

Facoltà di Ingegneria - Ingegneria informatica, elettronica e delle telecomunicazioni.

Ingegneria del Software 2022-2023

Online Wine Shop

Relazione progetto Ingegneria del Software

Alessandro Zoppi 318814 – Federico Zilioli 318817

28/02/2023

INDICE

Requisiti informali

L’obiettivo è sviluppare (usando il linguaggio Java) e documentare un sistema software per la vendita online di bottiglie di vino, utilizzando in modo appropriato le tecniche di riferimento della programmazione orientata agli oggetti. Il sistema realizzato sarà installato presso l’azienda che ha finanziato la realizzazione del sistema. Il sistema interagisce con clienti (persone che vogliono acquistare del vino), impiegati (persone che gestiscono la vendita e la spedizione e ricezione dei prodotti) e un amministratore (persona che controlla il lavoro dei dipendenti e gestisce le attività finanziarie dell’azienda). I clienti sono identificati da: nome, cognome, codice fiscale, indirizzo email, numero telefonico e indirizzo di consegna. Gli impiegati e l’amministratore sono identificati da: nome, cognome, codice fiscale, indirizzo email, numero telefonico e indirizzo di residenza. Per le consegne dei vini ai clienti e per la fornitura dei vini, l’azienda si serve dei servizi di un corriere e di un fornitore. Queste due persone sono identificate da: nome, cognome, indirizzo e-mail, numero telefonico, e codice fiscale e indirizzo della loro azienda. I vini sono identificati dai seguenti attributi: nome, produttore, provenienza, anno, note tecniche, e i vitigni da cui derivano. Ogni bottiglia ha un prezzo che dipende dalla qualità del vino e dal suo numero di vendite. Per poter utilizzare il sistema, un nuovo cliente deve registrarsi sul sistema inserendo nome utente e password e le altre informazioni richieste dal sistema. Un cliente registrato può fare ricerca dei vini per nome e anno di produzione e acquistare bottiglie di vino dopo un accesso autenticato. Ogni volta che un cliente accede nel sistema, il sistema informa il cliente sulle promozioni in corso. Anche gli impiegati e l’amministratore devono accedere al sistema tramite nome utente e password. e possono modificare la loro password. In aggiunta, l’amministratore ha una password iniziale predefinita che può comunque modificare. Inoltre l’amministratore deve registrare (con nome utente e password) gli altri dipendenti, e resettare e cancellare i loro dati di registrazione quando diventa necessario. Gli impiegati e l’amministratore possono fare la ricerca dei clienti per cognome, dei vini per nome e/o anno di produzione, degli ordini di vendita e di acquisto e delle proposte di acquisto per intervallo di date. Inoltre, l’amministratore deve preparare un report mensile sullo stato dell’azienda (introiti, spese, numero bottiglie vendute e disponibili alla vendita, numero di vendite per i diversi vini, valutazione dei dipendenti). Un cliente può acquistare le bottiglie di vino con confezioni da 1 a 5 bottiglie e da casse da 6 e 12 bottiglie. Ognuna di queste casse contiene solo bottiglie dello stesso tipo di vino. La vendita di casse di vino di 6 e 12 bottiglie è valorizzata da uno sconto del 5 e 10%. La vendita di più di una cassa è valorizzata da un ulteriore sconto del 2 e 3%. Il sistema deve tenere traccia per ogni tipo di vino del numero di bottiglie vendute e di quelle ancora disponibili per la vendita. Un cliente può visualizzare i vini in vendita e selezionare uno o più vini e, per ogni vino, selezionare il numero di bottiglie (default 1). Selezionate le bottiglie, il cliente può decidere di acquistarle. Il sistema permette questa operazione fornendo la possibilità di pagare tramite un bonifico o una carta di credito. Fatto il pagamento, il cliente riceve un ordine di vendita contenente le informazioni su: il cliente che ha fatto l’acquisto, i vini acquistati (nome vino, numero bottiglie e prezzo), l’indirizzo di consegna e la data di consegna. Quando non c’è un numero sufficiente di bottiglie di uno o più vini per soddisfare la richiesta del cliente del numero di bottiglie che vorrebbe acquistare: quando le bottiglie richieste saranno disponibili, il sistema notificherà la loro disponibilità all’utente. Nel caso in cui il numero di bottiglie di uno o più vini non riesce a soddisfare la richiesta del cliente, allora il cliente può compilare una proposta di acquisto contenente: i vini che vuole acquistare (nome vino, numero bottiglie) e l’indirizzo di consegna. Ricevuta la proposta di acquisto, il sistema invia una copia dell’ordine di vendita ad uno degli impiegati. Questo impiegato preparerà un ordine di acquisto e lo invia al fornitore. Ricevuto l’ordine, il fornito prepara il materiale ed effettua la consegna. Ricevuto il materiale dal fornitore, l’impiegato chiede al sistema di generare un ordine di vendita partendo dai dati della proposta di acquisto, dai prezzi dei vini e dalla data di consegna (definita con il corriere) e invia l’ordine di vendita. Ricevuto l’ordine di vendita, il cliente può decidere se acquistarlo o no. Se si, il cliente deve pagare tramite il sistema; se no, il cliente deve annullare l’acquisto. Il sistema e il dipendente che ha seguito la possibile vendita devono completare le attività necessarie in base alla scelta fatta dal cliente. Il sistema mantiene e gestisce gli ordini di vendita ricevuti dai clienti. In particolare, ogni volta che viene effettuato un acquisto, il sistema aggiorna il numero di vendita e disponibilità dei vini, e invia una copia dell’ordine di vendita ad uno degli impiegati. Questo impiegato deve esaminare l’ordine, preparare il materiale per la spedizione, concordare con il corriere la data di prelievo e consegna del materiale e informare il cliente sulla data di consegna. Dopo la consegna del materiale al corriere, il dipendete deve completare l’ordine di vendita firmandolo digitalmente. Il sistema deve garantire che per ogni tipo di vino ci sia sempre un numero ragionevole di bottiglie. Per gestire questo vincolo, ad ogni arrivo di un ordine di acquisto il sistema controlla se il numero di bottiglie di alcuni vini è sceso sotto la soglia (il valore della soglia può dipendere dal vino e dai suoi numeri di vendita). Se ciò avviene, il sistema informa uno dei dipendenti sulle quantità disponibili di questi vini. Ricevuta l’informazione, l’impiegato preparerà un ordine di acquisto e lo invia al fornitore. Ricevuto l’ordine, il fornito prepara il materiale ed effettua la consegna. Ricevuto il materiale dal fornitore, l’impiegato deve completare l’ordine di acquisto firmandolo digitalmente, e caricare l’ordine sul sistema. Il sistema elabora l’ordine e aggiorna le quantità di vini. Il sistema è basato su un’architettura client server. Il server deve supportare l’accesso in parallelo di diversi nodi client. L’interfaccia grafica dei client deve essere implementata con JavaFX. Il database del sistema deve mantenere le informazioni di almeno una decina di vini differenti. Il numero di impiegati al lavoro deve essere almeno tre e il sistema assegna le attività agli impiegati con una politica “round-robin” ovviamente tenendo conto degli impiegati al momento impegnati. Se l’attività non è terminata entro un tempo prestabilito, il sistema l’assegna ad un altro impiegato. Il sistema mantiene le informazioni sulle attività completate e non completate dai dipendenti.

