StreamDream

26/02/2023 Relazione di Maria Roggio ed Elio Vinciguerra

Sommario

Capitolo 1: Introduzione

StreamDream permette di tener traccia dei contenuti multimediali selezionati o cercati dall'utente, esso si basa su un'architettura client e server, in cui il client è scritto in linguaggio C# e Python e il server è scritto interamente in linguaggio GoLang. I dati che il server dovrà immagazzinare saranno memorizzati in un database MongoDB.
L'utente, accedendo tramite le proprie credenziali previa registrazione, potrà utilizzare il servizio. La home renderà visibile all'utente i Top 9 delle serie tv e dei film, estratti dalle API di IMDB. L'utente, nella stessa schermata può cercare contenuti o, tramite il bottone che appare sotto il poster del contenuto, può ottenere informazioni riguardo allo stesso. In quest'ultima finestra l'utente ha la possibilità di condividere opinioni riguardo al contenuto stesso con altri utenti.
L'utente ha anche la possibilità di visualizzare le informazioni relative al proprio account e modificare i dati personali (nominativo, password o email), oltre alla possibilità di cancellare il proprio account.
Infine, l'utente ha la possibilità di visualizzare e gestire il proprio catalogo personale, ovvero l'insieme di contenuti multimediali che esso ha aggiunto personalmente.

In generale i linguaggi C# e Python comunicano tramite IronPython, invece Python e il server (scritto in GoLang) comunicano tramite chiamate Rest API.

Per quanto concerne la suddivisione del lavoro svolto, Roggio si è occupata della componente C#, Vinciguerra della componente Python, il server Go è stato scritto da entrambe le parti.

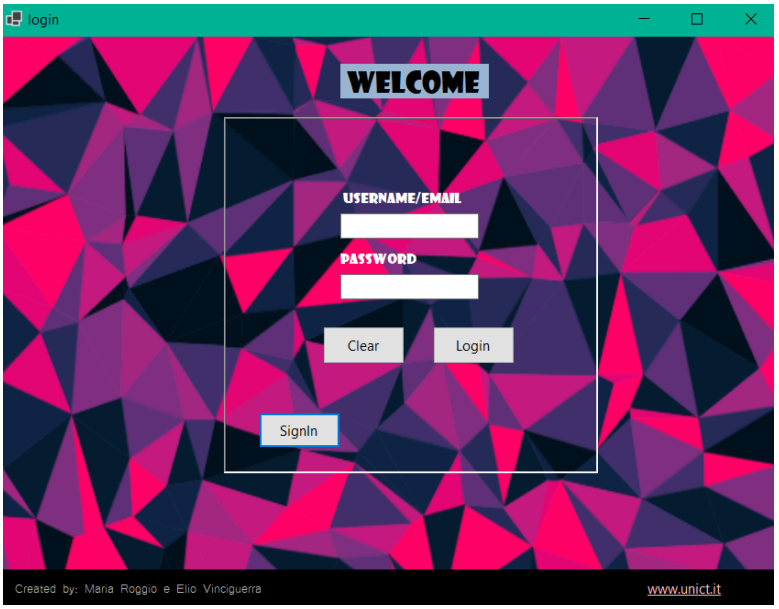
Il progetto e la guida all'installazione sono reperibili sulla repository github https://github.com/eliovinciguerra/portfolio e nell'apposito README.

Capitolo 2: Registrazione e Login

Nel capitolo che segue verrà approfondita la realizzazione delle sezioni relative alla registrazione e al login dell'utente.

Login

In questa sezione viene realizzata l'autenticazione di un utente al servizio.
L'utente deve digitare le proprie credenziali, ovvero lo username o la mail utilizzata nell'atto della registrazione e immettere la password. Se l'utente è sprovvisto di un account è possibile generarne uno tramite apposito form Registrazione, di cui parleremo in seguito.
Di fianco la schermata grafica di Login che verrà restituita all'utente. Nella sezione footer è presente un link di reindirizzamento alla webpage dell'ateneo dell'Università degli Studi di Catania. L'interfaccia grafica è stata realizzata in linguaggio C#.



Per quanto riguarda la logica di tale sezione, essa viene realizzata in linguaggio Python. Dalla componente C# vengono inviate le informazioni di accesso ad uno script Python, esso avvierà una request tramite API Rest al server, che, dopo aver effettuato i dovuti controlli, risponderà con una response. Tale risposta viene gestita; qualora l'esito sia positivo, viene restituito lo username alla componente C#, che permetterà di accedere alla schermata Home, altrimenti verranno cancellati i campi e verrà visualizzato un testo di errore sopra il campo "username/email".

