

Online Wine Shop

Progetto di ingegneria del software

Realizzato da

Matteo Gianvenuti Matr. 321490

matteo.gianvenuti@studenti.unipr.it

Sommario

1. Introduzione	3
1.1 Descrizione del sistema (Requisiti informali)	3
2. Storico delle revisioni	5
3. Requisiti	5
3.1 Stakeholders del sistema	5
3.2 Requisiti funzionali	6
3.3 Requisiti non funzionali	8
3.3.1 Requisiti di processo	8
3.3.2 Requisiti di prodotto	8
3.3.3 Requisiti di archiviazione	8
4. Documentazione Use Cases	8
4.1 Documentazione use case customer	8
4.1.1 Diagramma use case customer	15
4.2 Documentazione use cases employee	15
4.2.1 Diagramma use case employee	22
4.3 Documentazione use cases administrator	23
4.3.1 Diagramma use case administrator	26
5. Architectural Model	27
5.1 Architecture diagram	27
6. Detailed Software Model	28
6.1 Class diagram(s)	28
6.1.1 Diagramma complessivo	28
6.1.2 Diagrammi delle singole classi	29
7. Test Funzionali	38
7.1 Actor class tests	38
7.2 Notification class tests	39
7.3 Promotion class tests	39
7.4 Wine class tests	40
8. Manuale di sistema	41
8.1 Installazione	41
8.1.1 Installare XAMPP	41
8.1.2 Installare Eclipse IDE	41
8.1.3 Scaricare i pacchetti aggiuntivi	41
8.2 Configurazione	41
8.2.1 Avviare XAMPP	41
8.2.2 Avviare Eclipse IDE	41
8.2.3 Importare i pacchetti aggiuntivi	41
8.2.4 Avviare il sistema	41
8.3 Guida utente	41

8.3.1 Guida clienti	41
8.3.1.1 Registrazione	41
8.3.1.2 Accesso	42
8.3.1.3 Cercare vini	42
8.3.1.4 Creare un ordine di vendita	42
8.3.1.5 Cancellare un ordine di vendita	42
8.3.1.6 Creare una proposta d'acquisto	42
8.3.1.7 Cancellare un aproposta d'acquisto	43
8.3.1.8 Rifiutare un ordine di vendita generato da una proposta d'acquisto	43
8.3.1.9 Confermare un ordine di vendita generato da una proposta d'acquisto	43
8.3.1.10 Richiedere una notifica di disponibilità	43
8.3.1.11 Visualizzare lo storico ordini	43
8.3.1.12 Menu e logout	44
8.3.2 Guida impiegati	44
8.3.2.1 Accesso	44
8.3.2.2 Cambio password	44
8.3.2.3 Cerca clienti per cognome	44
8.3.2.4 Cerca vini per nome e/o anno di produzione	44
8.3.2.5 Cerca odini per intervallo di date	44
8.3.2.6 Crea promozione	44
8.3.2.7 Completa attività	44
8.3.2.7.1 Completa ordine di vendita	45
8.3.2.7.2 Completa proposta d'acquisto	45
8.3.2.7.3 Completa ordine d'acquisto	45
8.3.3 Guida amministratore	45
8.3.3.1 Accesso	45
8.3.3.2 Aggiungi impiegato	45
8.3.3.3 Gestisci impiegato	45
8.3.3.4 Crea/cerca report	46

1. Introduzione

1.1 Descrizione del sistema (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 email, numero telefonico, 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 (entro 10 giorni lavorativi dalla data del pagamento).

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, il cliente può chiedere di essere notificato dal sistema quando le bottiglie saranno disponibili: 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 (viene utilizzato dal sistema l'indirizzo specificato in fase di registrazione). Ricevuta la proposta di acquisto, il sistema invia una copia della proposta di acquisto ad uno degli impiegati. Questo impiegato preparerà un ordine di acquisto e lo invia al fornitore. Ricevuto l'ordine, il fornitore 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 (entro 10 giorni lavorativi dalla data di pagamento del cliente) e invia l'ordine di vendita. Ricevuto l'ordine di vendita, il cliente può decidere se acquistarlo o no. Se sì, 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, informare il corriere quando il materiale è pronto per essere prelevato indicando la data limite di consegna del materiale (entro 10 giorni lavorativi dalla data di pagamento del cliente). Dopo la consegna del materiale al corriere, il dipendente 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 vendita 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 fornitore 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 (3 minuti), il sistema l'asigna ad un altro impiegato. Il sistema mantiene le informazioni sulle attività completate e non completate dai dipendenti.

2. Storico delle revisioni

Versione	Data	Autore/i	Descrizione
1.0	16/12/2022	Matteo Gianvenuti	Aggiunti introduzione, requisiti funzionali e non, definizione degli stakeholder.
1.1	19/12/2022	Matteo Gianvenuti	Use cases, documentazione cliente.
1.2	12/03/2023	Matteo Gianvenuti	Scrittura del manuale di sistema.
1.3	13/03/2023	Matteo Gianvenuti	Use cases, documentazione impiegato, amministratore.
1.4	14/03/2023	Matteo Gianvenuti	Documentazione risultati dei test funzionali.
1.5	17/03/2023	Matteo Gianvenuti	Documentazione class diagrams, architecture diagram.

3. Requisiti

Nel seguente capitolo vengono definiti gli stakeholder del sistema, i requisiti funzionali ed i requisiti non funzionali.

3.1 Stakeholders del sistema

- Cliente (customer): colui che accede al sistema tramite autenticazione per visualizzare i vini in vendita ed eventualmente acquistarli. Un nuovo cliente deve prima registrarsi al sistema.
- Impiegato (employee): colui che accede al sistema per gestire la vendita e la spedizione e ricezione dei prodotti. Attività principali: esegue gli ordini di acquisto, risponde alle proposte di acquisto con proposte di vendita, ordina i vini al fornitore prima che esauriscano, fornisce al corriere il vino e la data di consegna.
- Amministratore (administrator): colui che accede al sistema per controllare il lavoro dei dipendenti, gestire i dipendenti e gestire le attività finanziarie dell'azienda. Attività principali: registrare nuovi impiegati nel sistema, preparare un report mensile contenente i dati finanziari dell'azienda e visualizzare quanti ordini hanno completato o non completato gli impiegati.

3.2 Requisiti funzionali

Di seguito è riportato l'elenco dei requisiti funzionali. C indica i requisiti funzionali che coinvolgono i clienti del sistema, F indica i requisiti funzionali che coinvolgono gli impiegati e l'amministratore del sistema, S indica i requisiti funzionali riguardanti funzioni del sistema che devono essere eseguite automaticamente. Ogni requisito ha una priorità di implementazione che può essere Alta, Media o Bassa.

ID	Descrizione funzione/funzioni	Stato	Priorità
C1	Un utente generico (cliente, impiegato o amministratore) può accedere al sistema.	Implementato	Alta
C2	Un cliente può registrarsi nel sistema.	Implementato	Alta
C3	Un cliente registrato, dopo un accesso autenticato, può fare ricerca dei vini per nome e anno di produzione e può inserire uno o più vini nell'ordine, per ogni vino, può inserire il numero di bottiglie da acquistare, può pagare con bonifico o carta di credito.	Implementato	Alta
C4	Se non c'è la quantità di vino desiderata il cliente può compilare una proposta di acquisto, dove inserisce i vini come per l'ordine (C3).	Implementato	Alta
C5	Se non c'è la quantità di vino desiderata il cliente può chiedere di essere notificato quando ci sarà, per un determinato vino.	Implementato	Alta
C6	Ogni volta che un cliente accede nel sistema, il sistema informa il cliente sulle promozioni in corso e le eventuali notifiche di disponibilità.	Implementato	Alta
C7	Il cliente può inserire la quantità da acquistare nei formati bottiglie singole (0-5), casse da 6 (0-100), casse da 12 (0-100).	Implementato	Alta
C8	Il cliente può accettare e pagare oppure rifiutare un ordine di vendita generato da una proposta.	Implementato	Alta
F1	Impiegato e amministratore possono cambiare la loro password.	Implementato	Alta
F2	Un amministratore può registrare nuovi dipendenti nel sistema, bloccarli e sbloccarli.	Implementato	Alta
F3	Un amministratore può resettare la password dei dipendenti.	Implementato	Alta
F4	Gli impiegati e l'amministratore possono fare la ricerca dei clienti per cognome.	Implementato	Alta
F5	Gli impiegati e l'amministratore possono fare la ricerca dei vini per nome e/o anno di produzione.	Implementato	Alta

F6	Gli impiegati e l'amministratore possono fare la ricerca degli ordini di vendita e di acquisto e delle proposte di acquisto per intervallo di date.	Implementato	Alta
F7	L'amministratore può cercare/realizzare il report mensile.	Implementato	Alta
F8	L'impiegato può creare promozioni per i vini.	Implementato	Alta
F9	L'impiegato può firmare digitalmente e completare un ordine di acquisto.	Implementato	Alta
F10	L'impiegato può generare un ordine di acquisto per il fornitore e un ordine di vendita per il cliente a partire da una proposta.	Implementato	Alta
F11	L'impiegato può firmare digitalmente e completare un ordine di vendita.	Implementato	Alta
S1	Ogni volta che viene effettuato un acquisto, il sistema aggiorna il numero di vendita e disponibilità dei vini, mostra al cliente l'ordine di vendita (nello storico) con le informazioni sull'acquisto.	Implementato	Alta
S2	Il sistema assegna le attività agli impiegati online in modo circolare (RR).	Implementato	Alta
S3	Il sistema incrementa il prezzo di vendita in base alle vendite, per ogni vino.	Implementato	Alta
S4	Il sistema mantiene le attività completate e non completate.	Implementato	Alta
S5	Ad ogni arrivo di un ordine di acquisto il sistema controlla se il numero di bottiglie di alcuni vini è sceso sotto la soglia.	Implementato	Alta
S6	Se sotto la soglia (S5) il sistema crea un ordine d'acquisto.	Implementato	Alta
S7	Il sistema assegna un ordine di vendita confermato allo stesso impiegato che l'ha generato.	Implementato	Alta
S8	Se non viene completata nel tempo prestabilito un'attività viene assegnata ad un altro impiegato online.	Implementato	Alta
S9	Se un impiegato non completa l'ordine di vendita confermato generato dalla proposta precedentemente completata (S7) il sistema gestisce l'ordine di vendita come un qualsiasi altro.	Implementato	Alta

3.3 Requisiti non funzionali

3.3.1 Requisiti di processo

- Sviluppare il sistema usando il linguaggio di programmazione Java.
- Il sistema deve essere basato su un'architettura client server. Il server deve supportare l'accesso in parallelo di diversi nodi client. Quindi si deve usare una socket.

3.3.2 Requisiti di prodotto

- L'interfaccia grafica dei client deve essere implementata con JavaFX.

3.3.3 Requisiti di archiviazione

- Il sistema deve fare riferimento ad una memoria permanente, cioè un database.
- Il database del sistema deve mantenere le informazioni di almeno una decina di vini differenti.
- Tre impiegati ed un amministratore devono essere già presenti nel sistema quindi nel database.

4. Documentazione Use Cases

Di seguito la documentazione e i diagrammi dei casi d'uso (use cases), per semplicità i diagrammi sono stati riportati divisi per ogni attore.

4.1 Documentazione use case customer

Di seguito la documentazione use cases customer descrive le funzionalità che il sistema offre ad un cliente.

Nome use case: login.

Tabella dati tecnici:

Use Case ID:	0		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	19/12/2022	Date Last Updated:	19/12/2022

Descrizione: Il cliente registrato vuole accedere al sistema.

