

WineShop – Federico Risoli Elia Candida

Summary

Summary	1
Introduction	1
Revision History	2
Original Assignment	2-4
Requirements	5-6
Use Case Documentation	7-14
Architectural Model	15
Detailed Software Model	16-31
Functional Tests	32
System Manual:	33
Installation	33
Configuration	34
User Guide	35-48
Changelog	49-57

Introduction

WineShop è un'applicazione client-server che permette a clienti e dipendenti di un negozio di vini, di interfacciarsi con un Sistema che permette ai clienti di comprare e ordinare vini e ai dipendenti di gestire gli ordini, clienti e altro, tutto in un unico Sistema.

Revision History

Version	Date	Author(s)	Description
1	Fine dicembre	Elia Candida-Federico Risoli	Inizio documentazione e progetto
2	Metà gennaio	Elia Candida-Federico Risoli	Implementate metà delle funzionalità
3	Metà marzo	Elia Candida-Federico Risoli	Bugfix, fine Progetto e documentazione ultimata

Original Assigment

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, 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 acquistarla 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 consegnare del materiale e informare il cliente sulla data di consegna. 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 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.

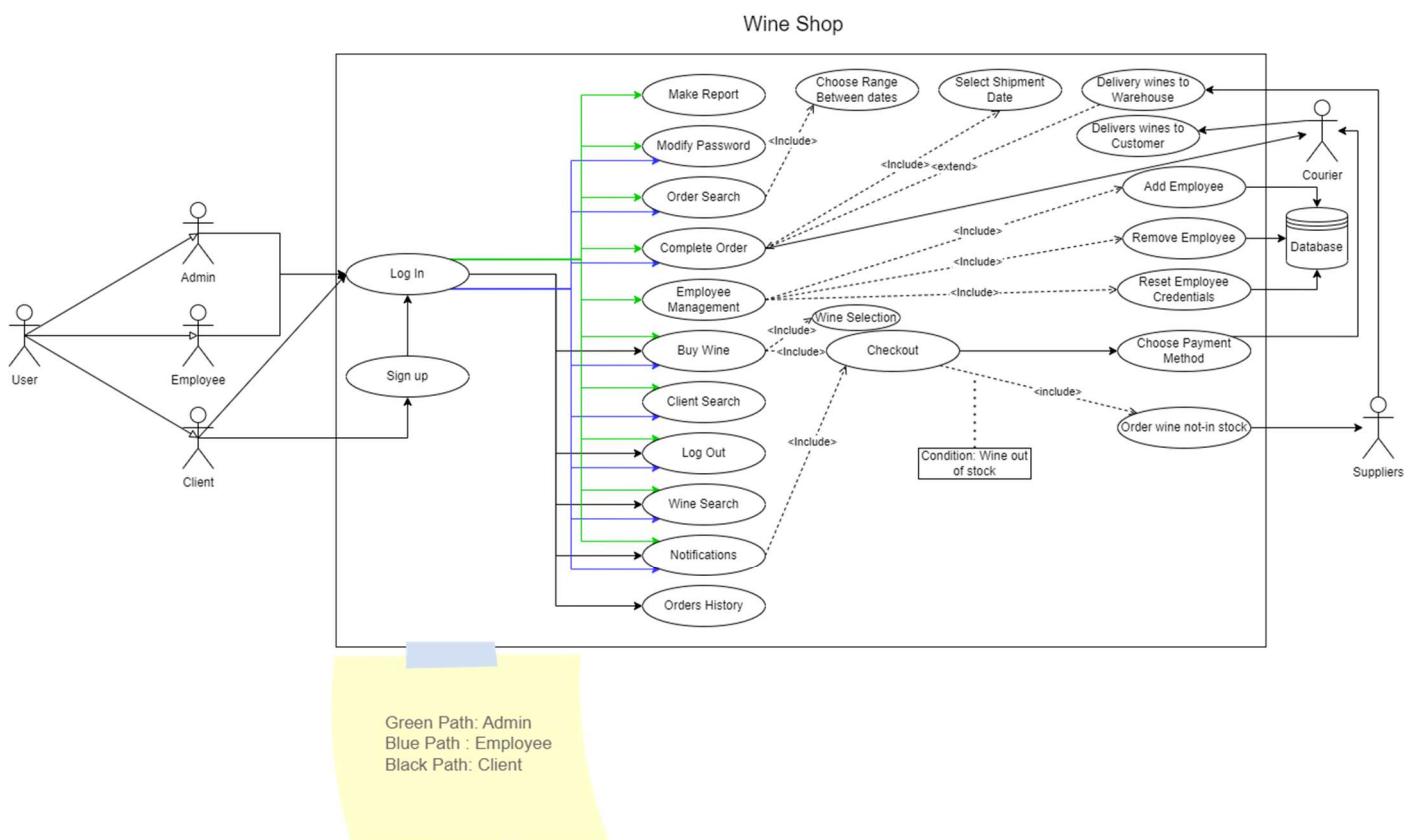
Il numero di impiegati al lavoro deve essere almeno tre e il sistema assegna le attività agli impiegati con una politica "round-robin" ovviamente tenendo conto degli impiegati al momento impegnati. Se l'attività non è terminata entro un tempo prestabilito, il sistema l'assegna ad un altro impiegato. Il sistema mantiene le informazioni sulle attività completate e non completate dai dipendenti.

Requirements

ID	Function name	Status	Priority
F1	Login	implemented	High
F2	User Registration	implemented	High
F3	Employee Registration	implemented	Medium
F4	Modify Password	implemented	Low
F5	Wine Search	implemented	High
F6	Variable Wine Price	implemented	Medium
F7	Popup Promozioni	implemented	Low
F8	Buy Wine	implemented	High
F9	Logout	implemented	Low
F10	My order	implemented	Low
F11	Notify	implemented	Medium
F12	Complete Order	implemented	High
F13	Client Search	implemented	Medium
F14	Order Search	implemented	Medium
F15	Delete Employee	implemented	Medium
F16	Reset Password Employee	implemented	Medium
F17	Generate Monthly Report	implemented	High
F18	Automatic Discount	implemented	Low
F19	Multiple Payment Methods	implemented	Medium

F20	Automatic Orders	implemented	Medium
F21	Order Wines not in Stock	implemented	High
F22	Automatic Orders to Supplier for easier use	Implemented	Medium
F23	Employee Can mark Order as arrived and change the price if needed	Implemented	High
F24	Clients can Buy Wines that were not in stock if they still wish to do this	Implemented	High
F25	Automatic Orders when Wines are Low	Implemented	High
F26	Client/Server Architecture	Implemented	High
F27	Round Robin Work Assignment	Implemented	High

Use Cases Documentation



Use cases diagram

Use Case ID:	1		
Use Case Name:	Login		
Created By:	Elia Candida	Last Updated By:	Federico Risoli
Date Created:	18/12/22	Date Last Updated:	27/02/23

Description:	The User can perform the Login into the system using the Username and the Password
Primary Actor:	User
Secondary actors:	Database
Pre-conditions:	The user was already Sign Up into the system by himself or an admin
Post-conditions:	The Username and Password must be the same of the ones into the Database
Trigger:	The User opens the application
Priority:	High
Frequency of Use:	High
Normal Course of Events:	The User inserts Username and the Password then clicks on Login
Alternative Courses:	The User inserts Username and the Password then clicks on Login but receives an error saying incorrect user/password and must try again

Use Case ID:	2		
Use Case Name:	Signup		
Created By:	Elia Candida	Last Updated By:	Federico Risoli
Date Created:	18/12/22	Date Last Updated:	20/12/22

Description:	The User can Sign up into the System
Primary Actor:	Clients
Secondary actors:	Database
Pre-conditions:	The Client was not already Sign Up into the system
Post-conditions:	The text inserted is not Blank
Trigger:	The User Clicks on “Registrati”
Priority:	High
Frequency of Use:	Low
Normal Course of Events:	The Client writes in all the fields and then sends the form.
Alternative Courses:	The User inserts non-valid text in the fields and so it gets an error and must correct the non-valid text

Use Case ID:	3		
Use Case Name:	Make Report		
Created By:	Elia Candida	Last Updated By:	Elia Candida
Date Created:	19/02/23	Date Last Updated:	03/03/23

Description:	The Admin generates a report detailing various things about the System
Primary Actor:	Admin
Secondary actors:	Database
Pre-conditions:	The Admin has logged-in
Post-conditions:	None
Trigger:	The Admin Clicks the button "Genera Report Mensile"
Priority:	High
Frequency of Use:	Low
Normal Course of Events:	The Admin Clicks the button "Genera Report Mensile"
Special Requirements:	The App must have the ability to write in its own folder

Use Case ID:	4		
Use Case Name:	Modify Password		
Created By:	Federico Risoli	Last Updated By:	Federico Risoli
Date Created:	21/12/22	Date Last Updated:	29/12/22

Description:	The Admin can change the password of everyone, the Employee can change his own password
Primary Actor:	Admin, Employee
Secondary actors:	Database
Pre-conditions:	The user was already Sign Up into the system by The Admin or was the Admin and they did the login successfully
Trigger:	The User clicks on "ModifyPSW"
Priority:	Low
Frequency of Use:	Low
Normal Course of Events:	The User fills and sends the form
Alternative Courses:	The User makes an error and must recompile or the user press "indietro"

Use Case ID:	5		
Use Case Name:	Order Search		
Created By:	Federico Risoli	Last Updated By:	Elia Candida
Date Created:	10/01/23	Date Last Updated:	13/03/23

Description:	The User can Search through the various Orders the one that they want to see
Primary Actor:	Admin, Employee
Secondary actors:	Database
Pre-conditions:	The User is logged in
Post-conditions:	Dates must be in a valid format and the time interval must be valid
Trigger:	The User goes in the "Ordini" tab
Priority:	Medium
Frequency of Use:	Medium
Normal Course of Events:	The User goes in the "Ordini" tab and can search by date interval the order they want
Alternative Courses:	The User goes in the "Ordini" tab and search through all the orders the one they want by ordering how they prefer using the table. The User goes in the "Ordini" tab and inserts an invalid date and must do it again in order to get the right order they are searching
Includes:	Choose Range Between Dates

Use Case ID:	6		
Use Case Name:	Complete Order		
Created By:	Federico Risoli	Last Updated By:	Elia Candida
Date Created:	10/01/23	Date Last Updated:	13/03/23

Description:	The Employee and the Admin can Complete the Orders with Price and date of Delivery
Primary Actor:	Employee, Admin
Secondary actors:	Server, Database, Supplier, Courier
Post-conditions:	The Date and Price selected are valid and not blank
Trigger:	The Employee or the Admin makes the Login
Priority:	High
Frequency of Use:	High
Normal Course of Events:	The User Contacts the Supplier and the Courier and establish a price and a delivery date with them, then compiles the form and press "Completa Ordine", in the end the courier will deliver the wines
Alternative Courses:	The User press "Completa Dopo" to skip this order for now and gets a new one. The User takes too much time to complete the order and receives a new one to complete from the server. The User completes the form incorrectly and must do it again.
Includes:	Select Shipment Date
Extends:	Delivery Wines to warehouse
Special Requirements:	The Wines must be arrived to us otherwise the User will press "Completa Dopo"
Notes and Issues:	The Database and server must have enough RAM and be able to support all the connections otherwise the server will stop working and assigning the new works

Use Case ID:	7		
Use Case Name:	Employee Management		
Created By:	Federico Risoli	Last Updated By:	Federico Risoli
Date Created:	18/12/22	Date Last Updated:	31/12/22

Description:	The Admin can Add, Remove and Reset Employees
Primary Actor:	Admin
Secondary actors:	Database, Employee
Pre-conditions:	Admin logged in
Post-conditions:	The Forms compiled must be valid and not blank
Trigger:	The Admin clicks on "Gestisci Dipendenti"
Priority:	High
Frequency of Use:	Low
Normal Course of Events:	The Admin clicks on "Gestisci Dipendenti", selects "Registra", "Cancella" or "Resetta", fills the form and press the button to complete the action
Alternative Courses:	<p>The Admin changes his mind and press "Indietro" to cancel the process.</p> <p>The Admin makes an error compiling the form and must do it again</p>
Includes:	Add Employee, Remove Employee, Reset Employee Credentials

Use Case ID:	8		
Use Case Name:	Buy Wine		
Created By:	Elia Candida	Last Updated By:	Federico Risoli
Date Created:	26/12/22	Date Last Updated:	27/02/23

Description:	All the Users can choose a wine to buy
Primary Actor:	Client
Secondary actors:	Database, Server, Employee, Admin
Pre-conditions:	User performs the login
Post-conditions:	Selected at least one wine.
Trigger:	The Client makes the login or the Employee (or Admin) goes into the "Vini" tab after the login
Priority:	High
Frequency of Use:	High
Normal Course of Events:	The User chooses some wines to buy and the quantity, clicks "compra", chooses a method of payment, clicks "Procedi al Pagamento", pays and get his wines delivered
Alternative Courses:	<p>The User inserts a quantity of wines that is greater than the one in stock, the User then can decide to order them and will pay when they arrive.</p> <p>The User makes an error in the payment and must do it again to buy the Wines.</p> <p>The User changes his mind and press "Indietro" after selecting the wines</p>
Includes:	Wines Selection, Checkout, Choose Payment Method, Order Wines not-in Stock
Notes and Issues:	The Order in any case is send directly to the Supplier if it is not in stock to have an easier User Experience

Use Case ID:	9		
Use Case Name:	Client Search		
Created By:	Federico Risoli	Last Updated By:	Federico Risoli
Date Created:	03/01/23	Date Last Updated:	09/03/23

Description:	The Employee and the Admin can search for a Client
Primary Actor:	Employee, Admin
Secondary actors:	Database
Pre-conditions:	The User performs the Login
Post-conditions:	The form must not be blank
Trigger:	The User goes into the tab "Clienti"
Priority:	Medium
Frequency of Use:	Low
Normal Course of Events:	The User goes into the tab "Clienti", put a Surname and get the desired Client
Alternative Courses:	The User inserts a wrong or invalid input and must do it again to get what he wants

Use Case ID:	10		
Use Case Name:	Log Out		
Created By:	Elia Candida	Last Updated By:	Elia Candida
Date Created:	30/01/23	Date Last Updated:	09/03/23

Description:	The User can perform the Logout without losing its data
Primary Actor:	All the Users
Secondary actors:	Server
Pre-conditions:	The User performed the Login
Trigger:	The User clicks on "logout"
Priority:	High
Frequency of Use:	High
Normal Course of Events:	The User clicks on "Logout" and its data are saved, the server takes note if the user was doing something important and make a secure logout
Alternative Courses:	The User press ALT+F4 to logout, in any case the Server will handle that