|  |
| --- |
| Attore: cliente |
| Registrarsi sul sistema inserendo nome utente, password e le altre informazioni richieste dal sistema. |
| Fare ricerca dei vini per nome e anno di produzione |
| Acquistare bottiglie di vino dopo l'accesso autenticato |
| Ricevere informazioni sulle promozioni in corso ogni volta che accede nel sistema. |
| Acquistare bottiglie di vino in confezioni da 6, 12, o sfuse |
| Fare proposte di acquisto che possono essere gestite dagli impiegati |

|  |
| --- |
| Attore: impiegato |
| Accedere al sistema tramite nome utente e password. |
| Fare ricerca dei clienti per cognome, dei vini per nome e/o anno di produzione, degli ordini di vendita e di acquisto e delle proposte di acquisto per intervallo di date. |
| Gestire le proposte di acquisto ricevute dai clienti. |
| Gestire gli ordini di acquisto dei vini dai fornitori. |
| Aggiornare gli ordini di acquisto man mano che i vini vengono consegnati all'azienda. |

|  |
| --- |
| Attore: amministratore |
| Accedere al sistema tramite nome utente e password. |
| Registrare gli altri dipendenti, resettare e cancellare i loro dati di registrazione quando necessario. |
| Preparare un report mensile sullo stato dell'azienda. |
| Aggiungere nuovi vini, modificare o eliminare quelli esistenti e visualizzare le informazioni dettagliate su ogni vino. |
| Gestire le informazioni finanziarie dell'azienda. |
| Visualizzare il bilancio dell'azienda, le entrate e le uscite, le spese sostenute per l'acquisto dei vini e per l'invio dei prodotti ai clienti. |
|  |

2.1 Stakeholder principali (utenti):

|  |  |  |
| --- | --- | --- |
| Clienti: possono inviare proposte di acquisto di vini dall'azienda. Hanno accesso alle informazioni sui vini offerti e alle notifiche sullo stato delle loro richieste. | Impiegati: gestiscono le proposte di acquisto dei clienti, gestiscono gli ordini dai. Hanno anche accesso alle informazioni sul catalogo dei vini e sugli acquisti | Amministratore: gestisce il catalogo dei vini offerti dall'azienda, monitora le informazioni finanziarie |

Specifica dei requisiti NON funzionali

|  |
| --- |
| Di seguito è riportato l’elenco dei requisiti non funzionali del sistema.   * Usabilità: l’interfaccia utente del sistema è stata implementata cercando di garantire la massima operabilità, un veloce apprendimento e una facile localizzazione dei comandi da utilizzare. Viene garantita inoltre un’interfaccia coerente in tutte le sezioni  dell’applicazione. * Sicurezza: l’applicazione gestisce informazioni sensibili, pertanto deve garantire un determinato livello di sicurezza per preservarle. È stata perciò implementata una procedura di autenticazione che permette di separare i diversi profili utente garantendo in questo modo diversi livelli di privilegi e di funzioni utilizzabili. |

CASI D’USO



Login

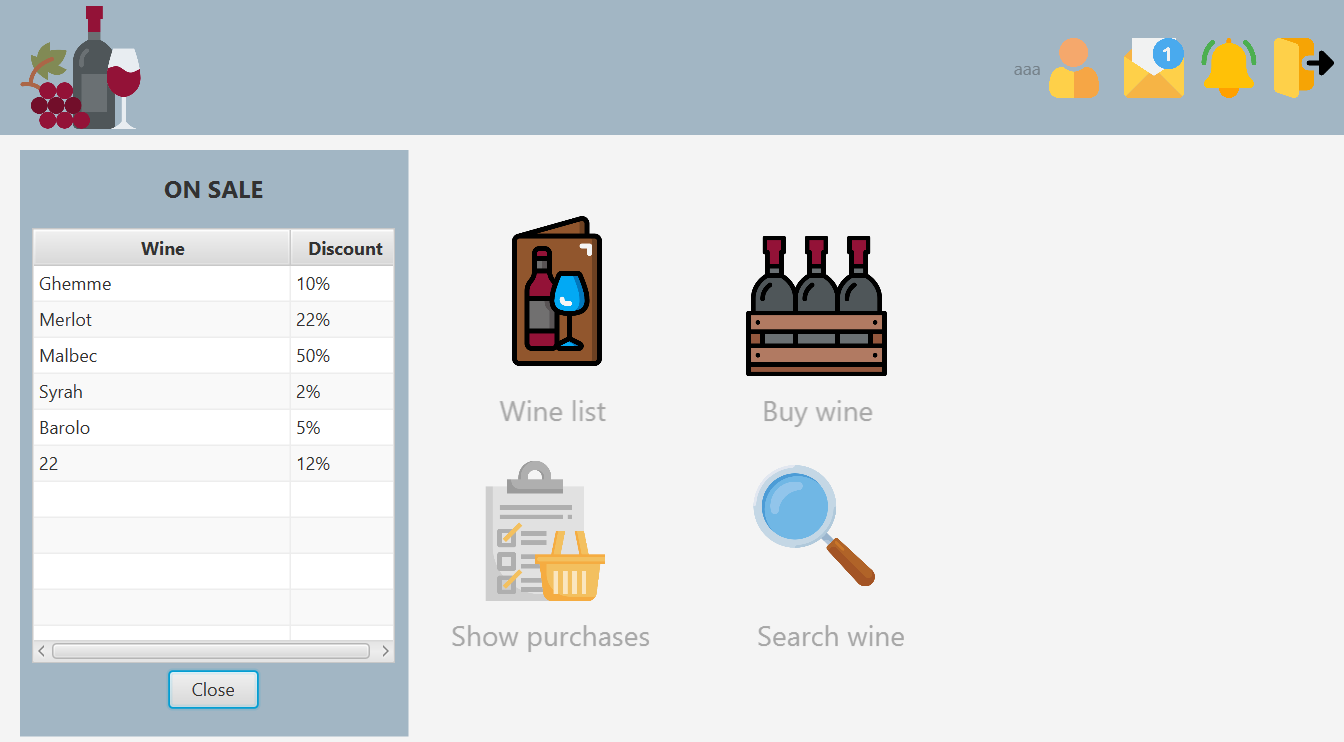
* Attori: Utente
* Scopo: Autenticare l'utente nel sistema
* Flusso principale:
  1. L'utente inserisce nome utente e password
  2. Il sistema verifica le credenziali e accede alla home page dell'utente in base al ruolo (cliente, dipendente o amministratore)
* Flusso alternativo:
  1. L'utente inserisce credenziali errate e il sistema restituisce un messaggio di errore

1. Recupero password

* Attori: Utente
* Scopo: Permettere all'utente di recuperare la password dimenticata
* Flusso principale:
  1. L'utente seleziona il link per il recupero password
  2. Il sistema mostra la pagina di recupero password
  3. L'utente inserisce username e numero di telefono associato all'account
  4. Il sistema mostra la password password
* Flusso alternativo:
  1. L'utente inserisce un indirizzo username o numero di telefono non valido e il sistema restituisce un messaggio di errore

1. Registrazione nuovo utente

* Attori: Utente
* Scopo: Registrare un nuovo utente nel sistema
* Flusso principale:
  1. L'utente seleziona il link per la registrazione
  2. Il sistema mostra la pagina di registrazione
  3. L'utente inserisce i propri dati personali
  4. Il sistema verifica la validità dei dati evitando username duplici e crea l'account utente
* Flusso alternativo:
  1. L'utente inserisce dati non validi e il sistema restituisce un messaggio di errore



1. Use case: Effettuare un acquisto online

Attori: Cliente, Sistema di pagamento, Magazzino

Flusso principale:

* Il cliente accede al sito web e naviga fino al prodotto che vuole acquistare
* Il cliente aggiunge il prodotto al carrello
* Il cliente accede alla pagina di checkout e inserisce le informazioni di pagamento
* Il sistema di pagamento elabora le informazioni e verifica la disponibilità del prodotto nel magazzino
* Il magazzino verifica la disponibilità del prodotto e lo prepara per la spedizione
* Il sistema di pagamento autorizza la transazione
* Il magazzino spedisce il prodotto al cliente

Flusso alternativo:

* Il sistema di pagamento rifiuta la transazione a causa di informazioni di pagamento errate
* Il prodotto non è disponibile nel magazzino e il cliente viene informato
* Il magazzino spedisce un prodotto difettoso o sbagliato e il cliente deve richiedere un rimborso o una sostituzione

1. Use case: Prenotare una stanza d'albergo

Attori: Cliente, Receptionist, Sistema di prenotazione

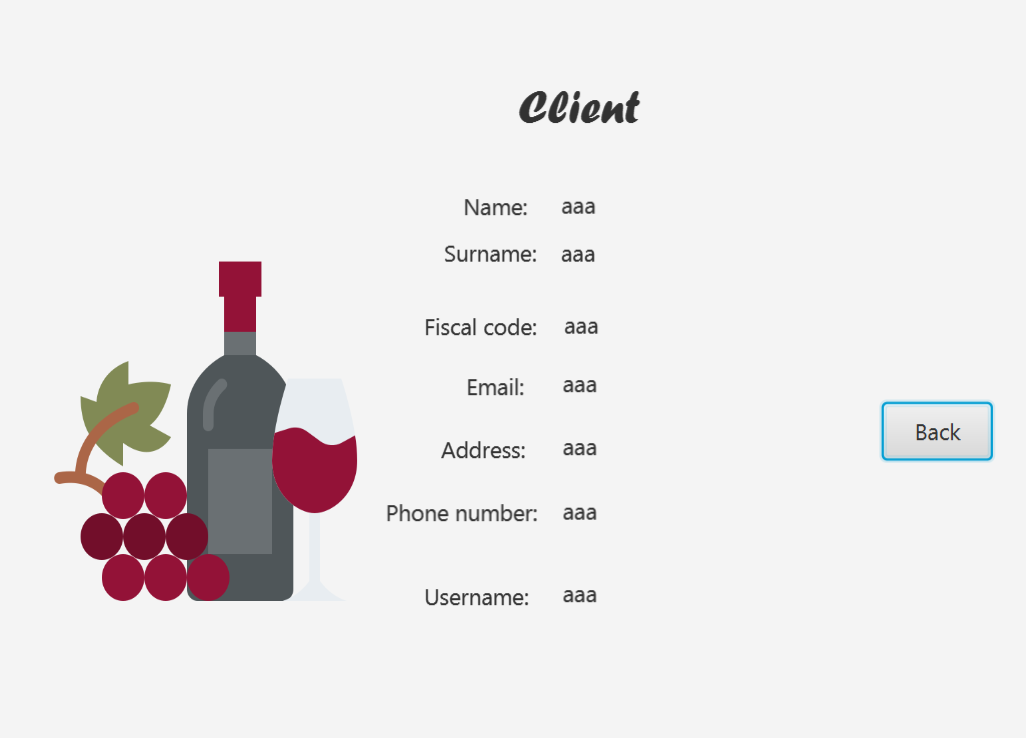
Flusso principale:

* Il cliente contatta la reception o accede al sito web dell'hotel per prenotare una stanza
* Il sistema di prenotazione mostra le disponibilità e il cliente seleziona le date e il tipo di stanza desiderati
* Il sistema di prenotazione verifica la disponibilità e conferma la prenotazione
* La reception riceve la prenotazione e la registra
* Il cliente arriva in hotel e si registra alla reception
* La reception assegna la stanza al cliente

Flusso alternativo:

* Il sistema di prenotazione non riesce a confermare la prenotazione a causa di disponibilità limitata
* Il cliente richiede una stanza diversa da quella prenotata
* Il cliente richiede servizi aggiuntivi come il parcheggio o il servizio in camera

Spero che questi esempi possano aiutarti a capire meglio come gli attori e i flussi principali e alternativi vengono utilizzati nei casi d'uso.



Attore principale: Cliente Attori secondari:

* Sistema di gestione del database
* Sistema di autenticazione e autorizzazione

Scopo del caso d'uso:

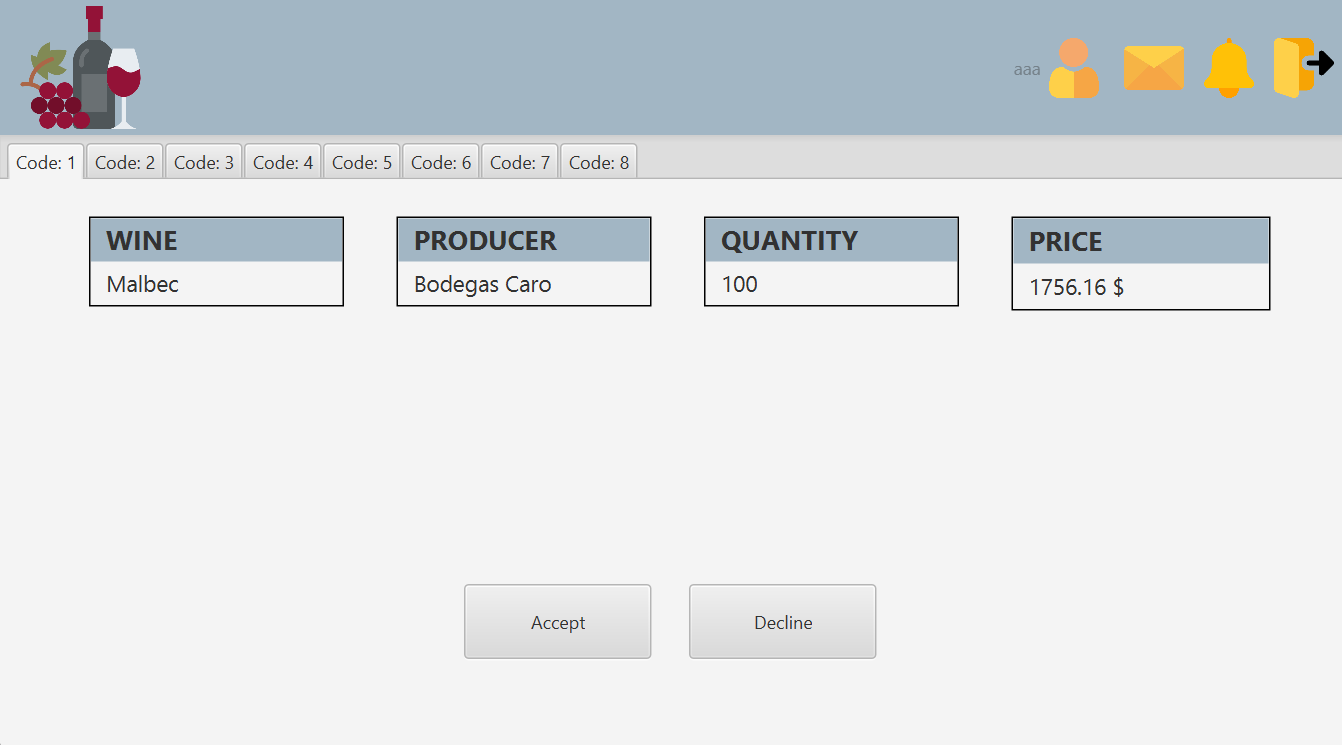
* Permettere al cliente di visualizzare i propri dati personali registrati nel sistema, inclusi nome, cognome, codice fiscale, indirizzo, numero di telefono, email, nome utente e password.
* Consentire al cliente di tornare alla home page della sua area personale.

Flusso principale del use case:

1. L'utente clicca sul pulsante "Back" per tornare alla schermata Home\_client.
2. Il metodo **back\_btnIsClicked()** viene chiamato e carica la schermata Home\_client.fxml utilizzando un oggetto FXMLLoader.
3. La schermata Home\_client.fxml viene visualizzata all'utente.

Flusso alternativo:

1. Il metodo **initialize()** viene chiamato per inizializzare la schermata Clientdata.fxml.
2. Viene creato un socket per stabilire una connessione con il server.
3. Viene inviata una richiesta al server per ottenere i dati del cliente corrente tramite il socket creato in precedenza.
4. I dati vengono recuperati dal server e visualizzati sulla schermata tramite i vari metodi **setText()**.
5. Se la connessione con il server fallisce, viene visualizzato un messaggio di errore tramite **printStackTrace()**.

