

Progetto Ristorante

Gaetano Romeo - Vincenzo Figliolino - Domenico Riso

Febbraio 2024

Introduzione

Si vuole realizzare un sistema informatico per simulare la gestione di un ristorante.

I clienti devono poter autenticarsi, comunicare il numero di posti di cui hanno bisogno, ottenere un tavolo (se possibile), visualizzare il menu, effettuare una serie di ordini, richiedere il conto e pagare. Lo chef deve poter personalizzare a proprio piacimento il menù, aggiungendo e rimuovendo ordini e stilando il listino dei prezzi.

Il sistema deve effettuare il controllo delle disponibilità del ristorante, gestire il rilascio dei tavoli, consegnare gli ordini ai clienti e calcolare il conto di ciascun tavolo.

Deve inoltre garantire tempi di attesa e di utilizzo accettabili e fornire una modalità di pagamento al cliente.

Idea

L'idea per risolvere il problema proposto è di utilizzare Java per realizzare un software di gestione del ristorante e garantirne l'utilizzo a clienti e chef mediante la creazione di interfacce grafiche in JavaFX.

Per la memorizzazione permanente delle credenziali di accesso degli utenti e del menù verrà creato un database MySQL.

Per gestire le comunicazioni tra i vari processi verranno utilizzate le socket.

La gestione della preparazione e della consegna degli ordini sarà affidata a due processi Chef e Waiter, che genereranno un nuovo thread per gestire ogni richiesta separatamente.

Per il rilascio periodico dei tavoli verrà dichiarato uno scheduler.

Il controllo delle disponibilità del ristorante e l'effettiva preparazione e consegna degli ordini richiesti da parte dei clienti saranno affidati a entità gestite automaticamente dal sistema.

Attori

All'interno del progetto è possibile riconoscere quattro attori, ognuno dei quali ha proprie responsabilità e compiti.

- Il cliente, intorno al quale è incentrata la realizzazione del software, avrà accesso al sistema di prenotazione dei tavoli solo dopo essersi autenticato. Egli potrà visualizzare il menù, effettuare ordini e richiedere il conto una volta terminato. Tutte le azioni saranno permesse al cliente mediante interfacce grafiche;
- Lo chef avrà il compito di personalizzare il menù e preparare gli ordini che gli arriveranno. Il primo compito gli sarà garantito mediante un'interfaccia grafica, mentre il secondo sarà affidato al sistema;
- Il receptionist avrà il compito di gestire le richieste di prenotazione di tavoli dei clienti, visualizzando le disponibilità del ristorante e assegnando un tavolo ai clienti se possibile. Inoltre, si occuperà del rilascio periodico dei tavoli. La gestione dei compiti del receptionist è affidata al sistema;
- Il cameriere avrà il compito di prendere le ordinazioni dei clienti, portarle allo chef e riconsegnarle una volta pronte. La gestione dei compiti del cameriere è affidata al sistema.

Gestione Interfacce Grafiche

Per la gestione delle interfacce sono stati creati dei metodi che adattano dinamicamente la scena in funzione dello stato del sistema.

L'esecuzione del sistema per il cliente inizia con la visualizzazione dell'interfaccia di Login, dalla quale si può giungere a quella di Registrazione, se si clicca sull'apposito pulsante, o a quella per la richiesta dei posti, se si esegue correttamente l'accesso.

Per quanto riguarda lo chef, la prima interfaccia grafica visualizzata è quella che permette di personalizzare il menù.

Il meccanismo di cambio di scena, eseguito in seguito alla pressione dei pulsanti da parte dell'utente o all'occorrenza di eventi, prevede di selezionare la scena corrente catturando un suo elemento e di sostituirla, caricando il contenuto del file FXML di un'altra interfaccia grafica.

In Java è possibile specificare le azioni da compiere in relazione all'occorrenza di un evento, come l'aver cliccato su di un pulsante nell'interfaccia.

All'interno del progetto, è necessario catturare eventi scaturiti dall'interazione tra l'utente e l'interfaccia grafica.

Ciò è realizzato mediante la dichiarazione di un gestore degli eventi all'interno delle classi CustomerController e ChefController, che permette di definire le istruzioni da eseguire in relazione ad un evento, in questo caso l'aver cliccato su di un ordine del menù.

Ogni volta che sarà caricato il contenuto di un'interfaccia grafica, sarà necessario inizializzarne i campi.

Il metodo initialize delle classi CustomerController e ChefController sarà utilizzato per inizializzare le componenti della loro prima interfaccia grafica visualizzata.

Login Interface

L'interfaccia grafica di login, definita nel file LoginInterface.fxml, è la prima ad essere visualizzata dal cliente.

Essa permette al cliente di autenticarsi, mostrando dei campi di testo in cui inserire l'username e la password e un pulsante per eseguire l'accesso.

E' presente anche un collegamento all'interfaccia di registrazione, nel caso in cui il cliente non si sia mai autenticato.

Il campo della password è realizzato con un PasswordField per oscurare la password inserita.

Se l'accesso va a buon fine, al cliente è mostrata l'interfaccia grafica per richiedere i posti.

Register Interface

L'interfaccia grafica di registrazione, definita nel file RegisterInterface.fxml, permette al cliente di effettuare la registrazione al ristorante.

Analogamente all'interfaccia di login, mostra dei campi di testo in cui inserire l'username e la password, il pulsante per eseguire la registrazione, il collegamento all'interfaccia di login nel caso il cliente abbia già eseguito la registrazione una volta ed un Text per eventuali errori, ma presenta un ulteriore campo per la conferma della password.

Anche in questo caso, i campi delle password sono realizzati con un PasswordField.

Se la registrazione va a buon fine, al cliente è mostrata l'interfaccia grafica per il login.

Get Seats Interface

L'interfaccia grafica per richiedere i posti al receptionist e ottenere un tavolo è definita nel file `GetSeatsInterface.fxml`.

Essa contiene un campo di testo per permettere al cliente di comunicare il numero di posti di cui ha bisogno ed un pulsante per inviare la richiesta.