Attore principale: Cliente.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. Il cliente si è registrato nel sistema.

Post-condizioni:

1. Il cliente dopo l'accesso autenticato visualizza il menu principale.

Trigger: Il cliente clicca sul pulsante accedi.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. Il cliente inserisce nome utente e password.
2. Il cliente clicca il pulsante per accedere.
3. Il sistema mostra il menu clienti.

Flussi alternativi:

1. Nome utente errato o inesistente.
 - a) Il sistema informa il cliente che ha sbagliato password/nome utente.

Eccezioni: Il cliente o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: signin.

Tabella dati tecnici:

Use Case ID:	1		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	19/12/2022	Date Last Updated:	19/12/2022

Descrizione: Il cliente vuole registrarsi nel sistema. Quindi dopo aver avviato il trigger inserisce i suoi dati e salva. I dati vengono salvati nel database del sistema.

Attore principale: Cliente.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.

Post-condizioni:

1. I dati del cliente sono stati registrati nel database.

Trigger: Il cliente clicca il pulsante, nella pagina iniziale di accesso, per registrarsi.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. Il cliente clicca il pulsante registrati.
2. Il sistema mostra al cliente il form per inserire i dati di registrazione.
3. Il cliente inserisce i dati.
4. Il cliente clicca il pulsante per registrarsi e salvare i dati.
5. il sistema salva i dati nel database tramite il DBMS.

Flussi alternativi:

1. Il cliente annulla l'operazione.
 - a) In un qualsiasi momento il cliente interrompe l'operazione tornando alla pagina di accesso chiudendo l'applicazione.
 - b) I dati non vengono salvati.
2. Nome utente, email o codice fiscale sono già stati usati.
 - a) Il cliente clicca il pulsante registrati.
 - b) Il sistema lo informa che nome utente, email o codice fiscale è già in uso.
 - c) Il cliente inserisce un nome utente, email o codice fiscale diverso e clicca nuovamente il pulsante per registrarsi.

Eccezioni: Il cliente o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: wine research by name and/or year.

Tabella dati tecnici:

Use Case ID:	2		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	19/12/2022	Date Last Updated:	19/12/2022

Descrizione: Il cliente vuole visualizzare i vini presenti nel sistema ed eventualmente comprarli. Quindi il cliente inserisce i dati del vino da cercare e avvia la ricerca. Il sistema mostra il vino cercato. Sempre è offerta la possibilità, per un qualsiasi vino presente, di inserire una quantità e acquistarla se disponibile.

Attore principale: Cliente.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. Il DBMS, server e client sono in esecuzione.
2. Il cliente registrato ha fatto accesso al sistema.

Post-condizioni:

1. Nessuna.

Trigger: Il cliente clicca il pulsante, nel menu principale, per fare la ricerca/acquisto di un vino.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. Il cliente clicca il pulsante per cercare/acquistare il vino.
2. Il cliente inserisce nome e/o anno di produzione.
3. Il sistema mostra il vino cercato se esiste.

Flussi alternativi:

1. Il vino non esiste.

- a) Il sistema informa il cliente che non ci sono risultati
2. Il cliente annulla l'operazione.
 - a) Il cliente in un qualsiasi momento torna al menu.

Eccezioni: Il cliente o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: make/cancel proposal purchase.

Tabella dati tecnici:

Use Case ID:	3		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	19/12/2022	Date Last Updated:	19/12/2022

Descrizione: Il cliente vuole acquistare un vino ma la quantità desiderata è maggiore di quella disponibile quindi invia una proposta d'acquisto.

Attore principale: Cliente.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. Il cliente registrato ha fatto accesso al sistema.
3. La quantità desiderata dal cliente è maggiore di quella disponibile.

Post-condizioni:

1. La proposta d'acquisto è stata salvata nel database.

Trigger: Il cliente clicca il pulsante, nel menu principale, per fare la ricerca/acquisto di un vino.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. Il cliente inserisce uno o più vini nella proposta di acquisto in quantità maggiore di quella disponibile.
2. Il cliente clicca il pulsante per inviare la proposta.

Flussi alternativi:

1. Il cliente annulla l'operazione.
 - a. Il cliente clicca il pulsante per cancellare la proposta d'acquisto.
2. Il cliente cerca di inserire un vino con una quantità minore di quella disponibile.
 - a. Il sistema lo informa che la quantità deve essere maggiore di quella disponibile.

Eccezioni: Il cliente o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: make/cancel sales order.

Tabella dati tecnici:

Use Case ID:	4		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	19/12/2022	Date Last Updated:	19/12/2022

Descrizione: Il cliente vuole acquistare un vino e la quantità desiderata è minore o uguale di quella disponibile quindi invia un ordine di vendita.

Attore principale: Cliente.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. Il cliente registrato ha fatto accesso al sistema.
3. La quantità desiderata dal cliente è minore o uguale di quella disponibile.

Post-condizioni:

1. L'ordine di vendita è stato salvato nel database.

Trigger: Il cliente clicca il pulsante, nel menu principale, per fare la ricerca/acquisto di un vino.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. Il cliente inserisce uno o più vini nell'ordine di vendita in quantità minore di quella disponibile.
2. Il cliente clicca il pulsante per inviare l'ordine e pagare.
3. Il cliente visualizza la pagina di pagamento e paga con il metodo che preferisce.
4. Il sistema informa il cliente: l'ordine si trova nello storico.

Flussi alternativi:

1. Il cliente annulla l'operazione.
 - a. Il cliente clicca il pulsante per cancellare l'ordine o non paga.
2. Il cliente cerca di inserire un vino con una quantità maggiore di quella disponibile.
 - a. Il sistema lo informa che la quantità deve essere minore o uguale di quella disponibile.

Eccezioni: Il cliente o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: ask availability notification.

Tabella dati tecnici:

Use Case ID:	5		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	19/12/2022	Date Last Updated:	19/12/2022

Descrizione: Il cliente vuole acquistare un vino ma la quantità desiderata non è sufficiente quindi chiede di essere notificato per quella quantità.

Attore principale: Cliente.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. Il cliente registrato ha fatto accesso al sistema.

Post-condizioni:

1. La richiesta di notifica del cliente è stata salvata nel database.

Trigger: Il cliente clicca il pulsante per essere notificato quando il vino sarà disponibile.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. Il cliente dalla pagina cerca acquista inserisce i dati di un vino e una quantità non disponibile.
2. il cliente clicca il pulsante per essere notificato.

Flussi alternativi:

1. Il cliente ha inserito una quantità minore di quella disponibile.
 - a) Il sistema lo informa che la quantità deve essere maggiore di quella disponibile.

Eccezioni: Il cliente o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: confirm/reject sales order.

Tabella dati tecnici:

Use Case ID:	6		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	19/12/2022	Date Last Updated:	19/12/2022

Descrizione: Il cliente riceve una richiesta di conferma di un ordine di vendita nello storico, per una proposta di acquisto fatta precedentemente.

Attore principale: Cliente.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. Il cliente registrato ha fatto accesso al sistema.
3. Il cliente ha precedentemente fatto una proposta di acquisto, use case ID 3.

Post-condizioni:

1. La decisione del cliente viene salvata nel database.

Trigger: Il cliente clicca il pulsante per confermare o rifiutare l'ordine.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. Il cliente, nello storico ordini, visualizza un ordine in attesa di conferma.
2. Il Cliente seleziona l'id ordine nel menu a tendina e clicca il pulsante conferma e paga.
3. Il cliente visualizza la pagina di pagamento e paga con il metodo che preferisce.

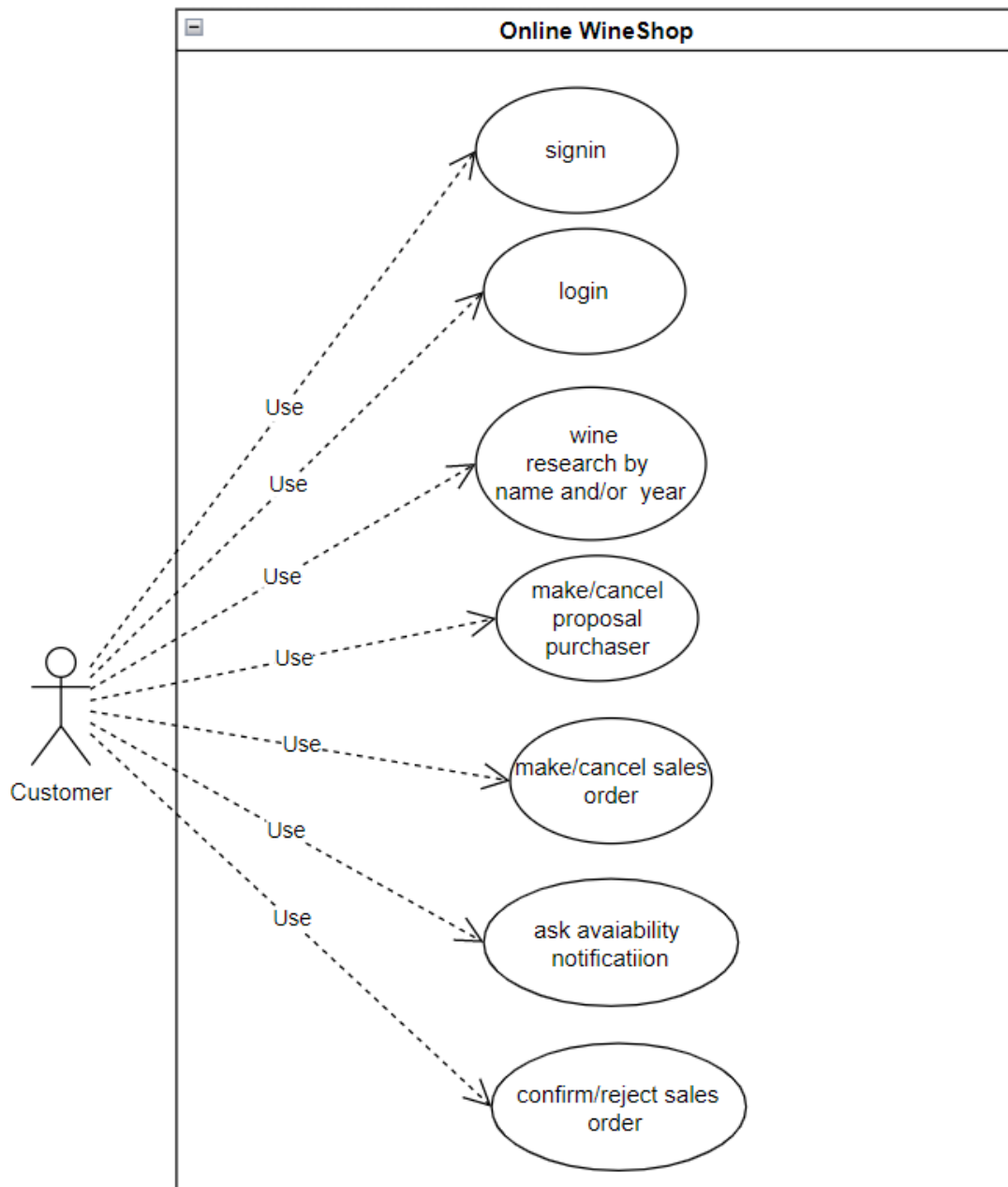
Flussi alternativi:

1. Il cliente rifiuta l'ordine.
 - a) Il cliente, nello storico ordini, visualizza un ordine in attesa di conferma.
 - b) Il Cliente seleziona l'id ordine nel menu a tendina e clicca il pulsante rifiuta ordine.
 - c) Il cliente visualizza nello storico l'ordine rifiutato.
2. Il cliente non paga.
 - a) Il cliente, nello storico ordini, visualizza un ordine in attesa di conferma.
 - b) Il Cliente seleziona l'id ordine nel menu a tendina e clicca il pulsante conferma e paga.
 - c) Il cliente visualizza la pagina di pagamento e non paga, l'ordine rimane non confermato, in attesa.