Use Case ID:	11		
Use Case Name:	Wine Search		
Created By:	Elia Candida	Last Updated By:	Federico Risoli
Date Created:	23/12/22	Date Last Updated:	27/01/23

Description:	The Users can search for a specific wine by date and/or name
Primary Actor:	All the Users
Secondary actors:	Database
Pre-conditions:	The User must have logged in
Post-conditions:	The Wine exists in the Database
Trigger:	The User clicks on "seach"
Priority:	Medium
Frequency of Use:	High
Normal Course of Events:	The User inserts a name of a wines and/or selects a date and then clicks "Search"

Use Case ID:	12		
Use Case Name:	Check Notifications		
Created By:	Federico Risoli	Last Updated By:	Elia Candida
Date Created:	20/02/23	Date Last Updated:	03/03/23

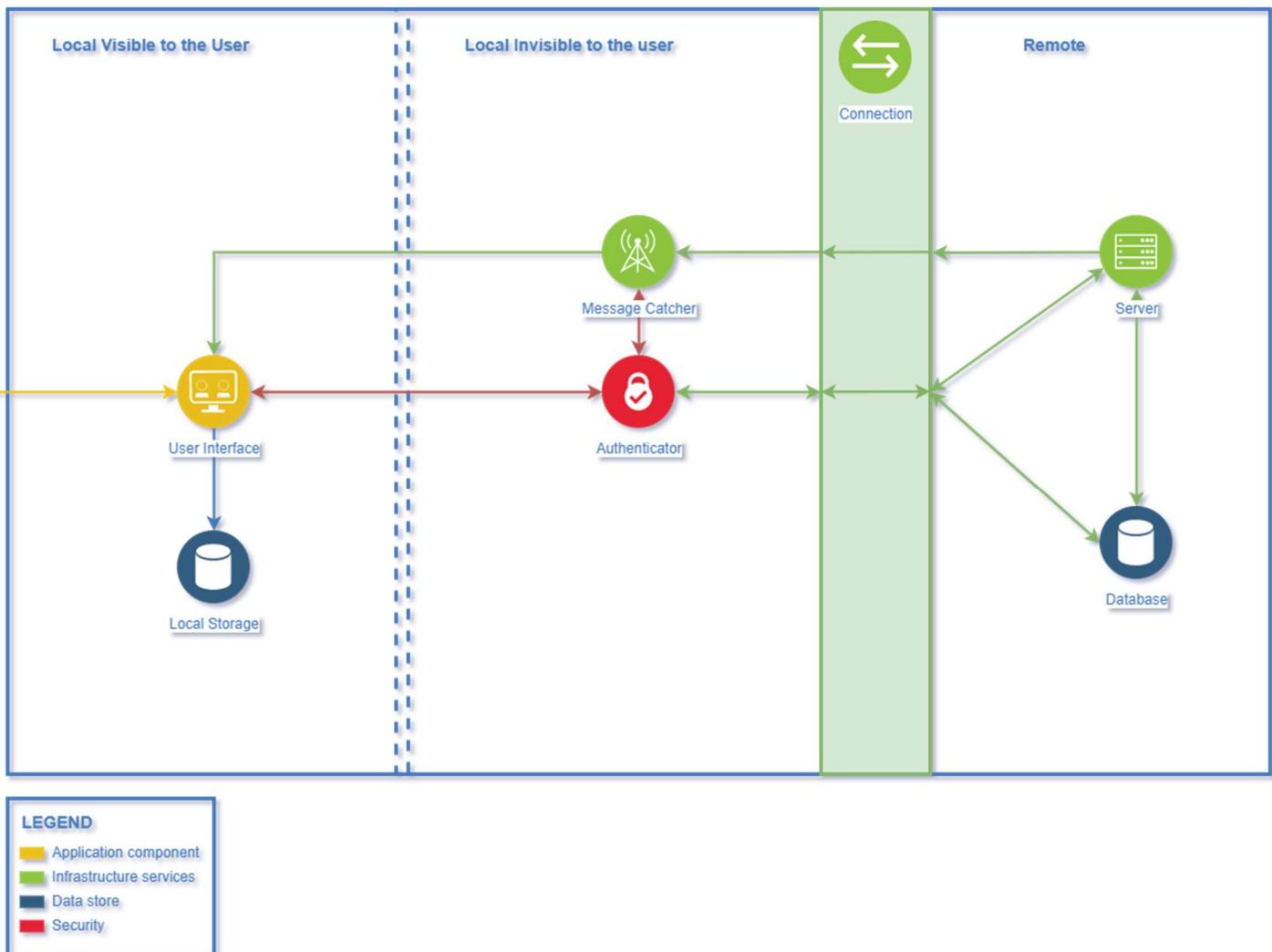
Description:	The User see if there are some notification for him and can take action accordingly
Primary Actor:	All the Users
Secondary actors:	Database
Pre-conditions:	The User logged in
Trigger:	The User Clicks on "Notifiche"
Priority:	Medium
Frequency of Use:	Medium
Normal Course of Events:	The User Clicks on "Notifiche" and if there are things to do can choose a payment method to get his wine or reject it or do nothing
Includes:	Checkout, Choose payment method
Special Requirements:	This will appear only if the Employee has marked the order as ready

Use Case ID:	13		
Use Case Name:	Orders History		
Created By:	Federico Risoli	Last Updated By:	Federico Risoli
Date Created:	23/01/23	Date Last Updated:	27/02/23

Description:	The User can check his Order History
Primary Actor:	All the Users
Secondary actors:	Database
Pre-conditions:	The User is logged in
Trigger:	The Client clicks on “I miei ordini” or the Employee/Admin search through all the Wines Histories
Priority:	Medium
Frequency of Use:	Low
Normal Course of Events:	The Client clicks on “I miei ordini” and sees his orders
Alternative Courses:	Employee/Admin clicks on the “Ordini” tab and search through all the Wines Histories

Architectural Model

Architecture diagram

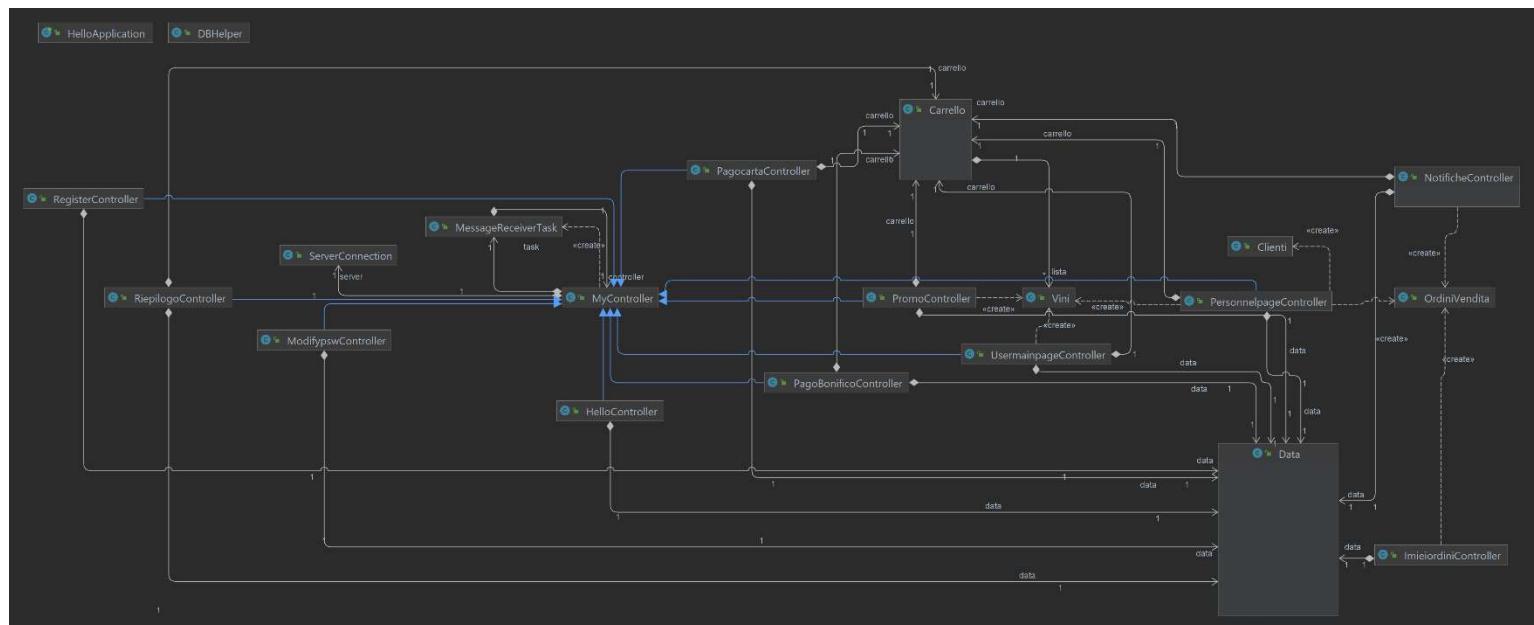


Detailed Software Model

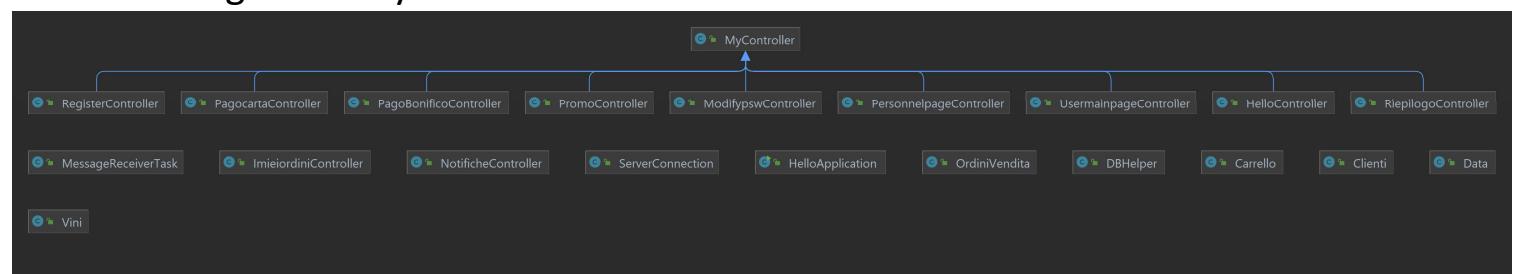
Class diagram(s)

The Full Class Diagram of the main application would be pretty hard to read and so it was decided to split it into multiple views each one highlighting something different.

- Class Diagram Without Methods, Constructors and variables (those can be seen later):



- Hilight of only the Inheritances:



- Main app Diagram with only Inheritances, Methods, Variables, Constructors :



- Zoom on the single classes:

These was extracted directly from the code and there might be a couple of words in Italian.

MyController	
m	MyController()
f	scene Scene
f	task MessageReceiverTask
f	stage Stage
f	root Parent
f	server ServerConnection
m	handleMessage(String) void
m	changeScene(String, ActionEvent) void
m	connect() void
m	createTask(Socket, MyController) void
m	killChildThread() void
m	sendMessage(String) void

MessageReceiverTask	
m	MessageReceiverTask(Socket, MyController)
f	instance MessageReceiverTask
f	controller MyController
f	shouldStop boolean
f	socket Socket
m	call() Void
m	stop() void



ImieiordiniController

m ImieiordiniController()

f o data Data

f 🔒 t_nome TableColumn<OrdiniVendita, String>

f 🔒 t_ordine TableColumn<OrdiniVendita, String>

f 🔒 t_dataconsegna TableColumn<OrdiniVendita, String>

f 🔒 back Button

f 🔒 tabella TableView<OrdiniVendita>

f 🔒 t_cognome TableColumn<OrdiniVendita, String>

f 🔒 t_indirizzo TableColumn<OrdiniVendita, String>

m o OnButtonClickBack(ActionEvent) void

m 🔒 initialize() void

Data		
m	lock	Data0
f	role	String
f	success	boolean
f	username	String
f	nome	String
f	id	int
f	cognome	String
f	promo	boolean
f	instance	Data
m	Setid(int)	void
m	Getrole()	String
m	Setusername (String)	void
m	Setsurname (String)	void
m	Getusername ()	String
m	SetSuccess (boolean)	void
m	reset()	void
m	GetPromo ()	boolean
m	GetId()	int
m	SetPromo (boolean)	void
m	GetName()	String
m	SetRole(String)	void
m	Getsurname ()	String
m	getInstance()	Data
m	GetSuccess ()	boolean
m	SetName (String)	void

UsermainpageController		
m	lock	UsermainpageController0
f	search	Button
f	t_nome	TableColumn <Vini, String>
f	t_vitigno	TableColumn <Vini, String>
f	log_out	Button
f	GestioneDipButton	Button
f	t_anno	TableColumn <Vini, String>
f	tabella	TableView <Vini>
f	t_qta	TableColumn <Vini, Spinner<Integer>>
f	nome_vino	TextField
f	error_text	Label
f	t_provenienza	TableColumn <Vini, String>
f	modifypswbutton	Button
f	t_note	TableColumn <Vini, String>
f	t_prezzo	TableColumn <Vini, Double>
f	t_selected	TableColumn <Vini, CheckBox>
f	compra	Button
f	carrello	Carrello
f	t_produttore	TableColumn <Vini, String>
f	RicercaclientiButton	Button
f	ImieiordiniButton	Button
f	annata	ComboBox <String>
f	data	Data
m	initialize()	void
m	OnButtonClearWineClick (ActionEvent)	void
m	OnButtonClickNotifiche (ActionEvent)	void
m	OnButtonClickBuy (ActionEvent)	void
m	OnButtonClickSearch (ActionEvent)	void
m	OnButtonClickLogOut (ActionEvent)	void
m	OnButtonClickImieiOrdini (ActionEvent)	void

RegisterController		
m	RegisterController 0	
f	indirizzo	TextField
f	cognome	TextField
f	data	Data
f	ActionBar	ButtonBar
f	cancradio	RadioButton
f	pass	PasswordField
f	usr	TextField
f	Registrati	Button
f	telefono	TextField
f	userselector	ComboBox <String>
f	regradio	RadioButton
f	cfiscale	TextField
f	mail	TextField
f	nome	TextField
f	resetradio	RadioButton
f	Indietro	Button
f	action	ToggleGroup
m	initialize()	void
m	Cancel (MouseEvent)	void
m	OnClickClose (ActionEvent)	void
m	OnClickRegister (ActionEvent)	void
m	Reset(MouseEvent)	void
m	registersel(MouseEvent)	void