Nel caso in cui nel ristorante, al momento della richiesta, non siano disponibili sufficienti posti, nell'interfaccia comparirà un messaggio per indicare al cliente il tempo necessario da attendere affinché i posti richiesti si liberino, e due pulsanti per permettere al cliente di accettare l'attesa oppure andarsene.

Se la richiesta di posti va a buon fine, o il cliente decide di attendere il tempo necessario per il loro liberamento, ad esso è mostrata l'interfaccia grafica per effettuare gli ordini.

Get Orders Interface

L'interfaccia grafica per richiedere gli ordini e il conto è definita nel file `GetOrderInterface.fxml`.

Essa mostra al cliente il numero del suo tavolo, il totale da pagare, il menù come lista di ordini, la lista degli ordini effettuati ed un pulsante per richiedere il conto.

Per richiedere un ordine, basterà cliccarci sull'interfaccia e confermare nella finestra di dialogo.

Una volta richiesto il conto, il cliente visualizzerà un menù a tendina nel quale dovrà scegliere il metodo di pagamento tra quelli previsti.

Credit Card Payment Interface

L'interfaccia grafica per permettere al cliente di pagare il conto con carta di credito è definita nel file `CreditCardPaymentInterface.fxml`.

Essa contiene dei campi di testo in cui il cliente dovrà inserire i dati della propria carta di credito come numero, intestatario, scadenza e CVV ed un pulsante per confermare il pagamento.

Se il pagamento va a buon fine, al cliente è mostrato un messaggio di conferma.

Chef Interface

L'interfaccia grafica per permettere allo chef di personalizzare il menù è definita nel file `ChefInterface.fxml`.

Essa è strutturata in due parti.

La prima, situata in alto, permette allo chef di aggiungere un ordine al menù e comprende due campi di testo, uno per il nome dell'ordine e uno per il prezzo, ed un pulsante per confermare l'aggiunta.

La seconda, situata in basso, mostra allo chef il menù in tempo reale e comprende due pulsanti, uno per annullare le modifiche effettuate ed uno per confermarle.

Per rimuovere un ordine dal menù, basterà cliccarci sopra e confermare nella finestra di dialogo.

Accesso e Registrazione

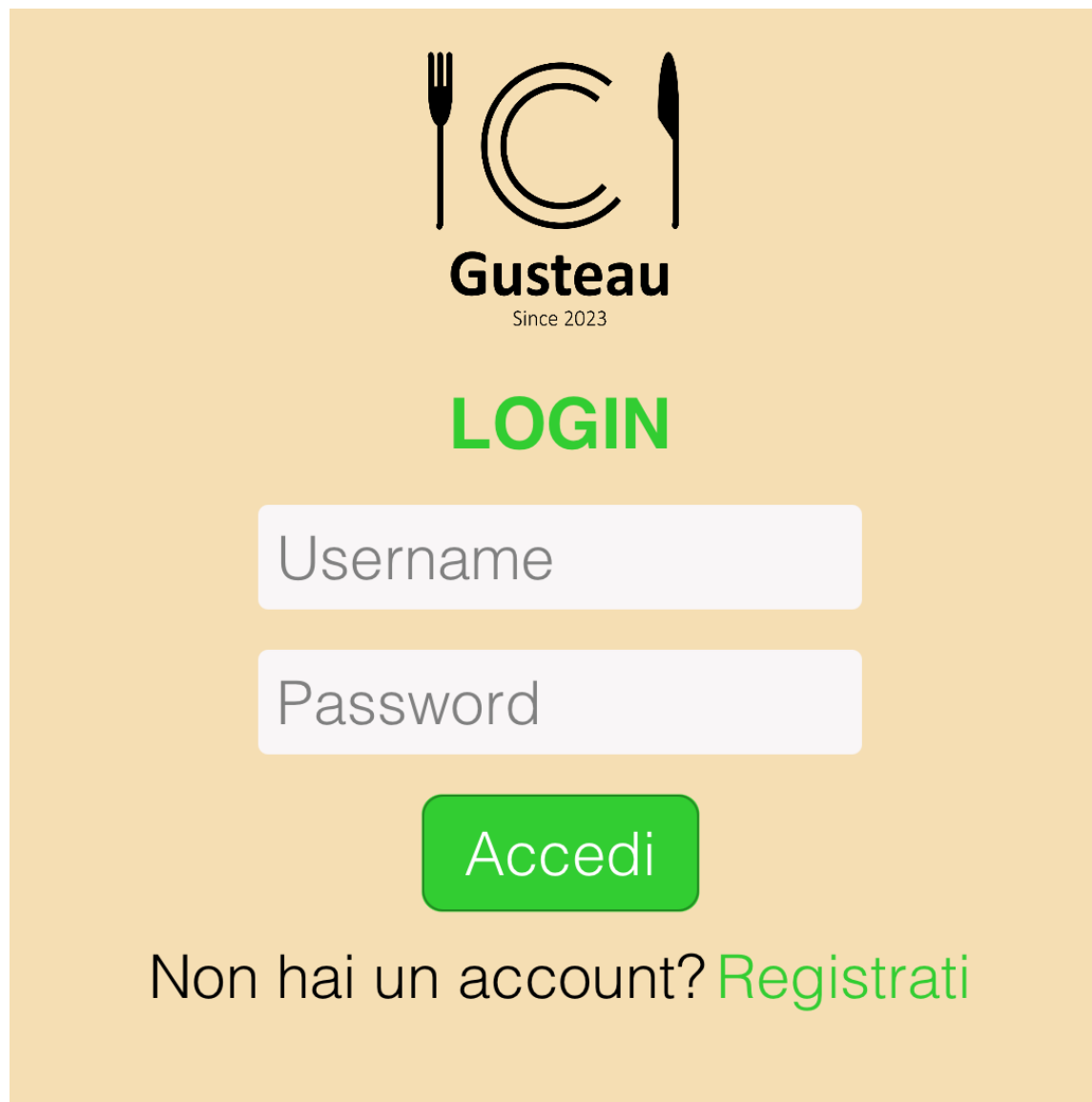
Per avere la possibilità di prenotare un tavolo, il cliente dovrà prima completare una fase di login. Egli dovrà comunicare un username univoco ed una password per la registrazione negli appositi campi dell'interfaccia grafica ed utilizzarli come credenziali di accesso per le successive volte.