Eccezioni: Il cliente o un bug termina il programma durante l'operazione, i dati non vengono salvati.

4.1.1 Diagramma use case customer

Di seguito è riportato il diagramma use case del cliente. Siccome gli attori DBMS e sistema sono sempre coinvolti, per semplicità non sono stati riportati nel diagramma.



4.2 Documentazione use cases employee

Di seguito la documentazione use cases employee descrive le funzionalità che il sistema offre ad un impiegato. L'impiegato accede al sistema come un cliente, previa registrazione da parte dell'amministratore, pertanto l'use case di accesso non è qui riportato.

Nome use case: modify password.

Tabella dati tecnici:

Use Case ID:	7		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'impiegato vuole cambiare la password di accesso al sistema.

Attore principale: Impiegato.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'impiegato registrato ha fatto accesso al sistema.

Post-condizioni:

1. L'eventuale nuova password è stata salvata nel database.

Trigger: L'impiegato clicca il pulsante cambia password dal menu.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'impiegato inserisce la vecchia password.
2. L'impiegato inserisce la nuova password.
3. L'impiegato ripete la nuova password.
4. L'impiegato clicca il pulsante cambia password.
5. Il sistema lo informa sul successo dell'operazione.

Flussi alternativi:

1. L'impiegato annulla l'operazione.
 - a) In un qualsiasi momento antecedente il punto 4 del flusso normale l'impiegato torna al menu.
2. La vecchia password non coincide.
 - a) Il sistema informa l'impiegato che la vecchia password non coincide.
 - b) L'impiegato riparte dal punto 1 del flusso normale oppure esegue il flusso alternativo 1.
3. La nuova password non coincide.
 - a) Il sistema informa l'impiegato che la nuova password non coincide.
 - b) L'impiegato riparte dal punto 2 del flusso normale oppure esegue il flusso alternativo 1.

Eccezioni: L'impiegato o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: wine research by name and/or year.

Tabella dati tecnici:

Use Case ID:	8
--------------	---

Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'impiegato cerca nel sistema i vini con nome e/o anno di produzione inseriti.

Attore principale: Impiegato.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'impiegato registrato ha fatto accesso al sistema.

Post-condizioni:

1. Nessuna.

Trigger: L'impiegato clicca il pulsante cerca vini dal menu.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'impiegato inserisce nome e/o anno del vino.
2. L'impiegato clicca il pulsante per la ricerca.
3. Il sistema se mostra i risultati.

Flussi alternativi:

1. Nessun risultato.
 - a) Il sistema informa l'impiegato che non ci sono risultati per tale nome e/o anno.
2. L'impiegato annulla l'operazione.
 - a) In un qualsiasi momento antecedente il punto 2 del flusso normale l'impiegato torna al menu.

Eccezioni: L'impiegato o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: customer research by surname.

Tabella dati tecnici:

Use Case ID:	9		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'impiegato cerca nel sistema tutti i clienti con il cognome da lui inserito.

Attore principale: Impiegato.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'impiegato registrato ha fatto accesso al sistema.

Post-condizioni:

1. Nessuna.

Trigger: L'impiegato clicca il pulsante cerca clienti dal menu.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'impiegato inserisce il cognome con cui intende fare la ricerca.
2. L'impiegato clicca il pulsante di ricerca.
3. Il sistema mostra i risultati.

Flussi alternativi:

1. Nessun risultato.
 - a) Il sistema informa l'impiegato che non ci sono risultati per tale cognome.
2. L'impiegato annulla l'operazione.
 - a) In un qualsiasi momento antecedente il punto 2 del flusso normale l'impiegato torna al menu.

Eccezioni: L'impiegato o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: search orders by date range.

Tabella dati tecnici:

Use Case ID:	10		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'impiegato cerca nel sistema ordini di vendita, ordini di acquisto o proposte d'acquisto per intervallo di date.

Attore principale: Impiegato.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'impiegato registrato ha fatto accesso al sistema.

Post-condizioni:

1. Nessuna.

Trigger: L'impiegato clicca il pulsante cerca ordini dal menu.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'impiegato inserisce la data d' inizio e fine per la ricerca.
2. L'impiegato clicca il pulsante per cercare tra gli ordini di vendita.
3. Il sistema mostra i risultati.

Flussi alternativi:

1. L'impiegato cerca tra ordini di acquisto.
 - a) L'impiegato inserisce la data d' inizio e fine per la ricerca.
 - b) L'impiegato clicca il pulsante per cercare tra gli ordini di acquisto.
 - c) Il sistema mostra i risultati.
2. L'impiegato cerca tra le proposte d'acquisto.
 - a) L'impiegato inserisce la data d' inizio e fine per la ricerca.
 - b) L'impiegato clicca il pulsante per cercare tra le proposte d'acquisto.
 - c) Il sistema mostra i risultati.
3. Nessun risultato.
 - a) Il sistema informa l'impiegato che non ci sono risultati per tale ricerca.
4. L'impiegato annulla l'operazione.
 - a) In un qualsiasi momento prima di cliccare il pulsante per la ricerca l'impiegato torna al menu.

Eccezioni: L'impiegato o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: create promotion.

Tabella dati tecnici:

Use Case ID:	11		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'impiegato crea una promozione per un vino.

Attore principale: Impiegato.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'impiegato registrato ha fatto accesso al sistema.

Post-condizioni:

1. La promozione è salvata nel database.

Trigger: L'impiegato clicca il pulsante, dal menu, per creare una promozione.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'impiegato inserisce la data di inizio e fine della promozione.
2. L'impiegato inserisce nome e anno del vino per cui intende creare la promozione.
3. L'impiegato inserisce lo sconto in percentuale.
4. L'impiegato clicca il pulsante per creare la promozione.
5. Il sistema lo informa del successo dell'operazione.

Flussi alternativi:

1. Nessun vino.
 - a) Il sistema informa l'impiegato che per tale nome e anno non esistono vini nel sistema.
2. Intervallo già usato.
 - a) Il sistema informa l'impiegato che esiste già una promozione per tale vino con intervallo che collide con quello da lui inserito.
3. Valori non coerenti.
 - a) Il sistema informa l'impiegato che deve inserire valori coerenti (data inizio precedente data di fine, sconto tra zero e cento, etc).
4. Operazione annullata.
 - a) L'impiegato in un qualsiasi momento, precedente al click del pulsante crea promozione, torna al menu.

Eccezioni: L'impiegato o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: complete activity.

Tabella dati tecnici:

Use Case ID:	12		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'impiegato completa un'attività che gli è stata assegnata.

Attore principale: Impiegato.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'impiegato registrato ha fatto accesso al sistema.
3. L'impiegato ha ricevuto un'attività.

Post-condizioni:

1. Il completamento/ non completamento dell'attività è salvato nel sistema (database).

Trigger: L'impiegato clicca, dal menu, il pulsante completa attività.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'impiegato visualizza l'ordine di vendita assegnatogli.
2. L'impiegato prepara il materiale.
3. L'impiegato contatta il corriere informandolo che il materiale è pronto.
4. L'impiegato completa l'ordine, firmandolo digitalmente, cliccando il pulsante completa ordine di vendita.
5. Il sistema salva l'attività completata.

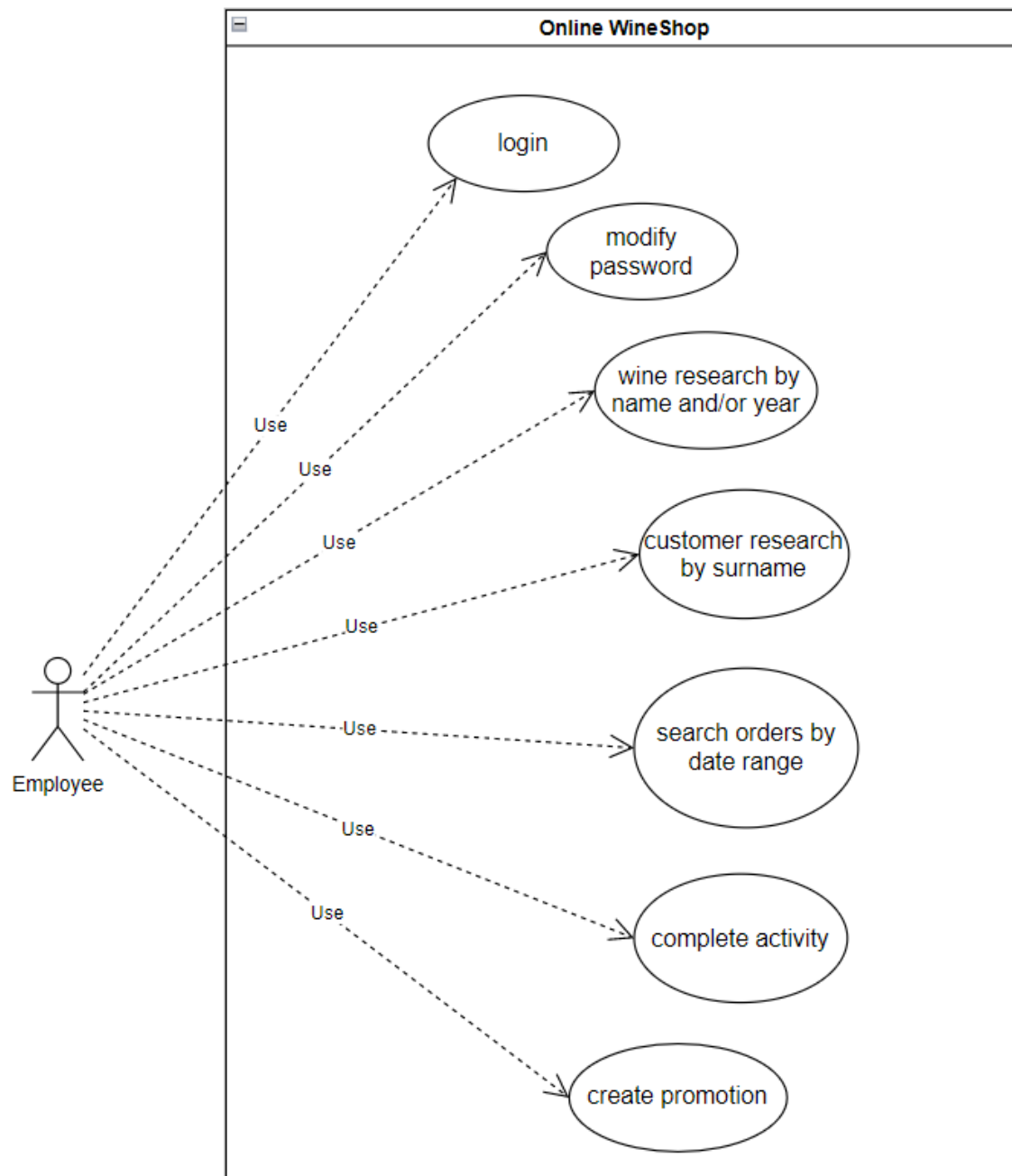
Flussi alternativi:

1. Ordine di acquisto.
 - a) L'impiegato visualizza l'ordine di acquisto assegnatogli.
 - b) L'impiegato contatta il fornitore ed effettua l'ordine.
 - c) L'impiegato, ricevuto il materiale, completa l'ordine firmandolo digitalmente tramite il pulsante completa l'ordine d'acquisto.
 - d) Il sistema salva l'attività completata.
2. Proposta d'acquisto.
 - a) L'impiegato visualizza la proposta d'acquisto assegnatagli.
 - b) L'impiegato, tramite il pulsante genera ordine d'acquisto, genera l'ordine per il fornitore partendo dai dati della proposta.
 - c) L'impiegato contatta il fornitore e gli invia l'ordine.
 - d) L'impiegato, ricevuto il materiale, clicca il pulsante completa ordine d'acquisto e genera ordine di vendita.
 - e) Se il cliente conferma l'ordine (se lo rifiuta non sono necessarie azioni da parte dell'impiegato), l'impiegato prepara il materiale.
 - f) L'impiegato contatta il corriere informandolo che il materiale è pronto.
 - g) L'impiegato completa l'ordine firmandolo digitalmente tramite il pulsante completa ordine di vendita.
 - h) Il sistema salva l'attività completata.
3. Attività non completata.
 - a) l'impiegato non completa l'attività assegnatagli per tempo.
 - b) Il sistema salva l'attività non completata dell'impiegato.