Il server che riceve una richiesta sull'URI "/login" catturerà i relativi parametri e li manderà ad una funzione apposita svolta da una GoRoutine, tale funzione si occuperà dei controlli relativi all'esistenza della mail immessa dallo user nel client all'interno del database MongoDB e della coincidenza della stessa con la password.

Registrazione

In questa sezione viene realizzata la registrazione dell'utente al servizio.
L'utente dovrà digitare le proprie informazioni, tra cui nome, cognome, email, username e password. In caso l'utente voglia generare uno username casuale, esso sarà generato dall'apposito bottone posto di fianco al campo username. Una volta riempiti tutti i campi

l'utente può proseguire cliccando il bottone di submit. Verranno effettuati controlli sulla coincidenza della password col proprio campo di conferma, inoltre all'atto del click sul bottone submit viene controllato che nessun campo sia vuoto. La componente grafica è stata realizzata interamente in C#. E, come nel form precedente, è presente un link di reindirizzamento alla webpage di ateneo.

Anche in questo caso, la componente logica della sezione è stata realizzata in Python. Dalla componente C# vengono inviate le informazioni di registrazione ad uno script Python, esso avvierà una request tramite API Rest al server, che, dopo aver effettuato i dovuti controlli e la registrazione nel database MongoDB, risponderà con una response. Tale risposta viene gestita; qualora l'esito sia positivo, viene restituito lo username alla componente C#, che permetterà di accedere alla schermata Home, altrimenti verranno cancellati i campi e verrà visualizzato un testo di errore sopra il campo "name". Per quanto concerne lo username randomico, all'atto del click sulla componente grafica, essa richiamerà una funzione Python che prenderà il nome, il primo carattere del cognome e aggiungerà tre numeri randomici, successivamente lo username ricavato sarà inviato al server per controllare che non esista già un utente registrato con tale username.

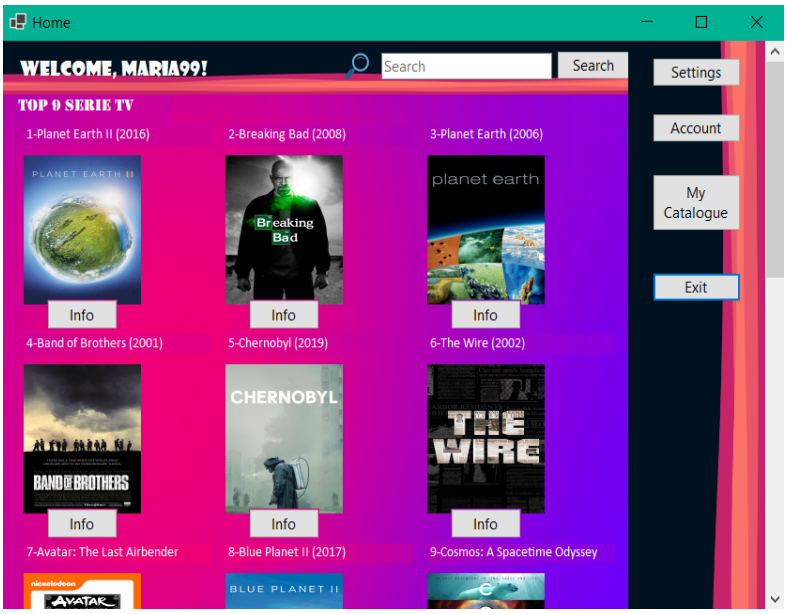
Il server che riceve una richiesta sull'URI "/signin" catturerà i relativi parametri e li manderà ad una funzione apposita svolta da una GoRoutine, tale funzione si occuperà dei controlli relativi all'assenza della mail e dello username immessi dall'utente nel client all'interno del database MongoDB oltre la validità della mail stessa. Verrà inoltre mandata una mail di avvenuta iscrizione al servizio dall'indirizzo progettoapl2022@gmail.com all'email inserita dall'utente.

Capitolo 3: Home e SearchContent

Nel capitolo che segue verrà approfondita la realizzazione della sezione relativa alla schermata home visualizzata dall'utente dopo aver effettuato l'accesso o la registrazione al servizio, verrà inoltre approfondita la realizzazione della componente di ricerca contenuti.