Quando un cliente tenta di eseguire una registrazione, il sistema verifica se l'username dell'utente è disponibile e se l'utente non si è già registrato, eseguendo una query di ricerca all'interno della tabella Utenti del database.

Se l'username è già utilizzato, viene visualizzato un messaggio di errore.

Se invece l'utente completa la fase di registrazione, il sistema memorizza le sue credenziali all'interno della tabella Utenti del database mediante una query di inserimento e lo indirizza al form di login.

Quando il cliente tenta di eseguire il login, il sistema verifica se in precedenza ha effettuato la registrazione mediante le credenziali inserite, utilizzando una query di ricerca all'interno della tabella Utenti del database.



The image shows a login form for 'Gusteau'. At the top is the logo, which consists of a fork, a plate, and a knife arranged around a central 'C' shape, with the text 'Gusteau' and 'Since 2023' below it. Below the logo is the word 'LOGIN' in large green letters. There are two input fields: 'Username' and 'Password', both with light gray placeholder text. Below these fields is a green button with the text 'Accedi'. At the bottom, there is a link that says 'Non hai un account? Registrati' in green text.

Form di Login

Richiesta Posti

Una volta che il cliente ha effettuato l'accesso, deve comunicare al receptionist il numero di posti di cui ha bisogno.

Il sistema cattura il numero inserito dal cliente nell'apposito campo dell'interfaccia grafica e lo invia al receptionist.

Il receptionist ha a disposizione il numero di posti e di tavoli disponibili del ristorante, ed usa queste informazioni per accettare o meno la richiesta del cliente.

Quando un cliente gli comunica un numero di posti, esso controlla se c'è un tavolo disponibile con i posti necessari e, in caso positivo, assegna il primo tavolo disponibile al cliente, altrimenti comunica il tempo di attesa da rispettare affinché si liberino dei posti.

A questo punto, il cliente potrà decidere se attendere e ottenere un tavolo al termine, oppure andarsene.

Per gestire il liberamento dei posti occupati, nel receptionist è definito uno scheduler che ad intervalli di tempo aleatori, che simulano l'abbandono del ristorante da parte dei clienti, libera i tavoli.

Un altro scheduler è utilizzato per calcolare, sempre in maniera casuale, e comunicare al cliente il tempo di attesa che deve rispettare il cliente affinché si liberino dei tavoli.

Lo scheduler farà in modo che se il cliente accetta di attendere, al termine dell'attesa otterrà un posto, simulando una prenotazione.

Per evitare tempi di attesa lunghi, l'unità di tempo utilizzata è il secondo.



Richiesta Posti



Esempio di Attesa

Ordinazioni

Una volta preso posto e scannerizzato il menù, il cliente può ordinare tutto ciò che vuole cliccando sugli ordini del menù dall'interfaccia grafica.

Il sistema cattura l'ordine selezionato, lo invia al cameriere e, una volta pronto, aggiunge l'ordine alla lista di ordini effettuati dal cliente e il suo prezzo al totale da pagare.

Quando ha terminato le ordinazioni, il cliente può richiedere il conto mediante l'apposito pulsante.

La scelta dell'ordine dal menù permette di evitare di controllare che l'utente cerchi di ordinare un piatto non preparato nel ristorante.

Siccome si tratta di una simulazione, gli ordini richiederanno un tempo di preparazione nullo e giungeranno in tempo reale al cliente.

TAVOLO TOTALE
1 €10,80

MENU

pasta al sugo	€5,50
pasta e cavolfiore	€9,50
pasta e fagioli	€5,80
sciallatielli ai frutti di mare	€5,00

ORDINI EFFETTUATI

pasta e fagioli
sciallatielli ai frutti di mare

Chiedi il conto

Ordinazioni

Personalizzazione Menu

Lo chef ha la possibilità di personalizzare a proprio piacimento il menù e di stilare i prezzi degli ordini.

Ogni volta che inserisce un ordine e il relativo prezzo negli appositi campi dell'interfaccia grafica, il sistema, prima di inserire l'ordine nel menù, controlla che non sia già presente eseguendo una query di ricerca all'interno della tabella Ordini del database.

Durante la sua scrittura, lo chef avrà la possibilità di visualizzare in tempo reale il menù.

Per rimuovere un ordine dal menù, basterà cliccare su di esso.

Una volta che ha terminato le proprie modifiche, lo chef potrà confermarle o annullarle.

Nel primo caso, le modifiche diventeranno effettive e permanenti mediante query di inserimento o cancellazione degli ordini del menù all'interno della tabella Ordini del database.

Memorizzare il menù all'interno del database garantisce al cliente una copia, anche se vecchia, dello stesso anche nel caso in cui lo chef abbia avuto problemi nel modificarlo.

ORDINE

€

Conferma piatto

MENU

pasta al sugo	€5,50
pasta e cavolfiore	€9,50
pasta e fagioli	€5,80
sciallatielli ai frutti di mare	€5,00

Annulla modifiche

Conferma menu

Personalizzazione Menu

Modalità di Pagamento

I clienti potranno decidere di pagare in contanti, comunicando il numero del tavolo in cassa, oppure mediante carta di credito.

La scelta sarà possibile mediante una finestra di dialogo.

Nel caso si scegliesse la modalità di pagamento con carta di credito, il cliente visualizzerà un'interfaccia grafica per l'inserimento dei dati della carta, che verranno opportunamente controllati dal sistema.

Per questioni di privacy, i dati delle carte degli utenti non sono memorizzati.



Dati Carta

Numero

Intestatario

Data Scadenza

CVV

Paga

Esempio di Pagamento

Gestione delle Password

Per garantire sicurezza, le password che i clienti utilizzano per l'accesso non saranno memorizzate in chiaro all'interno del database, nè saranno visibili al momento dell'inserimento nel campo dell'interfaccia.

