

# Progetto Programmazione Avanzata di Iacopo Massei

## Indice

<b>Idea iniziale</b> .....	<b>1</b>
<b>Database</b> .....	<b>1</b>
<b>Applicazione</b> .....	<b>2</b>
<i>Schermata di login</i> .....	2
<i>Registrazione</i> .....	2
<i>HomePage</i> .....	3
<i>Entrate</i> .....	4
<i>Uscite</i> .....	5
<i>Dialogo con il Server</i> .....	6
<b>Scelte progettuali</b> .....	<b>6</b>
<i>Formato data</i> .....	6
<i>Luogo</i> .....	6
<i>Unit test</i> .....	7
<b>Server</b> .....	<b>7</b>
<i>Dialogo DB</i> .....	7
<i>Unit test</i> .....	7

## Idea iniziale

Il progetto **BudgetManager** è un'applicazione Client-Server che permette la gestione del portafoglio di arbitri di pallacanestro. L'utilizzo è dunque limitato solo a quest'ultimi, in quanto l'idea nasce dalla necessità di un gestore di budget del sottoscritto.

## Database

Il database viene inizializzato tramite l'apposito tasto "**Download**" presente nella schermata di apertura (Login). In particolare, viene letto il file "**buildSQL**" riga per riga ed eseguite le query necessarie. Al fine di evitare ogni volta di perdere dati inseriti durante l'uso dell'applicazione (ad esempio un nuovo utente con delle gare e delle spese), la funzione "download" controlla, per prima cosa, se esistono già delle tabelle all'interno del database e, in particolare, se esiste almeno

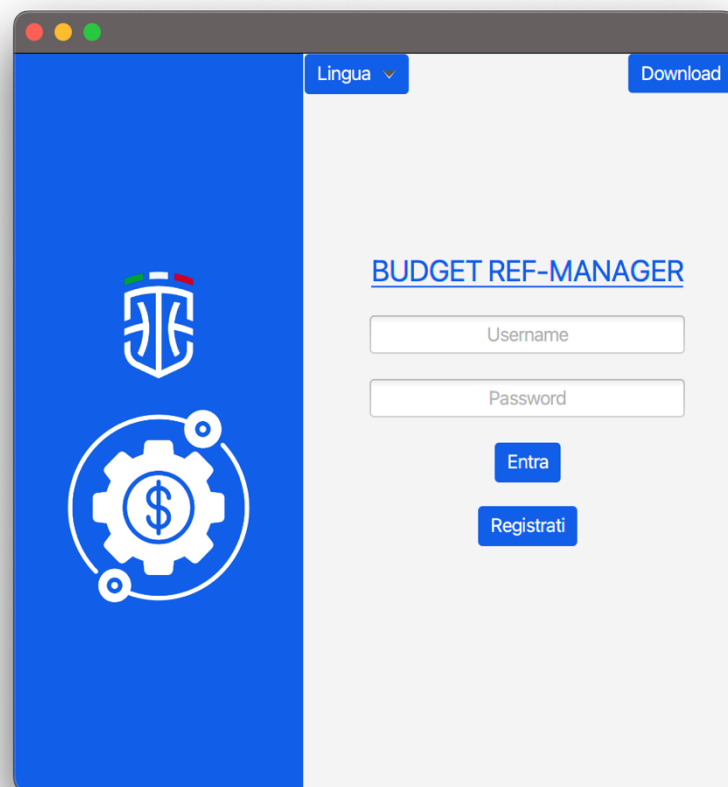
un utente. In caso affermativo non viene svuotato il database e si mantengono i dati già presenti. Il client si connette al database con i parametri:

IP -> localhost  
Port -> 3306  
Database -> 603217  
Username -> root  
Password -> root

