PROGRAMMAZIONE III

Progetto gestionale "Airplane ticket Reservation" Relazione

Partecipanti

Morlando Pasquale mat. 0124001791

Bencivenga Crescenzo mat. 0124001916

Descrizione del progetto

Si vuole realizzare e sviluppare un sistema per la gestione e la prenotazione di biglietti aerei. Si vogliono monitorare i dati persistenti del sistema attraverso un database SQLite. I dati da rendere persistenti sono quelli relativi ai voli, utenti, biglietti acquistati. L'accesso al sistema può avvenire secondo due modalità: amministratore e utente.

L'utente può:

- Visualizzare i voli disponibili
- Acquistare biglietti con più modalità di pagamento
- Visualizzare i biglietti acquistati

Una volta scelto il volo l'utente può effettuare l'acquisto del biglietto relativo ad esso con diversi metodi di pagamento, una volta acquistato il biglietto il sistema memorizza l'acquisto nella tabella bigliettiAcquistati

L'amministratore può:

- Aggiungere voli
- Eliminare voli
- Annullare l'ultima operazione effettuata

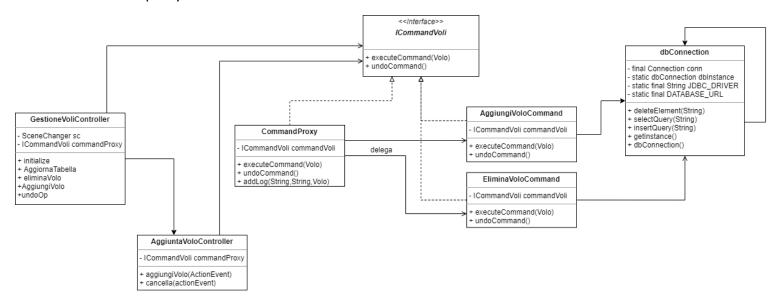
Design pattern

Design pattern utilizzati:

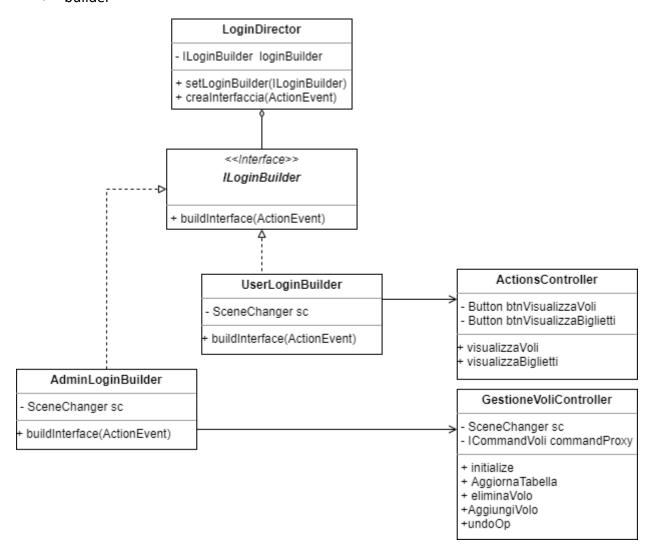
- Builder: è un pattern creazionale e viene utilizzato per separare la costruzione di un oggetto complesso dalla sua rappresentazione in modo tale che lo stesso processo di costruzione può creare differenti rappresentazioni, è stato implementato per la creazione degli oggetti controller delle diverse GUI per amministratori ed utenti.
- Strategy: L'obiettivo di questa architettura è isolare un algoritmo all'interno di un oggetto, in maniera tale da risultare utile in quelle situazioni dove sia necessario modificare dinamicamente gli algoritmi utilizzati da un'applicazione, è stato implementato per la gestione dei diversi metodi di pagamento.
- Command: Incapsula una richiesta all'interno di un oggetto, consentendo così di parametrizzare i client con richieste diverse, accodare o rintracciare le richieste, oppure supportare operazioni di undo, è stato implementato per l'esecuzione delle operazioni dell'amministratore e l'undo dell'ultima operazione effettuata.
- Proxy: fornire un surrogato per un altro oggetto per controllare l'accesso ad esso, è stato implementato per il logging delle operazioni effettuate dall'amministratore.