DBHelper		
m	DBHelper()	
m	execute(String)	void
m	idgetter(TextField)	int
m	query(String)	ResultSet
m	update(String)	void

ModifypswController		
m	ModifypswController 0	
f	oldpswtext	Text
f	confirmsw	PasswordField
f	mailto	Text
f	newpsw	PasswordField
f	mail	TextField
f	data	Data
f	oldpsw	PasswordField
f	Indietro	Button
m	initialize()	void
m	OnButtonModifyPSWClick (ActionEvent)	void
m	OnButtonClickBack (ActionEvent)	void

RiepilogoController		
m	RiepilogoController()	
f	bonifice	RadioButton
f	ordinanondisponibili	Button
f	stage	Stage
f	label_prezzo	Label
f	data	Data
f	textflow	TextFlow
f	metodo	ToggleGroup
f	adrrfield	TextField
f	pagtext	Label
f	root	Parent
f	scene	Scene
f	recap	ScrollPane
f	carrello	Carrello
f	annulla	Button
f	month	int
f	div1	Line
f	pagbuttonsbar	ButtonBar
f	year	int
f	procedi	Button
f	titletext	Label
m	OnClickAnnulla(ActionEvent)	void
m	round(double, int)	double
m	OnClickProcedi(ActionEvent)	void
m	OnClickOrdinaNonDisponibili(ActionEvent)	void
m	logout()	void
m	initialize()	void

		HelloApplication
		HelloApplication()
		start(Stage) void
		main(String[]) void

		HelloController
		HelloController()
		root Parent
		temps String
		tempn String
		scene Scene
		errorText Label
		stage Stage
		password PasswordField
		username TextField
		data Data
		onLoginButtonClick(ActionEvent) void
		OnClickOpenRegister(ActionEvent) void

		ServerConnection
		ServerConnection()
		out PrintWriter
		socket Socket
		instance ServerConnection
		getSocket() Socket
		getInstance() ServerConnection
		sendMessage(String) void



OrdiniVendita

m	◦	OrdiniVendita (String, String, String, String, String, String, String, double, String)	
f	🔒	Dataconsegna	SimpleStringProperty
f	🔒	Ordine	SimpleStringProperty
f	🔒	Id	SimpleStringProperty
f	🔒	check	CheckBox
f	🔒	Idcliente	SimpleStringProperty
f	🔒	DataOrdine	SimpleStringProperty
f	🔒	Nome	SimpleStringProperty
f	🔒	Cognome	SimpleStringProperty
f	🔒	Prezzo	double
f	🔒	Indirizzo	SimpleStringProperty
m	↳	getOrdine()	String
m	↳	getId()	String
m	↳	getNome()	String
m	↳	getCognome()	String
m	↳	getIdcliente()	String
m	↳	getPrezzo()	double
m	↳	getDataOrdine()	String
m	↳	getIndirizzo()	String
m	↳	getDataconsegna()	String
m	↳	getCheck()	CheckBox



Carrello

m	o	Carrello()	
f	l	totale	double
f	l	instance	Carrello
f	l	ordinenondisp	ObservableList<String>
f	l	nondisp	ObservableList<Vini>
f	l	ordine	ObservableList<String>
f	l	lista	ObservableList<Vini>
m	g	setNondisp(ObservableList<Vini>)	void
m	g	getinstance()	Carrello
m	g	clearOrdineNonDisp()	void
m	g	clearOrdine()	void
m	g	setTotale(double)	void
m	g	clear()	void
m	g	setCarrello(ObservableList<Vini>)	void
m	g	getTotale()	double
m	g	getCarrello()	ObservableList<Vini>
m	g	getNondisp()	ObservableList<Vini>
m	g	getOrdineNonDisp()	ObservableList<String>
m	g	addOrdineNonDisp(String)	void
m	g	setOrdineNonDisp(ObservableList<String>)	void
m	g	addOrdine(String)	void
m	g	getOrdine()	ObservableList<String>

PagoBonificoController		
m	PagoBonificoController 0	
f	iban	TextField
f	errore	Label
f	paga	Button
f	indietro	Button
f	o data	Data
f	intestatario	TextField
f	o text	String
f	o carrello	Carrello
f	o alert	Alert
m	o OnClickIndietro (ActionEvent)	void
m	o OnClickPaga (ActionEvent)	void

PagocartaController		
m	PagocartaController 0	
f	o data	Data
f	o carrello	Carrello
f	o year	int
f	o indietro	Button
f	o text	String
f	o paga	Button
f	o intestatario	TextField
f	o ncarta	TextField
f	o month	int
f	o errore	Label
f	o mese	ChoiceBox <String>
f	o anno	ChoiceBox <String>
f	o alert	Alert
f	o cvv	TextField
m	o OnClickIndietro (ActionEvent)	void
m	o initialize0	void
m	o OnClickPaga (ActionEvent)	void

PromoController		
m	PromoController 0	
f	o carrello	Carrello
f	t_note	TableColumn <Vini, String >
f	t_nome	TableColumn <Vini, String >
f	o data	Data
f	t_produttore	TableColumn <Vini, String >
f	t_provenienza	TableColumn <Vini, String >
f	t_anno	TableColumn <Vini, String >
f	t_prezzo	TableColumn <Vini, Double >
f	t_vitigno	TableColumn <Vini, String >
f	tabella	TableView<Vini >
m	o initialize0	void



NotificheController

m NotificheController()

f ord TableColumn<OrdiniVendita, String>

f Accetta Button

f scene Scene

f rbon RadioButton

f carrello Carrello

f notifiche TableView<OrdiniVendita>

f backbutton Button

f select TableColumn<OrdiniVendita, CheckBox>

f rcar RadioButton

f prezzo TableColumn<OrdiniVendita, String>

f consprev TableColumn<OrdiniVendita, String>

f data Data

F p double

m OnButtonClickAccetta(ActionEvent) void

m OnButtonClickIndietro(ActionEvent) void

m initialize() void

Vini

m	o	Vini(Integer, String, String, String, String, String, String, String, String, String, String)	
t	o	p1	SimpleStringProperty
t	o	p	SimpleStringProperty
t	o	quantita	SimpleStringProperty
t	o	p2	SimpleStringProperty
t	o	check	CheckBox
t	o	Nome	SimpleStringProperty
t	o	vendite	SimpleStringProperty
t	o	spin	Spinner<Integer>
t	o	not	SimpleStringProperty
t	o	disponibile	boolean
t	o	id	SimpleIntegerProperty
t	o	v	SimpleStringProperty
t	o	a	SimpleStringProperty
t	o	prezzo	SimpleDoubleProperty
m	g	getP1()	String
m	g	getDisponibilita()	Boolean
m	g	getSpin()	Spinner
m	g	getQuantita()	String
m	g	getV()	String
m	g	venditeProperty()	SimpleStringProperty
m	g	getSpinnerValueString()	String
m	g	idProperty()	SimpleIntegerProperty
m	g	getNome()	String
m	g	getId()	int
m	g	getP()	String
m	o	round(double, int)	double
m	g	getVendite()	String
m	g	getA()	String
m	g	getPrezzo()	double
m	g	getCheck()	CheckBox
m	g	getP2()	String
m	g	getNot()	String
m	g	getSpinnerValue()	int

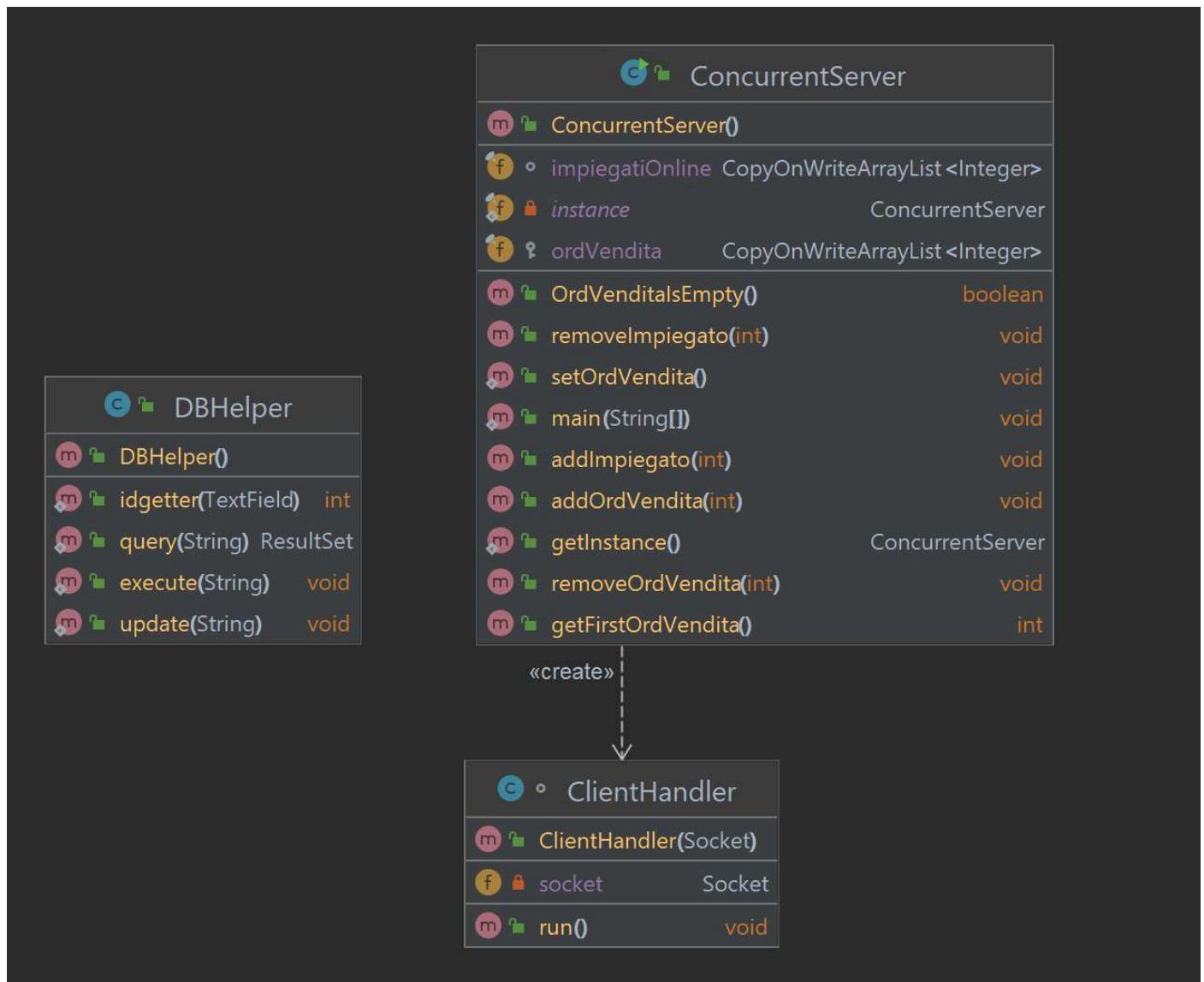
Clienti	
m	◦ Clienti(String, String, String, String, String, String, String)
f	🔒 Nome SimpleStringProperty
f	🔒 Username SimpleStringProperty
f	🔒 Mail SimpleStringProperty
f	🔒 Cognome SimpleStringProperty
f	🔒 Indirizzo SimpleStringProperty
f	🔒 Telefono SimpleStringProperty
f	🔒 CodiceFiscale SimpleStringProperty
m	getUsername() String
m	getCodiceFiscale() String
m	getNome() String
m	getTelefono() String
m	getIndirizzo() String
m	getCognome() String
m	getMail() String

PersonnelpageController was the biggest controller of all and it was split in 2 screenshots for better readability.

PersonnelpageController	
m PersonnelpageController()	
t lavoro_text_prenzzo	TextField
t data	Data
t search	Button
t modifypswbutton	Button
t esportjifen	Label
t ClearWine	Button
t nome_vino	TextField
t archivclient	Button
t Ordinivendita : TableColumn<Ordinivendita, String>	
t t_j : TableColumn<Ordinivendita, String>	
t text_lavoro_jifen	Label
t ClearOrder	Button
t selected : TableColumn<Vini, CheckBox>	
t nome : TableColumn<Vini, String>	
t Ordinivendita : TableColumn<Ordinivendita, String>	
t error_data	Label
t clientivendita : TableColumn<Clienti, String>	
t t_c : TableColumn<Ordinivendita, String>	
t clientmail	TableColumn<Clienti, String>
t compras	Button
t initialize	DatePicker
t tabella	TableView<Vini>
t CiderTableView	TableView<Ordinivendita>
t findata	DatePicker
t Ordinivendita : TableColumn<Ordinivendita, String>	
t imposta	TableColumn<Clienti, String>
t searchorder	Button
t ClearClient	Button
t Utadatossi	TableColumn<Ordinivendita, String>
t digitbutton	Button
t carrello	Carrello
t datapick	DatePicker
t prezzo	TableColumn<Vini, Double>
t surnamefield	TextField
t nome : TableColumn<Clienti, String>	
t log_out	Button
t GestioneDipButton	Button
t carico	TableColumn<Vini, String>
t Ordine : TableColumn<Ordinivendita, String>	
t Usain	TableColumn<Clienti, String>
t lavoro	TableView<Ordinivendita>
t t_o : TableColumn<Ordinivendita, String>	
t t_provenienza	TableColumn<Vini, String>
t t_qta	TableColumn<Vini, Spinner<Integer>>
t codicefiscale	TableColumn<Clienti, String>
t t_migno	TableColumn<Vini, String>
t annata	ComboBox<String>
t t_n : TableColumn<Ordinivendita, String>	
t ClientTableView	TableView<Clienti>
t tefone	TableColumn<Clienti, String>
t ReportMenùalebutton	Button
t note	TableColumn<Vini, String>
t procttore	TableColumn<Vini, String>

PersonnelpageController		
m PersonnelpageController()		
m OnButtonClickLogOut(ActionEvent)	void	
m OnButtonClickSearch(ActionEvent)	void	
m OnNotificheClick(ActionEvent)	void	
m OnButtonSkipClick(ActionEvent)	void	
m OnButtonClickSearchOrder(ActionEvent)	void	
m OnClearClientButtonClick(ActionEvent)	void	
m OnClearOrderButtonClick(ActionEvent)	void	
m roundToTwoDecimalPlaces(double)	double	
m OnButtonClickOrderComplete(ActionEvent)	void	
m initialize()	void	
m OnGestisciDipendentiClick(ActionEvent)	void	
m OnButtonClickBuy(ActionEvent)	void	
m handleMessage(String)	void	
m logout()	void	
m OnModifyPSWButtonClick()	void	
m OnButtonClickClearWineClick(ActionEvent)	void	
m OnClickGenerateReport(ActionEvent)	void	
m isValidDate(String)	boolean	
m OnButtonClickSearchClient(ActionEvent)	void	