Gli attori dell'applicazione sono:

* Cliente: è la persona che accede all'applicazione per visualizzare le proposte di vendita.
* Server: è il server che gestisce le richieste del cliente e gli restituisce le informazioni richieste.

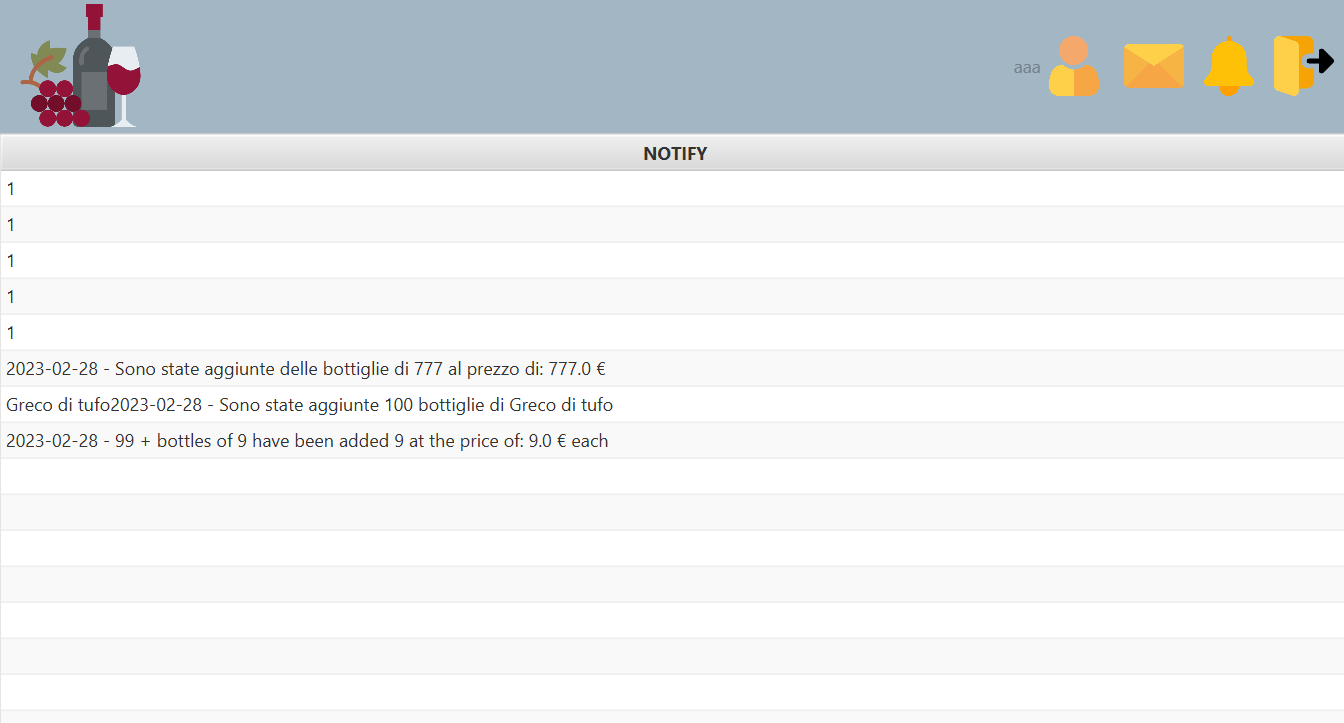
Lo scopo dell'applicazione è quello di mostrare al cliente le proposte di vendita di vini che il server ha predisposto per lui, e di consentirgli di accettare o rifiutare le proposte.

Il flusso principale dell'applicazione prevede le seguenti attività:

1. Connessione al database per ottenere le informazioni sulle proposte di vendita.
2. Ottenimento del nome del cliente dal server e visualizzazione del nome nella schermata principale.
3. Richiesta al server delle proposte di vendita per il cliente corrente.
4. Visualizzazione delle proposte di vendita in schede separate.
5. Per ogni scheda proposta, visualizzazione delle informazioni di base sul vino (nome, produttore e quantità).
6. Possibilità per il cliente di accettare o rifiutare ogni proposta di vendita tramite i pulsanti appropriati.
7. Invio della risposta del cliente al server.

Il flusso secondario dell'applicazione prevede le seguenti attività:

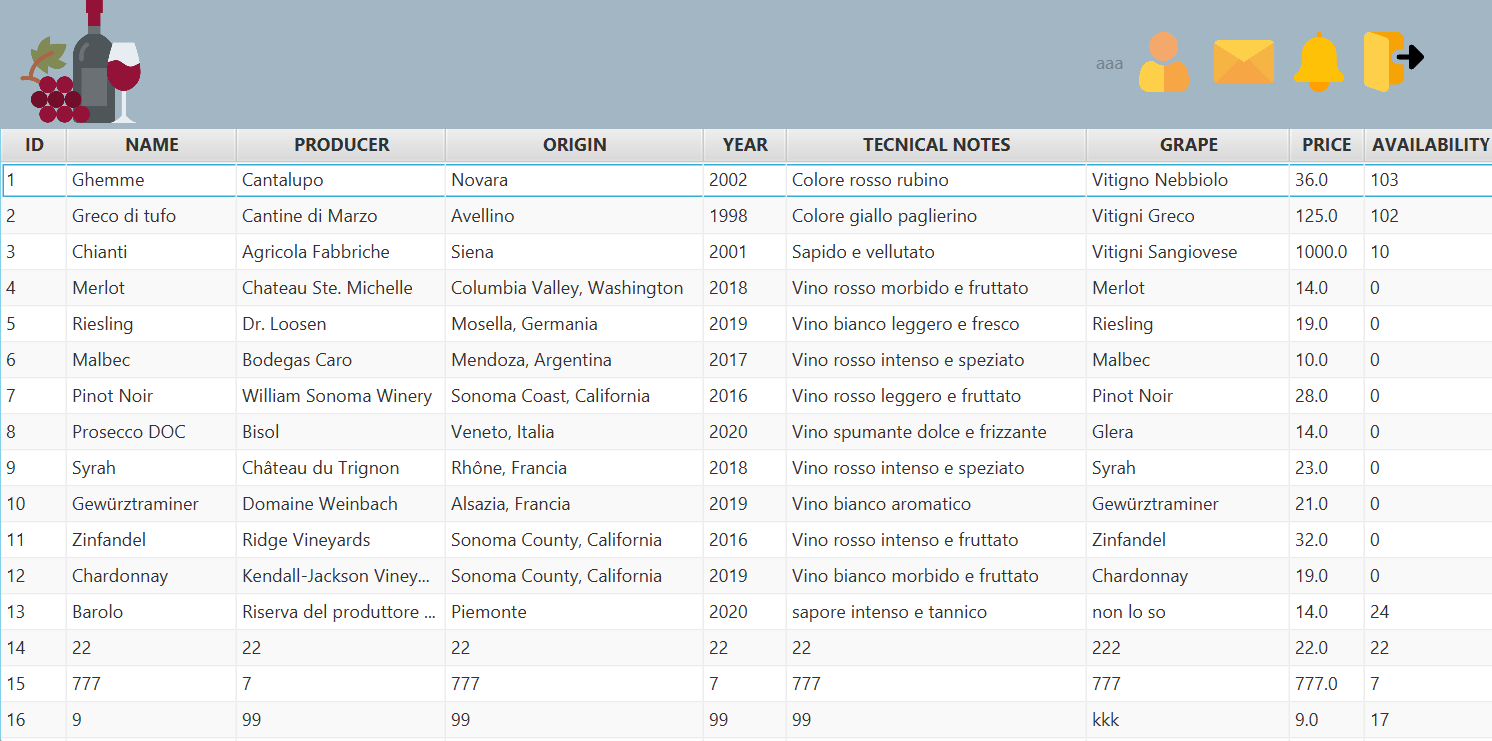
1. Nel caso in cui non ci siano proposte di vendita disponibili per il cliente, viene visualizzato un messaggio di "No results".
2. Ogni volta che viene cambiata la scheda proposta, il sistema mostra il prezzo totale della scheda precedentemente visualizzata.