Eccezioni: L'impiegato o un bug termina il programma durante l'operazione, i dati non vengono salvati.

4.2.1 Diagramma use case employee

Di seguito è riportato il diagramma use case dell'impiegato. Siccome gli attori DBMS e sistema sono sempre coinvolti, per semplicità non sono stati riportati nel diagramma.



4.3 Documentazione use cases administrator

Di seguito la documentazione use cases administrator descrive le funzionalità che il sistema offre ad un amministratore. L'amministratore accede al sistema come un cliente, l'amministratore ha credenziali di default, pertanto l'use case di accesso non è qui riportato.

Le funzionalità cambia password, cerca vini per nome e/o anno, cerca ordini per intervallo di date e cerca clienti per cognome dell'impiegato sono analoghe per l'amministratore, pertanto non sono qui riportate.

Nome use case: register new employee.

Tabella dati tecnici:

Use Case ID:	13		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'amministratore registra un nuovo impiegato nel sistema.

Attore principale: Amministratore.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'amministratore ha fatto accesso al sistema.

Post-condizioni:

1. Il nuovo impiegato è salvato nel database.

Trigger: L'amministratore, dal menu, clicca il pulsante registra impiegato.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'amministratore inserisce i dati che identificano un impiegato.
2. L'amministratore clicca il pulsante registra.
3. Il sistema salva l'impiegato nel database e informa sull'esito dell'operazione l'amministratore.

Flussi alternativi:

1. Dati già in uso.
 - a) Il sistema informa l'amministratore che nome utente, codice fiscale o email sono già in uso nel sistema.
 - b) L'amministratore ripete l'inserimento con dati aggiornati.
 - c) Il sistema informa l'amministratore sull'esito.
2. Operazione annullata.
 - a) In un qualsiasi momento precedente al punto due del flusso normale l'amministratore torna al menu, l'operazione è annullata.

Eccezioni: L'amministratore o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: handle employee.

Tabella dati tecnici:

Use Case ID:	14		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'amministratore effettua il reset della password a un impiegato, blocca un impiegato o sblocca un impiegato.

Attore principale: Amministratore.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'amministratore ha fatto accesso al sistema.

Post-condizioni:

1. L'operazione eseguita è salvata con successo nel database.

Trigger: L'amministratore clicca il pulsante gestisci impiegato dal menu.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'amministratore inserisce il nome utente dell'impiegato che intende gestire.
2. L'amministratore clicca il pulsante resetta password.
3. Il sistema informa l'amministratore del successo dell'operazione.

Flussi alternativi:

1. Blocca impiegato.
 - a) L'amministratore inserisce il nome utente dell'impiegato che intende gestire.
 - b) L'amministratore clicca il pulsante per bloccare l'impiegato.
 - c) Il sistema informa l'amministratore sull'esito dell'operazione.
2. Sblocca impiegato.
 - a) L'amministratore inserisce il nome utente dell'impiegato che intende gestire.
 - b) L'amministratore clicca il pulsante per sbloccare l'impiegato.
 - c) Il sistema lo informa sull'esito dell'operazione.
3. Operazione annullata.
 - a) in un qualsiasi momento antecedente il click sul pulsante dell'operazione l'amministratore torna al menu e l'operazione è annullata.

Eccezioni: L'impiegato o un bug termina il programma durante l'operazione, i dati non vengono salvati.

Nome use case: create/search report.

Tabella dati tecnici:

Use Case ID:	15		
Created By:	Matteo Gianvenuti	Last Updated By:	Matteo Gianvenuti
Date Created:	13/03/2023	Date Last Updated:	13/03/2023

Descrizione: L'amministratore crea o visualizza il report mensile per un mese e anno selezionati. Se il report esiste già il sistema lo mostra, se non esiste il sistema lo crea e lo mostra.

Attore principale: Amministratore.

Attori secondari: DBMS, sistema.

Pre-condizioni:

1. DBMS, server e client sono in esecuzione.
2. L'amministratore ha fatto accesso al sistema.

Post-condizioni:

1. Il nuovo report è salvato con successo nel database.

Trigger: L'amministratore clicca il pulsante, nel menu, crea/cerca report.

Priorità: Alta.

Frequenza d'uso: Ignota.

Flusso normale:

1. L'amministratore selezione un mese e anno (il giorno è ininfluente).
2. L'amministratore clicca il pulsante per creare/cercare il report.
3. Il sistema mostra il report.

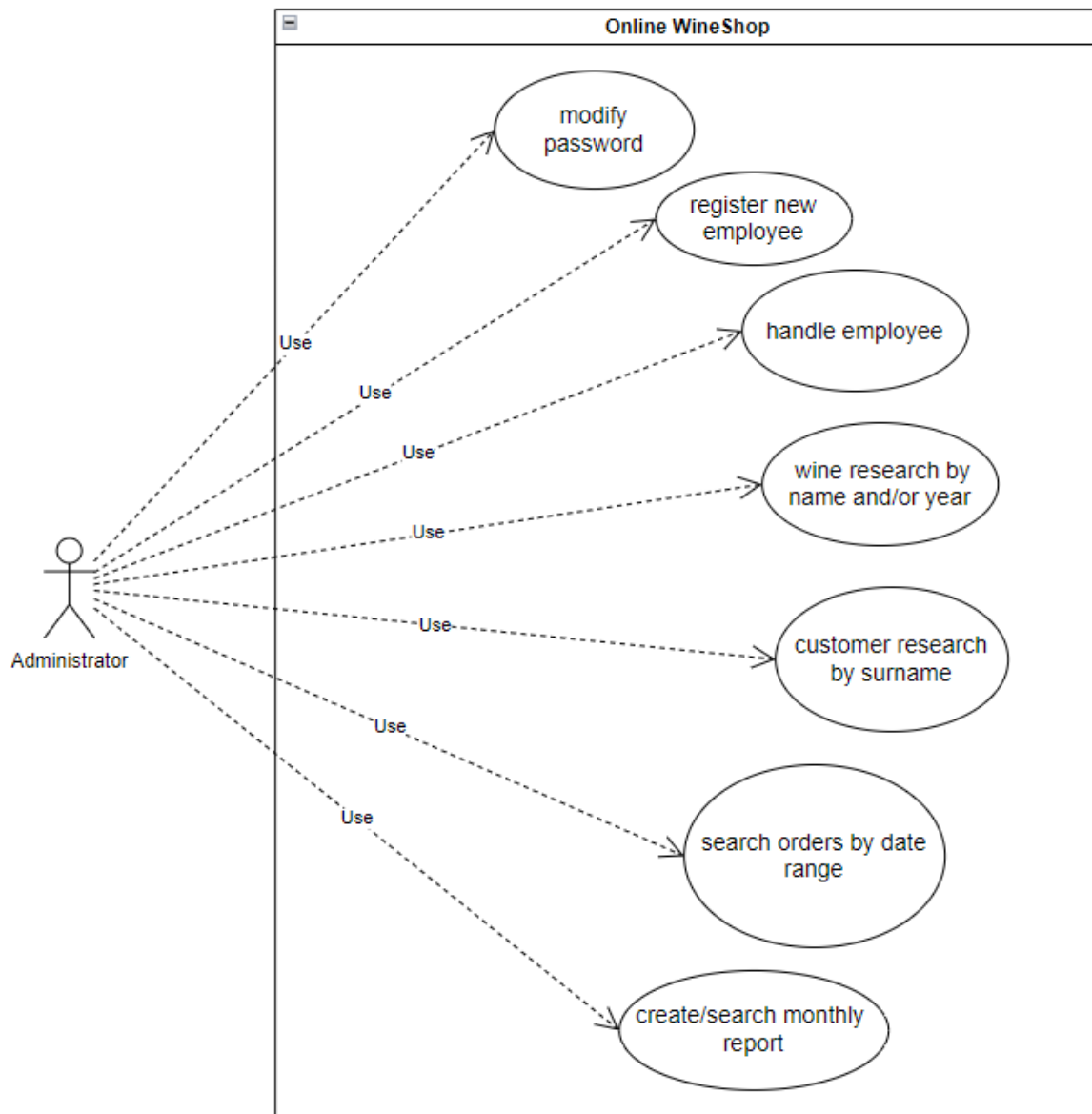
Flussi alternativi:

1. Il report non esiste.
 - a) L'amministratore seleziona un mese e anno.
 - b) L'amministratore clicca il pulsante per creare/cercare il report.
 - c) Il sistema crea il report e lo salva nel database.
 - d) Il sistema mostra il report.
2. Input incoerente.
 - a) L'amministratore selezione un mese e anno (un mese non concluso).
 - b) Il sistema informa l'amministratore che il mese deve essere concluso per effettuare il repost delle finanze.
3. Operazione annullata.
 - a) L'amministratore in qualsiasi momento precedente al click sul pulsante crea/cerca report torna al menu, l'operazione è annullata.

Eccezioni: L'amministratore o un bug termina il programma durante l'operazione, i dati non vengono salvati.

4.3.1 Diagramma use case administrator

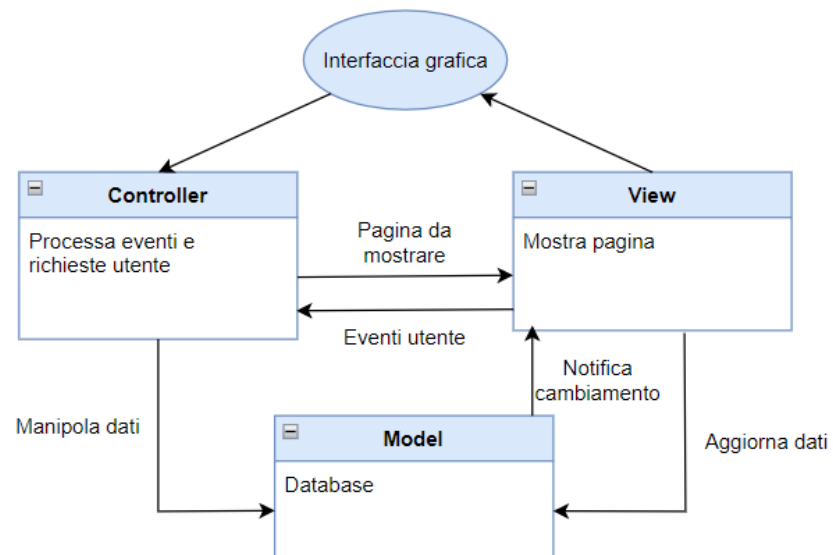
Di seguito è riportato il diagramma use case dell'amministratore. Siccome gli attori DBMS e sistema sono sempre coinvolti, per semplicità non sono stati riportati nel diagramma.