Diagrammi delle classi dei design pattern:

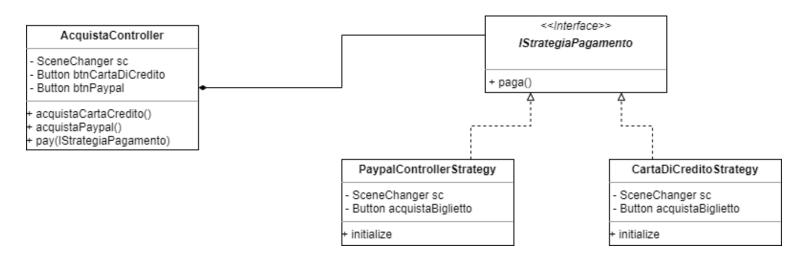
Command e proxy



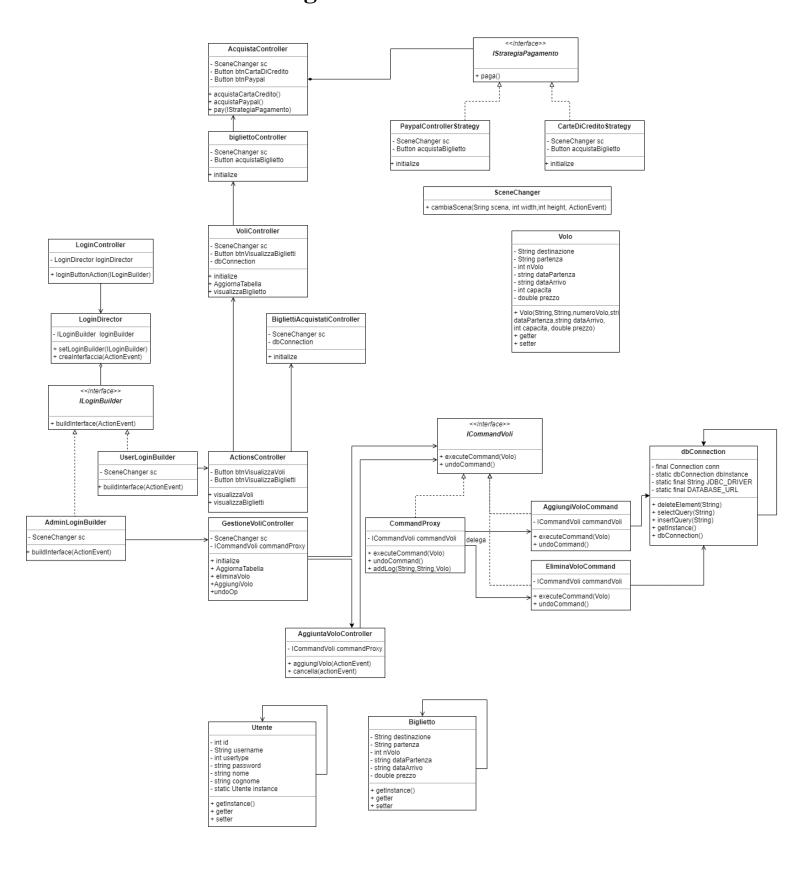
• builder



Strategy



Diagrammi delle classi:



Database:

Le tabelle usate per memorizzare i dati all'interno del Detabase sono:

- Volo contenente i campi: nVolo, partenza, destinazione, dataPartenza, dataArrivo, capacita e prezzo
- User contenente i campi: username, password, nome, cognome, user_id,userType
- o BigliettiAcquistati contenente nVolo e idUser

Credenziali admin:

Username: adminPassword: admin

Credenziali utente:

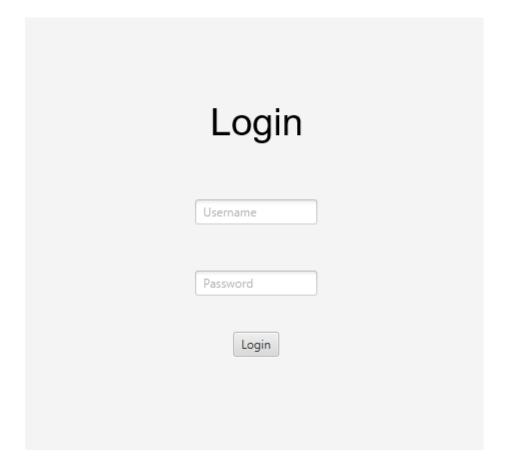
Username: userPassword: user

Connessione al database:

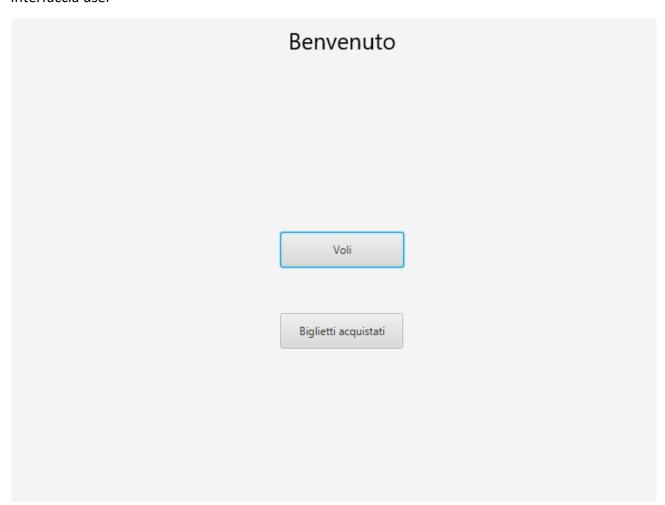
Per effettuare la connessione e le query al database è stata creata la classe singleton dbConnection