Gli attori principali coinvolti sono quindi l'utente che utilizza l'interfaccia grafica e il server remoto che comunica con l'applicazione tramite socket. Il controller ha il compito di visualizzare le notifiche dell'utente, gestire il logout, la visualizzazione del profilo, la visualizzazione delle proposte e la gestione della visualizzazione delle notifiche.

Il flusso principale di questo codice è il caricamento dei dati dal database e la visualizzazione delle notifiche dell'utente nella tabella presente nell'interfaccia grafica. In particolare, il metodo "initialize" si occupa di connettersi al database, di recuperare le notifiche dell'utente dal server tramite una socket e di visualizzarle nella tabella. Inoltre, il metodo avvia un thread che controlla periodicamente se ci sono nuove notifiche o proposte da parte del server e, se presenti, cambia l'immagine del campanello o della lettera per indicare la presenza di nuovi elementi.

Il flusso secondario del codice è rappresentato dalle diverse funzionalità implementate dagli altri metodi. Ad esempio, il metodo "logoutIsClicked" gestisce l'evento di clic sul pulsante di logout, il metodo "alarmIsClicked" gestisce l'evento di clic sul pulsante per visualizzare le proposte, e così via.



Gli attori coinvolti sono:

* L'utente che utilizza l'applicazione per cercare i vini
* Il database che contiene le informazioni sui vini
* Il socket che gestisce la connessione tra il client e il server

Lo scopo del codice è quello di fornire all'utente un'interfaccia grafica per cercare i vini nel database e visualizzarli in una tabella. In particolare, il metodo **buttonClick** viene chiamato quando l'utente preme il pulsante "Cerca". Questo metodo si connette al database, esegue una query per recuperare i vini corrispondenti ai criteri di ricerca e li visualizza nella tabella.

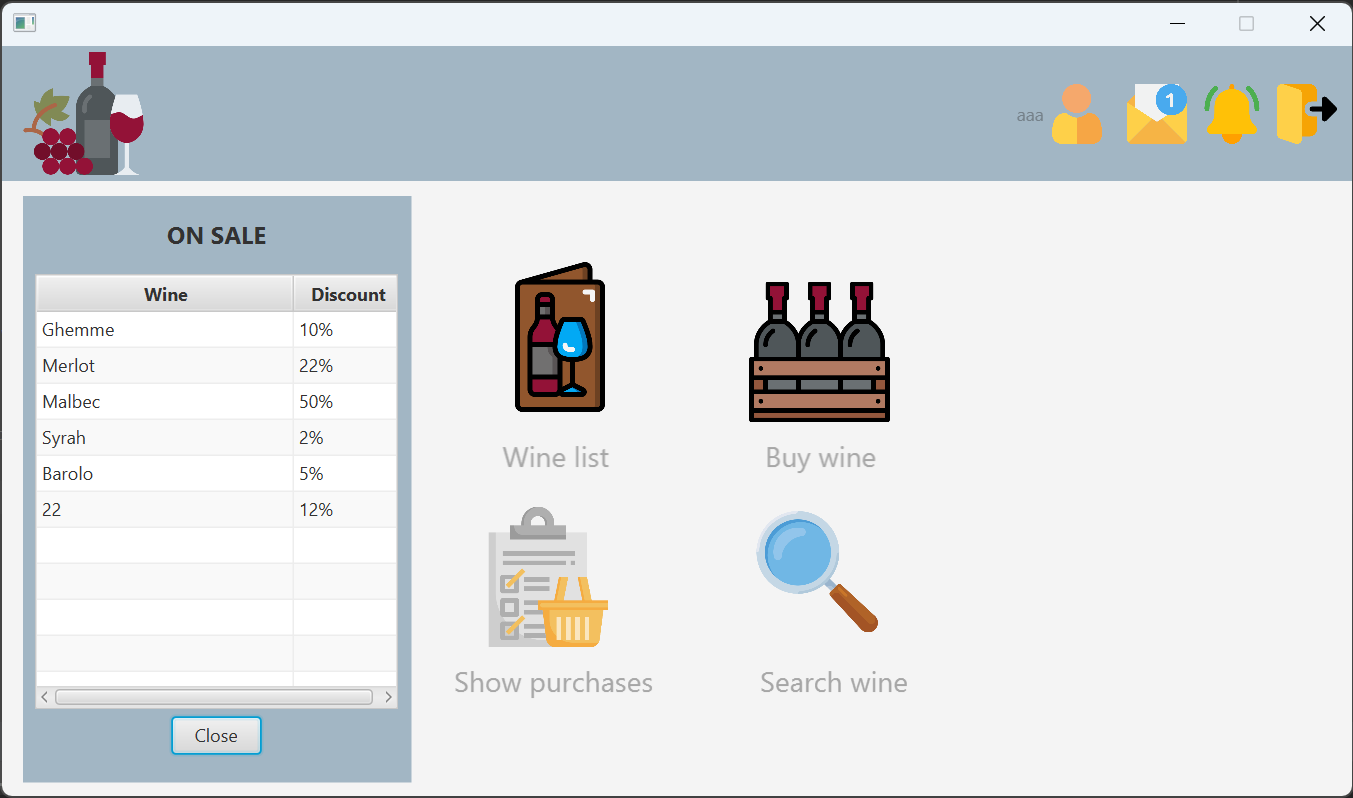
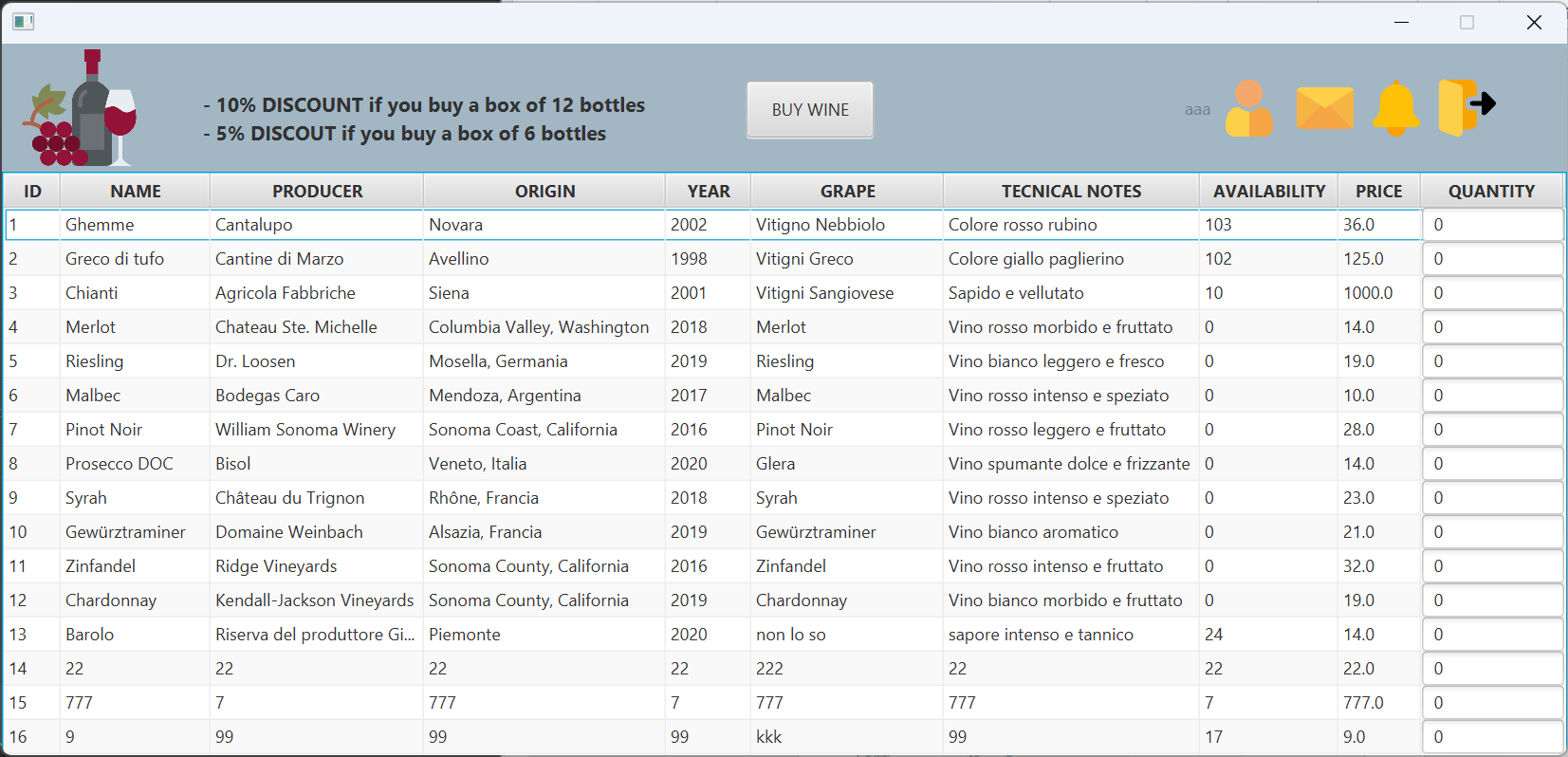
Il codice utilizza inoltre un thread che ciclicamente controlla se ci sono nuove notifiche o messaggi per l'utente. Se ci sono notifiche, viene mostrata un'icona di campana nella finestra dell'applicazione. Se ci sono messaggi, viene mostrata un'icona di lettera.