5. Architectural Model

5.1 Architecture diagram

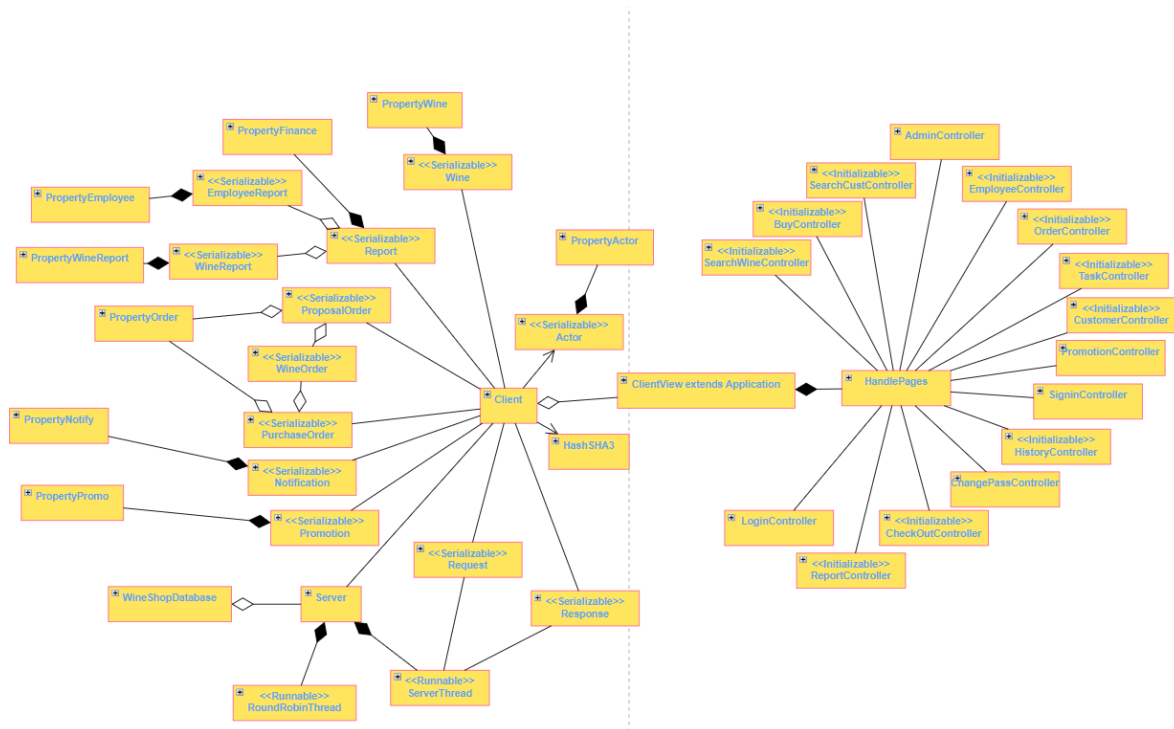
L'architettura utilizzata è Model-View-Controller(MVC).



6. Detailed Software Model

6.1 Class diagram(s)

6.1.1 Diagramma complessivo



6.1.2 Diagrammi delle singole classi



<div> <div></div> <div> <div><<Serializable>></div> <div>EmployeeReport</div> </div> </div>	
<div> <div>- serialVersionUID: long</div> <div>- email: String</div> <div>- numCompleted: int</div> <div>- numExpired: int</div> </div>	
<div> <div>+ EmployeeReport(String,int,int)</div> <div>+ getEmail(): String</div> <div>+ getNumCompleted(): int</div> <div>+ getNumExpired(): int</div> </div>	

<div> <div></div> <div>PropertyEmployee</div> </div>	
<div> <div>- email: SimpleStringProperty</div> <div>- numCompleted: SimpleIntegerProperty</div> <div>- numExpired: SimpleIntegerProperty</div> </div>	
<div> <div>+ PropertyEmployee(String,int,int)</div> <div>+ getEmail(): String</div> <div>+ getNumCompleted(): int</div> <div>+ getNumExpired(): int</div> </div>	

<div> <div></div> <div> <div><<Serializable>></div> <div>WineReport</div> </div> </div>	
<div> <div>- serialVersionUID: long</div> <div>- wineName: String</div> <div>- wineYear: int</div> <div>- numSell: int</div> <div>- numAv: int</div> <div>- income: double</div> </div>	
<div> <div>+ WineReport(String,int,int,int,double)</div> <div>+ getWineName(): String</div> <div>+ getWineYear(): int</div> <div>+ getNumSell(): int</div> <div>+ getNumAv(): int</div> <div>+ getIncome(): double</div> </div>	

<div> <div></div> <div>PropertyWineReport</div> </div>	
<div> <div>- wineName: SimpleDoubleProperty</div> <div>- wineYear: SimpleStringProperty</div> <div>- numSell: SimpleDoubleProperty</div> <div>- numAv: SimpleIntegerProperty</div> <div>- income: SimpleIntegerProperty</div> </div>	
<div> <div>+ PropertyWineReport(String,int,int,int,double)</div> <div>+ getWineName(): String</div> <div>+ getWineYear(): int</div> <div>+ getNumSell(): int</div> <div>+ getNumAv(): int</div> <div>+ getIncome(): double</div> </div>	

<div> <div></div> <div> <div><<Serializable>></div> <div>Wine</div> </div> </div>	
<div> <div>- serialVersionUID: long</div> <div>- name: String</div> <div>- producer: String</div> <div>- name: String</div> <div>- origin: String</div> <div>- year: int</div> <div>- TechnicalNotes: String</div> <div>- GrapeVariety: String</div> <div>- quality: double</div> <div>- price: double</div> <div>- supplierPrice: double</div> <div>- quantity: int</div> <div>- SalesNumber: int</div> </div>	
<div> <div>+ public PropertyWine(String,String,String,int,String,String,double,double,double,int,int)</div> <div>+ getName(): String</div> <div>+ getProducer(): String</div> <div>+ getOrigin(): String</div> <div>+ getYear(): int</div> <div>+ getTechnicalNotes(): String</div> <div>+ getGrapeVariety(): String</div> <div>+ getQuality(): double</div> <div>+ getPrice(): double</div> <div>+ getSupplierPrice(): double</div> <div>+ getQuantity(): int</div> <div>+ getSalesNumber(): int</div> </div>	

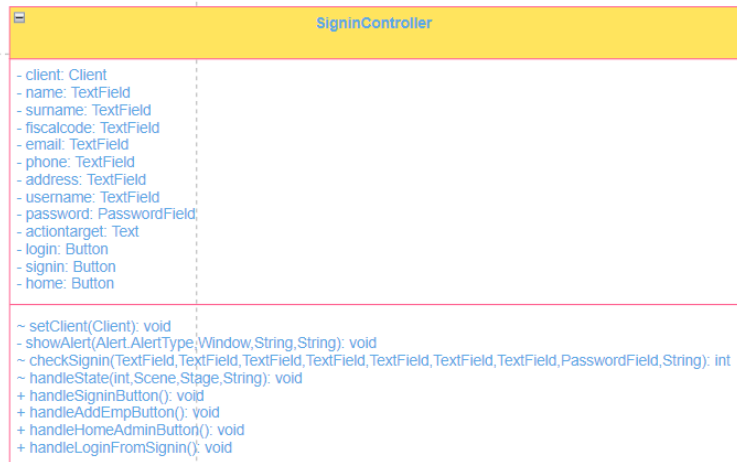
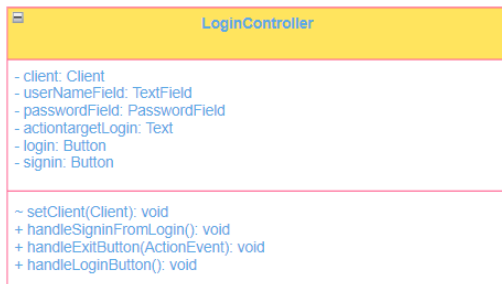
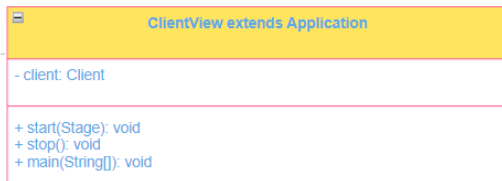
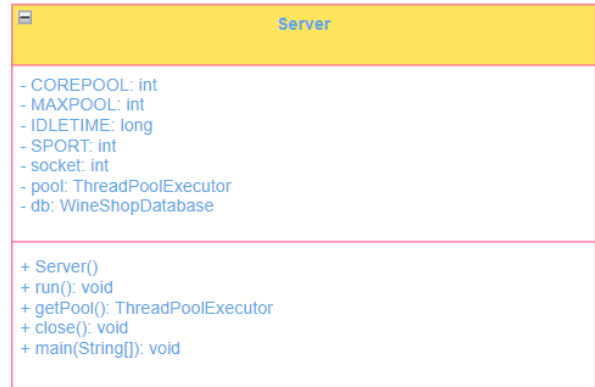
<div> <div></div> <div>PropertyWine</div> </div>	
<div> <div>- name: SimpleStringProperty</div> <div>- producer: SimpleStringProperty</div> <div>- name: SimpleStringProperty</div> <div>- origin: SimpleStringProperty</div> <div>- year: SimpleIntegerProperty</div> <div>- TechnicalNotes: SimpleStringProperty</div> <div>- GrapeVariety: SimpleStringProperty</div> <div>- quality: SimpleDoubleProperty</div> <div>- price: SimpleDoubleProperty</div> <div>- supplierPrice: SimpleDoubleProperty</div> <div>- quantity: SimpleIntegerProperty</div> <div>- SalesNumber: SimpleIntegerProperty</div> </div>	
<div> <div>+ public PropertyWine(String,String,String,int,String,String,double,double,double,int,int)</div> <div>+ getName(): String</div> <div>+ getProducer(): String</div> <div>+ getOrigin(): String</div> <div>+ getYear(): int</div> <div>+ getTechnicalNotes(): String</div> <div>+ getGrapeVariety(): String</div> <div>+ getQuality(): double</div> <div>+ getPrice(): double</div> <div>+ getSupplierPrice(): double</div> <div>+ getQuantity(): int</div> <div>+ getSalesNumber(): int</div> </div>	

