Università degli Studi di Bari – Dipartimento di Informatica *[ITPS A-L]*

Caso di studio 4

SERATE MUSICALI

AUTORI:

Cellammare Gabriel (g.cellammare1@studenti.uniba.it)

Ingusci Paolo (p.ingusci@studenti.uniba.it)

SAL

25/05/2021: [ANALISI] Aggiornare la documentazione modificando l'analisi, i requisiti funzionali e i casi d'uso.

27/05/2021: [ANALISI COMPLETATA]

3/06/2021: Terminare la progettazione e iniziare i casi di test.

10/06/2021: Terminare le conclusioni.

21/06/2021: Consegna del caso di studio 4.

Sommario

Analisi

Il caso di studio in questione, si occupa della progettazione di un software in grado di **organizzare delle serate musicali.**

Precisamente dovremo gestire la vendita di biglietti, la gestione di ogni evento e di ogni persona registrata.

Secondo il nostro ragionamento, il software in questione dovrà essere utilizzato dai proprietari dei locali, quindi tutte l'operazione di registrazione "Utente", verranno effettuate da loro. (E ovviamente, saranno comprese tutte le altre funzioni richieste nella traccia).

La struttura del software, ci ha fatto subito pensare ad un database gestito ***in linguaggio SQL***.

Infatti, i dati saranno gestiti da tre macro-strutture di tipo ***struct***, che conterranno le informazioni dell'utente, dell'evento e della prenotazione. Saranno tutte collegate tra loro, poiché tutte le struct avranno delle chiavi primarie ed esterne che permetteranno il collegamento tra loro.

Inoltre, durante l'avvio del software viene controllata l'esistenza dei file .csv, che serviranno per tener traccia di tutte le operazioni. Se non vengono rilevati, il software avviserà l'utente e saranno creati automaticamente in base alle scelte di quest'ultimo.

Le funzioni da noi create rispondono correttamente a quelle richieste.

Inoltre, è stata implementata una funzione aggiuntiva che durante la registrazione dell'evento, controlla se esiste un evento simile

(ricercaEvento(int,int,char[],int,char[],char[]).

In questo modo, vengono evitate le registrazioni di eventi uguali, poiché un'artista non può esibirsi lo stesso giorno in due luoghi differenti, oppure non possono esistere due eventi con lo stesso *ID* (***CODICE UNIVOCO***). Durante la stampa dei prenotati ad un certo evento, vengono stampate anche le poltrone prenotate da loro e la fila corrispondente, in modo tale da avere una visione più ampia e precisa

delle prenotazioni. È stata creata la procedura “Istogramma()”, per visualizzare graficamente il numero dei biglietti venduti per ogni evento.

Requisiti funzionali

Codice	***Nome***	***Descrizione***
R00	*Visualizzazione menu*	Il programma consente di visualizzare il menu delle scelte, e prende in input un numero che decreterà la scelta.
R01	*Registrazione utente*	Il programma acquisirà tutti i dati dell’utente da registrare, e li trasferirà nel file “Utenti.csv”
R02	*Modifica dati utente*	Il programma permette all’utente di modificare alcuni parametri di registrazione, tranne il nickname.
R03	*Registra evento*	Il programma permette di registrare un evento attraverso alcuni parametri forniti in input, e di scrivere tutto all’interno del file “Evento.csv”.
R04	*Cerca evento in base alla data o al nome dell’artista.*	Il programma permette di ricercare un evento attraverso alcuni parametri dati in input, e permette di modificare alcuni parametri, o di eliminarlo completamente dal file.
R05	*Stampa dei partecipanti di un dato evento*	Il programma permette di visualizzare a video, la lista dei partecipanti ad un evento fornito in input.
R06	*Stampa la lista degli eventi a cui ha partecipato un utente, in ordine di data o costo.*	Il programma permette di stampare la lista degli eventi a cui ha partecipato un utente, dando la possibilità di scegliere se stampare il tutto, in ordine cronologico, o in base al costo dei biglietti comprati.
R07	*Vende un biglietto ad un utente*	Il programma permette di vendere un biglietto di un evento selezionato, ad un utente.
R08	*Verifica la disponibilità di una poltrona*	Il programma verifica se la poltrona di una certa fila di un evento, è disponibile o meno.

R09	*Stampa l'incasso complessivo di un evento*	Il programma permette di stampare l'incasso complessivo di un evento.
R10	*Stampa i riconoscimenti del software*	Il programma permette di stampare tutti i riconoscimenti del software.

Tabella 1. Requisiti funzionali.

Casi d'uso

Codice	***Nome***	***Descrizione***
CU00	Menù iniziale	Il programma acquisirà da tastiera la scelta dell'utente.
Pre-condizioni	L'utente deve schiacciare un tasto qualsiasi dopo la comparsa del menù di presentazione.	
Post-condizioni	*Successo*	Il sistema richiama la procedura corrispondente al numero selezionato dall'utente.
	Fallimento	● Il sistema mostra un messaggio di errore; ● Si viene reindirizzati alla stampa del menù
Scenario di base	Il sistema richiama correttamente la procedura selezionata dall'utente.	
Scenario alternativo	Il sistema comunica l'errore, e reindirizza al menu.	

Codice	***Nome***	***Descrizione***
CU01	Registrazione utente	Il programma acquisirà tutti i dati dell'utente da

		registrare e li trasferirà nel file “Utenti.csv”
Pre-condizioni	Il nickname dell’utente non deve essere già presente all’interno del file “Utenti.csv”.	
Post-condizioni	*Successo*	Il sistema registra l’utente attraverso l’inserimento da tastiera di tutti parametri richiesti.
	Fallimento	● Il sistema mostra un messaggio di errore; ● Si viene reindirizzati al menu principale.
Scenario di base	Il sistema registra correttamente il profilo utente, se il nickname non è già registrato.	
Scenario alternativo	Il sistema comunica l’errore, e reindirizza al menu principale.	

Tabella 2. Caso d’uso del requisito C02.

Codice	**Nome**	**Descrizione**
CU02	Modifica dati utente	Il programma permette all’utente di modificare alcuni parametri di registrazione, tranne il nickname.
Pre-condizioni	Il nickname dell’utente deve essere già presente all’interno del file “Utenti.csv”.	
Post-condizioni	*Successo*	Il sistema riconosce l’utente dal nickname, e permette la modifica di alcuni dati.
	Fallimento	● Il sistema mostra un messaggio di errore; ● Si viene reindirizzati al menu principale.