Verrà infatti utilizzato un algoritmo di hashing come forma di crittografia.

L'uso dell'algoritmo di hashing, che è irreversibile (non è possibile tornare alla password originale partendo solo da quella crittografata) e rende ipoteticamente impossibile a chi non possiede la password di potervi accedere.

Siccome un algoritmo di hashing è deterministico (a parità di password iniziale genera la stessa sequenza esadecimale), quando un cliente tenterà di eseguire la fase di login, il sistema, prima di verificare che le sue credenziali siano corrette ricercandole all'interno del database, utilizzerà lo stesso algoritmo di hashing applicato durante la fase di registrazione per convertire la password.

La mancata visibilità della password al momento dell'inserimento è garantita dall'elemento PasswordField di JavaFX.

Per garantire un'ulteriore sicurezza, gli utenti dovranno utilizzare password contenenti almeno 8 caratteri, tra cui almeno un carattere speciale, una lettera maiuscola e un numero.

Tale controllo è eseguito mediante l'utilizzo di una espressione regolare.

Rispettare lo standard aiuta a ridurre le probabilità che un utente malintenzionato si appropri della password di un altro utente e aumenta la difficoltà di un approccio brute-force.

SQL Injections

Le SQL Injections sono un tipo comune di attacco informatico con la quale un utente malintenzionato accede a dati riservati memorizzati all'interno di un database mediante comandi SQL.

Per far fronte a questo problema, all'interno del codice del progetto non sono presenti query concatenate ma query parametrizzate.

Quando viene eseguita una query per cercare una determinata informazione all'interno del database, la condizione non viene esplicitata ma rappresentata dal simbolo "?", che verrà in seguito sostituito mediante l'utilizzo di appositi statement.

Errori Accidentali

Il sistema deve far fronte al caso, assolutamente plausibile, in cui un utente esegua un'azione indesiderata.

Per evitare che ciò avvenga, tutte le operazioni fornite dal sistema all'utente saranno controllate da una finestra di conferma.

Un ulteriore controllo effettuato, per verificare che l'utente inserisca correttamente le credenziali di accesso al momento della registrazione, è quello della conferma della password.

Al momento della registrazione, oltre ad inserire la password, il cliente dovrà confermarla in un altro campo.

Se le due password non combaceranno, il sistema lo segnalerà con un messaggio di errore.

Design Patterns

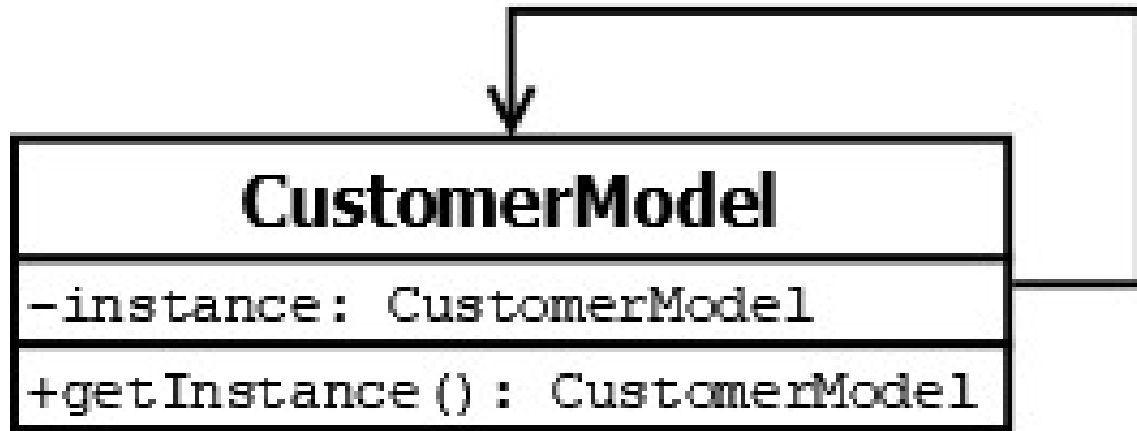
Per garantire estendibilità e manutenibilità al software, sono stati applicati i seguenti design pattern.

Singleton

Il pattern Singleton è utilizzato per garantire un unico punto di accesso ad un oggetto.

Il suo utilizzo è presente all'interno della classe CustomerModel per istanziare solo una volta un oggetto di tale classe e poter applicare la logica funzionale ai componenti presenti all'interno della classe CustomerController.

E' stata utilizzata la Lazy Initialization.



Singleton

Model View Controller

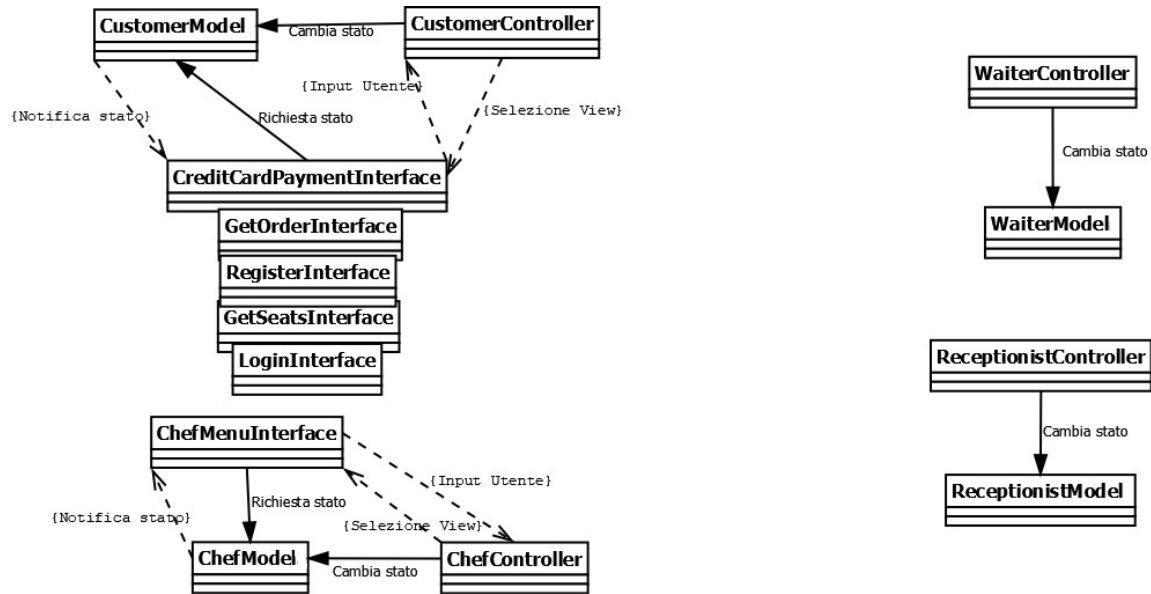
Il pattern MVC è utilizzato per separare le componenti software tra funzionalità di business, logica di presentazione e controlli delle funzionalità.