Il flusso principale del codice è la funzione **buttonClick**, che viene chiamata quando l'utente preme il pulsante "src\_wine\_by\_name\_btn". All'interno di questa funzione, viene stabilita una connessione al database e viene inviata una richiesta al server tramite socket per recuperare i dati dei vini corrispondenti al nome inserito dall'utente. I dati vengono quindi visualizzati nella tabella nella GUI.

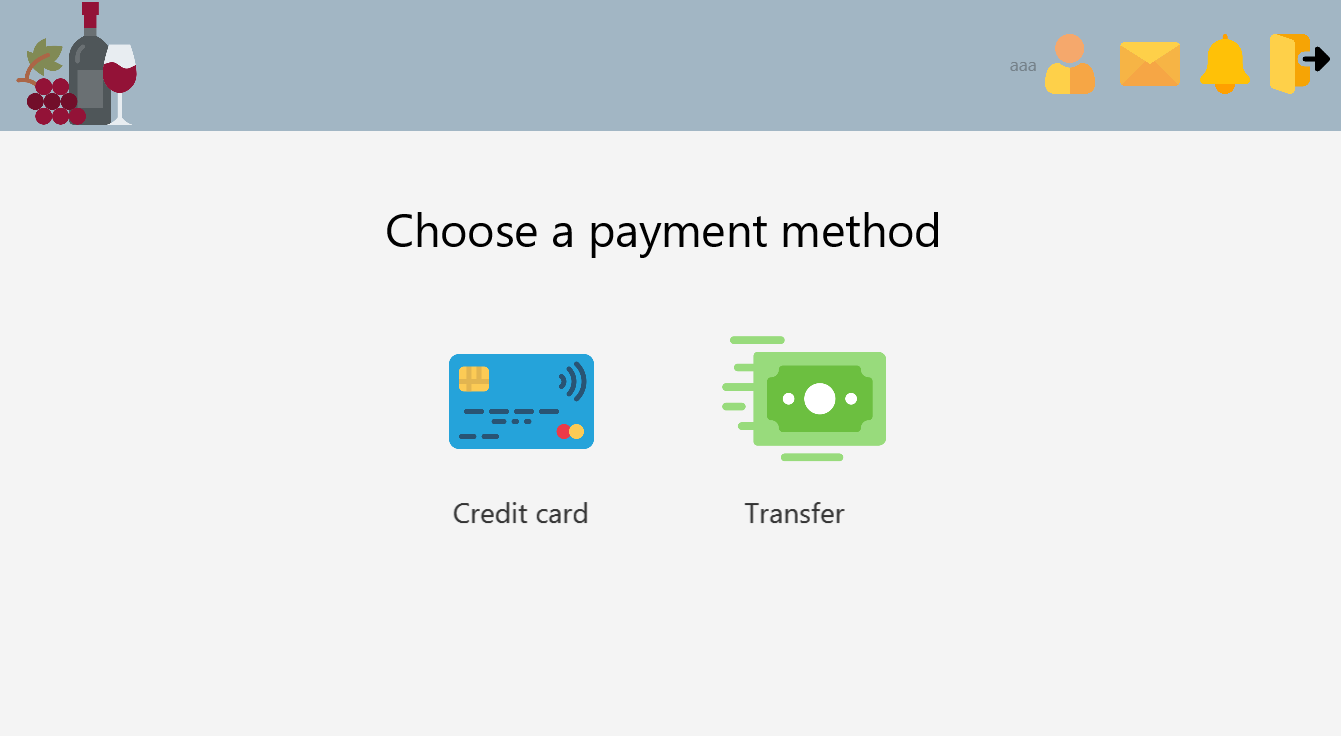
Il flusso secondario riguarda il thread che viene avviato nella funzione **initialize** e che viene utilizzato per controllare periodicamente la presenza di nuove notifiche dal server. Se viene rilevata una nuova notifica, viene visualizzata l'icona della campanella nella GUI. Lo stesso controllo viene effettuato per la presenza di nuove proposte, che viene segnalata dall'icona della lettera. Questo thread viene eseguito in background in modo da non interferire con l'interazione dell'utente con la GUI.