Scenario di base	Il sistema modifica correttamente il profilo utente, se il nickname è già registrato.
Scenario alternativo	Il sistema comunica l'errore, e reindirizza al menu principale.

Codice	***Nome***	***Descrizione***
CU03	Registra evento	Il programma permette di registrare un evento attraverso alcuni parametri forniti in input, e di scrivere tutto all'interno del file "Evento.csv".
Pre-condizioni	L'id dell'evento non deve essere già presente all'interno del file "Evento.csv", e inoltre il sistema, controlla che nello stesso locale, non coesistano due eventi simili.	
Post-condizioni	*Successo*	Il sistema registra l'evento attraverso l'inserimento da tastiera di tutti parametri richiesti.
	Fallimento	● Il sistema mostra un messaggio di errore; ● Si viene reindirizzati al menu principale.
Scenario di base	Il sistema registra correttamente l'evento, se l'id utente non è già registrato.	
Scenario alternativo	Il sistema comunica l'errore, e reindirizza al menu principale.	

Codice	***Nome***	***Descrizione***
CU04	Cerca evento in base alla data o al nome dell'artista.	Il programma permette di ricercare un evento attraverso alcuni

		parametri dati in input, e permette di modificare alcuni campii, o di eliminarlo completamente dal file.
Pre-condizioni	L'evento deve essere registrato all'interno del file "Evento.csv".	
Post-condizioni	*Successo*	Il sistema, dopo aver riconosciuto la data, permette all'utente di poter modificare quell'evento o di eliminarlo. Il sistema dopo aver riconosciuto il nome dell'artista permette la modifica o l'eliminazione.
	Fallimento	Il sistema non riconosce la data, o il nome dell'artista, e reindirizza l'utente al menu principale.
Scenario di base	Il sistema modifica correttamente il profilo utente, se il nickname è già registrato.	
Scenario alternativo	Il sistema comunica l'errore, e reindirizza al menu principale.	

Codice	***Nome***	***Descrizione***
CU05	*Stampa dei partecipanti di un dato evento*	Il programma permette di visualizzare a video, la lista dei partecipanti ad un evento fornito in input.
Pre-condizioni	All'interno del file "Prenotazioni.csv" deve esserci almeno una prenotazione dell'evento richiesto.	
Post-condizioni	Successo	Il sistema, dopo essersi accertato dell'esistenza di

		una prenotazione dell'evento, mostra all'utente tutti i nickname delle persone prenotate.
	Fallimento	Il sistema non riconosce il codice evento, all'interno del file "Prenotazioni.csv".
Scenario di base	Il sistema stampa correttamente tutti i partecipanti di quel dato evento.	
Scenario alternativo	Il sistema comunica l'errore, e reindirizza al menu principale.	

Codice	***Nome***	***Descrizione***
CU06	*Stampa la lista degli eventi a cui ha partecipato un utente, in ordine di data o costo*.	Il programma permette di stampare la lista degli eventi a cui ha partecipato un utente, dando la possibilità di scegliere se stampare il tutto, in ordine cronologico, o in base al costo dei biglietti comprati.
Pre-condizioni	All'interno del file "Prenotazioni.csv" deve essere presente il nickname dell'utente da controllare	
Post-condizioni	*Successo*	Il sistema, dopo essersi accertato dell'esistenza di una prenotazione all'evento, mostra all'utente tutti gli eventi a cui ha partecipato, in base al filtro scelto.
	Fallimento	Il sistema non riconosce il nickname, all'interno del file "Prenotazioni.csv".

Scenario di base	Il sistema stampa correttamente tutti gli eventi a cui ha partecipato l'utente.
Scenario alternativo	Il sistema comunica l'errore, e reindirizza al menu principale.

Codice	**Nome**	**Descrizione**
CU07	*Vende un biglietto ad un utente*	Il programma permette di vendere un biglietto di un evento selezionato ad un utente, selezionando l'evento, la fila e la poltrona.
Pre-condizioni	All'interno del file "Prenotazioni.csv" non possono esistere due Id di prenotazioni uguali. L'esistenza del nickname inserito dall'utente nel file "Utenti.csv", e l'esistenza dell'id evento nel file "Evento.csv". Il sistema controlla che ci siano posti disponibili per quell'evento, che la fila abbia almeno un posto, e che la poltrona scelta sia libera.	
Post-condizioni	Successo	Il sistema, registra nel file "Prenotazioni.csv" la vendita del biglietto, e aggiorna l'incasso dell'evento nel file "Evento.csv" attraverso il relativo codice evento.
	Fallimento	Il sistema ritorna al menu principale
Scenario di base	Il sistema, registra nel file "Prenotazioni.csv" la vendita del biglietto, e aggiorna l'incasso dell'evento nel file "Evento.csv" attraverso il relativo codice evento.	
Scenario alternativo	Il sistema comunica l'errore, e reindirizza al menu principale.	

Codice	***Nome***	***Descrizione***
CU08	*Verifica la disponibilità di una poltrona*	Il programma verifica se la poltrona di una certa fila di un evento, è disponibile o meno.
Pre-condizioni		
Post-condizioni	*Successo*	Il sistema, dopo aver controllato la fila esatta, e il numero della poltrona, stampa un messaggio di disponibilità.
	Fallimento	Il sistema ritorna al menu principale
Scenario di base	Il sistema, dopo aver controllato la fila esatta, e il numero della poltrona, stampa un messaggio di disponibilità.	
Scenario alternativo	Il sistema comunica la non disponibilità, e reindirizza al menu principale.	

Codice	***Nome***	***Descrizione***
CU09	Stampa l'incasso complessivo di un evento	Il programma permette di stampare l'incasso complessivo di un evento.
Pre-condizioni	L'evento deve essere registrato all'interno del file "Evento.csv".	
Post-condizioni	*Successo*	Il sistema, dopo aver riconosciuto l'evento, stampa un messaggio con il relativo incasso totale.
	Fallimento	Il sistema ritorna al menu principale
Scenario di base	Il sistema, dopo aver riconosciuto l'evento, stampa un messaggio con il relativo incasso totale.	
Scenario alternativo	Il sistema non avendo trovato l'id evento, ritorna al menu principale.	