All'interno del progetto, le classi Controller rappresentano i controlli delle funzionalità e permettono di richiamare una funzionalità di business in relazione ad eventi che l'utente scatuisce interagendo con le componenti dell'interfaccia grafica.

Le classi Model rappresentano la logica di business, ovvero le funzionalità implementate dai componenti presenti nelle classi Controller.

Le classi View rappresentano la visualizzazione effettiva delle interfacce e sono rappresentate dai file FXML.

Nelle classi Controller è presente un riferimento al Model.



MVC

Memento

Il pattern Memento è utilizzato per memorizzare lo stato precedente di un oggetto e permetterne l'accesso senza violare il principio di incapsulamento.

Nel progetto, esso è utilizzato per memorizzare la versione iniziale del menù, e per permettere di annullare eventuali modifiche apportate dallo chef.

L'interfaccia MenuMemento dichiara il metodo da utilizzare per ritornare allo stato precedente del menù.

La classe MenuOriginator memorizza lo stato corrente del menù.

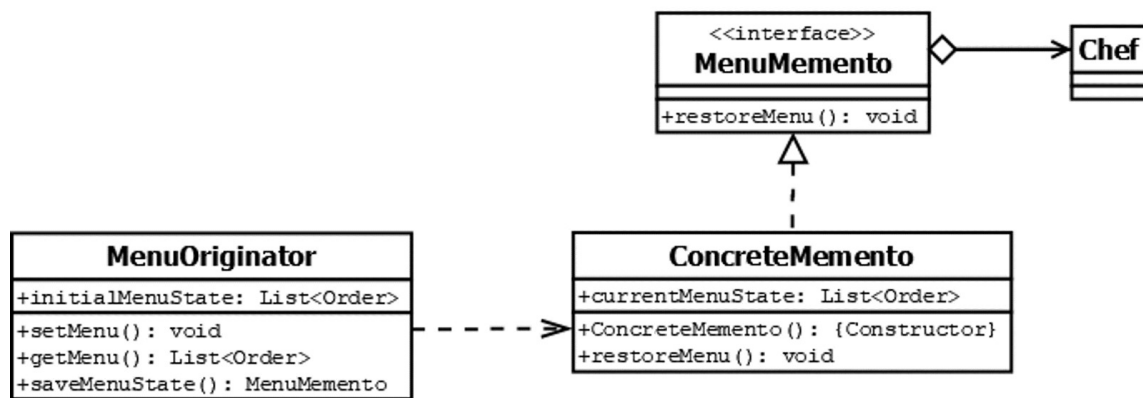
La classe ConcreteMenuMemento implementa l'interfaccia MenuMemento e permette l'accesso allo stato precedente del menù.

Quando è eseguita l'interfaccia grafica per la personalizzazione del menù, viene salvata una copia del menù mediante un oggetto MenuOriginator.

Le modifiche adoperate dallo chef influenzeranno solo questa copia.

Se lo chef decide di confermare il menù, per applicare le modifiche la copia memorizzata con l'oggetto MenuOriginator è memorizzata all'interno del database.

Se lo chef decide di annullare le modifiche, viene chiamato il metodo restoreMenu della classe MenuMemento per riportare il menù allo stato iniziale.



Memento

State

Il pattern State è utilizzato per modificare il comportamento di un oggetto in funzione del suo stato interno.

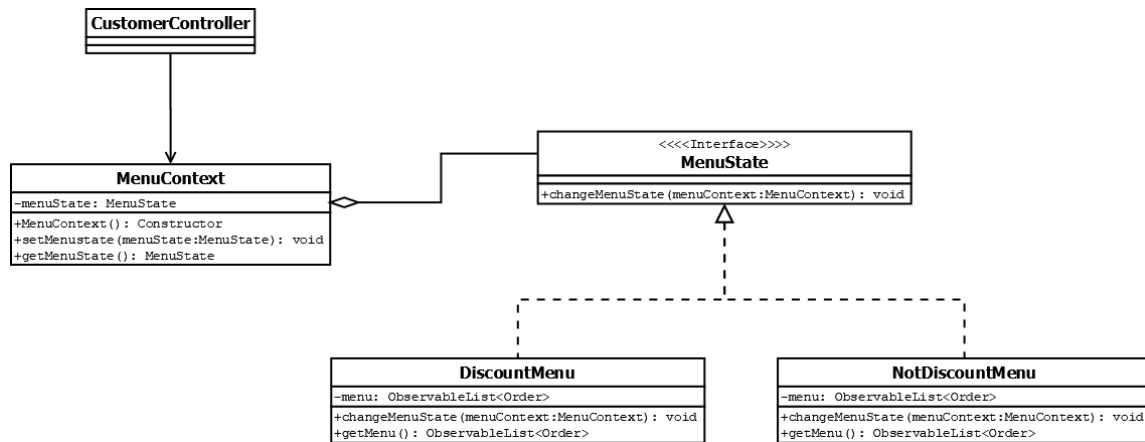
Nel progetto è stato utilizzato per visualizzare una versione scontata del menù durante il weekend. L'interfaccia MenuState dichiara il metodo da utilizzare per modificare lo stato del menù da scontato a non e viceversa.