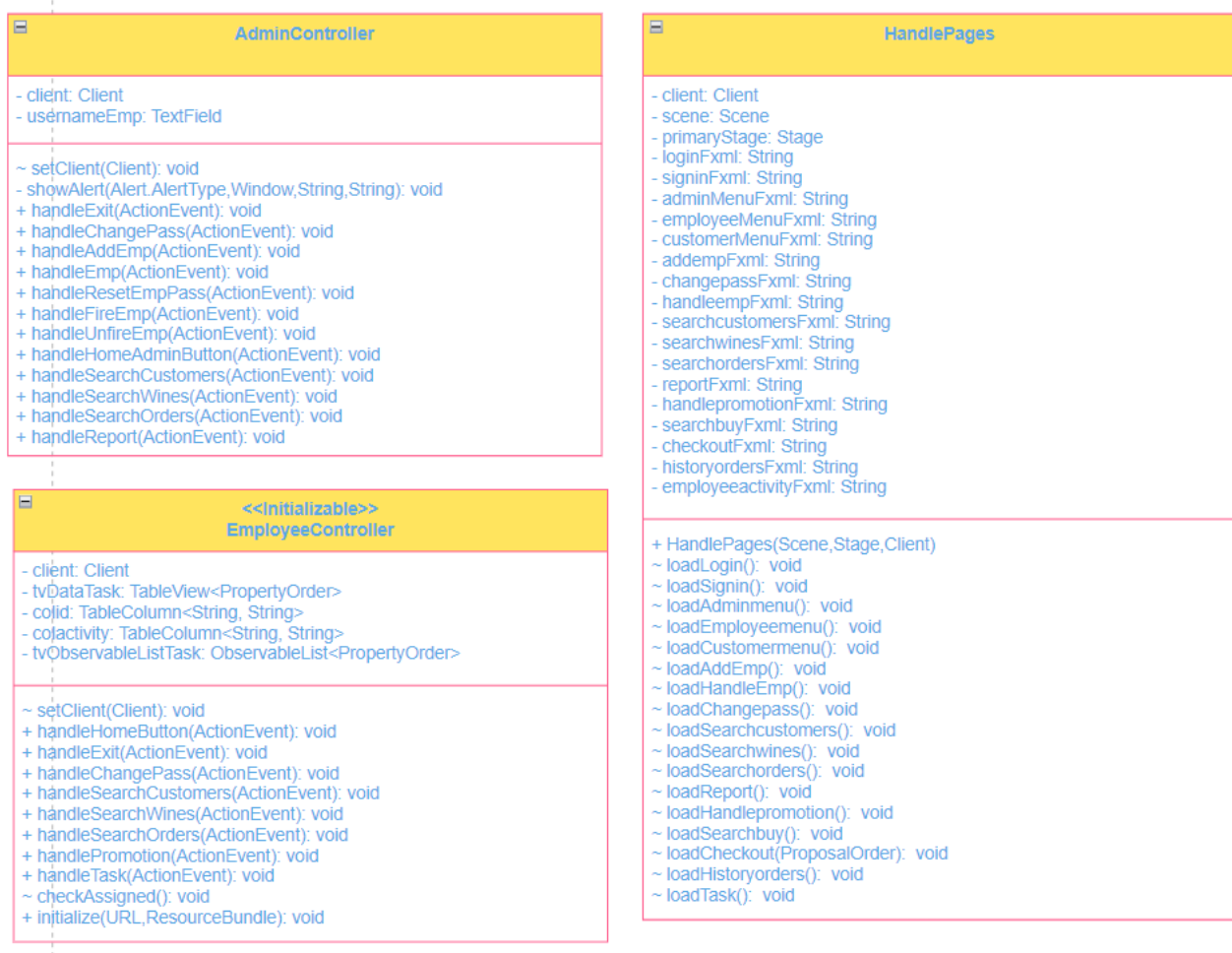


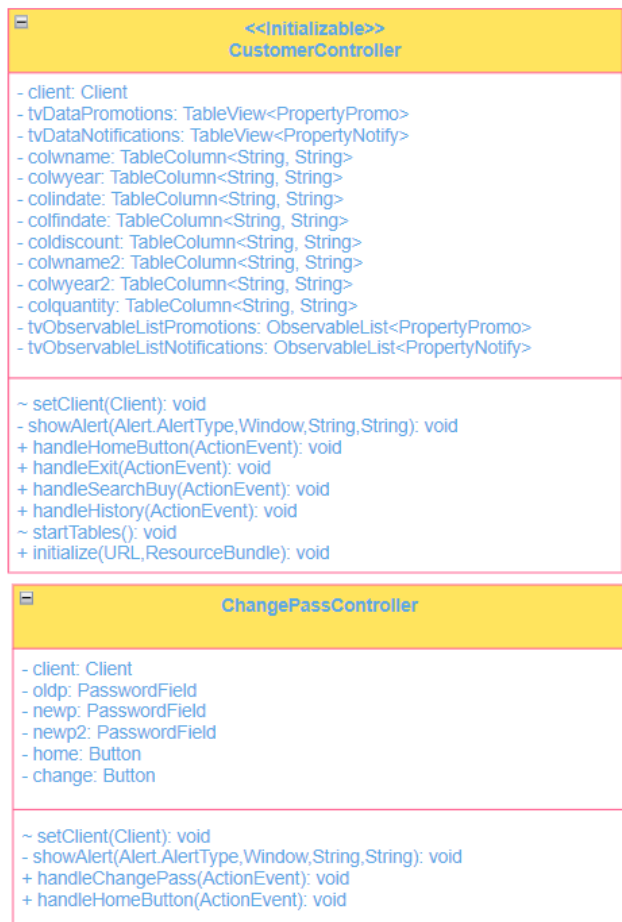
WineShopDatabase
<ul style="list-style-type: none"> - DBURL: String - ARGS: String - LOGIN: String - PASSWORD: String - employees: List<Actor> - orders: List<ProposalOrder> - id_newword: int - confirmEmps: List<Actor> - assigned: int - index: int
<ul style="list-style-type: none"> + WineShopDatabase() - executeUpdate(String): void - start_users(): void - start_wines(): void - start_salesOrders(): void - start_notifications(): void - start_proposalsPurchase(): void - start_purchaseOrders(): void - start_newOrders(): void - start_ids(): void - start_reports(): void - start_employeeereports(): void - start_winereports(): void - start_promotions(): void - start_assignments(): void ~ findWine(String,int): List<Wine> ~ logoutEmployee(String): void ~ login(String,String): Actor ~ signin(Actor): Actor ~ changePass(String,String,String): boolean ~ resetPass(String): boolean ~ searchCustomers(String): List<Actor> ~ getOperators(): List<Actor> ~ fireEmployee(String,String): boolean ~ searchProposalOrders(LocalDate,LocalDate,String): List<ProposalOrder> ~ searchPurchaseOrders(LocalDate,LocalDate): List<PurchaseOrder> ~ getAssignment(String): List<ProposalOrder> ~ report(LocalDate): Report ~ SelectReport(LocalDate,Report,ResultSet): Report ~ CreateReport(LocalDate, Report): Report ~ createPromotion(String,int,int,LocalDate,LocalDate): boolean ~ getPromotions(): List<Promotion> ~ insertProposalOrder(ProposalOrder,String): void ~ checkExpired(int): boolean ~ handleProposalOrder(ProposalOrder,String): boolean ~ generatePurchaseOrder(ProposalOrder): List<PurchaseOrder> ~ insertPurchaseOrder(ProposalOrder): boolean ~ checkQuantity(LocalDate): void ~ makeProposalOrder(ProposalOrder): boolean ~ checkAssignments(): void ~ assignments(): void ~ getCircleActor(): Actor ~ RoundRobin(): void ~ getNewOrders(int): void ~ createNotification(String,int,String,int): boolean ~ getNotifications(String): List<Notification> ~ getHistory(String): List<ProposalOrder>

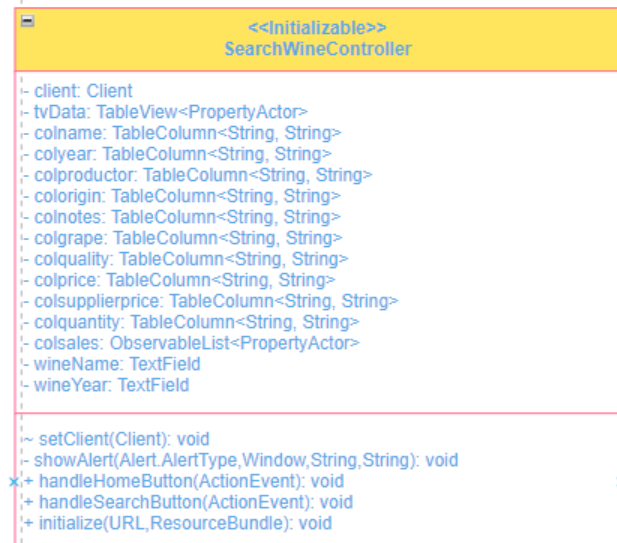
Client
<ul style="list-style-type: none"> - SPORT: int - SHOST: String - client: Socket - inputStream: ObjectInputStream - outputStream: ObjectOutputStream - winesOrder: List<WineOrder> - winesProposal: List<WineOrder> - myuser: Actor
<ul style="list-style-type: none"> + Client() ~ getUser(): Actor ~ hashFunction(String): String ~ sendRequest(Request): void ~ readResponse(): Response ~ login(String,String): Actor ~ logout(): boolean ~ close(): boolean ~ signin(String,String,String,String,String,String,String,String,String): boolean ~ searchWine(String,int): List<Wine> ~ changePass(String,String,String): boolean ~ resetPass(String): boolean ~ searchCustomers(String): List<Actor> ~ getOperators(): List<Actor> ~ searchCustOrders(LocalDate,LocalDate,String): List<ProposalOrder> ~ fireEmployee(String): boolean ~ unfireEmployee(String): boolean ~ searchPurchaseOrders(LocalDate,LocalDate): List<PurchaseOrder> ~ makeReport(LocalDate): Report ~ createPromotion(String,int,int,LocalDate,LocalDate): boolean ~ getPromotions(): List<Promotion> ~ makeProposalOrder(boolean): int ~ completeSalesOrder(ProposalOrder): boolean ~ completeProposalPurchase(ProposalOrder): boolean ~ completePurchaseOrder(ProposalOrder): boolean ~ generatePurchaseOrder(ProposalOrder): List<PurchaseOrder> ~ rejectSalesOrder(ProposalOrder): boolean ~ confirmSalesOrder(ProposalOrder): boolean ~ getNotifications(): List<Notification> ~ createNotification(String,int,int): int ~ checkOrderLenght(boolean): boolean ~ cancelProposalOrder(boolean): void ~ addWineToProposalOrder(boolean,String,int,int,int,int): int ~ getWineProposalOrder(boolean): List<WineOrder> ~ getHistory(): List<ProposalOrder> ~ getAssignment(): List<ProposalOrder>