Codice	***Nome***	***Descrizione***
CU10	Stampa la copertina del software	Il programma permette di stampare tutti i riconoscimenti all'interno del software.
Pre-condizioni	Avvio del main	
Post-condizioni	*Successo*	Il sistema, stampa tutti i messaggi preimpostati.
	Fallimento	
Scenario di base	Il sistema, stampa la copertina del software.	
Scenario alternativo		

Codice	***Nome***	***Descrizione***
CU11	Stampa l'istogramma delle prenotazioni ad un evento.	Il programma disegna un istogramma in base ai biglietti venduti per ogni evento.
Pre-condizioni	Avvio del main	
Post-condizioni	*Successo*	Il sistema, stampa l'istogramma.
	Fallimento	Il sistema, legge il file "Prenotazioni.csv", e rilevandolo vuoto, avvisa l'utente dell'assenza di biglietti venduti.
Scenario di base	Il sistema, stampa l'istogramma di tutti gli eventi con almeno un biglietto venduto, leggendo le informazioni dal file "Prenotazioni.csv".	
Scenario alternativo	Il sistema, non stampa l'istogramma poiché non esistono ancora biglietti venduti all'interno del file "Prenotazioni.csv".	

Progettazione

Principali variabili, strutture dati e file

Nome	***Tipologia***	***Descrizione***	***Tipi/campi/valori***
Utenti	*Struct*	Tipo di dato definito per descrivere le caratteristiche dell'utente.	char nickname[MAX]; char email[MAX]; char cognome[MAX]; char nome[MAX]; char genere[MAX]; char datan[MAX]; char ntelefono[MAX]; char gmusicale[MAX];
evento	Struct	Tipo di dato per descrivere le caratteristiche dell'evento.	int id; int anno; char mese[MAX]; int giorno; int ora_i; int minuti_i; int ora_f; int minuti_f; char nomelocale[MAX]; char artista[MAX]; double costo_b_primaf; double costo_b_ultimaf; int n_file; int n_posti_perfila; double incasso_complessivo;

prenotazione	Struct	Struttura utilizzata per gestire tutte le prenotazioni.	int idp; int id_e; char nick[MAX]; int fila; int n_pol; double importo_p;
listautenti	Struct	Struttura d'appoggio, utilizzata per gestire il requisito funzionale R06.	char nickname[MAX]; int id_evento; int anno; int mese; int giorno; double costo_biglietto;
Utenti.csv	File	File utilizzato per registrare tutti gli utenti.	
Eventi.csv	File	File utilizzato per registrare tutti gli eventi.	
Prenotazioni.csv	File	File utilizzato per registrare tutte le prenotazioni.	

Tabella 3. (Tipi di dato e strutture dati).

Librerie e funzioni

Abbiamo progettato e realizzato le seguenti librerie:

- **Evento.o**
- **Prenotazioni.o**
- **Funzioni.o**
- **Utenti.o**

Le funzioni all'interno della libreria "**Evento.o**" sono:

➔ **void** registraEvento();
➔ **int** ricercaEvento(int, int, char[],int, char[], char[]);
➔ **void** cercaEvento();

- ➔ **void** ModificaEvento(int);
- ➔ **int** Menu_modifica_evento();
- ➔ **void** EliminaEvento(int);
- ➔ **int** ricercaIDEvento(int);
- ➔ ***void*** VisualizzaIncasso();
- ➔ ***void*** Istogramma*();*

Le funzioni all'interno della libreria "**Prenotazioni.o**" sono:

- ➔ **void** venditaBiglietto();
- ➔ **int** ControllaFilaPoltrona(int,int,int,int,int);
- ➔ **int** ControllaPosti(int);
- ➔ **int** ricercaPrenotazione(int);
- ➔ **void** AggiornaIncasso(int,double);
- ➔ **void** ControlloPoltrona();
- ➔ **void** StampaUtentiFiltro();
- ➔ **void** StampaPrenotati();
- ➔ **int** ControlloMesi(char []);

Le funzioni all'interno della libreria "**Funzioni.o**" sono:

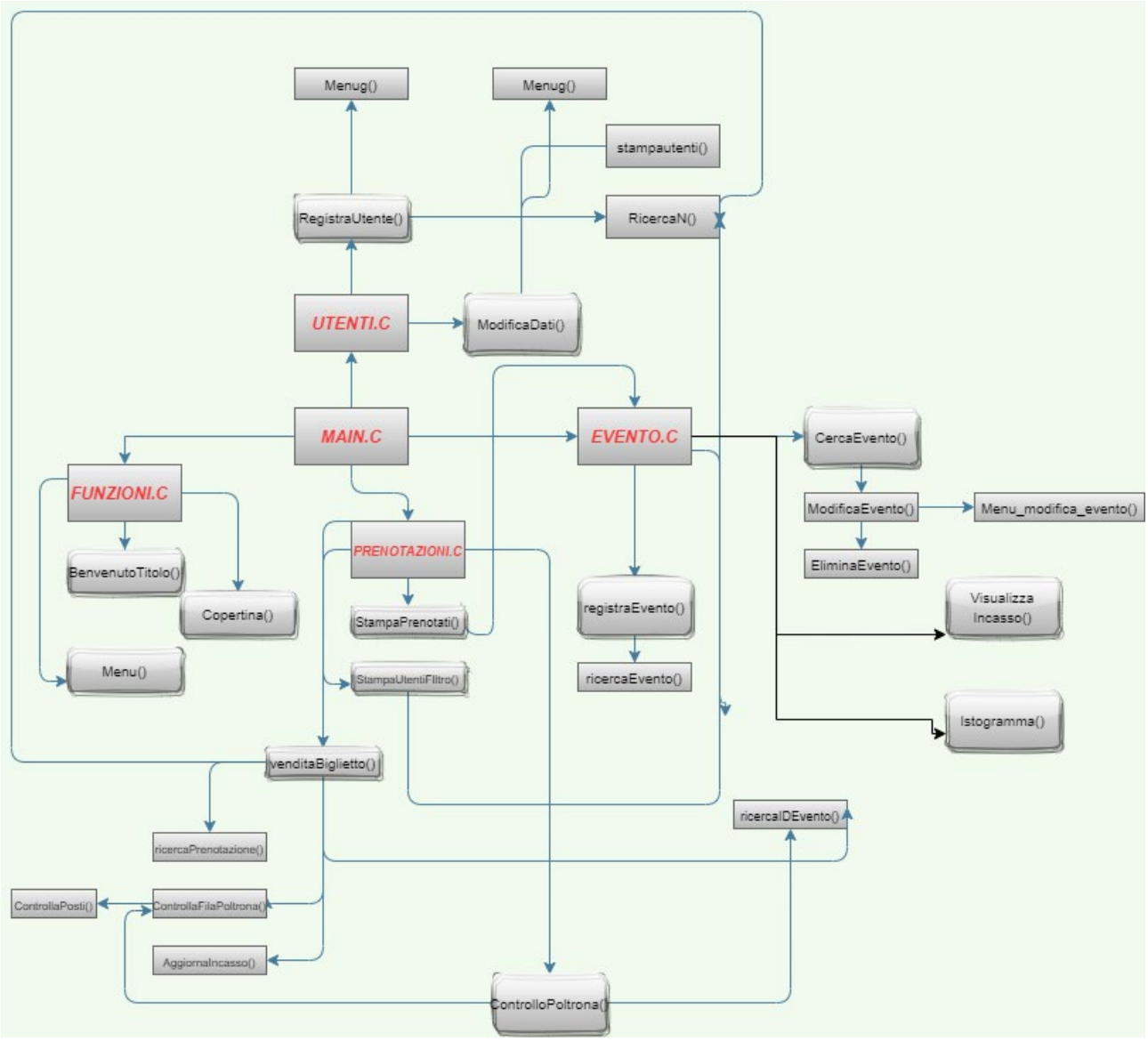
- ➔ **void** menu();
- ➔ **void** BenvenutoTitolo();
- ➔ **void** SetColor(unsigned short);
- ➔ **void** Copertina();
- ➔ **void** Colora_Menu();