Interfaccia Utente:

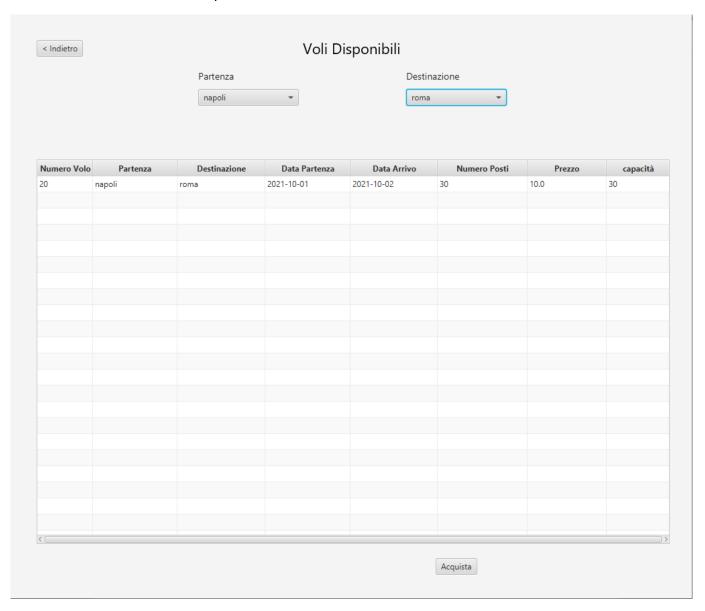
Login



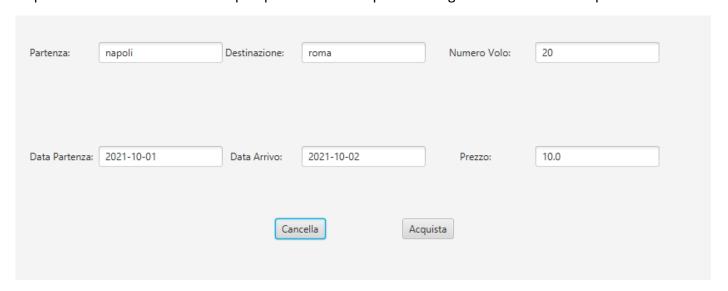
Interfaccia user



Interfaccia contenente i voli disponibili



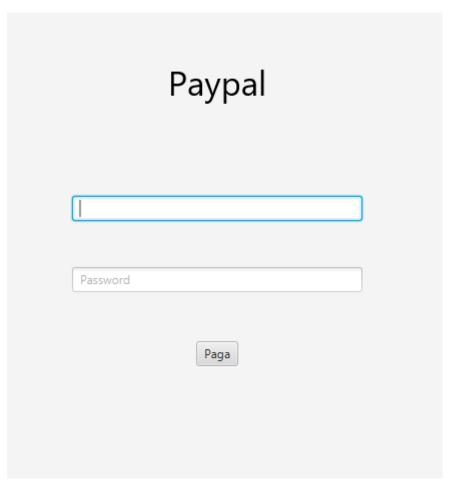
Dopo aver scelto un volo l'utente può procedere all'acquisto del biglietto cliccando su acquista



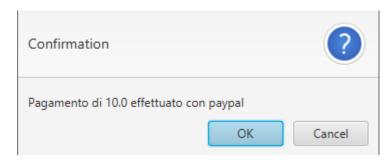
Scelta del metodo di pagamento



Inserimento dei dati per il pagamento



Conferma del pagamento



Interfaccia utente per visualizzare i biglietti acquistati

< Indietro

Biglietti Acquistati

Numero Volo	Partenza	Destinazione	Data Partenza	Data Arrivo
0	napoli	roma	2021-10-01	2021-10-02
}	milano	roma	2021-10-07	2021-10-27

Voli Disponibili

Numero Volo	Partenza	Destinazione	Data Partenza	Data Arrivo	Numero Posti	Prezzo
20	napoli	roma	2021-10-01	2021-10-02	29	10.0
37	roma	milano	2021-10-23	2021-10-24	100	19.0
38	milano	roma	2021-10-07	2021-10-27	0	9.0
39	milano	torino	2021-10-06	2021-10-10	100	23.0
<						

Form per l'aggiunta di un nuovo volo

Partenza:	Destinazione:	Prezzo:	
Data Partenza:	Data Arrivo:	Capacità:	
	Cancella	Aggiungi Volo	