- Class Diagram Concurrent Server:



Functional Tests

ID	Test name	Steps	Expected result	Test result	Comments
1.	Unit Test Carrello.java	1. Setup pre test 2. Testo i metodi set/get/clear/non-disponibile della classe	pass	pass	
2.	Unit Test Data.java	1. Testo tutte le funzioni per vedere se funzionano correttamente	pass	pass	
3.	Unit Test PersonnelPageController.java	1. Testo nei casi limite le funzioni roundtodecimalplaces e isValidDate	Pass	pass	
4	Unit Test ConcurrentServer.java	1. Setup con clear delle variabili che rimangono da un test all'altro. 2. Testo le funzioni che Aggiungono/rimuovono/ottengono gli impiegati e gli ordini di vendita registrati nel server	Pass	pass	
5	Test UI	1. Entrare in una pagina 2. Testare il corretto funzionamento di tutti i suoi elementi 3. Ripetere per ogni Controller	Pass	Pass	
6	Test Generale Sviluppatori	1. Testare tutto il sistema in maniera insolita per evidenziarne i bug	Pass	Pass	Eseguito da Elia e Federico non sono stati trovati bug malgrado ci fossero; quindi il test è passato malgrado bug nascosti (i test non assicurano l'assenza di bug)
7	Alpha Test 1	1. Testare tutto il sistema in maniera insolita per evidenziarne i bug	Pass	Fail	Eseguito da un amico degli sviluppatori per avere visioni differenti sul programma. Ha evidenziato un bug quindi il test conta come fallito
8	Alpha Test 2	1. Testare tutto il sistema in maniera insolita per evidenziarne i bug	Pass	Pass	Eseguito dopo l'Alpha Test 1 e dopo il fix del bug trovato.

System Manual

Installation

- Installare XAMPP sul proprio sistema da
<https://www.apachefriends.org/it/index.html>
- Installare Java 19.0.2 sul proprio sistema da
<https://www.oracle.com/java/technologies/javase/jdk19-archive-downloads.html>
- Andare sulla pagina GitHub del progetto e scaricare l'ultima Release
<https://github.com/FedericoRisoli/Assegnamento>
- Estrarre il file .zip dove si preferisce sul proprio sistema, questa sarà la cartella dell'applicazione
- Nella cartella d'installazione aprire la cartella javafx-sdk-19.0.2.1 e copiare il percorso della cartella "lib"
- Modificare il file RunClient.bat sostituendo [PERCORSO CARTELLA LIB] con quello appena copiato. Il risultato dovrebbe essere simile a questo:

```
#!/bin/bash
java --module-path
C:\Users\Elia\IdeaProjects\AssegnamentoWineShop\out\artifacts\Client\javafx
-sdk-19.0.2.1\lib --add-modules javafx.controls,javafx.fxml -jar Client.jar
```

- Aprire XAMPP, attivare Apache e MySQL, premere sul pulsante Admin di MySQL
- Creare un nuovo Database nella pagina appena aperta nel Browser
- Prendere il file wineshop.sql dalla cartella dell'applicazione e trascinarlo nel browser, questo farà l'upload del Database
- (opzionale) cancellare il file wineshop.sql dalla cartella dell'applicazione

Configuration

Il sistema è già pronto per essere usato, tuttavia ci sono dei passaggi opzionali che sono consigliati:

- Aprire XAMPP, attivare Apache e MySQL, premere sul pulsante Admin di MySQL per sostituire il database di esempio con quello che effettivamente verrà utilizzato
- Se non si vuole eseguire il passaggio qui sopra si consiglia almeno di: aprire XAMPP, attivare Apache e MySQL. Aprire RunServer.bat e successivamente RunClient.bat. Eseguire l'accesso con le credenziali dell'amministratore (usr: admin, psw: admin) e modificare la password dell'amministratore di sistema.

Per far partire il sistema bisogna comunque aprire XAMPP, attivare Apache e MySQL e aprire RunServer.bat

User Guide

Indice

Clienti

1. Login
2. Comprare/Ordinare/Ricerca Vini
3. Visualizzare I Propri Ordini
4. Notifiche-Ordini Pronti

Impiegati

1. Login
2. Lavoro
3. Comprare/Ordinare/Ricerca Vini
4. Notifiche-Ordini Pronti
5. Ricerca Ordini
6. Ricerca Cliente
7. Modifica Password

Amministratore

Per funzionalita' base vedere manuale dipendente

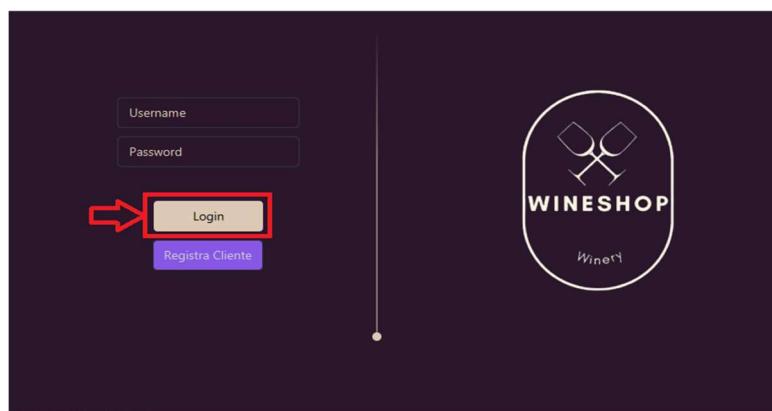
Per funzionalita' specifiche admin

1. Generare Report Mensile
2. Gestione Dipendenti
 - a. Registrare un dipendente
 - b. Cancellare un dipendente
 - c. Resettare Credenziali
3. Modifica Password

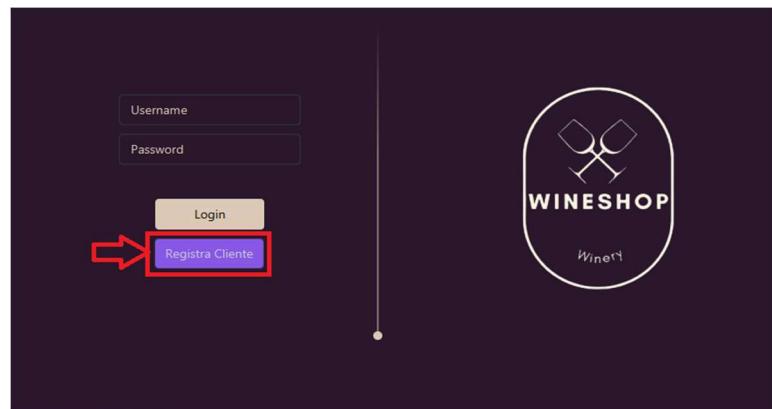
Clienti

1-Login

- Utenti Già Registrati
 1. Aprire RunClient.bat
 2. Inserire username e password
 3. Cliccare su Login

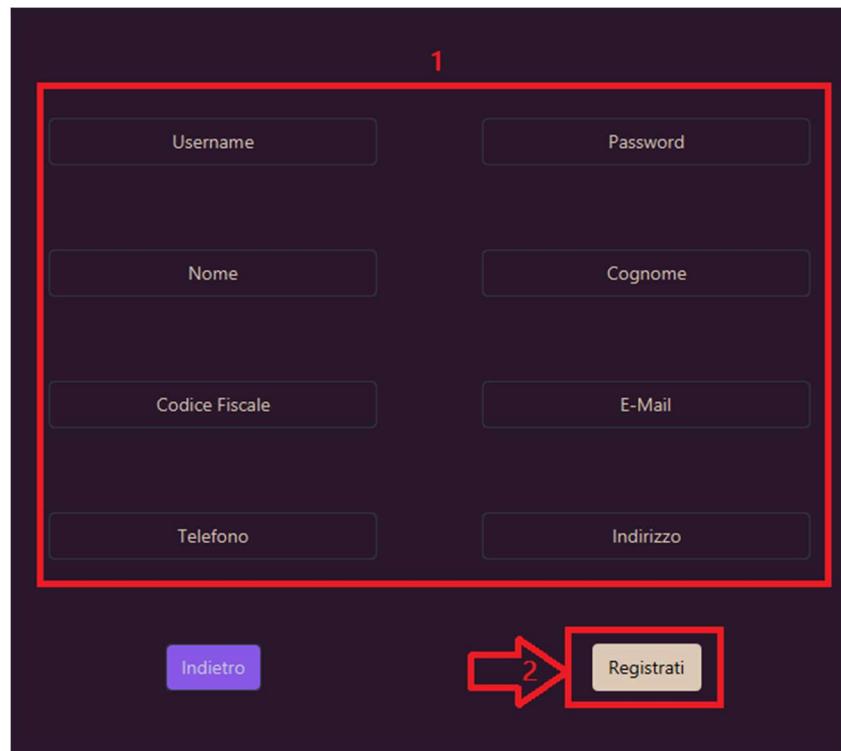


- Utenti non Registrati
 1. Aprire RunClient.bat
 2. Cliccare su "Registrati Cliente"



3. Compilare tutti i campi

4. Cliccare su “Registrati”



2-Comprare/Ordinare/Ricercare Vini

1. Selezionare i vini che si desiderano comprare con l'apposita checkbox sul lato sinistro della tabella
 - a. Si possono ricercare I vini compilando gli apposite campi "Nome Prodotto" o selezionare l'annata
 - b. Per Resettare la ricerca cliccare su "Clear"
2. Inserire la quantità per ogni prodotto selezionato
3. Cliccare il tasto "Compra"

The image shows the Wineshop application interface. On the left, there is a sidebar with icons for 'I Miei Ordini', 'Notifiche', and 'Log Out'. The main area has a search bar at the top with fields for 'Nome Vino' and 'Annata', and buttons for 'Clear' and 'Search'. Below the search bar is a table with columns: Selezionato, Q.tà, Nome, Produttore, Anno, Vitigno, Note, Prezzo €, and Provenienza. The 'Selezionato' column contains checkboxes, and the 'Q.tà' column contains dropdown menus. A red arrow labeled '1' points to the 'Selezionato' column, and a green arrow labeled '2' points to the 'Q.tà' column. A yellow arrow labeled '3' points to the 'Prezzo €' column. At the bottom right is a large orange arrow pointing upwards, with a box labeled 'Compra' above it.

Selezionato	Q.tà	Nome	Produttore	Anno	Vitigno	Note	Prezzo €	Provenienza
<input type="checkbox"/>	1	test	pr1	2000	vit1	text	37.72	0
<input type="checkbox"/>	1	Francia Co...	1701 Franc...	1990	Albana	Frizzante, f...	8.29	Francia
<input type="checkbox"/>	1	Barbaresco...	Abrigo Orl...	1995	Barbera	Frizzante, f...	38.59	Italia
<input type="checkbox"/>	1	Greco di T...	Conti Di B...	2000	Biancolella	Rosso, fer...	22.54	Italia
<input type="checkbox"/>	1	Pink Desire	Desirè Win...	1999	Pink Magic	Rosè, frut...	23.51	Inghilterra
<input type="checkbox"/>	1	francia corta	unipr	2022	rnewgserg	sergsdrgzxcv	50.1	parma
<input type="checkbox"/>	1	testa	pr1	2000	vit1	text	30.3	
<input type="checkbox"/>	1	Barbaresco...	Abrigo Orl...	2021	Barbera	Rosso, fer...	30.95	Italia
<input type="checkbox"/>	1	Francia Co...	1701 Franc...	1987	Albana	Frizzante, f...	50.9	Francia
<input type="checkbox"/>	1	Ghemme	Alamos	1999	Barolo bia...	Frizzante, f...	10.05	Italia
<input type="checkbox"/>	1	Lison	Dosnon	2019	Bonarda	Rosso, fer...	10.0	Italia

4. Selezionare il metodo di pagamento e cliccare su “procedi al pagamento”

Ordine di Vendita:

VINI NEL CARRELLO:

test
Confezione da 1 bottiglie, prezzo 37.72 €
0 casse da 6 bottiglie, prezzo 0.0 €
0 casse da 12 bottiglie, prezzo 0.0 €

• **Metodo di Pagamento:** Bonifico Carta di Credito 4

TOTALE: 37.72 €

Indietro Procedi al Pagamento

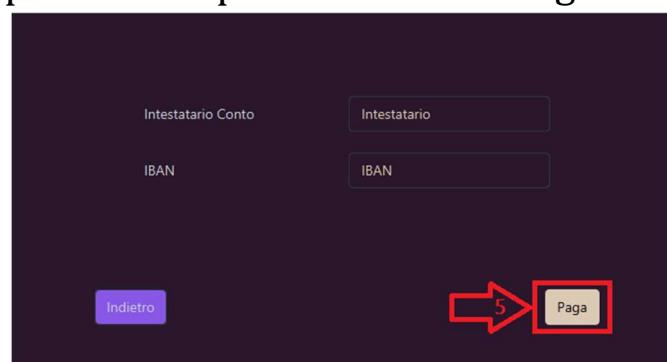


5. Compilare i campi e cliccare su “Paga”

Intestatario Conto

IBAN

Indietro 5 Paga



- **NB se i vini selezionati non sono disponibili**
 1. Inserire l'indirizzo di consegna nell' apposito campo
 2. Cliccare su “Ordina Vini Non Disponibili”

Proposta di Acquisto:

VINI NEL CARRELLO:

VINI NON DISPONIBILI:

Nome: Francia Corta Q.tà: x 129

• **Inserisci indirizzo di Consegna** 1

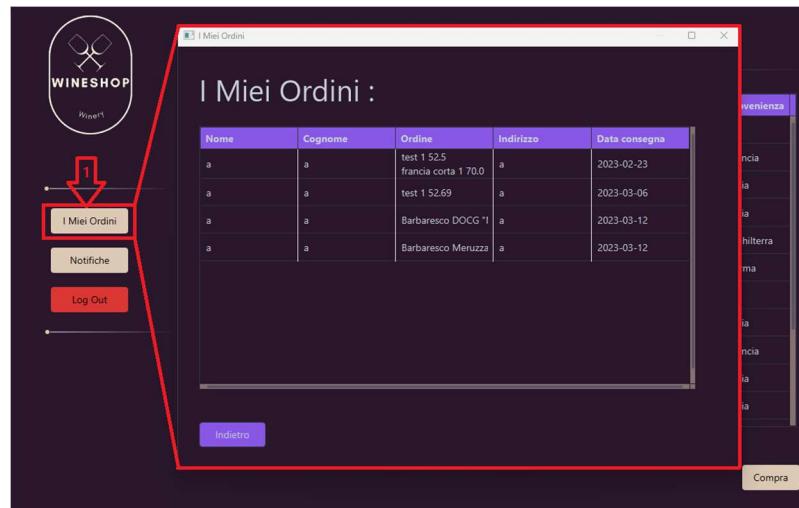
TOTALE: 0.0 €

Indietro 2 Ordina Vini Non Disponibili



3-Visualizzare i propri ordini

1. Nella pagina principale cliccare su “I miei Ordini”



4-Notifiche-Ordini Pronti

1. Cliccare su Notifiche
2. Selezionare metodo di pagamento se si vuole acquistare e cliccare su “Paga”



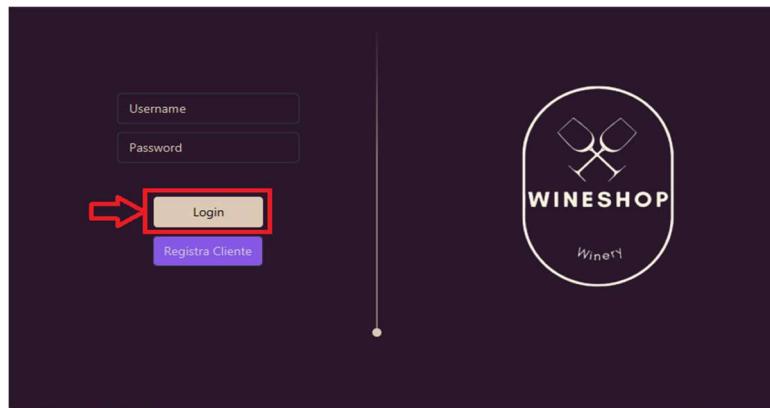
3. Compilare i campi richiesti dal metodo di pagamento e cliccare su “Paga”

The screenshot shows a form for entering payment details. It has two input fields: 'Intestatario Conto' and 'IBAN', each with a corresponding 'Intestatario' and 'IBAN' placeholder. At the bottom are 'Indietro' and 'Paga' buttons. A red arrow labeled '3' points to the 'Paga' button.

Impiegati

1-Login

1. Aprire RunClient.bat
2. Inserire username e password
3. Cliccare su Login



2-Lavoro

Per Completare il Lavoro (merce arrivata)

1. Selezionare data di consegna
2. Inserire Prezzo a cui vendere il vino
3. Cliccare su "Completa Ordine"
 - a. Se la merce non e' arrivata cliccare su "Completa Dopo"

Nome	Cognome	Ordine	Indirizzo
Tito	Fiorentino	[Francia Corta, 17]	qwerty

Data di Consegna:

24/03/2023

Prezzo:

€ 2

Completa Ordine

Completa Dopo

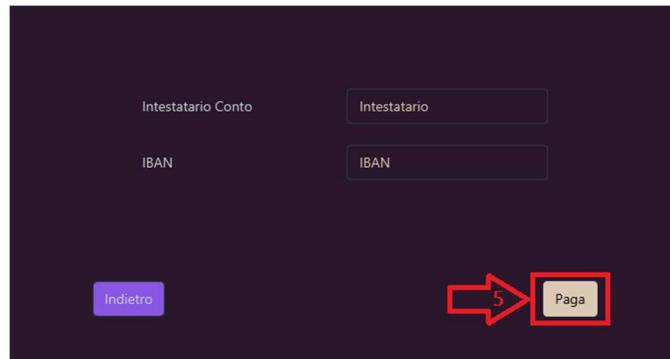
a

3-Comprare/Ordinare/Ricercare Vini

- 1.Selezionare i vini che si desiderano comprare con l'apposita checkbox sul lato sinistro della tabella
 - b. Si possono ricercare I vini compilando gli apposite campi "Nome Prodotto" o selezionare l'annata
 - c. Cliccare su "Search"
 - d. Per Resettare la ricerca cliccare su "Clear"
- 2.Inserire la quantità per ogni prodotto selezionato
- 3.Cliccare il tasto "Compra"

- 4.Selezionare il metodo di pagamento e cliccare su “procedi al pagamento”

5. Compilare i campi e cliccare su “Paga”



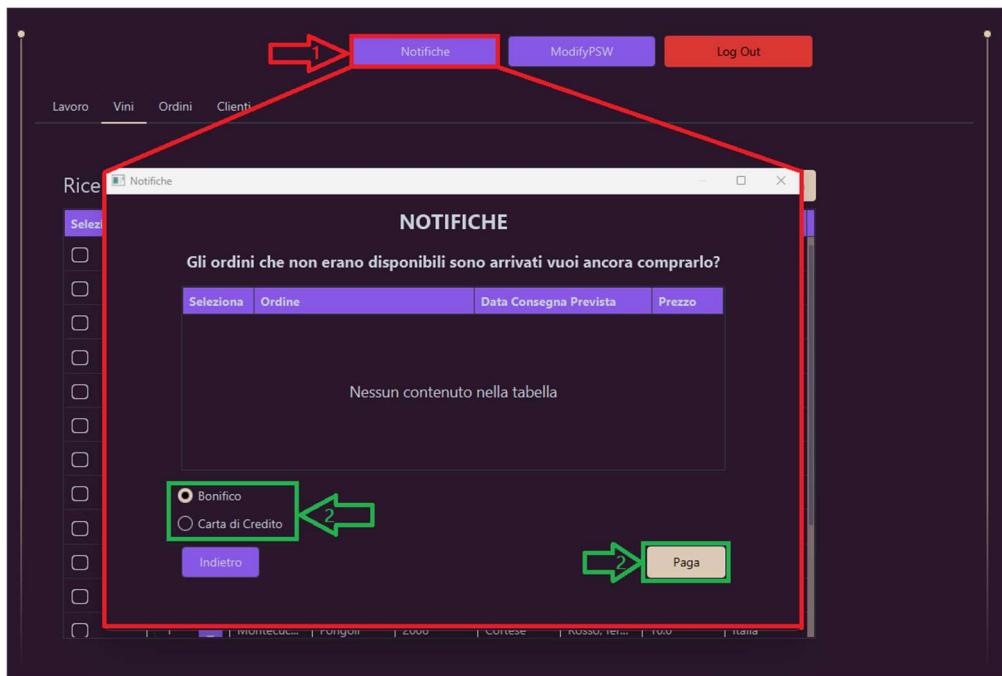
- **NB se i vini selezionati non sono disponibili**
 1. Inserire l'indirizzo di consegna nell' apposito campo
 2. Cliccare su “Ordina Vini Non Disponibili”



4-Notifiche-Ordini Pronti

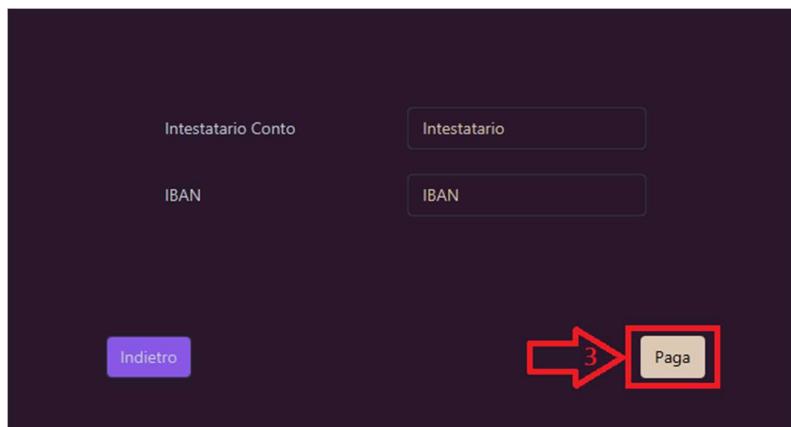
1.Cliccare su Notifiche

2.Selezionare metodo di pagamento se si vuole acquistare e



cliccare su “Paga”

3.Compile i campi richiesti dal metodo di pagamento e cliccare su “Paga”



5-Ricerca Ordini

- 1.Inserire Data iniziale e data finale negli appositi campi per definire l'intervallo di ricerca
- 2.Cliccare su “Search”
 - a. Cliccare su “Clear” per resettare la ricerca

The screenshot shows a user interface for searching orders. At the top, there are tabs for 'Lavoro', 'Vini', 'Ordini', and 'Clienti'. Below the tabs is a search bar labeled 'Ricerca ordini' with two input fields for date ('Da' and 'A') and a search button ('Search'). A green arrow points from the number '2' to the 'Search' button. To the right of the search bar are two buttons: 'Clear' (orange) and 'a' (grey). Below the search bar is a table with columns: 'Data Ordine', 'Nome', 'Cognome', 'Ordine', and 'Indirizzo'. The table contains several rows of order data.

Data Ordine	Nome	Cognome	Ordine	Indirizzo
1800-01-01			francia corta 30 2100.0	
1800-01-01			francia corta 30 2101.5	
1800-01-01			Francia Corta 15 236.25	
2023-03-09	null	null	[Francia Corta, 33]	ehsi
2023-03-09	null	null	[Francia Corta, 11]	ezz
2023-03-09	Tito	Fiorentino	[Francia Corta, 17]	qwerty
2023-03-09	Tito	Fiorentino	Francia Corta 16 11.0	cappolo
2023-03-09	Agostino	Poggi	Francia Corta 15 11.0	zzz
1800-01-01	a	a	francia corta 1 70.0	a
1800-01-01	a	a	test 1 52.54	a
1800-01-01	a	a	test 1 52.57	a
1800-01-01	a	a	testa 1 50.25	a

6-Ricerca Clienti

- 1.Inserire Cognome nell'apposito campo
- 2.Cliccare su “Search”
 - a. Cliccare su “Clear” per resettare la ricerca

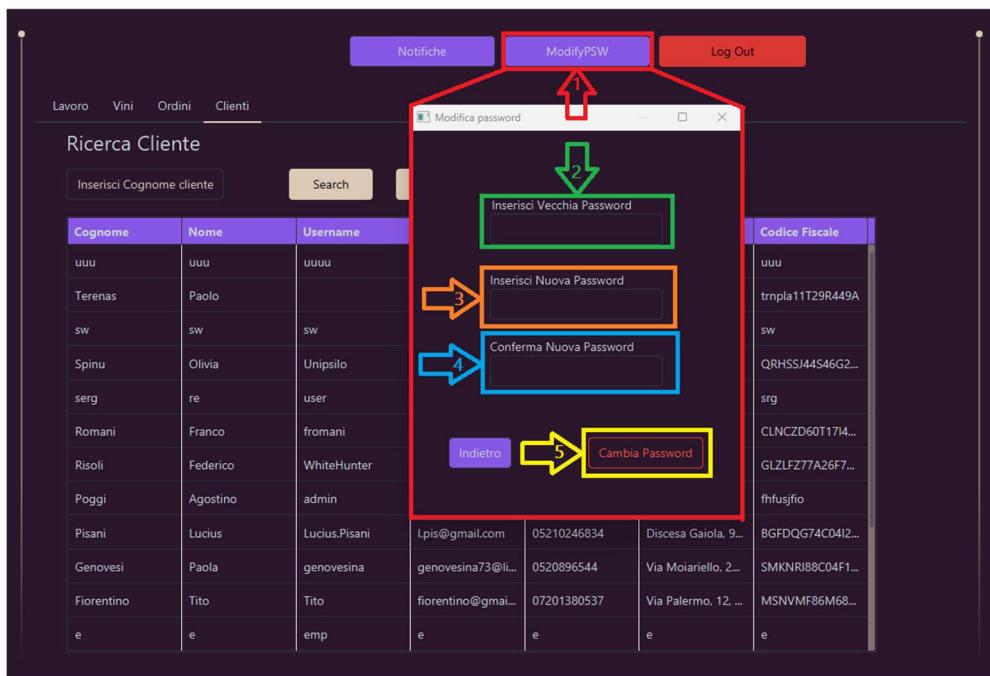
The screenshot shows a user interface for searching clients. At the top, there are tabs for 'Lavoro', 'Vini', 'Ordini', and 'Clienti'. Below the tabs is a search bar labeled 'Ricerca Cliente' with an input field for 'Inserisci Cognome cliente' and a search button ('Search'). A green arrow points from the number '2' to the 'Search' button. To the right of the search bar are two buttons: 'Clear' (orange) and 'a' (grey). Below the search bar is a table with columns: 'Cognome', 'Nome', 'Username', 'Mail', 'Telefono', 'Indirizzo', and 'Codice Fiscale'. The table contains several rows of client data.