Le funzioni all'interno della libreria "**Utenti.o**" sono :

- ➔ **void** registra();
- ➔ **void** modificadati();

➔ **void** menug();
➔ **int** ricercaN(char[]);
➔ **void** stampautenti(char[]);

Dipendenze tra funzioni



Descrizione

MAIN.C

Quando il software viene avviato, la funzione "***int main()",*** richiama dapprima le funzioni BenvenutoTitolo() e Copertina(), che appartengono alla libreria "***Funzioni.o***". Dopodichè, viene richiamato il menù appartenente ancora alla libreria "***Funzioni.o",*** che permetterà all'utente di scegliere come navigare all'interno del software.

UTENTI.C

La procedura "*RegistraUtente()*", appartenente alla libreria ***"Utenti.o",*** richiama la funzione *"RicercaN()"* per controllare la registrazione di quel dato nickname. La procedura *"modificadati()"* richiama nuovamente la funzione *"ricercaN"* per controllare la presenza di quell'utente, la procedura *"stampautenti()"*, che stampa tutte le informazioni di quel dato utente e *"menug()"* che stampa la lista di tutti i generi musicali disponibili.

EVENTO.C

La procedura *RegistraEvento()*, che appartiene alla libreria "***Evento.o",*** attraverso la funzione *ricercaEvento()*, stabilisce le condizioni per registrare il nuovo evento.

La procedura *ricercaEvento(),* utilizza le funzioni *ModificaEvento()* e *EliminaEvento(),* per permettere all'utente di modificare o eliminare un determinato evento. *ModificaEvento()* a sua volta, richiama la procedura *Menu_modifica_evento()*, che stampa tutti i campi modificabili.

PRENOTAZIONI.C

La procedura *stampaPrenotati()*, richiama la funzione *ricercaIDEvento()*, che assicura l'esistenza di un dato evento all'interno del file "Eventi.csv".

La procedura *StampaUtentiFiltro()*, utilizza le procedura *ricercaN().*

La procedura *venditaBiglietto()*, utlizza le procedure: *ricercaN(), ricercaIDEvento(), ricercaPrenotazione(), ControllaFilaPoltrona() e AggiornaIncasso(),*

La funzione *ControllaFilaPoltrona()* a sua volta richiama la funzione *ControllaPosti().*

Infine, VisualizzaIncasso().

Pseudo-codice

All’inizio della funzione main, vengono richiamate le funzioni BenvenutoTitolo() e Copertina(), che presenteranno il programma e stamperanno i vari riconoscimenti.
Viene effettuato un controllo con il puntatore ai vari file per controllare che essi esistano, questa condizione non implica il funzionamento del sotfware, poiché se non vengono rilevati, quest’ultimo provvederà a crearli da sé.
Dopodichè, viene richiamata la funzione menu, che permette all'utente di navigare all'interno del software e di scegliere come procedere. All’interno del nostro codice la funziona che viene richiamata più volte è la SetColor(unsigned short int), che permette di colorare l’output sul cmd a nostro piacimento.

Successivamente, viene controllato l'input inserito all'interno della funzione [***PROGRAMMAZIONE SICURA***] e controllato in uno switch, dove si potrà scegliere tra:

RegistraUtente(), modificadati(), registraEvento(), cercaEvento(), StampaUtentiFiltro(), venditaBiglietto(), VisualizzaIncasso(), ControlloPoltrona(), StampaPrenotati() e Istogramma(), che è una funzionalità aggiuntiva.

La procedura RegistraUtente() permette di registrare tutti i dati richiesti dal software

all'interno del file "Utenti.csv", nel caso il nickname inserito inizialmente, non sia già presente.

Per effettuare questo controllo, viene richiamata la funzione RicercaN(), che accetta come parametro il

nickname inserito dall'utente, e controlla appunto la sua presenza all'interno del file "Utenti.csv".

Dopo aver oppurtanamente controllato questo dato, si continua con la registrazione.
Tutti i dati vengono registrati momentaneamente nella struct utente, dopodichè verranno scritti in una nuova riga all’interno del file “File//Utenti.csv”.

All'interno della funzione RegistraUtente(), viene richiamata inoltre la funzione menug(), che stampa

tutti i generi musicali a disposizione per la registrazione.

Dopo aver completato tutti i campi, il file verrà aggiornato/creato attraverso la fprintf.

La procedura ModificaDati() permette di modificare un parametro di un utente registrato, ogni volta che essa

viene richiamata. Viene chiamata al suo interno nuovamente la funzione ricercaN(), per accertarsi che l'utente esista. Dopodichè, vengono stampate tutte le attuali informazioni dell'utente, e viene richiesto il numero del campo da modificare. Ovviamente, il campo del nickname non può essere modificato, poiché trattasi di un identificativo. Vengono ricopiate tutte le informazioni aggiornate in nuovo file "Utenti2.csv".

Infine, il file non aggiornato viene eliminato attraverso la funzione remove(), e il file "Utenti2.csv", viene rinominato "Utenti.csv".Se così non fosse, la funzione riporta l'utente all'interno del main(); La funzione stampautenti, attraverso il nickname, stampa tutti i campi di quel dato nickname.

