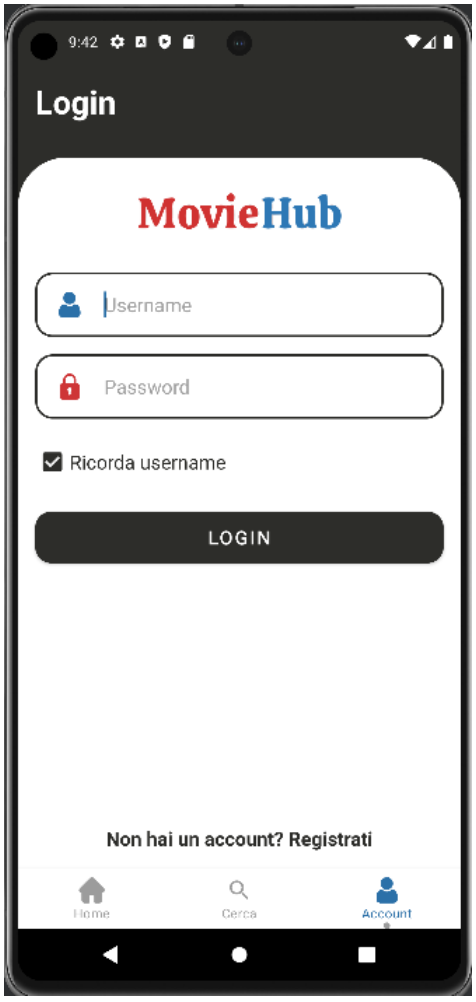

*Figura 9: **Login UI***

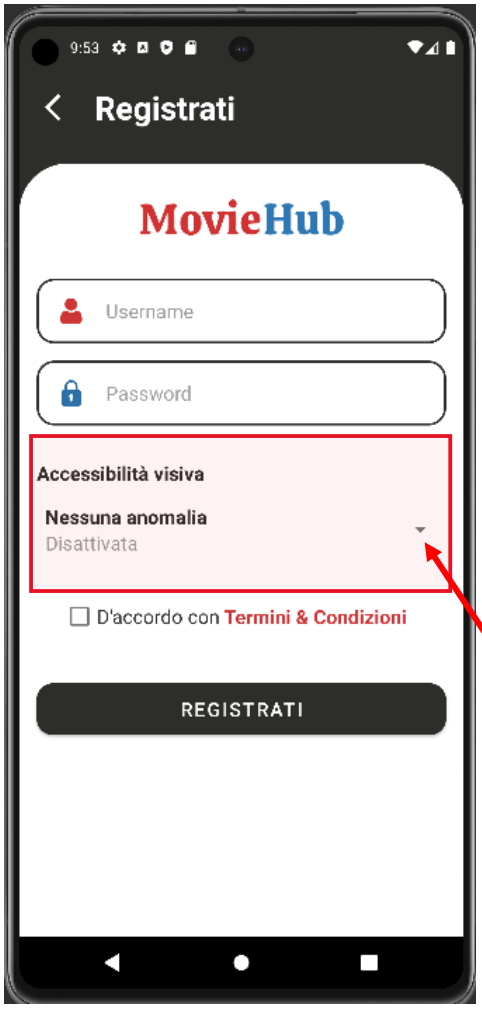

Figura 10: ***Registrazione UI***

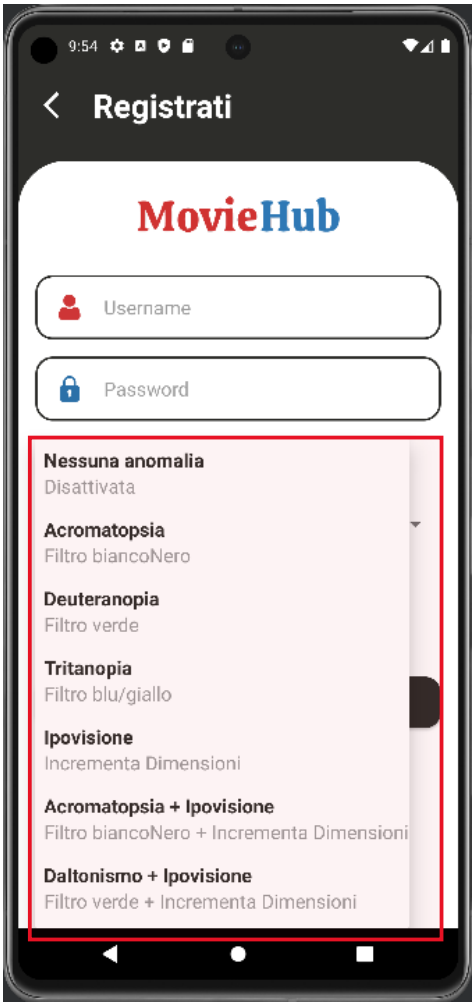

Figura 11: ***Spinner Accessibilità aperto***

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.

1

Figura 12: ***Ricerca film o serie tv***

2

Figura 13: ***Inserimento film o serie tv***

3

Figura 13: ***Risultati Visualizzati***

4

Figura 14: ***Descrizione dettagliata del film***