Immagine che contiene tavolo

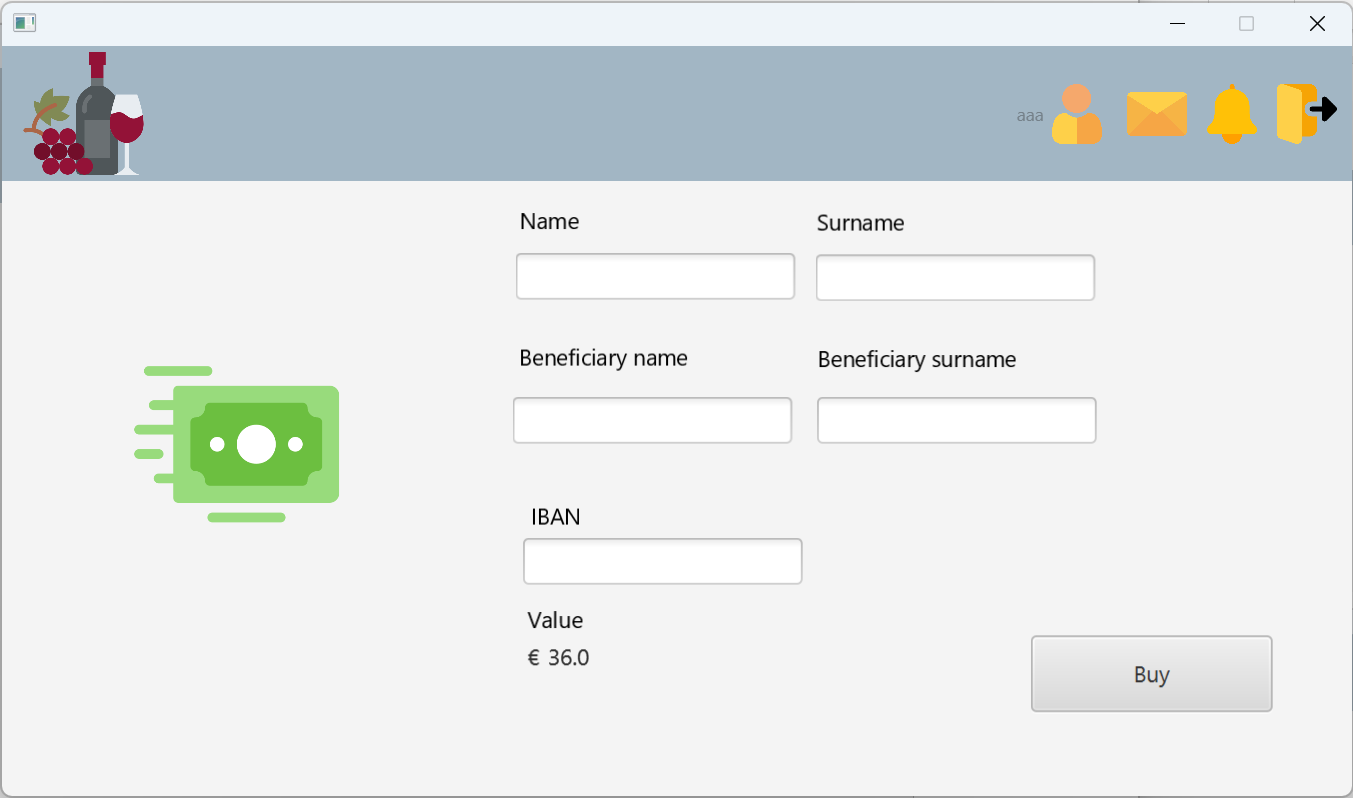
Descrizione generata automaticamente

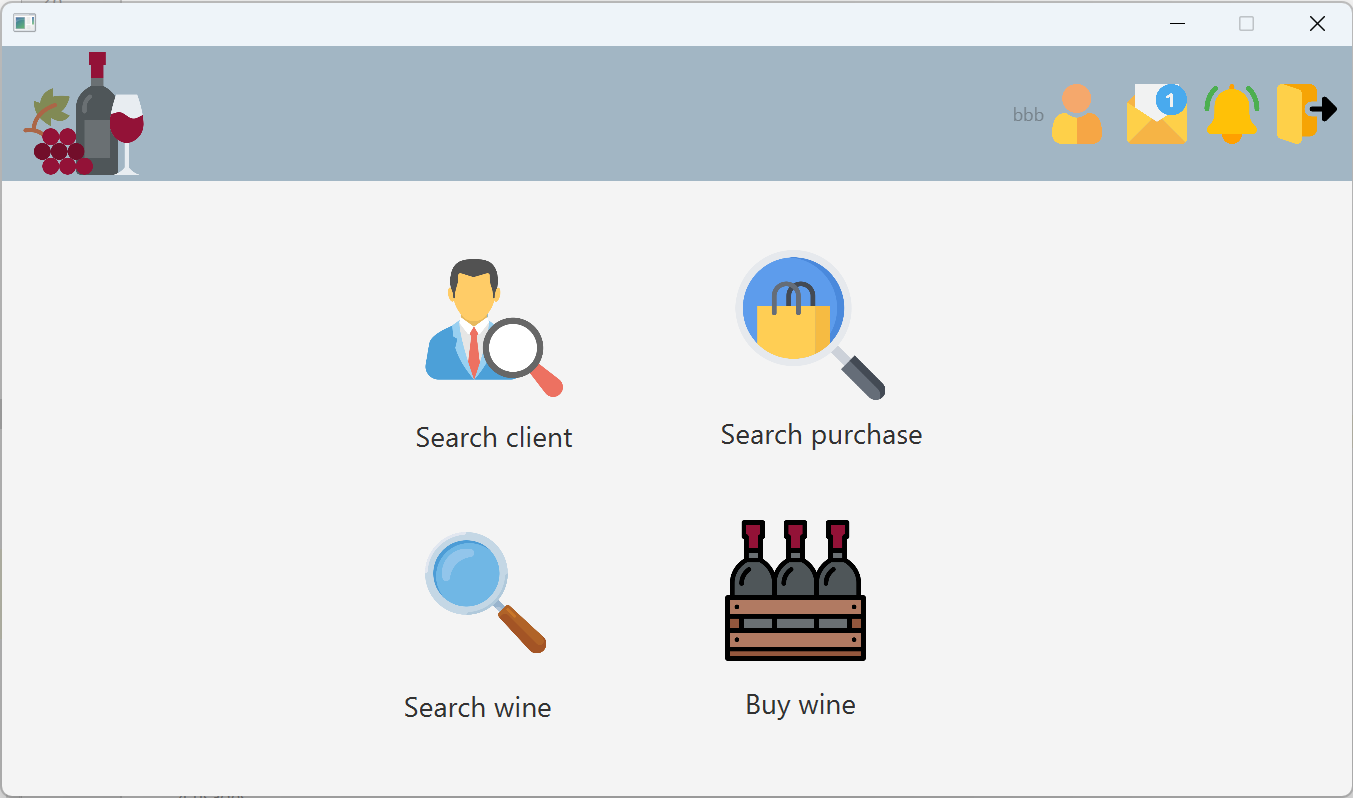
aaaaImmagine che contiene tavolo

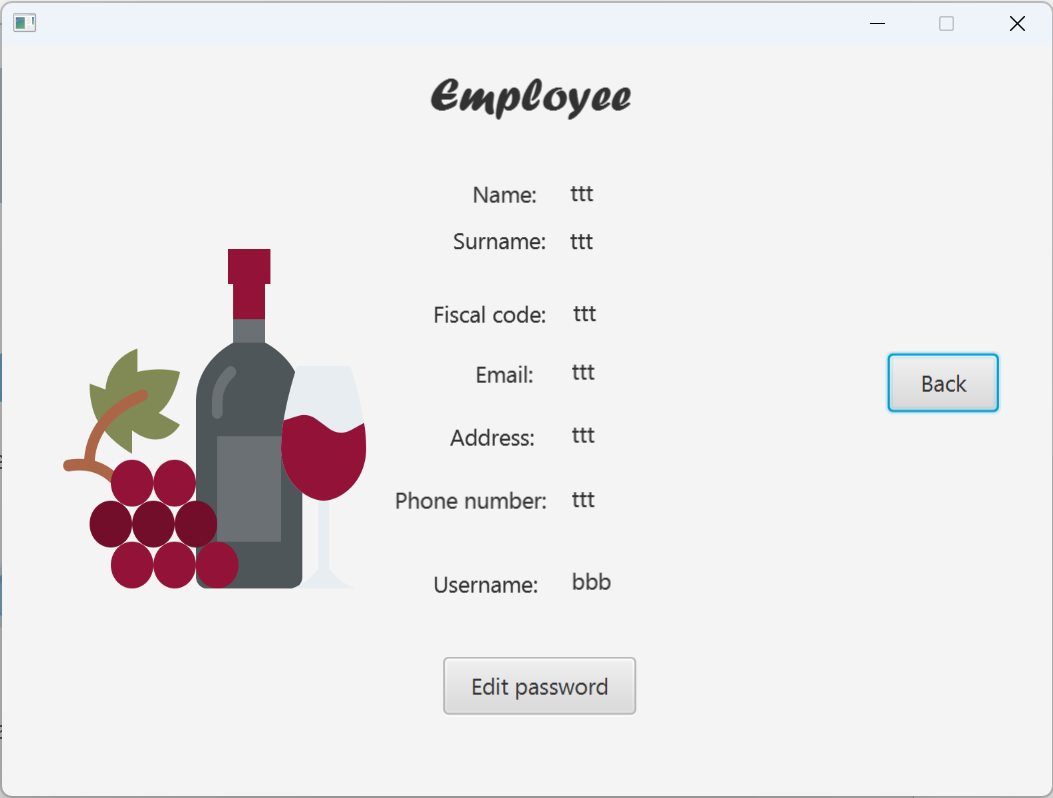
Descrizione generata automaticamente











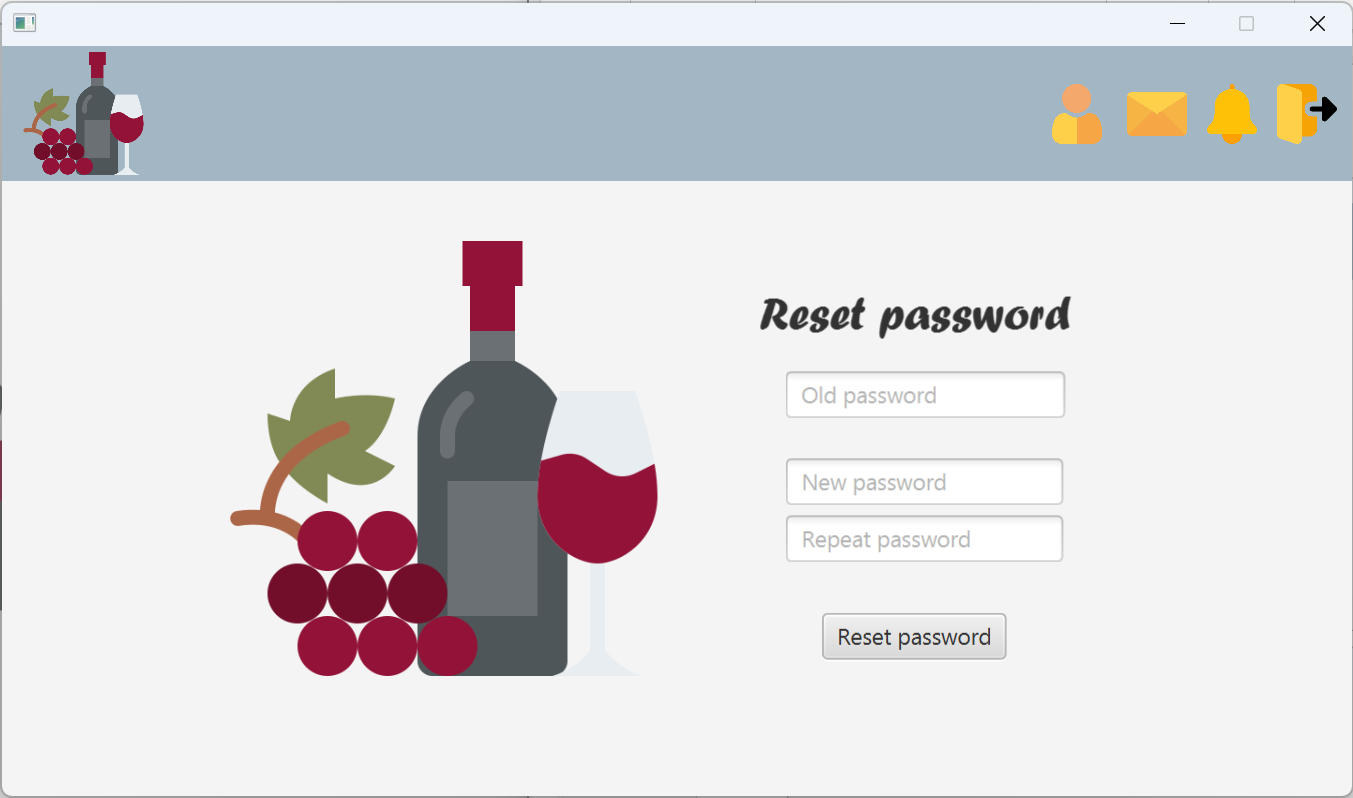