La procedura registraEvento() permette all'utente di registrare il proprio evento nel file "Evento.csv",

ma prima di scrivere sul file, viene richiamata la funzione ricercaEvento(), che prende come parametri in input l'id inserito, l'anno, il mese, il giorno, il nome del locale e dell'artista.

La funzione ricercaEvento(), si occupa di scoprire eventuali incongruenze all'interno dell'evento che si vuole registrare, con quelli già presenti all'interno del file "Evento.csv". Infatti non possono esistere all'interno del database degli eventi che abbiano lo stesso artista che si esibisce lo stesso giorno o nello stesso locale, oppure un id evento uguale.
Successivamente, se l'algoritmo trova degli elementi non conformi, ritorna al main, altrimenti continua la registrazione dell'evento, andando infine, a scrivere le informazioni inserite all'interno del file "Eventi.csv".

La procedura cercaEvento(), permette all'utente di ricercare un evento in base alla data o al nome dell'artista ospite. In entrambi casi, dopo aver verificato che la data corrisponda ad un evento, o che l'artista corrisponda ad un evento registrato, l'utente può scegliere di modificare o di eliminare quel dato evento.

In caso di modifica, viene richiamata la funziona modifcaEvento, che riceve come parametro l'identificatore dell'evento (ID (intero)) e ricerca all'interno del file "Eventi.csv" quest'ultimo. Se la ricerca sarà andata a buon fine, il software restituirà i dati attuali e richiederà il numero del campo da modificare, attraverso la funzione Menu_modifica_evento(). Verrà poi scritto un nuovo file "Eventi2.csv", nel quale saranno presenti tutti i campi e i record aggiornati. Come avvenuto per il database aggiornato degli utenti, viene utilizzata nuovamente la funzione remove(), per eliminare il vecchio file, e quella rename() per rinominare quello nuovo.

La funzione Menu_modifica_evento() stampa tutti i campi modificabili, e restituisce la scelta.

Dopo aver inserito il nuovo dato, il software provvederà a trascrivere tutto all'interno del file "Eventi.csv".

In caso di eliminazione dell'evento, la funziona cercaEvento(), richiamerà la funzione EliminaEvento(). Quest'ultima prenderà come parametro l'identificatore dell'evento (ID (intero)) e andrà ad eliminare quell'evento dal file "Evento.csv".

La procedura StampaUtentiFiltro() permette di stampare l'elenco degli eventi a cui ha partecipato un utente, in ordine cronologico, o in base al costo del biglietto da lui comprato. Inizialmente viene richiamata la funzione ricercaN(), che controlla l'esistenza del nickname all'interno del file "Utenti.csv". In seguito, la funzione continua l'esecuzione, e richiama la funzione ControlloMesi() per convertire il mese letto dal file come stringa, in un numero. Dopodichè, dopo aver trasferito tutti gli eventi in un vettore di struct, si procede all'ordinamento. Però, se la variabile conta, che si occupa di contare tutti gli eventi a cui ha partecipato quel dato utente, è uguale a 0, la funzione restituirà un messaggio di avviso.

La procedura VenditaBiglietto() permette all'utente registrato all'interno del file "Utenti.csv" di prenotare un biglietto per un evento. Ovviamente, la funzione obbliga l'utente di inserire dapprima il nickname, e successivamente il codice evento. La funzione ricercaN() e la funzione ricercaIDEvento() controllano che entrambi i campi funzionino correttamente. L'id prenotazione viene generato

automaticamente, e viene controllato richiamando la funzione ricercaPrenotazione() che accetta come parametro il dato appena generato. Quando viene controllato il numero di fila e di poltrona inserito dall'utente, viene richiamata la funzione ControllaFilaPoltrona(). Quest'ultima controlla che l'evento abbia ancora posti liberi, richiamando la funzione ControllaPosti(), che accetta come parametro l'id evento e conta il numero di biglietti venduti per quell'evento. Successivamente, il controllo torna alla funzione ControllaFilaPoltrona(), che controlla se la fila abbia almeno un posto libero, e infine che la poltrona selezionata dall'utente non sia già occupata. Ovviamente se la fila selezionata corrisponde alla prima o all'ultima, il costo rimane fisso. Invece, nel caso l'utente dovesse selezionare file intermedie, il costo viene dinamicamente calcolato attraverso la formula:

$$C_n = C_{prima} - \frac{n-1}{N_{ultima}-1} * (C_{prima} - C_{ultima}).$$

Dopo aver concluso il tutto, viene richiamata la funzione AggiornaIncasso(), che aggiorna automaticamente l'incasso dell'evento sul file "Evento.csv".

La procedura VisualizzaIncasso() permette all'utente di stampare l'incasso complessivo di un evento, inserendo l'opportuno codice. Viene richiama la funzione ricercaIDEvento() per controllare che esso esista. La funzione ricercaIDEvento(), apre il file "Evento.csv" e trasferisce l'incasso parziale in una variabile.