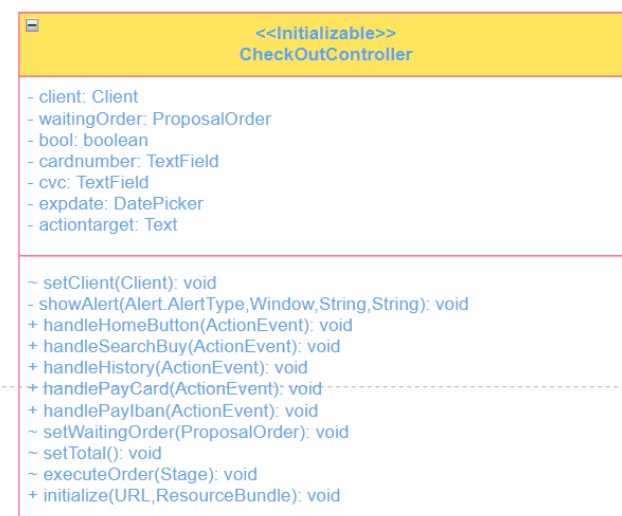
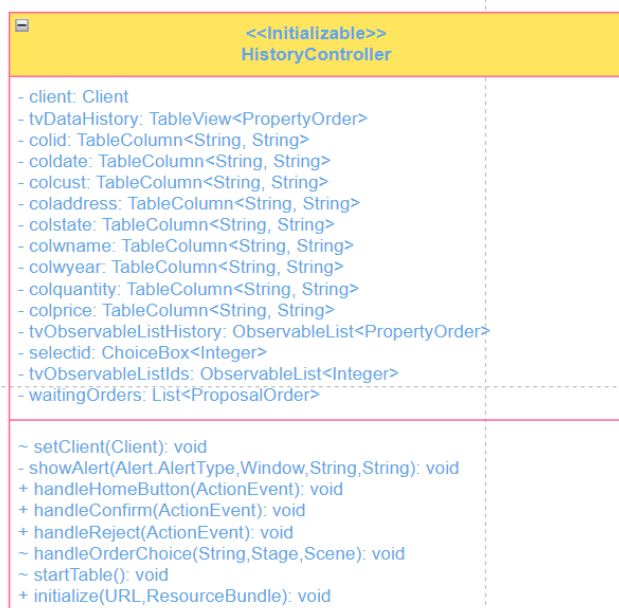
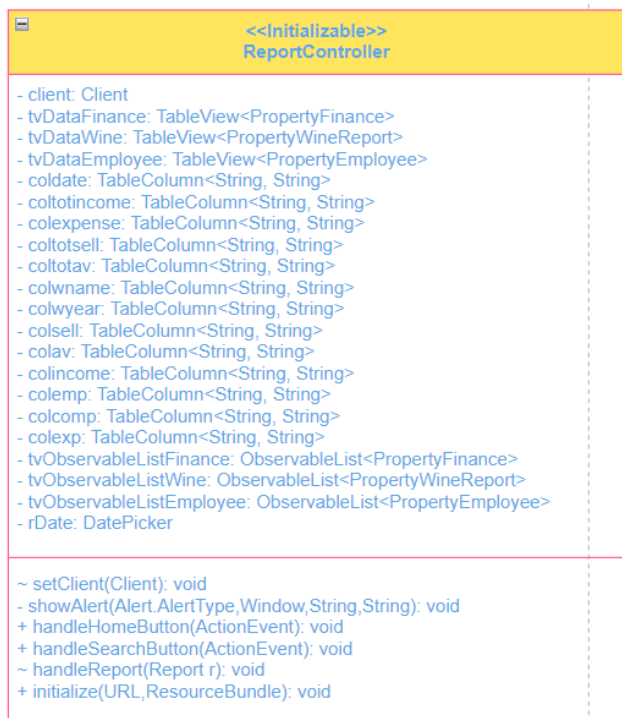
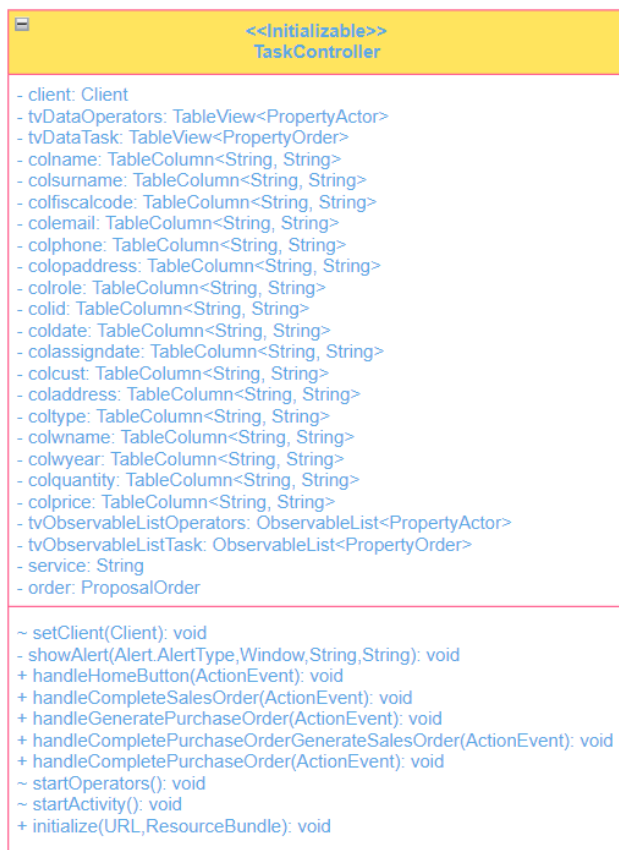
Hash SHA3
<ul style="list-style-type: none"> + executeHash(String): String











7. Test Funzionali

Di seguito sono riportati gli unit test delle principali classi base. I test sono stati eseguiti tramite Junit5.

7.1 Actor class tests

Di seguito i test dei metodi della classe Actor.

ID	Nome del test	Passi	Risultato aspettato	Riusltato del test	Commenti
0.	getActorName	Avviare la classe ProjectTests	Passato	Passato	
1.	getActorSurname	Avviare la classe ProjectTests	Passato	Passato	
2.	getActorUsername	Avviare la classe ProjectTests	Passato	Passato	
3.	getActorPassword	Avviare la classe ProjectTests	Passato	Passato	
4.	getActorDefaultPassword	Avviare la classe ProjectTests	Passato	Passato	
5.	getActorFiscalCode	Avviare la classe ProjectTests	Passato	Passato	
6.	getActorEmail	Avviare la classe ProjectTests	Passato	Passato	
7.	getActorPhone	Avviare la classe ProjectTests	Passato	Passato	
8.	getActorAddress	Avviare la classe ProjectTests	Passato	Passato	
9.	getActorRole	Avviare la classe ProjectTests	Passato	Passato	

7.2 Notification class tests

Di seguito i test dei metodi della classe Notification.

ID	Nome del test	Passi	Risultato aspettato	Riusltato del test	Commenti
10.	getNotificanWineName	Avviare la classe ProjectTests	Passato	Passato	
11.	getNotificationWineYear	Avviare la classe ProjectTests	Passato	Passato	
12.	getNotificationQuantity	Avviare la classe ProjectTests	Passato	Passato	
13.	getNotificationEmail	Avviare la classe ProjectTests	Passato	Passato	

7.3 Promotion class tests

Di seguito i test dei metodi della classe Promotion.

ID	Nome del test	Passi	Risultato aspettato	Riusltato del test	Commenti
14.	getPromotionInDate	Avviare la classe ProjectTests	Passato	Passato	
15.	getPromotionFinDate	Avviare la classe ProjectTests	Passato	Passato	
16.	getPromotionDiscount	Avviare la classe ProjectTests	Passato	Passato	
17.	getPromotionWineName	Avviare la classe ProjectTests	Passato	Passato	
18.	getPromotionWineYear	Avviare la classe ProjectTests	Passato	Passato	

7.4 Wine class tests

Di seguito i test dei metodi della classe Wine.

ID	Nome del test	Passi	Risultato aspettato	Riusltato del test	Commenti
19.	getWineName	Avviare la classe ProjectTests	Passato	Passato	
20.	getWineYear	Avviare la classe ProjectTests	Passato	Passato	
21.	getWineProductor	Avviare la classe ProjectTests	Passato	Passato	
22.	getWineOrigin	Avviare la classe ProjectTests	Passato	Passato	
23.	getWineTecNotes	Avviare la classe ProjectTests	Passato	Passato	
24.	getWineGrape	Avviare la classe ProjectTests	Passato	Passato	
25.	getWinePrice	Avviare la classe ProjectTests	Passato	Passato	
26.	getWineSupplierPrice	Avviare la classe ProjectTests	Passato	Passato	
27.	getWineQuality	Avviare la classe ProjectTests	Passato	Passato	
28.	getWineQuantity	Avviare la classe ProjectTests	Passato	Passato	
29.	getWineSalesNumber	Avviare la classe ProjectTests	Passato	Passato	

8. Manuale di sistema

8.1 Installazione

8.1.1 Installare XAMPP

Installare una versione recente di XAMPP.

XAMPP fornisce un database MySQL al sistema, attivo sulla porta 3306. In alternativa si può installare un qualsiasi altro provider di database con le stesse regole SQL, attivo sulla stessa porta, tenendo in considerazione però che XAMPP permette di visualizzare in localhost il database tramite phpMyAdmin.

8.1.2 Installare Eclipse IDE

Installare una versione recente di Eclipse IDE.

Eclipse IDE, è lo strumento di sviluppo che è stato utilizzato per sviluppare il software, testare e provare il sistema.

8.1.3 Scaricare i pacchetti aggiuntivi

Scaricare open jfx 19 pacchetto necessario per la grafica. Scaricare un connector, mysql-connector-j, Java MySQL necessario per potersi connettere al database tramite Java.

8.2 Configurazione

8.2.1 Avviare XAMPP

Aprire il control panel di XAMPP e cliccare il pulsante start per MySQL (avvia il database). Cliccare il pulsante start per Apache se si vuole visualizzare tramite phpMyadmin il database in localhost.

8.2.2 Avviare Eclipse IDE

Aprire Eclipse IDE, importare il progetto WineShop.

8.2.3 Importare i pacchetti aggiuntivi

Aggiungere le librerie JavaFX (open jfx 19) e mysql-connector-j per il progetto WineShop in Build Path -> Libraries -> Classpath.

Per la classe ClientView.java che implementa Application è necessario, per poter utilizzare JavaFX, andare su Run As -> Run Configuration -> Arguments -> VM arguments, scrivere: --module-path "percorso assoluto di javafx-sdk-19\lib" --add-modules javafx.controls,javafx.fxml.

8.2.4 Avviare il sistema

Mettere in esecuzione le classi Server.java e ClientView.java in questo ordine.

8.3 Guida utente

8.3.1 Guida clienti

8.3.1.1 Registrazione

Dopo aver avviato il sistema cliccare sul pulsante “Registrati”, questo caricherà la pagina di registrazione. Per potersi registrare con successo un nuovo utente deve inserire: nome, cognome, codice fiscale, indirizzo email, numero di telefono, indirizzo di spedizione, nome utente e password, con un limite di 30 caratteri escluso l’indirizzo che ha limite 500.

Ovviamente email, codice fiscale e nome utente devono essere univoci quindi il sistema non permetterà la registrazione se uno di questi è stato già utilizzato.

Nell'operazione di registrazione bisogna porre particolare attenzione all’indirizzo di spedizione che vuole inserire siccome a questo indirizzo verrà spedito il materiale acquistato.

8.3.1.2 Accesso

Dopo aver avviato il sistema un utente registrato può accedere tramite nome utente e password, definiti in fase di registrazione, inserendo i dati nelle apposite textbox e cliccando il pulsante “Accedi”, dopodichè si visualizza il menu clienti con le sue funzionalità quali possibilità di cercare vini, comprare vini tramite ordini di vendita e inviare proposte di acquisto e chiedere di essere notificato quando un vino sarà disponibile in quantità desiderata, inoltre l’utente registrato può visualizzare lo storico degli ordini di vendita, confermare e rifiutare ordini di vendita generati dalle proposte fatte. In più dal menu può visualizzare le promozioni in corso e le notifiche di disponibilità dei vini richieste.

8.3.1.3 Cercare vini

Un cliente dopo un accesso autenticato al sistema clicca sul pulsante “Cerca/compra vini”, dopodichè nella parte superiore della pagina visualizza due textbox una dove inserire il nome del vino e una dove inserire opzionalmente l’anno del vino a questo punto cliccando il pulsante “Cerca” visualizza per tutti i vini (con tale nome e/o anno): nome del vino, anno di produzione, produttore, provenienza, note tecniche, vitigni, qualità in percentuale (0-100, indicata dal produttore), prezzo di una bottiglia (senza tenere conto di promozioni o sconti) e unità disponibili.

8.3.1.4 Creare un ordine di vendita

Un cliente dopo un accesso autenticato al sistema può cliccare sul pulsante “Cerca/compra vini”, dopodichè nella parte inferiore della pagina visualizza due textbox una dove inserire il nome del vino e una dove inserire l’anno del vino per identificare univocamente il vino che intende aggiungere all’ordine di vendita. Il cliente deve inserire la quantità desiderata, che deve essere minore o uguale a quella disponibile, in singole bottiglie da zero a cinque (default uno, può scendere a zero se incrementa il numero di casse, che hanno default 0), numero di casse da sei bottiglie e numero di casse da 12 bottiglie. Inserito il nome del vino, anno del vino e quantità desiderata il cliente cliccando il pulsante “Aggiungi a ordine di vendita” inserisce il vino e la quantità nell’ordine. L’operazione può essere ripetuta quante volte si vuole. Una volta inseriti i vini che si intende acquistare il cliente, cliccando il pulsante “Invia ordine di vendita e paga”, visualizza la pagina di pagamento dove può decidere se pagare tramite IBAN o carta di credito. Fatto il pagamento può visualizzare l’ordine nello storico, accessibile dal menu e dalla stessa pagina di pagamento tramite il pulsante “Storico ordini”. il cliente può anche tornare alla pagina di acquisto/ricerca dei vini tramite il pulsante “Torna ad acquistare”. Se il cliente non paga tramite una delle due opzioni l’ordine non viene eseguito.