Home

Graficamente, in tale sezione, nella barra superiore viene visualizzato un messaggio di benvenuto e lo username dell'utente, è inoltre presente una barra di ricerca contenuti. Per quanto concerne la barra laterale, essa permette di orientarsi all'interno del servizio, al click sul bottone l'utente verrà reindirizzato all'apposita sezione.



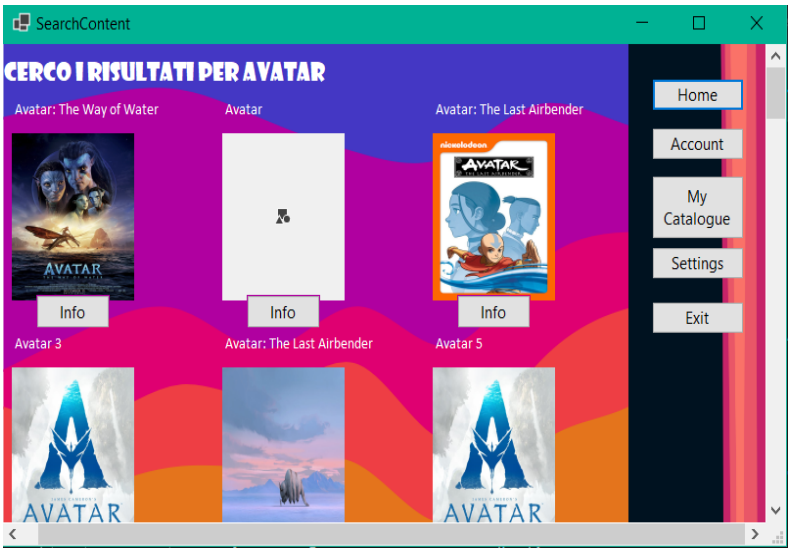
Il cuore di questa schermata riguarda la visualizzazione dei contenuti in vetta alle classifiche secondo IMDB, sia per i film che per le serie tv verranno mostrati i primi 9 contenuti (i componenti riguardanti titolo, immagine locandina, e bottone informazioni, inizialmente non esistono, vengono generati in base al numero di elementi che vogliamo mostrare, permettendo una facile scalabilità nel caso si volessero mostrare più o meno contenuti). L'utente cliccando l'apposito bottone "Info" avrà accesso alle informazioni del contenuto selezionato. Tale componente è interamente realizzato in C#.

Per quanto concerne la logica della sezione, essa è stata realizzata in Python. La componente C# invocherà due funzioni Python, una relativa ai contenuti in vetta alle classifiche di categoria serie tv, la seconda per i film. Gli elementi che verranno estratti dalla richiesta API Rest al server IMDB riguardano i dati del contenuto, come l'identificatore, il titolo, il link dell'immagine e il rank.
Per quanto riguarda il nostro progetto abbiamo preferito rendere visibile solo i primi 9 elementi per ogni categoria, ma è possibile estendere fino a 250 elementi.

SearchContent

Questa componente permette all'utente di cercare contenuti, egli deve inserire il nome del contenuto ricercato nell'apposito campo della schermata Home, successivamente verrà mostrata una finestra che raccoglie tutti i contenuti che riguardano tale nominativo. Graficamente questa sezione del servizio è molto simile alla Home. In seguito ne viene mostrata una illustrazione.

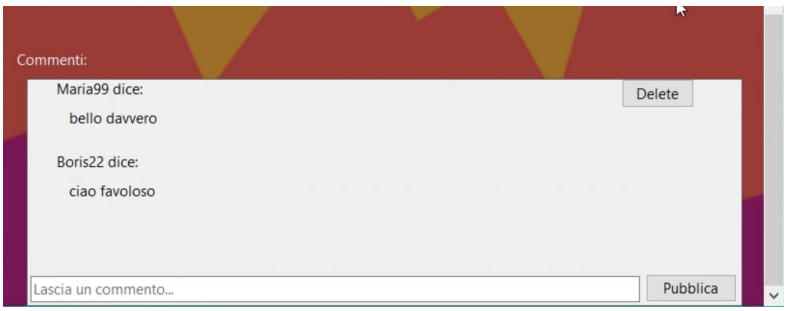
Per quanto concerne la logica implementativa, essa è stata realizzata interamente in Python, dalla componente grafica (scritta in C#) viene chiamata una funzione Python che si occupa di fare una chiamata API Rest al server IMDB per ottenere i risultati compatibili con ciò che l'utente ha immesso. Successivamente l'insieme di tali risultati e i relativi dati (titolo contenuto, locandina e identificativo) vengono restituiti alla componente grafica.