## Applicazione

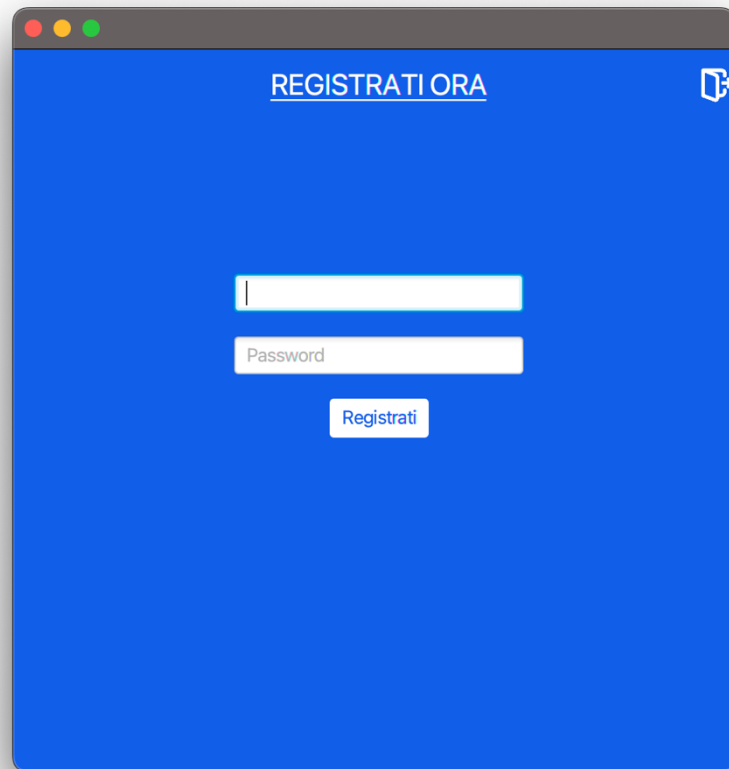
### Schermata di login

All'avvio dell'applicazione si apre la pagina di login dove si può scegliere la lingua desiderata (quella di default è l'italiano), entrare con un username ed una password, oppure andare alla pagina di registrazione. Se l'username o la password sono errati si presenta un messaggio di errore.



### Registrazione

Nella pagina di registrazione si può registrare un nuovo utente tramite username e password. L'applicazione controllerà che non esistano già utenti con l'username scelto: in caso positivo restituirà un messaggio di errore. Se la registrazione avverrà con successo, comparirà una scritta di benvenuto con un link per tornare alla pagina di login.

A screenshot of a web browser window displaying a registration form. The window has a blue background. At the top, the text "REGISTRATI ORA" is displayed in white, underlined, and centered. To the right of this text is a small white icon of a document with an arrow pointing out. Below the header, there are two white input fields. The first field is for the username, with a cursor visible inside. The second field is for the password, with the placeholder text "Password" visible. Below these fields is a white button with the text "Registrati" in blue. The browser window has a standard macOS-style title bar with red, yellow, and green window control buttons.