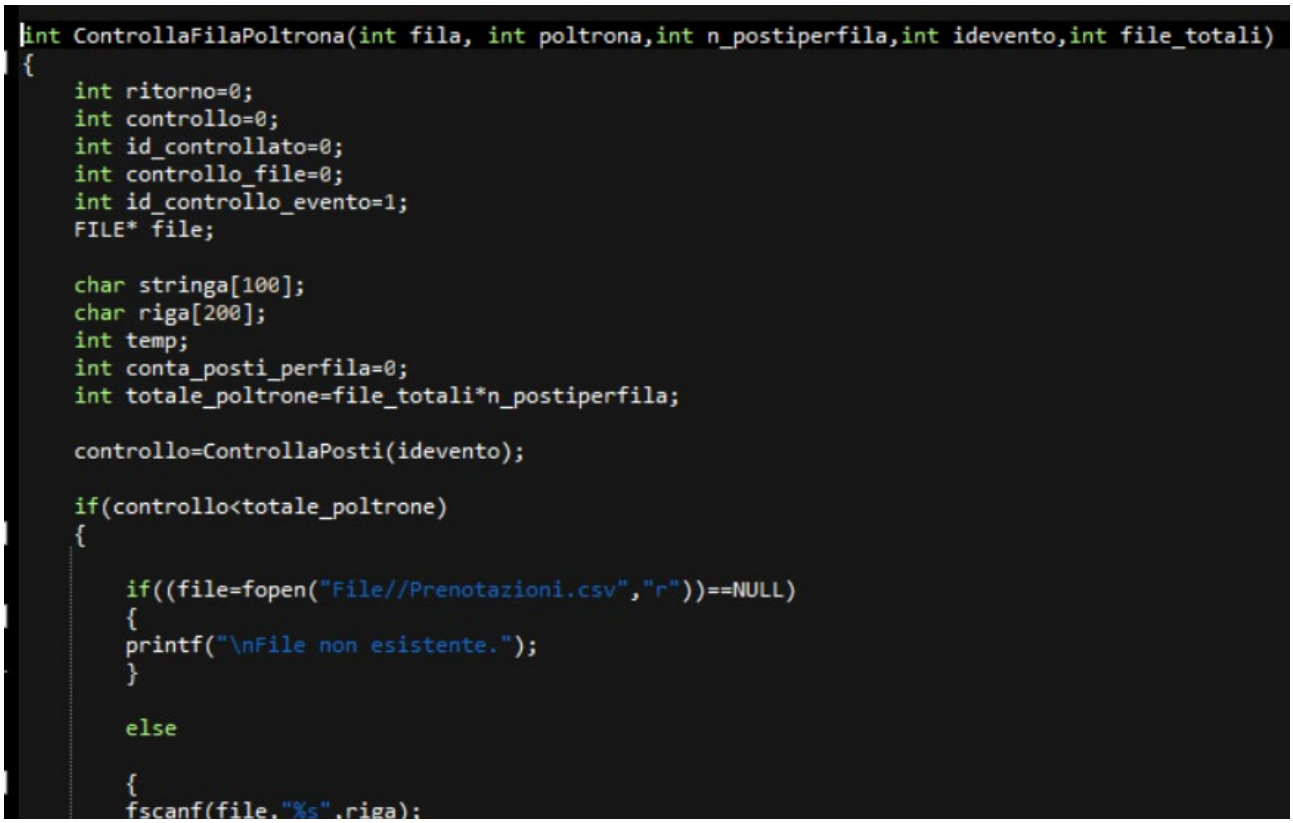
La procedura ControlloPoltrona() permette all'utente di controllare che una determinata poltrona di un evento, posta in una N fila, sia libera. Si comporta esattamente come la funzione VenditaBiglietto(), senza richiamare la funzione ricercaN(), poiché non viene inserito nessun nickname. Quindi richiama la funzione ricercaIDEvento() per controllare l'esistenza dell'evento selezionato, e ne controlla tutti i parametri inseriti, attraverso la funzione ControllaFilaPoltrona().

La procedura StampaPrenotati() permette di stampare tutti i nickname dei prenotati ad un determinato evento. Inizialmente viene richiamata la funzione ricercaIDEvento(), per controllare che esista quest'utlimo. Dopodichè, si leggono tutte le informazioni necessarie all'interno del file "Prenotazioni.csv".

La procedura Istogramma() permette di creare un istogramma, disegnato con degli '*', che permette di visualizzare il numero di biglietti venduti per ogni evento. Ovviamente, non compariranno tutti gli eventi, ma bensì gli eventi con almeno un biglietto venduto.

Codifica

La funzione in questione *(ControllaFilaPoltrona),* è una parte fondamentale del software, poiché permette all'utente durante la vendita del biglietto, di verificare se la poltrona da lui scelta è libera o meno. Viene richiamata anche durante il controllo della poltrona, che esegue esattamente la medesima cosa, senza però vendere il biglietto.



```
int ControllaFilaPoltrona(int fila, int poltrona,int n_postiperfila,int idevento,int file_totali)
{
    int ritorno=0;
    int controllo=0;
    int id_controllato=0;
    int controllo_file=0;
    int id_controllo_evento=1;
    FILE* file;

    char stringa[100];
    char riga[200];
    int temp;
    int conta_posti_perfila=0;
    int totale_poltrone=file_totali*n_postiperfila;

    controllo=ControllaPosti(idevento);

    if(controllo<totale_poltrone)
    {

        if((file=fopen("File//Prenotazioni.csv","r"))==NULL)
        {
        printf("\nFile non esistente.");
        }

        else

        {
        fscanf(file,"%s",riga);
```

Come prima cosa, viene richiamata un'altra funzione che si occupa di controllare che sia disponibile almeno un posto, contando tutti quelli occupati. Se così fosse, si passa ad un ulteriore controllo. Viene aperto il file in lettura, attraverso il puntatore *"file"* e dopodiché viene letta ogni riga usando la funzione *"fscanf",* e inserendo il contenuto all'interno della variabile *"riga".*

```
else

{
fscanf(file,"%s",riga);

while(!feof(file) && ritorno==0)
{
    strcpy(stringa, strtok(riga, ","));
    strcpy(stringa, strtok(NULL, ","));
    id_controllato=atoi(stringa); // id evento
    strcpy(stringa, strtok(NULL, ","));
    strcpy(stringa, strtok(NULL, ",")); //nfile
    temp=atoi(stringa);

    if(id_controllato==idevento)
    {
        id_controllo_evento=1;

        if(temp==fila)
        {
            controllo_file=1;
            conta_posti_perfila++;

            if(conta_posti_perfila<n_postiperfila)
            {
                strcpy(stringa, strtok(NULL, ","));
                temp=atoi(stringa);


                if(temp==poltrona)
                {
```

Viene poi, utilizzata la funzione "*strtok*", che legge il contenuto della variabile "*riga"*, fino a che non incontra l'identificatore, ovvero la "*,*".

All'interno di questo algoritmo, ci serve controllare dapprima che l'id evento inserito dall'utente, corrisponda a quella riga. Se così non fosse, il ciclo riparte, e legge la riga successiva.

Se invece all'interno della riga letta, l'id evento corrisponde, il software entra all'interno dell'if, segnalando con una variabile quello appena accaduto. Dopodichè, controlliamo se esista un biglietto venduto all'interno del file *"Prenotazioni.csv",* che ha la stessa fila inserita dall'utente.
Sappiamo che ogni fila ha dei posti limitati, quindi controlliamo che in quelle fila ci sia almeno una poltrona disponibile.

```
if(temp==fila)
{
    controllo_file=1;
    conta_posti_perfila++;

    if(conta_posti_perfila<n_postiperfila)
    {
        strcpy(stringa, strtok(NULL, ","));
        temp=atoi(stringa);


        if(temp==poltrona)
        {
            ritorno=1;
        }

        else

        {
            ritorno=0;
        }
    }

    else
    {
        system("CLS");
        BenvenutoTitolo();
        printf("\n\nIn questa fila, tutte le poltrone sono esaurite.\n");
    }
}
```

Se i posti per quella fila sono esauriti, il programma avviserà immediatamente l'utente, altrimenti se i posti di quella fila sono disponibili, si confronterà il numero della poltrona inserito dall'utente, con quello scritto nel file.