Cognome	Nome	Username	Mail	Telefono	Indirizzo	Codice Fiscale
uuu	uuu	uuuu	uu	uu	uu	uuu
Terena	Paolo		terenaSRL@gm...	6974456622	Barriera di Milan...	tmpila11T29R449A
sw	sw	sw	sw	sw	sw	sw
Spinu	Olivia	Unipsilo	ocimicidae.mate...	9874563218	via Campus 3, P...	QRHSSI44S46G2...
serg	re	user	dsfg	sdfg	sdfg	srg
Romani	Franco	fromani	mciglia@gmail.c...	03161986517	Via Vicenza, 15 , ...	CLNCZD60T17I4...
Risol	Federico	WhiteHunter	risolia@gmail.com	4832185138	via Campus 2, P...	GLZLFZ77A26F7...
Poggi	Agostino	admin	A.Poggi@unipr.it	3333333333	via Campus 1	ffhusjfo
Pisani	Lucius	Lucius.Pisani	l.pis@gmail.com	05210246834	Discesa Gaiola, 9...	BGFDQG74C04I2...
Genovesi	Paola	genovesina	genovesina73@li...	0520896544	Via Molariello, 2...	SMKNRI88C04F1...
Fiorentino	Tito	Tito	fiorentino@gmai...	07201380537	Via Palermo, 12 ...	MSNVMF86M68...
e	e	emp	e	e	e	e

7-Modifica Password

- 1.Cliccare su “Modify PSW”
- 2.Inserire Password attuale
- 3.Inserire nuova Password
- 4.Confermare Nuova Password
- 5.Cliccare su “Cambia Password”

NB. Se non si ricorda la password contattare l'admin per il Reset delle credenziali

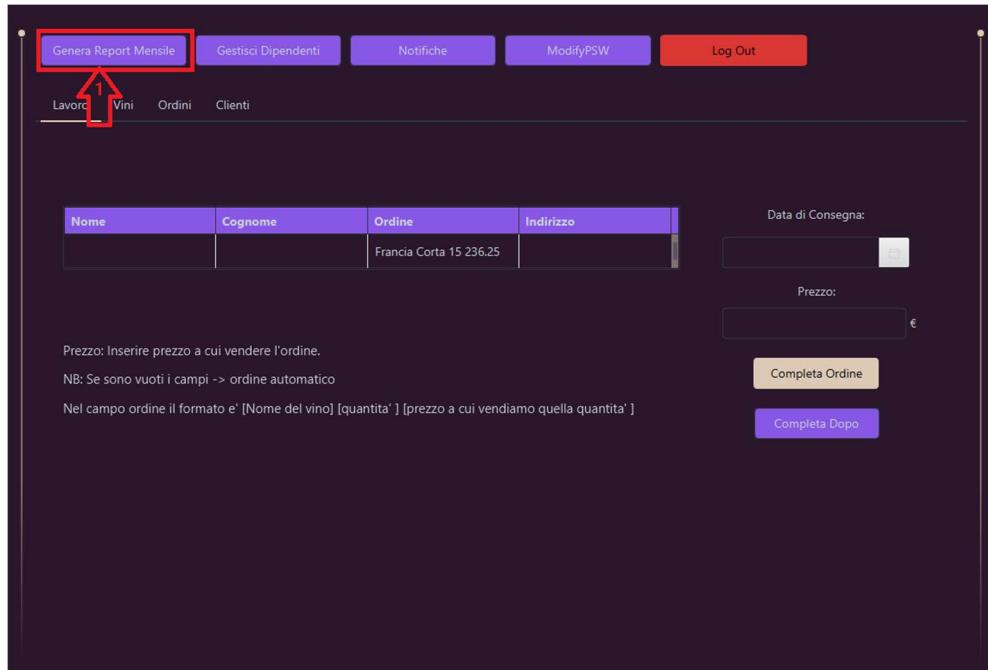


Amministratore

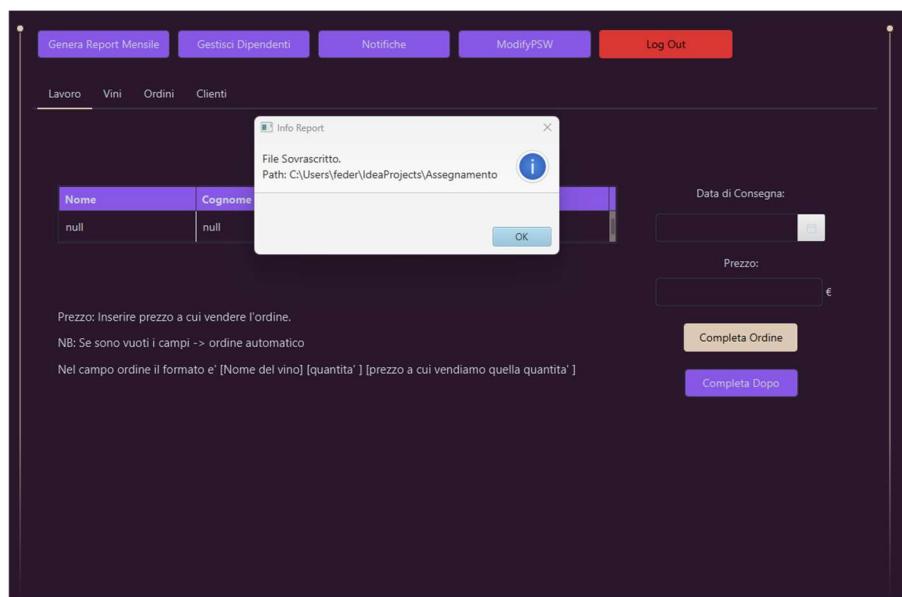
L' amministratore è considerato come un impiegato particolare, per le funzioni base Vedere manuale utente per impiegati, qui sotto verrà mostrato il manuale delle funzionalità specifiche dell'Admin.

1-Generare Report Mensile

1.Cliccare su “Genera report Mensile”



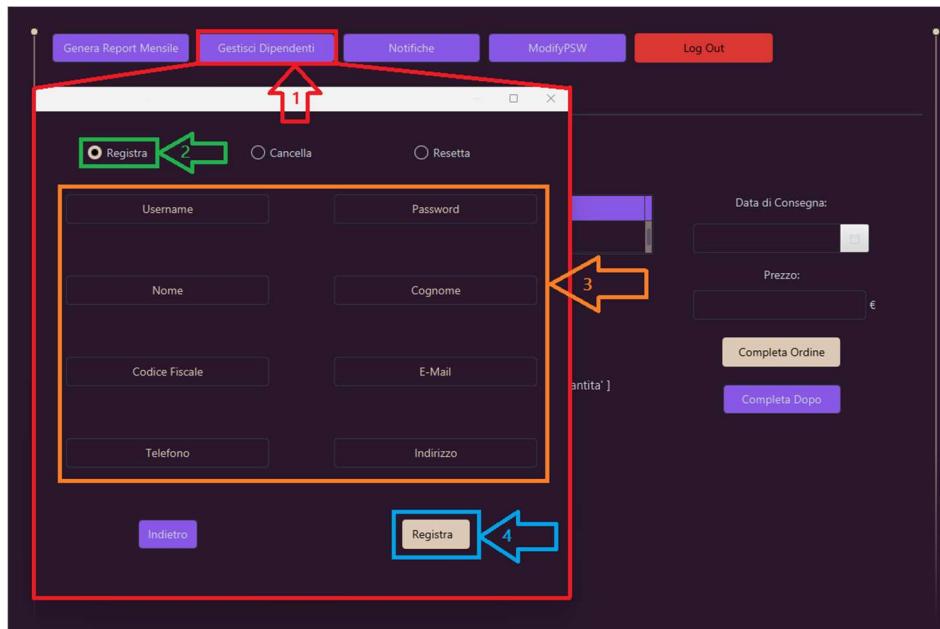
Il report si trova nel percorso specificato dal popup



2-Gestione dipendenti

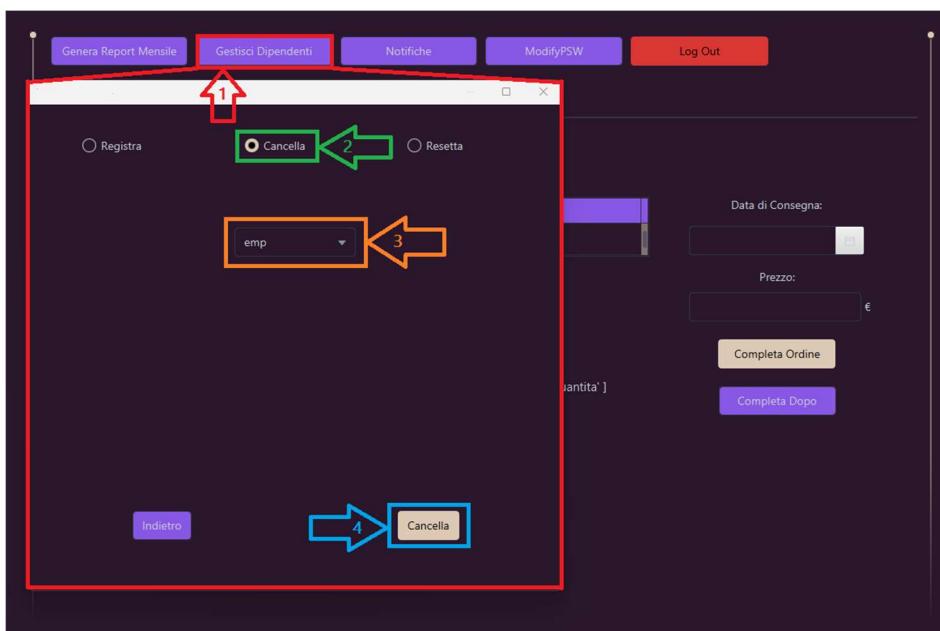
Registrazione dipendente

- 1.Cliccare su “Gestione Dipendenti”
- 2.Selezionare il radio “Registra”
- 3.Compilare i campi richiesti
- 4.Cliccare su “Registra”



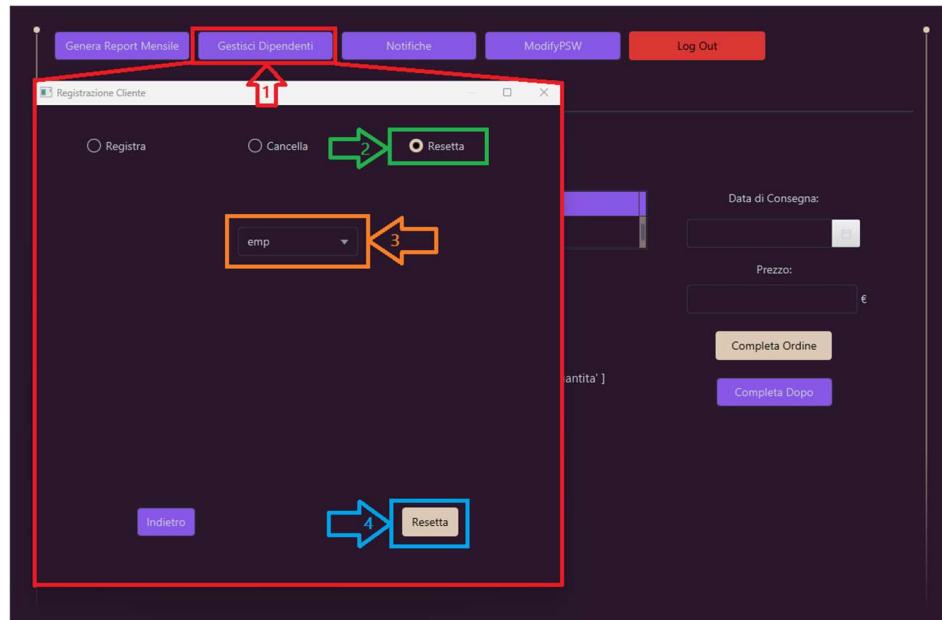
Cancellazione dipendente

- 1.Cliccare su “Gestione Dipendenti”
- 2.Selezionare il radio “Cancella”
- 3.Selezionare il dipendente da eliminare
- 4.Cliccare su “Cancella”



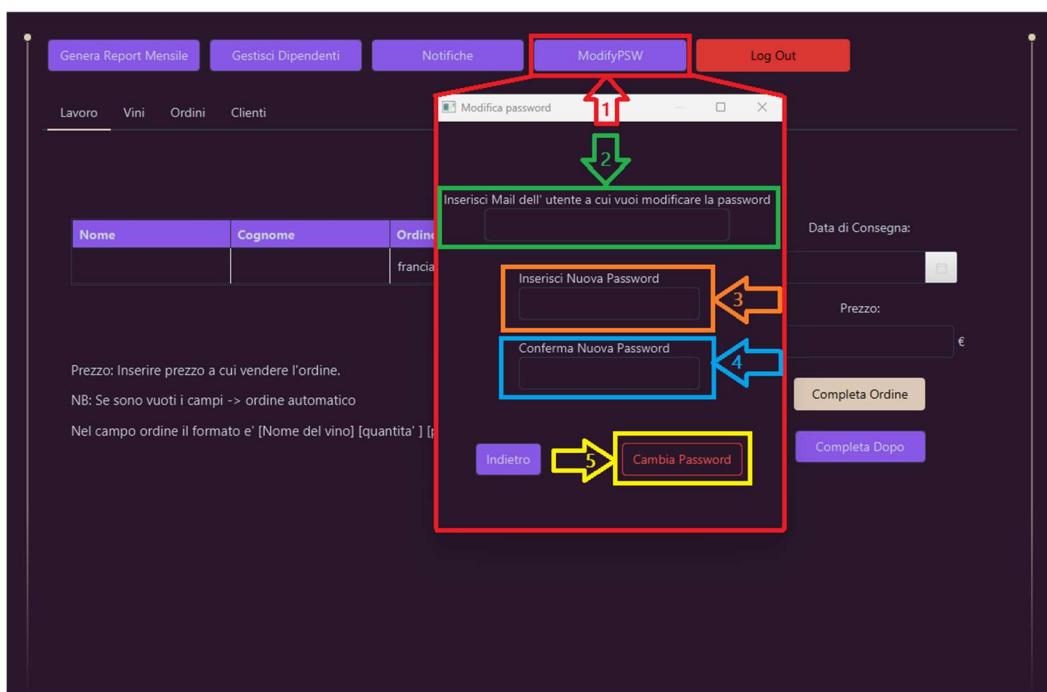
Reset Credenziali dipendente

- 1.Cliccare su “Gestione Dipendenti”
- 2.Selezionare il radio “Resetta”
- 3.Selezionare il dipendente a cui resettare le credenziali
- 4.Cliccare su “Resetta”



3-Modifica password

- 1.Cliccare su “ModifyPSW”
- 2.Inserire la mail dell’utente a cui si vuole cambiare password
- 3.Inserire Nuova password
- 4.Confermare la password
5. Cliccare su “Cambia password”



Changelog

In questa sezione mostreremo i cambiamenti apportati al programma con tanto di data e nome dell'autore in ordine dal più nuovo al più vecchio. Per maggiori informazioni visitare la pagina Github <https://github.com/FedericoRisoli/Assegnamento>