La classe MenuContext memorizza lo stato del menù e permette di modificarlo.

Le classi DiscountMenu e NotDiscountMenu implementano l'interfaccia MenuState e rappresentano il comportamento che assume il menù in relazione al suo stato.

Nella classe CustomerController è definito un oggetto di tipo MenuContext per memorizzare lo stato del menù (inizialmente non scontato).

Quando viene eseguito il metodo showMenu per rendere visualizzabile il menù al cliente, il sistema controlla il giorno corrente e in base ad esso utilizza il metodo setMenuState per impostare il menù a scontato durante il weekend e a non scontato durante la settimana.



State

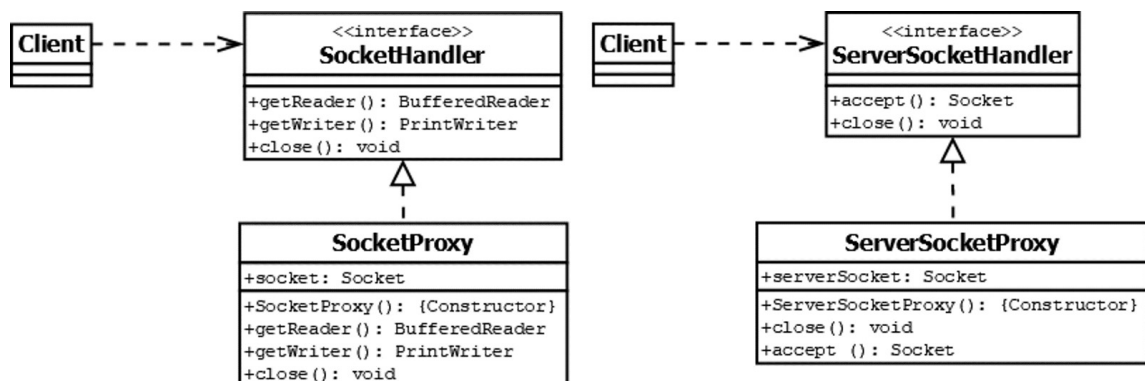
Proxy

Il pattern Proxy è utilizzato per fornire un'interfaccia per controllare l'accesso ad un oggetto.

All'interno del progetto è stato utilizzato per controllare l'accesso alle socket, garantendo una gestione sicura delle connessioni e del loro utilizzo.

Le interfacce SocketHandler e ServerSocketHandler dichiarano i metodi di chiusura della socket e di ottenimento degli stream di input e output.

Le classi SocketProxy e ServerSocketProxy implementano tali interfacce e i loro metodi.



Proxy

Factory Method

Il pattern Factory Method è utilizzato per nascondere la logica implementativa di un oggetto e per creare una famiglia di prodotti.

Nel progetto è utilizzato per rappresentare gli ordini che fanno parte del menù.

L'interfaccia Order dichiara il metodo da utilizzare per visualizzare l'ordine sottoforma di string nome-prezzo all'interno del menù.

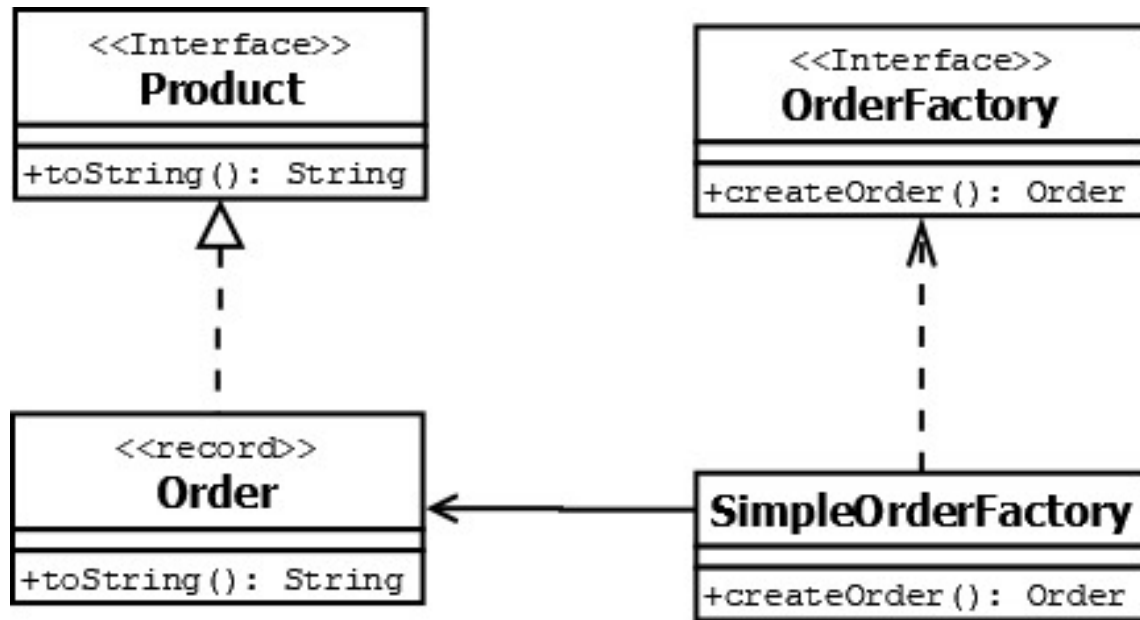
L'interfaccia OrderFactory dichiara il metodo da utilizzare per creare un ordine.

La classe SimpleOrderFactory implementa l'interfaccia OrderFactory e il suo metodo.

La classe ConcreteOrder implementa l'interfaccia Order e il suo metodo e memorizza nome e prezzo degli ordini.

All'interno della classe ChefController è dichiarato un oggetto di tipo SimpleOrderFactory.

Quando lo chef inserisce il nome e il prezzo di un ordine all'interno dell'interfaccia grafica, viene chiamato il metodo createOrder dell'oggetto SimpleOrderFactory, passandogli come parametri il nome e il prezzo inseriti dallo chef.