Può capitare che quella fila non sia presente all'interno del file, questo vuol dire che non è stato acquistato ancora un biglietto per quella fila.

Quindi automaticamente, verrà comunicata la disponibilità.

Test

MAIN

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R00	0.1	Menù iniziale	Scelta n.1	1	Registrazione utente	Registra utente
R00	0.2	Menù iniziale	Scelta n.2	1	Modifica dati utente	Modifica utente
R00	0.3	Menù iniziale	Scelta n.3	3	Registra Evento	Registra evento
R00	0.4	Menù iniziale	Scelta n.4	4	Cerca evento	Ricerca evento
R00	0.5	Menù iniziale	Scelta n.5	5	Stampa prenotati	Stampa prenotati
R00	0.6	Menù iniziale	Scelta n.6	6	Stampa Utenti Filtro	Stampa eventi di un utente
R00	0.7	Menù iniziale	Scelta n.7	7	Vendita biglietto	Vende un biglietto ad un utente
R00	0.8	Menù iniziale	Scelta n.8	8	Controllo poltrona	Controllo della poltrona di un evento
R00	0.9	Menù iniziale	Scelta n.9	9	Visualizza incasso	Visualizza l'incasso di un evento
R00	0.10	Menù iniziale	Scelta n.10	10	Copertina	Visualizza la copertina del software
R00	0.11	Menù iniziale	Scelta n.11	11	Istogramma	Stampa un istogramma degli eventi
R00	0.12	Menù iniziale	Scelta errata	Numero minore di 1 e maggiore di 10	Messaggio di errore	Ritorno al menù

REGISTRA UTENTE

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R01	1.0	Registrazione utente	Inserimento del nickname	Nickname	Proseguimento della registrazione	Completamento della registrazione
R01	1.1	Registrazione utente	Inserimento di un nickname già esistente.	Nickname	Interruzione della registrazione	Reindirizzamento al menù

MODIFICA UTENTE

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R02	2.0	Modifica dati utente	Inserimento del nickname	Nickname	Proseguimento della modifica	Modifica di un campo dell'utente
R02	2.0	Modifica dati utente	Inserimento di nickname non registrato	Nickname	Richiesta di un nuovo nickname	Nuova richiesta di un nickname

REGISTRA EVENTO

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R03	3.0	Registrazione di un evento	Inserimento di un Id evento già registrato, o di una data coincidente con un altro spettacolo	Id evento Anno evento Mese evento Giorno evento Nome locale Nome dell'artista	Nuova richiesta di tutti i dati.	Richiesta di un nuovo id evento e di tutti gli altri input utili alla registrazione.

			dello stesso artista.			
R03	3.1	Registrazione di un evento	Inserimento di un Id evento non registrato, o di date non in conflitto tra loro	Id evento Anno evento Mese evento Giorno evento Nome locale Nome dell'artista	Richiesta di tutti i dati rimanenti per completare la registrazione.	Completamento della registrazione dell'evento.

CERCA EVENTO

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R04	4.0	Ricerca di un evento	Ricerca di un determinato evento attraverso la data, che potrà in seguito essere modificato o eliminato.	1	Scelta successiva del campo da modificare o eliminare	Scelta di quale campo modificare o eliminare
R04	4.1	Ricerca di un evento	Ricerca di un determinato evento attraverso l'artista, che potrà in seguito essere modificato o eliminato.	2	Scelta successiva del campo da modificare o eliminare	Scelta di quale campo modificare o eliminare

R04	4.2	Ricerca di un evento	Ricerca di un determinato evento attraverso la data o l'artista, che potrà in seguito essere modificato o eliminato.	<1 || >2	Ritorno al menù di scelta.	Richiesta nuovamente di un input di scelta.

STAMPA PRENOTATI

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R05	5.0	Stampa prenotati	Stampa tutti gli utenti che hanno acquistato un biglietto per quell'evento.	Id_evento registrato	Stampa di tutti i nickname dei prenotati a quell'evento.	Stampa del nickname di tutti gli utenti che hanno acquistato un biglietto.
R05	5.1	Stampa prenotati	Stampa tutti gli utenti che hanno acquistato un biglietto per quell'evento	Id_evento non registrato.	Ritorno all'inserimento dell'id evento.	Ritorno al menù di scelta

STAMPA EVENTI

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R06	6.0	StampaEventiFiltro	Stampa tutti gli eventi a cui ha partecipato un utente in	Nickname	Viene richiesta la stampa di tutti gli eventi, in	Il software permette all'utente di scegliere come

			ordine cronologico o in base al costo del biglietto		ordine cronologico o in base al costo del biglietto.	effettuare la stampa.
R06	6.0.1	StampaEventiFiltro	Stampa tutti gli eventi a cui ha partecipato un utente in ordine cronologico o in base al costo del biglietto	1	Stampa tutti gli eventi in ordine cronologico.	Il software stampa gli eventi a cui ha partecipato l'utente in ordine cronologico.
R06	6.0.2	StampaEventiFiltro	Stampa tutti gli eventi a cui ha partecipato un utente in ordine cronologico o in base al costo del biglietto	2	Stampa tutti gli eventi in base al costo dei biglietti.	Il software stampa gli eventi a cui ha partecipato l'utente in base al costo dei biglietti.
R06	6.0.3	StampaEventiFiltro	Stampa tutti gli eventi a cui ha partecipato un utente in ordine cronologico o in base al costo del biglietto	<1 || >2	Richiede nuovamente la scelta da inserire.	Il software richiede la scelta.
R06	6.1	StampaEventiFiltro	Stampa tutti gli eventi a cui ha partecipato un utente in ordine cronologico o in base al costo del biglietto	Nickname non registrato	Viene richiesto nuovamente di inserire un nickname valido e registrato.	Il software riprende in input il nickname