Elia Candida alias Elia, elia2904
Federico Risoli alias Fede, Lyfaen666

Update DOC installazione e configurazione	origin & master	Elia	Today 13:23
Non riceviamo i messaggi, committo questo codice per salvarlo in vista di revert ora si possono rifiutare le notifiche	origin & message_error	Elia	Today 11:48
Update DOC Finiti use case		Elia	17/03/2023 11:57
Aggiunto firma digitale		Lyfaen666	17/03/2023 11:54
Cambiato Titolo alle pagine		Lyfaen666	17/03/2023 11:34
usecase definitivo		Lyfaen666	17/03/2023 10:39
Update DOC		Elia Candida*	16/03/2023 13:32
fix #72 minor changes ai titoli		Federico Risoli	16/03/2023 12:11
Update Doc		Elia Candida*	16/03/2023 01:32
Update DOC		Elia Candida*	16/03/2023 01:32
Update DOC		Elia	15/03/2023 15:44
Merge remote-tracking branch 'origin/master'		Elia	15/03/2023 13:55
Aggiunte immagini Class Diagram		Elia	15/03/2023 13:55
usecaseaggiornato		Federico Risoli	15/03/2023 13:54
Merge remote-tracking branch 'origin/master'		Lyfaen666	15/03/2023 13:32
finito manuale utente admin		Lyfaen666	15/03/2023 13:32
Aggiungi Class Diagram		Elia	15/03/2023 13:29
Merge remote-tracking branch 'origin/master'		Elia	15/03/2023 12:31
Added to tracked files Architecture.drawio.png		Elia	15/03/2023 12:31
minor change		Elia	15/03/2023 12:29
Merge remote-tracking branch 'origin/master'		Lyfaen666	15/03/2023 12:26
iniziato manuale utente admin		Lyfaen666	15/03/2023 12:26
Architecture Diagram Added		Elia Candida*	15/03/2023 12:07
Finito manuale utente impiegato		Lyfaen666	15/03/2023 11:49
Finito manuale utente Cliente		Lyfaen666	14/03/2023 09:47
Merge remote-tracking branch 'origin/master'		Elia	13/03/2023 14:42
Update Doc		Elia	13/03/2023 14:42
Merge remote-tracking branch 'origin/master'		Lyfaen666	13/03/2023 14:41
minor changes		Lyfaen666	13/03/2023 14:41
Removed CSS		Elia	13/03/2023 14:34
Update doc		Elia	13/03/2023 14:32
Merge remote-tracking branch 'origin/master'		Lyfaen666	13/03/2023 14:25
Ui finita (sistemare solo testo sotto taella lavoro personale)		Lyfaen666	13/03/2023 14:25
Rimosso CSS superfluo		Elia	13/03/2023 13:38
Merge remote-tracking branch 'origin/master'		Lyfaen666	13/03/2023 13:33
cambiamento UI e Logo		Lyfaen666	13/03/2023 13:33

Documentati Test	Elia	13/03/2023 13:00
Aggiunto Unit Test	Elia	13/03/2023 12:31
rimosso vecchio commento	Elia	13/03/2023 12:31
tolta estensione	Elia	13/03/2023 11:38
UPDATE UI	Elia	13/03/2023 11:17
Tolto pulsante per cambio CSS	Elia	13/03/2023 11:13
aggiunta possibilità di ordinare vini non disponibili dopo aver comprato	Elia	13/03/2023 11:12
Update DB Tolti record problematici	Elia	13/03/2023 10:50
bugfix data picker Ordini	Elia	13/03/2023 10:45
bugfix data picker	Elia	13/03/2023 10:43
Update CSS	Elia Candida*	13/03/2023 10:34
Merge remote-tracking branch 'origin/master'	Federico Risoli	13/03/2023 09:59
Update documentazione	Federico Risoli	13/03/2023 09:58
Merge pull request #70 from FedericoRisoli/bugfix_vari	Elia Candida*	13/03/2023 00:25
bugfix data picker	Elia	12/03/2023 13:59
tolto popup	Elia	11/03/2023 15:12
tolto totale prezzo	Elia	11/03/2023 15:12
aggiunta negazione nei 4 if del commit precedente	Elia	11/03/2023 14:52
tolta chiamata inutile a server per clienti	Elia	11/03/2023 14:49
se lavori vuoti ora non da errore	Elia	11/03/2023 14:49
Iniziato Documentazione (iniziato userguide+function implementation)	Federico Risoli	10/03/2023 18:09
aggiunto logout sicuro e eliminati duplicati thread	Elia	09/03/2023 17:12
modifica addimpiegato previene duplicati	Elia	09/03/2023 17:09
Update DataTest.java	elia2904*	09/03/2023 15:23
Merge remote-tracking branch 'origin/master'	Elia	09/03/2023 15:21
i dipendenti ottengono i nuovi lavori senza ricaricare il server	Elia	09/03/2023 15:20
risolto bug prezzo	Lyfaen666	09/03/2023 15:13
rimosso import non utilizzato	Elia	09/03/2023 14:58
Aggiunto clear e usermainpage controller e migliorato gli altri clear	Lyfaen666	09/03/2023 14:57
bug fix ordini in caso di vino mancante ora funzionano cambiati prezzi	Elia	09/03/2023 13:08
data nome e cognome non sono piu>null per gli impiegati	Lyfaen666	09/03/2023 12:09
reset = null invece che vuoto	Lyfaen666	09/03/2023 11:42
Bug Fix i dipendenti possono comprare vini senza problemi, carrello.clear() aggiustato	Elia	09/03/2023 11:27
Bug Fix i dipendenti possono tornare alla loro schermata ora invece che a quella dei clienti	Elia	09/03/2023 11:07
Bug Fix search date	Elia	09/03/2023 10:58

bug fix date	Lyfaen666	09/03/2023 10:41
Bug Fix e aggiunte funzioni utili	Elia	09/03/2023 10:15
aggiunto round	Lyfaen666	09/03/2023 10:13
alpha Use Case 0.1	Federico Risoli	06/03/2023 10:07
Aggiorna Usecase.drawio.png	Federico Risoli	06/03/2023 09:32
Aggiorna Usecase.drawio.png	Federico Risoli	06/03/2023 09:21
Aggiorna Usecase.drawio.png	Federico Risoli	06/03/2023 09:14
Aggiorna Usecase.drawio.png	Federico Risoli	06/03/2023 09:11
Usecase.png	Federico Risoli	06/03/2023 08:54
Bug Fix	Elia	04/03/2023 12:36
update UI	Elia	03/03/2023 17:02
diversi cambiamenti: Update DB Cambio UI personnel.fxml per ricerca tramite data di ordine, il controller Per Elia	Elia	03/03/2023 16:51
update DB	Elia	03/03/2023 16:08
fix #66 , fix #68 + minor changes	Federico Risoli	03/03/2023 15:32
arrotondato report Corretto bug stringhe vuote	Elia	03/03/2023 15:28
fix #65 e migliorati i popup Nota: in generale mai usare alwaysontop	Elia	03/03/2023 14:54
rimosso codice test vecchio ora era inutile	Elia	03/03/2023 14:38
fix #67 Update Readme	Elia	03/03/2023 14:37
fix #59	Elia	02/03/2023 17:19
minor changes	Elia	02/03/2023 17:18
Fix #54 errori di login tolti (ho sbagliato non era 52)	Lyfaen666	27/02/2023 12:31
Fix #52 errori di login tolti	Lyfaen666	27/02/2023 12:26
Reso i radio mutualmente esclusivi e migliorato UI, (bug negli alert)	Federico Risoli	21/02/2023 10:03
Notifiche funzionano da rifinire alcune cose come visualizzazione pagine, e sistemare fxml	Federico Risoli	21/02/2023 09:26
Aggiunto rime a notifiche	Lyfaen666	20/02/2023 18:51
Merge pull request #64 from FedericoRisoli/report	elia2904*	20/02/2023 12:08
Merge branch 'master' into report	elia2904*	20/02/2023 12:07
Implementato Report tramite file invece che in console e ristretto l'uso solo per l'admin	Elia	20/02/2023 11:51
Creato Sezione notifiche (modificando DB e classe ordini di vendita) potrebbero esserci bug	Lyfaen666	20/02/2023 11:49
bug fix #63	Elia	20/02/2023 10:21
bug fix PersonnelpageController	Elia	20/02/2023 10:13
update DB	Elia	19/02/2023 18:03
creato report in funzione Initialize, visibile in console da fare: creare pagina per report	Elia	19/02/2023 17:54
fix #61	Elia	19/02/2023 17:03
fixato errore	Elia	19/02/2023 16:52
aggiunte funzioni utili	Elia	19/02/2023 16:45
cambiamenti minori	Elia	19/02/2023 16:37

aggiunto ordine automatico quando q.ta vini bassa	Elia	14/02/2023 17:52
fix dato dalla precedente modifica al DB e migliorato DB	Elia	14/02/2023 17:52
Aggiunta funzione per lavoro completato e per skippare lavoro ConcurrentServer e PersonnelpageController	Elia	13/02/2023 15:11
Eliminata colonna non utile	Elia	13/02/2023 11:20
prima bozza di round robin	Elia	12/02/2023 15:52
Bug Fix problemi di gestione lavori Aggiunti commenti utili per testing manuale Rimosse variabili inutili e bug	Elia	12/02/2023 14:58
Risolto problema messaggi nulli	Elia	12/02/2023 13:35
presunto miglioramento struttura codice	Elia	12/02/2023 01:57
CODICE NON FUNZIONANTE URGENTE BUG FIX NECESSARIO funzione logout-concurrent server ha un probi	Elia	01/02/2023 14:25
bug fix logout	Elia	01/02/2023 14:04
migliorato logout dipendenti/admin	Elia	01/02/2023 13:58
Update README.md	elia2904*	31/01/2023 16:31
Merge pull request #56 from FedericoRisoli/logout_testing	elia2904*	31/01/2023 16:28
Implementato logout sicuro e migliorata assegnazione lavori	Elia	31/01/2023 16:25
aggiunto ID alla classe OrdiniVendita e fix query errata	Elia	31/01/2023 15:59
rimosso commento	Elia	31/01/2023 15:28
continuato lavoro per log-out sicuro	Elia	30/01/2023 15:35
Merge pull request #55 from FedericoRisoli/lavori	elia2904*	30/01/2023 14:44
Merge remote-tracking branch 'origin/lavori' into lavori	Elia	30/01/2023 14:43
aggiornato DB	Elia	30/01/2023 14:42
assegnazione lavoro completata	Elia	30/01/2023 14:00
assegnazione lavoro completata	Elia	30/01/2023 14:00
testing logout tutto commentato perchè non funziona, causa null stage in MyController	Elia	30/01/2023 00:02
rimosso commento	Elia	29/01/2023 23:57
comunicazione emp-server per assegnazione lavori iniziata	Elia	29/01/2023 23:56
aggiunte e modificate funzioni	Elia	29/01/2023 23:53
Sto aggiungendo funzione che permette di completare gli ordini (help per ficcare l' id dell ordine nella query in base)	Lyfaen666	27/01/2023 15:18
Fix #46 ora ricerca vini funziona come deve	Lyfaen666	27/01/2023 14:52
Merge remote-tracking branch 'origin/master'	Lyfaen666	25/01/2023 16:06
Creata Tabella nella main page del personale per visualizzare gli ordini da completare e possibilita' di selezionarli, (Lyfaen666)	Lyfaen666	25/01/2023 16:06
Merge pull request #53 from FedericoRisoli/errore_query	Federico Risoli*	25/01/2023 12:33
#fix 52 ora il server aggiorna i job falliti	Lyfaen666	25/01/2023 12:31
Add files via upload	elia2904*	25/01/2023 11:42
provata a fare query ma errore. ascoltare audio whatsapp	Elia	24/01/2023 13:25
Dovrei aver fixato il bug dei vini testo oggi pomeriggio	Lyfaen666	23/01/2023 11:48
Aggiunta possibilita di controllare i propri ordini per gli user	Lyfaen666	23/01/2023 11:28

fix ai test data minori	Lyfaen666	23/01/2023 10:23
Aggiunto i test alle aggiunte della classe Data	Lyfaen666	23/01/2023 10:19
Ora i vini non disponibili se vengono ordinati aggiungono al db un ordine di vendita con data di consegna inizializzata	Lyfaen666	23/01/2023 10:15
Aggiunto a Data metodi per settare e gettare nome e cognome utente, provato a fare query per aggiungere in ordine corretto test	Lyfaen666	22/01/2023 19:27
Merge remote-tracking branch 'origin/master'	elia2904	19/01/2023 21:08
aggiunti test e corretto errore indice	elia2904	19/01/2023 20:48
Tolto idea tabella e modificato riepologo per mostrare informazioni giuste	Lyfaen666	19/01/2023 20:41
Update README.md	elia2904*	19/01/2023 18:02
Testing (#51) Aggiunto primo test automatico e fatto setup per github	elia2904*	19/01/2023 17:54
Update pom.xml	elia2904*	18/01/2023 02:50
Update maven.yml	elia2904*	17/01/2023 18:38
Update maven.yml	elia2904*	17/01/2023 18:30
Create maven.yml	elia2904*	17/01/2023 18:22
Update README. Fix #47	elia2904*	15/01/2023 14:03
Creato Pop-up (tentato di popolare tabella ma non sono riuscito in tempo)	Lyfaen666	13/01/2023 12:09
cambiata ui minor changes	elia2904	12/01/2023 22:51
cambiata ui, iniziato controller, non finito minor changes	elia2904	12/01/2023 21:56
Merge remote-tracking branch 'origin/master'	elia2904	12/01/2023 20:31
corretto e espanso readme	elia2904	12/01/2023 20:19
corretto e espanso readme	elia2904	12/01/2023 20:19
aggiunto fornitore e corriere al DB	elia2904*	12/01/2023 20:05
corretto errore, stavolta veramente	elia2904*	12/01/2023 18:26
corretto errore	elia2904*	12/01/2023 18:24
minor change	elia2904	12/01/2023 18:24
Merge pull request #45 from FedericoRisoli/concorrente	elia2904*	12/01/2023 18:19
update README.md	elia2904	12/01/2023 18:12
eliminato codice da test	elia2904	12/01/2023 18:12
ATTENZIONE aggiunto metodo getSocket() che è necessario per alcuni commit precedenti	elia2904	12/01/2023 18:12
Aggiunti/cambiati diversi metodi cambiato handleMessage aggiunti sendMessage, killChildThread, changeScene	elia2904	12/01/2023 18:11
Aggiunto modo per ottenere istanza e modo per bloccare il thread	elia2904	12/01/2023 18:08
Passo ruolo e id di chi logga al server minor changes	elia2904	12/01/2023 18:06
aggiunto codice per segnarsi l'id dell'impiegato/admin che fa login attenzione funziona solo al primo login minor	elia2904	12/01/2023 18:04
update DB	elia2904	12/01/2023 17:58
Eliminato file non più utile	elia2904	12/01/2023 17:48
tolto codice per test	elia2904	12/01/2023 17:46
abbozzata struttura per scambio dati e messi commenti per cosa fare in futuro	elia2904	10/01/2023 18:57