Factory Method

Strategy

Il pattern Strategy è utilizzato per definire una famiglia di algoritmi intercambiabili che portano allo stesso risultato.

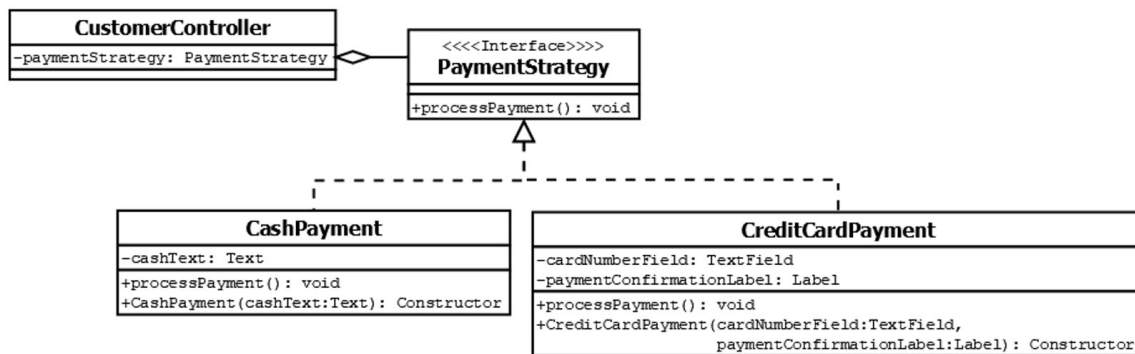
All'interno del progetto è utilizzato per garantire diversi metodi di pagamento ai clienti.

L'interfaccia PaymentStrategy dichiara il metodo da utilizzare per eseguire il pagamento.

Le classi CashPayment e CreditCardPayment implementano l'interfaccia PaymentStrategy e il suo metodo per permettere pagamenti in contanti e con carta di credito.

Quando il cliente richiede il conto, in base al metodo di pagamento che seleziona verrà utilizzato l'algoritmo adatto alla gestione del pagamento.

Supponendo che il cliente scelga di pagare in contanti, viene richiamato il metodo processPayment della classe CashPayment, che comunica al cliente di presentarsi in cassa con il numero del tavolo. Se invece il cliente decide di pagare con carta, viene richiamato il metodo processPayment della classe CreditCardPayment, che mostra al cliente l'interfaccia grafica per inserire i dati della carta.



Strategy

Compilazione ed Esecuzione (IntelliJ per MacOS)

Per il corretto funzionamento del software, è stata realizzata una classe Main che si occupa di eseguire i processi nel corretto ordine.

Nel caso si volesse verificare il comportamento del progetto in maniera autonoma, è opportuno rispettare il seguente ordine di esecuzione dei processi:

- Per gestire correttamente l'accesso e la registrazione da parte del cliente, è necessario che il database server sia attivo sulla macchina di utilizzo;
- Per permettere al cliente di richiedere i posti, il processo Receptionist deve essere eseguito prima della comunicazione dei posti all'interno dell'interfaccia grafica GetSeatsInterface.fxml;
- Per permettere al cliente di effettuare ordini, il processo Waiter deve essere eseguito prima che il cliente clicchi sugli ordini nell'interfaccia grafica GetOrderInterface.fxml;
- Per potersi collegare correttamente al processo Chef, il processo Waiter deve essere eseguito solamente dopo che lo chef ha terminato la personalizzazione del menù dall'interfaccia grafica ChefInterface.fxml.

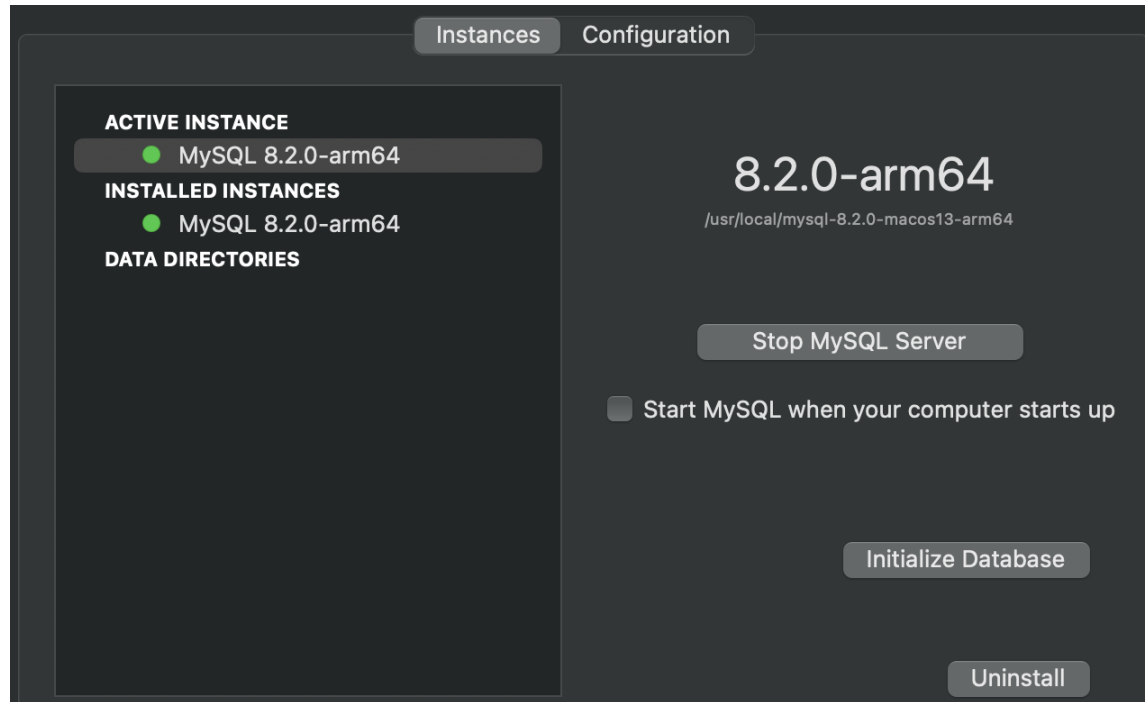
Il progetto necessita di alcuni plugin per essere compilato ed eseguito correttamente.