## HomePage

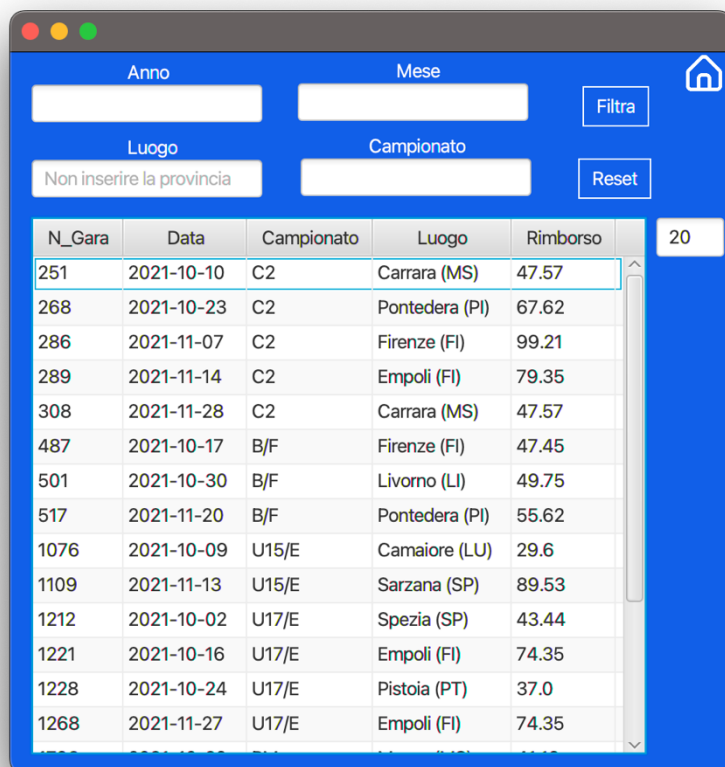
Quando un utente riesce ad accedere si ritrova nella propria homepage. All'interno di questa homepage si possono visualizzare le entrate e le uscite del mese corrente, insieme al budget rimasto. L'utente potrà visualizzare i budget mensili di un intero anno solare. In alto, in mezzo alla

pagina, oltre all'username dell'utente comparirà anche il nome del mese con il relativo anno associato. L'utente tramite questa schermata potrà anche aggiungere una gara o una spesa. Una gara è l'unica fonte di guadagno che si può inserire, in quanto l'applicazione, come detto in precedenza, è pensata solo per arbitri di pallacanestro che guadagnano arbitrando. Oltre a poter cambiare la visione mensile tramite l'apposito menù in alto a sx, l'utente può anche andare a vedere tutte le sue entrate ed uscite, che sono all'interno del database, tramite il menù **"Naviga"**.

The screenshot shows a web application interface for managing basketball arbitrage. The interface is titled "Luglio 2023" and displays the username "iacopo.massei". The main content area is divided into two columns: "Entrate" (Income) and "Uscite" (Expenses). Both sections show a value of "0.0". Below the "Entrate" section, there is a button "Inserisci una nuova gara" and a form with fields for "Numero gara", "Data (yyyy-MM-dd)", "Comune (Provincia)", "Rimborso", and a "Campionato" dropdown menu. A button "Invia gara" is located below the form. Below the "Uscite" section, there is a button "Inserisci una nuova spesa" and a form with fields for "Data (yyyy-MM-dd)", "Causale", and "Costo". A button "Invia spesa" is located below the form. At the bottom center, there is a "Budget" section showing a value of "0.0".