Capitolo 4: InfoContent

Nel capitolo che segue verrà approfondita la realizzazione della sezione relativa alla schermata informativa del singolo contenuto visualizzata all'utente dopo aver cliccato sull'apposito bottone nella sezione Home o nella sezione SearchContent.

Graficamente, in tale sezione, viene visualizzata la locandina del contenuto, il nome, la descrizione e a lato una tabella con tutti i dati dei ratings (punteggi da 1 a 10, con relativo numero di utenti che hanno votato tale punteggio), oltre ad un bottone che permette all'utente di salvare contenuti nel proprio catalogo personale.
Successivamente vengono visualizzati i primi 12 attori facenti parte del cast.
Ultima componente di tale sezione riguarda i commenti, ogni utente può lasciare un feedback sul contenuto, che viene visualizzato come mostrato a lato. L'utente che ha creato il commento ha anche la possibilità di cancellare tale commento, tramite apposito tasto "Delete". Tale componente grafica viene realizzata in linguaggio C#.



La componente logica è realizzata interamente in linguaggio Python, alla creazione del form C# vengono lanciate alcune funzioni Python, tra cui:

- Funzione che si occupa del recupero della trama da IMDB, tramite apposita richiesta API Rest;
- Funzione che si occupa del recupero del cast da IMDB, tramite apposita richiesta API Rest;
- Funzione che si occupa del recupero dei ratings da IMDB, tramite apposita richiesta API Rest;
- Funzione che si occupa del recupero dei commenti dal database MongoDB, tramite apposita richiesta API Rest.

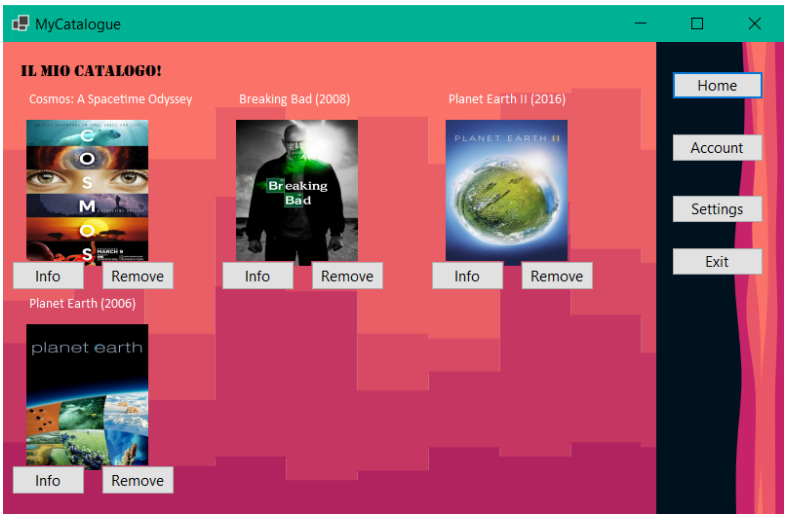
Quando l'utente desidera cancellare un commento, cliccando l'apposito bottone, viene invocata una funzione Python che richiede la cancellazione del suddetto commento al server. Stesso funzionamento per quanto riguarda l'aggiunta di un commento al contenuto o l'aggiunta di un contenuto al proprio catalogo personale.

Per quanto concerne il server, da questa sezione potrebbero arrivare diverse richieste; se ad esempio l'utente decide di eliminare un commento (URI "/deleteComment"), al server arriva l'identificativo del commento in questione, che verrà poi eliminato. L'utente potrebbe anche decidere di aggiungere un nuovo commento, (URI "/addComment"). Inoltre, se l'utente decidesse di aggiungere tale contenuto al proprio catalogo (URI "/addContent"), il server si occuperà di controllare se tale contenuto è già stato aggiunto al catalogo personale del dato utente, se non è presente, verrà memorizzato, altrimenti verrà restituito un messaggio di errore.

Capitolo 5: MyCatalogue

Nel capitolo che segue verrà approfondita la realizzazione della sezione relativa al catalogo personale dell'utente loggato.

Graficamente, in tale sezione, viene visualizzata la lista di elementi relativi al proprio catalogo, quindi per ogni contenuto viene visualizzato il titolo, la locandina e due bottoni, uno per visualizzare le informazioni (riporta alla sezione analizzata nel precedente capitolo) e uno per eliminare tale contenuto dal catalogo.
Tale componente grafica viene realizzata in linguaggio C#.