Creazione e Accesso al Database (MySQL per MacOS)

Per la gestione del database è necessaria l'installazione di un DBMS.

Come prima cosa va eseguito il server, cliccando sul pulsante "Start MySQL Server" e creato il database per la memorizzazione delle credenziali dell'utente e del menù in locale, cliccando sul pulsante "Initialize Database".

Nella schermata che comparirà, bisognerà stabilire una password per l'accesso al database.



Ora bisogna aprire MySQL Workbench e creare una connessione al server del database.

Nella seguente finestra, bisogna assegnare un nome arbitrario alla connessione (opportuno stabilire lo stesso nome del database), impostare il tipo di connessione (standard TCP/IP), inserire come Hostname l'indirizzo IP del dispositivo sul quale è eseguito il server (in questo caso "localhost" o 127.0.0.1 in quanto il server è in locale), stabilire la porta da utilizzare (di default la 3306) e l'username dell'utente con il quale si vuole eseguire l'accesso (lasciare "root" per indicare che si è l'amministratore del database).

Creata la connessione e inizializzato il database, è tempo di inizializzare le tabelle eseguendo il codice memorizzato nel file DatabaseCreationScript.sql.

Ora il database è pronto e per verificarne il funzionamento si possono eseguire query di esempio.

Setup New Connection

Connection Name: Type a name for the connection

Connection Method: **Standard (TCP/IP)** Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.

Esempio di creazione della connessione

Query 1

Limit to 1000 rows

```

1  USE RISTORANTE;
2
3  SELECT * FROM ORDINI;

```

100% 22:3

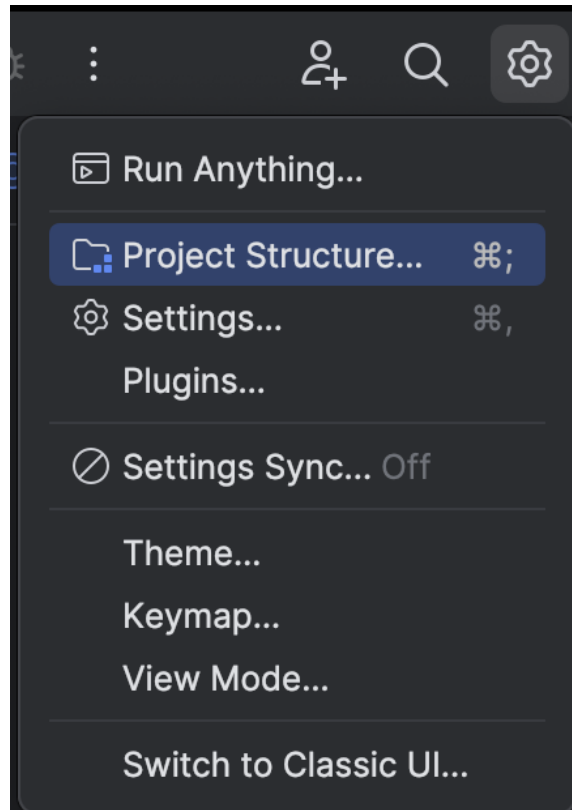
Result Grid Filter Rows: Search Edit:

	NOME	PREZZO
	pasta al sugo	8.50
	sciallatielli ai frutti di mare	10.50
	NULL	NULL

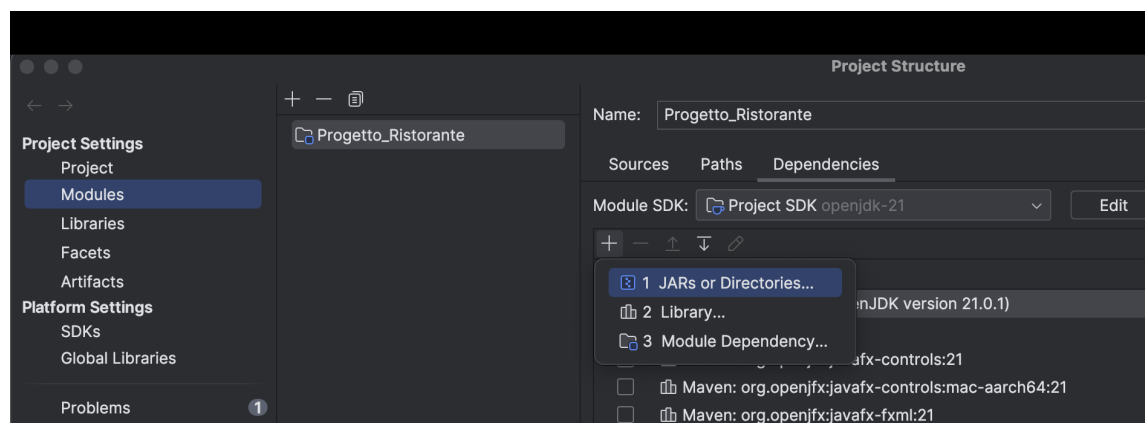
Esempio di query

JavaFX

Per il funzionamento delle interfacce grafiche sono necessarie le librerie di JavaFX. Successivamente bisogna includere le librerie scaricate nella struttura del progetto. Con IntelliJ aperto, aprire le Impostazioni in alto a destra e selezionare l'opzione "Project Structure" dal menù a tendina che compare.



Ora bisogna selezionare la sezione "Modules", cliccare sul simbolo "+", selezionare l'opzione "JARs or Directories" e selezionare il file JAR scaricato precedentemente per JavaFX.



Connector

Per garantire la connessione al database è necessario il plugin Connector.

La procedura per includere il plugin nel progetto è la medesima utilizzata per includere la libreria JavaFX.

Per verificare che la connessione al database e che l'inclusione della libreria JavaFX siano andate a buon fine, si può eseguire una tra le classi Customer.java e Chef.java ed utilizzare il software.