rimossi commenti inutili	elia2904	10/01/2023 18:56
duplicato in Server per poterlo far usare anche al Server concorrente	elia2904	10/01/2023 18:56
Preparato UI per ordini di vendita (non finito)	Lyfaen666	10/01/2023 18:06
Merge remote-tracking branch 'origin/master'	Lyfaen666	10/01/2023 18:00
Creato controller per proposta acquisto	Lyfaen666	10/01/2023 18:00
Merge remote-tracking branch 'origin/master'	elia2904	10/01/2023 17:56
aggiornato DB	elia2904	10/01/2023 17:55
Aggiunto UI per proposta acquisto	Lyfaen666	10/01/2023 17:55
Merge remote-tracking branch 'origin/master'	Lyfaen666	10/01/2023 10:01
Merge remote-tracking branch 'origin/master'	Lyfaen666	10/01/2023 10:01
Merge pull request #42 from FedericoRisoli/server	elia2904*	10/01/2023 03:08
aggiunte istruzioni per ottenere dai controller i messaggi inviati dal server	elia2904	10/01/2023 03:04
aggiunto commento	elia2904	10/01/2023 03:04
minor Fix	elia2904	10/01/2023 03:03
Test per ricevere i messaggi	elia2904	10/01/2023 02:48
snellito file	elia2904	10/01/2023 02:46
Creato Classe MessageReceiverTask per uso asincrono dei messaggi in arrivo	elia2904	10/01/2023 02:45
mando messaggio appena si connette il client per testare	elia2904	10/01/2023 02:43
Reso Controller sottoclasse di MyController	elia2904	09/01/2023 23:03
Creato Classe MyController, futuro padre di tutti i controller	elia2904	09/01/2023 23:02
Creato classe di utilità partendo dal client	elia2904	09/01/2023 16:05
Implementato scambio messaggi di testo tramite console	elia2904	09/01/2023 15:37
Abbozzata prima struttura Client-Server Concorrente	elia2904	09/01/2023 15:04
Fixed #33 I vini ora hanno il prezzo arrotondato dal momento in cui son definiti	elia2904	09/01/2023 14:08
Merge branch 'tmp'	Lyfaen666	08/01/2023 22:58
Merge pull request #40 from FedericoRisoli/tmp	elia2904*	08/01/2023 16:08
Merge branch 'master' into tmp	elia2904	08/01/2023 16:06
Aggiunto funzione ad impiegati e admin di ricercare cose secondo la consegna	Lyfaen666	08/01/2023 15:07
Cancellato tutto per ripartire da capo	Lyfaen666	08/01/2023 13:12
Ripristinato senza errori il Master	Lyfaen666	08/01/2023 13:10
Merge pull request #39 from FedericoRisoli/revert-38-tmp	elia2904*	07/01/2023 18:59
Revert "ottimizzato"	elia2904*	07/01/2023 18:59
Merge pull request #38 from FedericoRisoli/tmp	elia2904*	07/01/2023 18:59
ottimizzato	Lyfaen666	07/01/2023 18:52
Merge pull request #36 from FedericoRisoli/Errore-tabclienti	Federico Risoli*	07/01/2023 18:35
provato a fixare errori e ottimizzato codice	Lyfaen666	07/01/2023 18:33
Fixato promo,(HEIP non capisco perche non viene popolata la tabella clienti, ho meglio funziona ma non si vede il	Lyfaen666	07/01/2023 12:08

Merge pull request #31 from FedericoRisoli/ramo_di_elia	elia2904*	06/01/2023 17:44
fixati errori da commit e minor ui fx	elia2904	06/01/2023 17:42
Merge branch 'master' into ramo_di_elia	elia2904*	06/01/2023 17:32
dopo un pagamento modifico la quantità di bottiglie vendute e rimanenti predisposto il codice per gli ordini delle	elia2904	06/01/2023 17:26
Refactorato UI	Lyfaen666	06/01/2023 15:20
Fixato problema log in	Lyfaen666	06/01/2023 15:04
Popolato DB	elia2904*	05/01/2023 19:45
change minori	elia2904	05/01/2023 19:26
Fix errore programmazione	elia2904	05/01/2023 19:25
sistemato schermata personale (da finire)	Lyfaen666	05/01/2023 18:10
Rifatta Ui	Lyfaen666	05/01/2023 12:41
Rinnovato UI impiegati e admin	Lyfaen666	03/01/2023 17:54
Separato main page dipendenti da quella dagli utenti erano diventate troppe cose da gesire in una pagina	Lyfaen666	03/01/2023 15:31
PRomo completata	Lyfaen666	01/01/2023 21:13
Fixato problema pop-up promo	Lyfaen666	01/01/2023 20:57
tolta colonna offerta e UI change tabella	elia2904	01/01/2023 17:27
Fatto si che dopo il pagamento di un ordine (o annullamento) si ritorni alla mainpage	elia2904	01/01/2023 17:12
Risolto bug ordine anni Usavo apici errati Fix #9	elia2904	01/01/2023 16:48
Popolato DB	elia2904*	01/01/2023 16:35
Fixati pop-up rotti, aggiunta chiusura stage dove mancante, eliminati return; inutili	elia2904	31/12/2022 16:40
Fatto fxml e tabella per spammare la promo al' utente non riesco a popolarla vedo se riusiro) altrimenti ci sentiam	Lyfaen666	31/12/2022 13:36
Reset password da parte dell' admin verso gli impiegati completato (setta la password di default "emp")	Lyfaen666	31/12/2022 12:42
Pagamento con bonifico e carta finiti. Generano ordine di vendita	elia2904	30/12/2022 16:13
aggiornata classe per maggior versatilità	elia2904	30/12/2022 16:07
Ora quando un cliente logga gli compare lo spam delle promo (da finire)	Lyfaen666	30/12/2022 15:18
Dovrei aver fixato il problema del cognome	Lyfaen666	30/12/2022 15:11
aggiunta tabella	elia2904	29/12/2022 22:13
aggiunto commento, lavoro incompleto	elia2904	29/12/2022 22:13
creato popup per pagare con bonifico	elia2904	29/12/2022 21:46
Merge remote-tracking branch 'origin/master'	elia2904	29/12/2022 21:37
creato popup per pagare con la carta, bonifico non funzionante	elia2904	29/12/2022 21:35
Update README.md	elia2904*	29/12/2022 19:45
Merge remote-tracking branch 'origin/master'	elia2904	29/12/2022 19:45
aggiunta funzionalità al pulsante Annulla Ordine	elia2904	29/12/2022 19:44
Update README.md	elia2904*	29/12/2022 19:13
aggiunto riepilogo prima di procedere al pagamento	elia2904	29/12/2022 19:11
Merge remote-tracking branch 'origin/master'	elia2904	29/12/2022 17:28

Update README.md	elia2904*	29/12/2022 15:11
aggiunte checkbox alle cose da fare	elia2904*	29/12/2022 15:09
Merge remote-tracking branch 'origin/master'	elia2904	29/12/2022 14:28
aggiunto fxml per promo	Lyfaen666	29/12/2022 13:22
Modify password funzionante e completato	Lyfaen666	29/12/2022 12:58
fixato bug ui ui non completa	elia2904	28/12/2022 21:07
css fix	elia2904	28/12/2022 20:55
fixato bug ora admin puo cancellare solo dipendenti	Lyfaen666	28/12/2022 16:36
ora admin puo eliminare dipendenti	Lyfaen666	28/12/2022 16:33
Migliorato pagina gestione dipendenti	Lyfaen666	28/12/2022 14:33
Creato menu per selezionare l' azione da fare sui dipendenti (registra cancella reset)	Lyfaen666	28/12/2022 14:10
Ora l' admin registra i dipendenti	Lyfaen666	28/12/2022 13:31
Prende l' icona da un immagine su internet	Lyfaen666	28/12/2022 13:06
Fixato bug login (ora compare la scritta non errori sulla console)	Lyfaen666	28/12/2022 12:54
Unito i modify psw controller e fixato i bug relativi alla modifica password	Lyfaen666	28/12/2022 12:50
abbozzata UI. Non finita	elia2904	27/12/2022 22:28
modificato UsermainpageController.java per portare a riepilogo.fxml dopo aver premuto Compra riepilogo.fxml è	elia2904	27/12/2022 21:44
aggiunta classe Carrello per salvare da una scena all'altra l'ordine	elia2904	27/12/2022 21:41
aggiunto loop che indica quante bottiglie di che tipo si intende comprare è un placeholder per quando lo completeremo	elia2904	27/12/2022 15:41
rimosso codice duplicato	elia2904	27/12/2022 15:13
refactoring	elia2904	27/12/2022 15:10
Merge pull request #17 from FedericoRisoli/sicurezza	elia2904*	27/12/2022 14:52
Merge branch 'master' into sicurezza	elia2904*	27/12/2022 14:50
rifatta UI e controller per rispecchiare le richieste del prof aggiornato DB	elia2904	27/12/2022 14:40
creata classe Vini.java per ospitare i dati dei vini	elia2904	27/12/2022 14:40
aggiornato Database	elia2904*	26/12/2022 20:53
eliminato css superfluo e cambiato css in pagina admin	elia2904	26/12/2022 15:02
abbellite pagine con css	elia2904	26/12/2022 15:00
Merge pull request #15 from FedericoRisoli/re-style	elia2904*	25/12/2022 23:53
aggiunto css in maniera brutale ovvero senza rimettere in ordine alcune cose che si sono spostate	elia2904	25/12/2022 23:41
aggiunti file css utili IntelliJ dice che ci sono errori in questi file ma secondo me no, fonte dei file github	elia2904	25/12/2022 23:39
Eliminati file poco utili	elia2904	24/12/2022 14:22
aggiunto funzione al pulsante di log-out	elia2904	23/12/2022 23:10
aggiunto pulsante per gestire utenti. UI fix	elia2904	23/12/2022 22:22
Implementata ricerca vini, rifatta UI, aggiunta immagine, eliminati import non utili BUG linea 70 e 71 UsernameCon	elia2904	23/12/2022 20:49
Ridotto e ottimizzato numero di controller	Lyfaen666	23/12/2022 13:03

eliminato un controller così da gestire meno cose	Lyfaen666	23/12/2022 12:29
Sistemato MODifica password ora solo l'admin può modificare tutte le password e gli impiegati possono modificare	Lyfaen666	23/12/2022 00:42
Riuscito a passare dati tra una scena e l'altra con classe DATA, da capire meglio il funzionamento	Lyfaen666	23/12/2022 00:27
Creato funzione idgetter per prendere l'id ora devo capire come passare una variabile da una scena all'altra una volta	Lyfaen666	22/12/2022 20:52
MODifica psw funzionante (da fissare il fatto che si possono modificare tutte le psw), dobbiamo trovare il modo di	Lyfaen666	22/12/2022 18:18
Cerco il vino richiesto NON FINITO	elia2904	22/12/2022 16:04
cancellate linee vuote inutili	elia2904	22/12/2022 15:47
Aggiunti anni alla ComboBox	elia2904	22/12/2022 15:44
Aggiunto schermata diversa anche per admin e cercato di fissare il bug (a me funziona eventualmente ci sentiamo)	Lyfaen666	22/12/2022 09:35
Update wineshop.sql	elia2904*	21/12/2022 23:52
aggiunto user user	elia2904	21/12/2022 23:42
Sistemato funzione modifica password (la query non funziona e ora devo andare a lavoro) dobbiamo trovare un modo	Lyfaen666	21/12/2022 16:48
Creato modifica password sia controller che fxml per permettere ad admin e utenti di modificare la loro password	Lyfaen666	21/12/2022 11:38
Distinto le main page in base al ruolo, ora durante il login viene controllato il ruolo dell'utente e viene indirizzato a	Lyfaen666	21/12/2022 11:00
Reso Unresizable tutte le finestre (non so come specificare la scena) anyway ora dovrebbe essere fissato il bug più tardi	Lyfaen666	21/12/2022 09:59
Implementato cambio scena a mainpage.fxml dopo login	elia2904	21/12/2022 00:41
Eliminato SceneChanger.java NON FUNZIONAVA	elia2904	21/12/2022 00:39
creato SceneChanger.java NON FUNZIONA doveva esser un modulo per non scrivere sempre la funzione da capo,	elia2904	21/12/2022 00:38
aggiunto grandezza minima a mainpage.fxml	Fede	20/12/2022 19:59
modificata UI mainpage.fxml creato MainpageController.java, non finito	elia2904	20/12/2022 16:50
provato a creare una sorta di pagina da mostrare dopo il login non so mettere il controller poi ci guardo	Lyfaen666	20/12/2022 13:22
Aggiunto tabella vini e terze parti al DB	Lyfaen666	20/12/2022 13:06
una volta registrato torna alla schermata di login	Lyfaen666	20/12/2022 10:51
implementato DBHelper.update per registrare Clienti refactoring minore	elia2904	20/12/2022 01:41
aggiunta funzione per poter aggiungere nuovi record al DB (update)/ modificare forse, non ne sono sicuro. Refactored	elia2904	20/12/2022 01:38
Aggiunto controllo al pulsante registrati (Verifica presenza dell'username nel db), aggiunta query per registrare ma	Lyfaen666	20/12/2022 00:26
aggiunta funzione al pulsante Registra Clienti (ex registrati) che apre pop-up di registrazione	elia2904	19/12/2022 19:53
nessun cambiamento	elia2904	19/12/2022 19:52
Aggiunto RegisterController OnClickRegister (linea 53) ha la funzione per vedere se i campi sono vuoti, ne serve una	elia2904	19/12/2022 19:50
Aggiunta pagina per registrare utenti	elia2904	19/12/2022 19:46
Aggiunta interrogazione DB per vedere se un utente esiste nel DB. Migliorare la risposta dell'applicazione HelloController	elia2904	19/12/2022 16:56
aggiunto Database per poterlo usare insieme	elia2904	19/12/2022 16:51
creato DBHelper.java per contenere le funzioni del DB	elia2904	19/12/2022 16:50
nuova libreria tentativo di connessione collegato DB a intelliJ	elia2904	19/12/2022 14:51
Aggiunto pulsante "Registrati" Ancora da aggiungere anche al controller	elia2904	18/12/2022 19:00
Corretto errore di spelling	elia2904*	18/12/2022 18:16
Merge remote-tracking branch 'origin/master'	Lyfaen666	18/12/2022 17:27
primi passi con JavaFX. Fatta pagina login e implementato messaggio errore	elia2904	18/12/2022 14:37
Merge remote-tracking branch 'origin/master'	Lyfaen666	17/11/2022 15:32
help please	Lyfaen666	17/11/2022 15:31
gitignore update	elia2904	17/11/2022 15:28
rifatto progetto	elia2904	17/11/2022 15:16
Initial commit	Lyfaen666	17/11/2022 14:45