VENDITA BIGLIETTO

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R07	7.0	Vendita biglietto	Permette di vendere un biglietto di un determinato evento ad un utente.	Nickname registrato	Viene richiesto di inserire successivamente l'id dell'evento da prenotare.	Richiesta dell'id evento da prenotare.
R07	7.0.1	Vendita biglietto	Permette di vendere un biglietto di un determinato evento ad un utente.	Id evento registrato	Viene richiesto di inserire successivamente la fila da prenotare.	Richiesta della file da prenotare.
R07	7.0.2	Vendita biglietto	Permette di vendere un biglietto di un determinato evento ad un utente.	Id evento non registrato	Viene richiesto nuovamente l'id evento.	Viene richiesto un id evento registrato.
R07	7.1	Vendita biglietto	Permette di vendere un biglietto di un determinato evento ad un utente.	Nickname non registrato	Viene richiesto di inserire successivamente un nickname registrato.	Viene richiesto nuovamente il nickname per prenotare il biglietto.
R07	7.2	Vendita biglietto	Permette di vendere un biglietto di un determinato evento ad un utente.	N fila da prenotare	Viene richiesto di inserire successivamente il numero della poltrona	Richiesta del numero della poltrona da prenotare.
R07	7.2.1	Vendita biglietto	Permette di vendere un biglietto di un determinato	N fila da prenotare maggiore di quelle disponibili o minori.	Viene richiesto di inserire nuovamente la fila.	Richiesta nuovamente della fila da prenotare.

			evento ad un utente.			
R07	7.3	Vendita biglietto	Permette di vendere un biglietto di un determinato evento ad un utente.	N poltrona libera	Il biglietto viene venduto correttamente e viene aggiornato l'incasso dell'evento.	Vendita del biglietto e aggiornamento dell'incasso.
R07	7.3.1	Vendita biglietto	Permette di vendere un biglietto di un determinato evento ad un utente.	N poltrona occupata	Vendita fallita e aggiornamento non avvenuto.	Vendita del biglietto fallita.

CONTROLLO POLTRONA

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R08	8.0	Controllo Poltrona	Permette di controllare se una determinata poltrona di un evento, è libera o meno.	N fila	In seguito, viene richiesto il numero della poltrona.	Richiesta del numero della poltrona
R08	8.0.1	Controllo Poltrona	Permette di controllare se una determinata poltrona di un evento, è libera o meno.	N fila minore o maggiore di quello presente	In seguito, viene richiesta nuovamente la fila.	Nuova richiesta della fila.
R08	8.1	Controllo Poltrona	Permette di controllare se una determinata poltrona di un evento, è libera o meno.	Numero Poltrona	Stampa di un messaggio che attesta la condizione della poltrona.	Stampa di un messaggio.

R08	8.1.1	Controllo Poltrona	Permette di controllare se una determinata poltrona di un evento, è libera o meno.	N fila minore o maggiore di quello presente	In seguito, viene richiesta nuovamente la fila.	Nuova richiesta della fila.

VISUALIZZA INCASSO

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
RO9	9.0	Visualizza Incasso	Permette di controllare l'incasso di un determinato evento.	Id evento registrato	Stampa dell'incasso complessivo.	Stampa dell'incasso di quel determinato evento.
RO9	9.0	Visualizza Incasso	Permette di controllare l'incasso di un determinato evento.	Id evento non registrato	Richiesta di un nuovo id evento	Nuova richiesta di un id evento

COPERTINA

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R10	10.0	Copertina	Permette di stampare la copertina del software.		Stampa di tutti i riconoscimenti	Stampa di tutti i riconoscimenti dell'autore.

ISTOGRAMMA

Codice requisito	***Codice test***	***Nome***	***Descrizione test***	***Eventuale input***	***Risultato atteso***	***Risultato ottenuto***
R11	11.0	Istogramma	Permette di stampare l'istogramma degli eventi con almeno un biglietto venduto	Almeno un evento con un biglietto venduto all'interno del file "Prenotazioni.csv"	Stampa dell'istogramma	Stampa attuale dell'istogramma degli eventi, con almeno un biglietto venduto.
R11	11.1	Istogramma	Permette di stampare l'istogramma degli eventi con almeno un biglietto venduto	Nessun evento con un biglietto venduto all'interno del file "Prenotazioni.csv"	Avviso all'utente	Il software avvisa l'utente, poiché non esiste nessun evento con almeno un biglietto venduto all'interno del file "Prenotazioni.csv".

Tabella 4. Risultati dei test.

Matrice delle responsabilità

	Cellammare Gabriel	***Paolo Ingusci***
CODICE		
Indentazione	X	
Funzioni/Procedure	X	
Grafica	X	
Controlli sull'input		X
Algoritmi di ordinamento		X
Nomi e controllo variabili		X
DOCUMENTAZIONE		
Analisi		X
Progettazione	X	
Codifica	X	
Doxygen	X	
Test		X
Conclusioni		X

Conclusioni

Il software durante il suo funzionamento necessita obbligatoriamente di tre file ***csv***, ovvero “Utenti.csv”, ”Prenotazioni.csv” e ”Eventi.csv”.

Una limitazione da noi percepita, è quella della funzione strtok(), che non consente di individuare all’interno della stringa ulteriori spazi. Se per esempio, noi inserissimo un’artista (NOME COGNOME), il software smetterebbe di funzionare. Quindi, in una futura release, si potrebbe pensare di risolvere questo bug, utilizzando la fscanf() al posto della strtok(), che riconosca effettivamente i delimitatori.

Si prevede inoltre in futuro, di poter implementare una funzione che riconosca qual è il genere più in voga tra gli utenti registrati, e possa consigliare ad ogni proprietario del locale, che artista far partecipare in base all’utenza.

Alcune criticità importanti provengono da alcuni controllati non effettuati in alcuni dati d'input, poiché permetterebbero ad alcuni utenti distratti di inserire dati che poi andrebbero registrati all'interno dei file .csv.

L'utente, inoltre, potrebbe inserire alcuni caratteri al posto delle cifre durante la prenotazione del biglietto, e questo comporterebbe un loop/crash del software.

Il software risponde pienamente ai requisiti funzionali richiesti, e inoltre è stata aggiunta una funzione che permette di visionare graficamente (attraverso un istogramma creato con il carattere **'*'**) il numero di biglietti venduti per ogni evento. La registrazione di ogni singolo evento non avviene casualmente, controllando solo che non esista un ID uguale. Viene controllato in primis che l'artista non si stia esibendo altrove, e successivamente che non esista un evento simile con uno stesso id, o che si svolga effettivamente, nello stesso locale.

Dal punto di vista grafico, il software è molto intuitivo poiché la console viene pulita ogni volta, e le operazioni da eseguire risultano molto chiare.

Infine, si spera in futuro di creare altre librerie che si occupino principalmente di tutti i controlli sull'input e che il nostro software possa avere una grafica più avanzata.