La componente logica è realizzata in linguaggio Python, quando l'utente decide di consultare il catalogo, viene invocata una funzione Python che richiede al server la lista di contenuti da visualizzare. Inoltre, se l'utente decide di eliminare un contenuto dal proprio catalogo, verrà invocata una funzione Python apposita che richiederà al server la cancellazione del contenuto in questione.

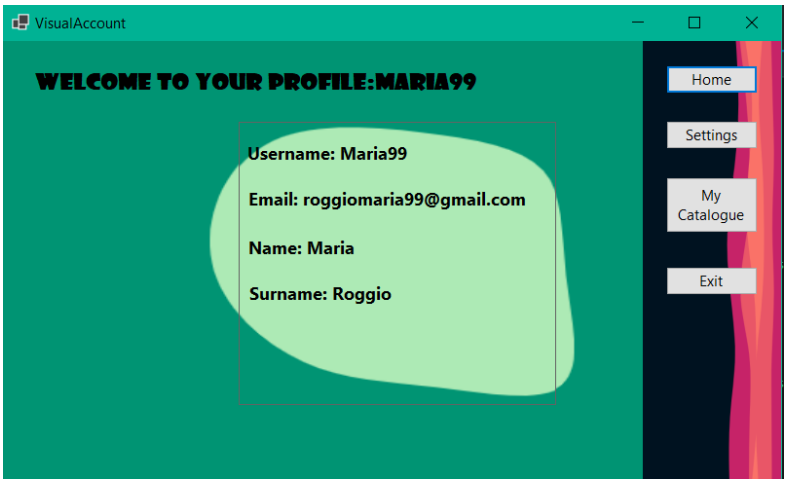
Per quanto concerne il server, esso potrà ricevere due tipologie di richieste da questa sezione, la prima (URI "/getCatalogo") che si occupa di restituire per intero il catalogo dell'utente dato, la seconda (URI "/deleteContentCat") che invece elimina il contenuto dato dal catalogo dell'utente in questione.

Capitolo 5: Account e Settings

Nel capitolo che segue verrà approfondita la realizzazione delle sezioni relative alle informazioni personali dell'utente e la gestione dell'account.

Account

Graficamente, in tale sezione, vengono visualizzate le informazioni relative all'utente loggato, come mostrato di fianco.



La componente logica, scritta in Python, si occupa di richiedere le informazioni dell'utente in questione al server tramite API Rest.

Il server riceve la richiesta (URI "/infoUser") e restituisce tutte le informazioni relative allo user.

Settings

Graficamente, in tale sezione, coesistono due parti, una prima parte per gestire la cancellazione del proprio account (Profilo Pubblico), una volta immessa la password e cliccato il bottone "Delete" l'account verrà cancellato. Nella parte del profilo privato vengono invece gestite le informazioni generali, quindi la variazione della mail, il cambio di nome o cognome inserito, oppure della password stessa.

La componente logica, scritta in Python, si occupa di effettuare richieste al server in base all'azione intrapresa; se viene cliccato il bottone "Delete" nel Profilo Pubblico verrà fatta una richiesta al server per tale meccanismo di cancellazione, lo stesso meccanismo viene intrapreso per il cambio email (bottone "Salva email"), per il cambio di Nome e/o Cognome registrato (bottone "Salva") e per la variazione della password (bottone "Conferma").

Per quanto concerne il server, esso potrà ricevere quattro tipologie di richieste da questa sezione, la prima (URI “/deleteAccount”) che si occupa di eliminare account, commenti e catalogo di un determinato utente, la seconda (URI “/changeEmail”) che controlla se l’email inserita risulta valida e successivamente la cambia, per poi mandare una mail di avvenuto aggiornamento della mail dall’indirizzo progettoapl2022@gmail.com all’email inserita dall’utente, la terza (URI “/changeNameSurname”) che aggiorna il nome e/o il cognome dell’utente con quelli ricevuti, se manca il nome viene aggiornato il solo cognome e viceversa, altrimenti vengono aggiornati entrambi, e la quarta (URI “/changePW”) che invece aggiorna la password vecchia con la nuova.

Capitolo 6: Altri elementi

In ogni schermata discussa precedentemente viene inserita una barra laterale, che permette la navigazione tra le diverse sezioni del programma, discusse precedentemente.

Quando l'utente clicca su "Exit" verrà effettuato il logout e verrà chiusa l'applicazione.