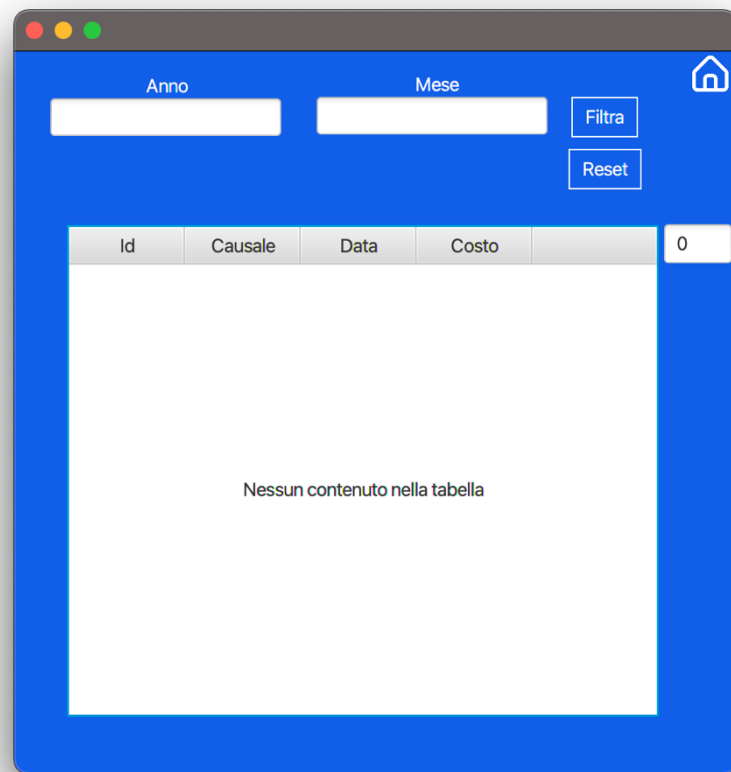
Entrate

In questa pagina si possono visualizzare tutte le gare fatte negli anni dall'utente. È possibile filtrare le gare per anno, mese, luogo e campionato: una volta inseriti i parametri di filtraggio desiderati basta premere il pulsante **"Filtra"** per visualizzare una nuova tabella con i valori filtrati; per tornare invece alla condizione iniziale, o per pulire i parametri di filtraggio, basta premere il tasto **"Reset"**. Accanto alla tabella si trova anche un contatore che ci indica quante gare esistono in base ai parametri di filtraggio.



N_Gara	Data	Campionato	Luogo	Rimborso
251	2021-10-10	C2	Carrara (MS)	47.57
268	2021-10-23	C2	Pontedera (PI)	67.62
286	2021-11-07	C2	Firenze (FI)	99.21
289	2021-11-14	C2	Empoli (FI)	79.35
308	2021-11-28	C2	Carrara (MS)	47.57
487	2021-10-17	B/F	Firenze (FI)	47.45
501	2021-10-30	B/F	Livorno (LI)	49.75
517	2021-11-20	B/F	Pontedera (PI)	55.62
1076	2021-10-09	U15/E	Camaione (LU)	29.6
1109	2021-11-13	U15/E	Sarzana (SP)	89.53
1212	2021-10-02	U17/E	Spezia (SP)	43.44
1221	2021-10-16	U17/E	Empoli (FI)	74.35
1228	2021-10-24	U17/E	Pistoia (PT)	37.0
1268	2021-11-27	U17/E	Empoli (FI)	74.35

Pagina molto simile a quella delle entrate. Qui l'utente può visualizzare tutte le proprie uscite e filtrarle per anno e mese. I pulsanti di filtraggio e di reset sono gli stessi della pagina "**Entrate**".



The screenshot shows a web application window with a blue header and a white body. The header contains two input fields labeled "Anno" and "Mese", a "Filtro" button, and a "Reset" button. A home icon is in the top right corner. Below the header is a table with columns "Id", "Causale", "Data", and "Costo". The table is currently empty, displaying the message "Nessun contenuto nella tabella". A small input field with the value "0" is located to the right of the table header.

## Dialogo con il Server

Il client comunica con il server con richieste HTTP, utilizzando la classe **URLConnection**. Le richieste e le risposte sono codificate in formato Json. All'arrivo della risposta il client legge il file e va ad analizzare il corpo della risposta.

## Scelte progettuali

### Formato data

Il formato delle date che è stato scelto è il seguente: **yyyy-MM-dd**. Qualsiasi altro formato non viene accettato e viene segnalato all'utente che il formato usato è errato.

## Luogo

Per inserire un luogo si è scelto il formato comune: **Comune (Provincia)**. Nel filtraggio, nella pagina **“Entrate”**, si può cercare solo utilizzando il formato **“Comune”**, senza la provincia.

## Unit test

Il client testa la connessione al database, la quale fallisce sicuramente se il server è spento.

## Server

Il server non può avviarsi se non esiste ancora il database.

Vengono utilizzati due controllori, **“HomeController”** e **“UserController”**, che permettono di gestire rispettivamente l’aggiunta di una nuova spesa/gara ed il login/registrazione di un utente.

## Dialogo DB

Per generare le risposte il server utilizza principalmente i metodi implementati dalla classe **CrudRepository**.

## Unit test

Il server testa la connessione al database, la quale fallisce se il server è spento.