8.3.1.5 Cancellare un ordine di vendita

Un ordine di vendita è un ordine di sessione, se non confermato tramite pagamento e viene cancellato automaticamente al logout, un cliente può comunque cancellare un ordine non confermato tramite pagamento dalla pagina di acquisto/ricerca dei vini (accessibile dal menu tramite il pulsante “Cerca/compra vino”) tramite il pulsante “Cancella ordine di vendita”.

8.3.1.6 Creare una proposta d’acquisto

Un cliente dopo un accesso autenticato al sistema andando nella pagina di acquisto/ricerca dei vini tramite il pulsante "Cerca/compra vino" visualizza nella parte inferiore della pagina due textbox una dove inserire il nome del vino e una dove inserire l'anno del vino per identificare univocamente il vino che intende aggiungere alla proposta d'acquisto. Scelto il nome, l'anno del vino e inserita la quantità desiderata, che deve essere maggiore di quella disponibile, in singole bottiglie da zero a cinque (default uno, può scendere a zero se incrementa il numero di casse, che hanno default 0), numero di casse da sei bottiglie e numero di casse da 12 bottiglie. Inserito il nome del vino, anno del vino e quantità desiderata il cliente cliccando il pulsante "Aggiungi a proposta d'acquisto" inserisce il vino e la quantità nella proposta. Il cliente può iterare il procedimento per inserire altri vini nella proposta. Il cliente può inviare la proposta tramite il pulsante "Invia proposta d'acquisto".

8.3.1.7 Cancellare un'acquisto

Una proposta d'acquisto come l'ordine di vendita è una proposta di sessione, se non inviata tramite l'apposito pulsante viene cancellata automaticamente al logout, un cliente può comunque cancellare una proposta non inviata dalla pagina di acquisto/ricerca dei vini (accessibile dal menu tramite il pulsante "Cerca/compra vino") tramite il pulsante "Cancella proposta d'acquisto".

8.3.1.8 Rifiutare un ordine di vendita generato da una proposta d'acquisto

Un cliente dopo un accesso autenticato può rifiutare un ordine di vendita generato a partire da una proposta andando nella pagina storico ordini dal menu tramite l'apposito pulsante "Storico ordini". Il cliente nella parte superiore della pagina visualizza una tabella con tutti gli ordini di vendita mentre nella parte inferiore della pagina trova un menu a tendina, per rifiutare un ordine seleziona l'id ordine e clicca il pulsante "Rifiuta ordine".

8.3.1.9 Confermare un ordine di vendita generato da una proposta d'acquisto

Un cliente dopo un accesso autenticato può confermare un ordine di vendita generato a partire da una proposta andando nella pagina storico ordini dal menu tramite l'apposito pulsante "Storico ordini". Il cliente nella parte superiore della pagina visualizza una tabella con tutti gli ordini di vendita mentre nella parte inferiore della pagina trova un menu a tendina, per confermare un ordine seleziona l'id ordine e clicca il pulsante "Conferma ordine e paga", visualizza la pagina di pagamento e paga tramite il sistema. L'ordine verrà considerato confermato solo dopo il pagamento.

8.3.1.10 Richiedere una notifica di disponibilità

Un cliente dopo un accesso autenticato al sistema andando nella pagina di acquisto/ricerca dei vini tramite il pulsante "Cerca/compra vino" visualizza nella parte inferiore della pagina due textbox una dove inserire il nome del vino e una dove inserire l'anno del vino per identificare univocamente il vino per cui intende essere notificato. Scelto il nome, l'anno del vino e selezionata la quantità desiderata, che deve essere maggiore di quella disponibile, in singole bottiglie da zero a cinque (default uno, può scendere a zero se incrementa il numero di casse, che hanno default 0), numero di casse da sei bottiglie e numero di casse da 12 bottiglie. Il cliente cliccando il pulsante "Notificami disponibilità" richiede la notifica per tale vino. Quando sarà disponibile la quantità desiderata il sistema mostrerà una notifica nella tabella notifiche del menu principale, visualizzata la notifica viene cancellata automaticamente.

8.3.1.11 Visualizzare lo storico ordini

Un cliente dopo un accesso autenticato al sistema cliccando il pulsante "Storico ordini" visualizza una tabella con tutti gli ordini di vendita effettuati.

8.3.1.12 Menu e logout

Per tornare al menu principale da qualsiasi pagina basta cliccare il pulsante “Casa”.
Per uscire dal sistema basta cliccare il pulsante “Esci” dal menu.

8.3.2 Guida impiegati

Menu e logout funzionano allo stesso modo del cliente, non sono quindi riportati.

8.3.2.1 Accesso

Per accedere al sistema prima deve essere stato registrato dall'amministratore. Un impiegato registrato accede con nome utente e password forniti dall'amministratore.

8.3.2.2 Cambio password

Dopo un accesso autenticato cliccando il pulsante “Cambia password” dal menu visualizza tre textbox una per inserire la vecchia password e due per inserire la nuova. Il testo inserito nelle due textbox della nuova password deve coincidere, essere lungo almeno un carattere e massimo 30.

Se l'impiegato non ricorda la vecchia password deve chiedere il reset della password all'amministratore.

8.3.2.3 Cerca clienti per cognome

Dopo un accesso autenticato cliccando il pulsante “Cerca clienti” dal menu visualizza una textbox per inserire il cognome del cliente, cliccando il pulsante “Cerca” vengono mostrati, nella tabella sottostante, nome utente, nome, cognome, codice fiscale, email, telefono, e indirizzo di spedizione per tutti i clienti con il cognome inserito.

8.3.2.4 Cerca vini per nome e/o anno di produzione

Dopo un accesso autenticato cliccando il pulsante “Cerca vini” dal menu visualizza due textbox per inserire nome e opzionalmente anno del vino. Cliccando il pulsante “Cerca” visualizza nella tabella sottostante per ogni vino con tale nome e/o anno: nome, anno, produttore, provenienza, note tecniche, vitigni, qualità, prezzo, prezzo fornitore, quantità e numero vendite.

8.3.2.5 Cerca ordini per intervallo di date

Dopo un accesso autenticato cliccando il pulsante “Cerca ordini” dal menu visualizza due datepicker per scegliere da di inizio e data di fine della ricerca, e tre pulsanti “Cerca ordini di vendita”, “Cerca proposte di acquisto” e “Cerca ordini di acquisto”. Selezionate le date, cliccando uno dei tre pulsanti, visualizza nella tabella sottostante per tutti gli ordini nell'intervallo (a seconda del tipo di ordine): id, data, email del cliente, email dell' impiegato, indirizzo di spedizione stato dell'ordine, nome vino, anno vino, quantità e prezzo.

8.3.2.6 Crea promozione

Dopo un accesso autenticato cliccando il pulsante “Crea promozione” dal menu visualizza due datepicker per inserire data di inizio e data di fine della promozione, tre textbox per nome del vino, anno del vino e sconto (in percentuale). Inseriti i dati clicca il pulsante “Crea” per creare la promozione.

8.3.2.7 Completa attività

Dopo un accesso autenticato aggiornando il menu visualizza nella tabella “Notifica attività assegnata”, se gli

è stata assegnata un'attività, l'id dell'ordine e il tipo. Per completare l'attività inizia cliccando il pulsante nel menu "Completa attività", visualizza una pagina con due tabelle da dove trova le informazioni sugli operatori (corriere e fornitore) e una dove trova informazioni a seconda dell'attività da completare (ordine di vendita/proposta d'acquisto/ordine d'acquisto): id, data, data assegnamento, email del cliente, indirizzo di spedizione, tipo ordine, nome vino, anno vino, quantità, prezzo/prezzo fornitore. Segue a seconda del tipo di ordine.

8.3.2.7.1 Completa ordine di vendita

Per completare un ordine di vendita visualizza i dati di recapito del corriere nella tabella operatori, e informa quest'ultimo quando ha preparato il materiale da spedire, indicato nella seconda tabella dove si trovano le informazioni sull'attività. Successivamente completa l'ordine firmandolo digitalmente cliccando il pulsante "Completa ordine di vendita".

8.3.2.7.2 Completa proposta d'acquisto

Per completare una proposta d'acquisto, l'impiegato clicca il pulsante "Genera ordine d'acquisto", visualizza nella tabella con informazioni sull'attività un ordine per il fornitore generato a partire dai dati della proposta, successivamente contatta il fornitore con i dati di recapito nella tabella operatori, ottiene il materiale dal fornitore, e clicca il pulsante "Completa ordine d'acquisto e genera ordine di vendita". La prima fase dell'attività è completata. Se il cliente rifiuta l'ordine non sono necessarie altre azioni, mentre se lo conferma l'impiegato riceve l'ordine generato confermato che va gestito come un qualsiasi altro ordine di vendita.

8.3.2.7.3 Completa ordine d'acquisto

Per completare una proposta d'acquisto l'impiegato contatta il fornitore tramite i dati nella tabella operatori, invia l'ordine al fornitore, ricevuto il materiale completa l'ordine firmandolo digitalmente cliccando il pulsante "Completa ordine d'acquisto".

8.3.3 Guida amministratore

Menu e logout funzionano allo stesso modo del cliente, non sono quindi riportati.

Cambia password, cerca clienti per cognome, cerca vini e cerca ordini per intervallo di date che funzionano allo stesso modo dell'impiegato, non sono qui riportate.

8.3.3.1 Accesso

L'amministratore come gli altri utenti accede al sistema tramite nome utente e password. L'amministratore ha credenziali di default.

8.3.3.2 Aggiungi impiegato

Per aggiungere un impiegato clicca il pulsante "Aggiungi impiegato" dal menu (dopo un accesso autenticato), visualizza una pagina dove può inserire nome, cognome, codice fiscale, email, numero di telefono, indirizzo di residenza e password. Email, codice fiscale e nome utente devono essere univoci. Inseriti i dati clicca il pulsante "Registra".

8.3.3.3 Gestisci impiegato

Dal menu cliccando il pulsante "Gestisci impiegato" visualizza una pagina dove trova una textbox per

inserire nome utente dell'impiegato che intende gestire, inserito nome utente può:

- Cliccando il pulsante "Reset password", impostare come password dell'impiegato la password di registrazione.
- Cliccando il pulsante "Cancella (blocca)", impedire ad un impiegato di accedere al sistema in futuro.
- Cliccando il pulsante "Sblocca", permettere ad un impiegato di accedere nuovamente al sistema.

8.3.3.4 Crea/cerca report

Dal menu cliccando il pulsante "Crea/cerca report" visualizza una pagina con tre tabelle una dove trova i dati finanziari (entrate, uscite, numero totale vendite, numero totale disponibili), un'altra con i dati dei singoli vini (nome, anno, numero vendite numero disponibili, entrate), e un'ultima dove trova i dati sugli impiegati (email, numero attività completate e non completate).

Selezionato un mese concluso nel datepicker a inizio pagina clicca il pulsante "crea/cerca" per ottenere il report (se già esiste viene mostrato direttamente, altrimenti viene creato e mostrato).