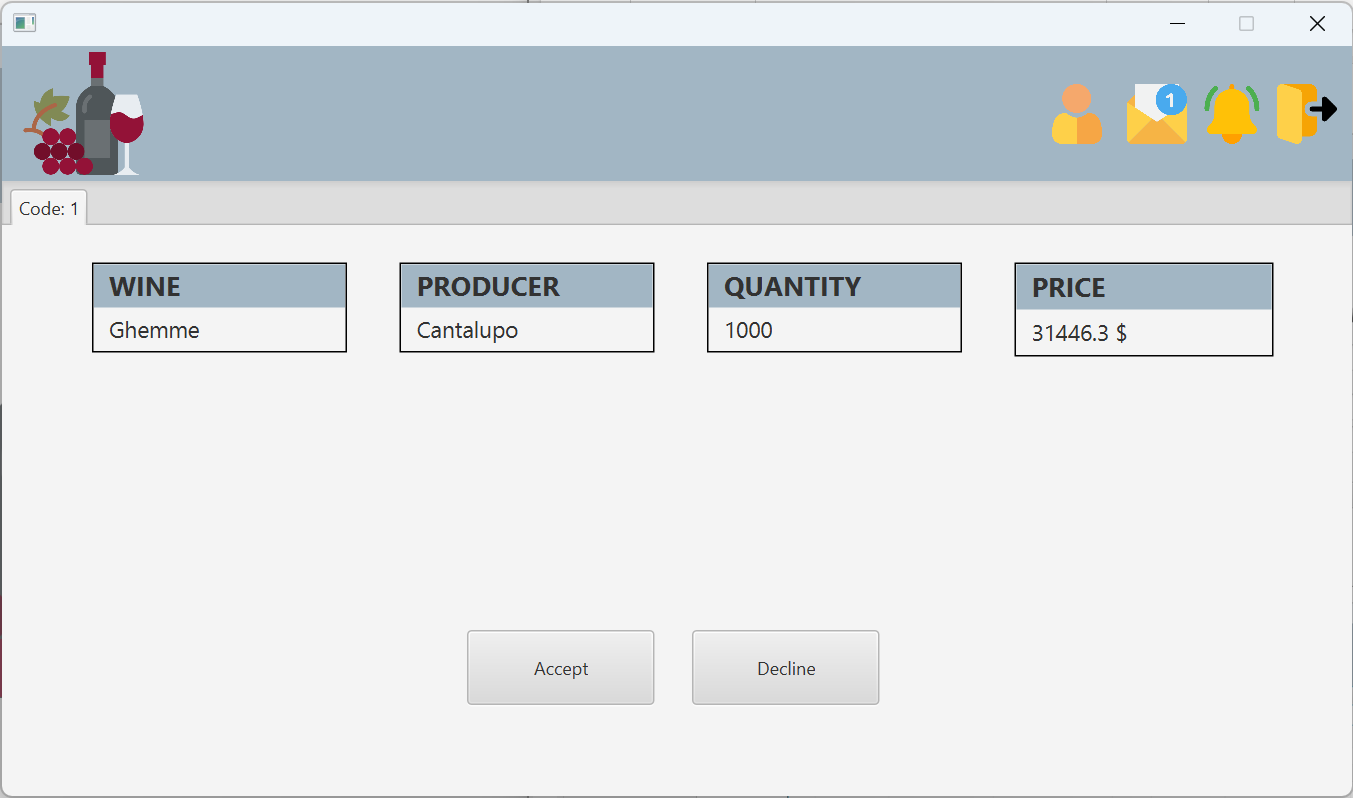


Immagine che contiene tavolo

Descrizione generata automaticamente  Immagine che contiene tavolo

Descrizione generata automaticamenteImmagine che contiene tavolo

Descrizione generata automaticamente

Immagine che contiene tavolo

Descrizione generata automaticamente

