

Restaurant Manager Report

Simone Di Cesare

July 2022

1 Introduzione

Il Progetto consiste nella realizzazione di un Gestore di un Ristorante con le seguenti funzionalità:

- Creazione e Gestione di un Menu di Portate strutturate in {ID, NOME, PREZZO}
- Gestione degli Ordini, dalla creazione alla preparazione in cucina di essi
- Gestione dei Pagamenti degli Ordini, con stampa degli Scontrini.

2 Descrizione Delle Classi

2.1 ImageButtonHighlighted

Parametri:

- **icon:**
- **highlightedIcon:**
- **action:**

Metodi:

- **setAction():**
- **mouseExited():**
- **mouseEntered():**
- **mouseClicked():**

2.2 TextButtonHighlighted

Parametri:

- **baseForeground:**
- **brightForeground:**
- **action:**

Metodi:

- **setAction():**
- **mouseExited():**
- **mouseEntered():**
- **mouseClicked():**

2.3 MenuScopeDialog

Parametri:

- **scope:**
- **shouldSave:**

Metodi:

- **showDialog():**

2.4 MainPanel

Parametri:

- **waiterSelectionButton:**
- **cookSelectionButton:**
- **chefSelectionButton:**
- **cashierSelectionButton:**
- **exitButton:**
- **employeeCount:**

Metodi:

- **addEmployeeButton():**

2.5 CashierPanel

Parametri:

- menu:
- currentOrder:
- receiptPanel:
- receiptPanelScrollPane:
- receiptTotalLabel:

Metodi:

- getBackgroundAssets():
- refreshReceiptPanel():

Classi Interne:

- DishPanel

Parametri:

- dish:

Metodi:

- setBackground():
- setOpaque():
- getFormattedName():

- ReceiptPanel

Descrizione ReceiptPanel

2.6 ChefPanel

Parametri:

- menu:
- menuPanel:

Metodi:

- addMenuScope():

Classi Interne:

- MenuScopePanel

Parametri:

- scope:

Metodi:

- getFormattedName():

- MenuPanel

Metodi:

- addMenuScope():

- updateMenuScope():

- removeMenuScope():

2.7 WaiterPanel

Parametri:

- menu:

- currentOrder:

- orderPanel:

Metodi:

- getBackgroundAssets():

- refreshOrderPanel():

Classi Interne:

- OrderPanel

Metodi:

- resetOrder():

- DishPanel

Parametri:

- dish

Metodi:

- getFormattedName():

2.8 CookPanel

Parametri:

- menu:
- currentOrder:
- orderPanelScrollPane:
- orderPanel:

Metodi:

- getBackgroundAssets():
- refreshOrderPanel():

Classi Interne:

- OrderPanel

Descrizione bla bla

- DishPanel

Metodi:

- getFormattedName():

2.9 AbstractPanel

Parametri:

- backgroundImage:

Metodi:

- paintComponent():
- getBackgroundAssets():

2.10 MainFrame

Parametri:

- content
- menu
- menuFile

Metodi:

- getMenu():
- setContent():

2.11 Order

Parametri:

- tableNumber
- dishes
- cookedDishes

Metodi:

- add():
- remove():
- get():
- toString():
- indexOf():
- contains():
- entries():
- getTotal():
- cook():
- getTableNumber():
- getCookedDishes():
- setTableNumber():
- decrementQuantity():
- incrementQuantity():
- getDishes():
- addCooked():

2.12 MenuUtil

Metodi:

- saveMenuOnFile():
- getMaxNameLength():
- loadMenuFromFile():

2.13 Dish

Parametri:

- scope
- quantity

Metodi:

- toString():
- setQuantity():
- getQuantity():
- getScope():
- fromText():
- toText():
- decrementQuantity():
- incrementQuantity():

2.14 MenuScope

Parametri:

- id
- name
- cost

Metodi:

- equals():
- toString():
- getName():
- getId():
- setName():
- getCost():
- setCost():
- fromText():
- toText():

2.15 Menu

Parametri:

- menu
- lastID

Metodi:

- add():
- remove():
- toString():
- entries():
- getMenuScope():
- setLastID():
- getLastID():
- getMenu():
- getMenuScopeFromName():
- getMenuScopeFromID():

2.16 OrderUtil

Parametri:

- ORDER_SEPARATOR

Metodi:

- saveOrder():
- getActiveOrders():
- sendOrder():
- availableOrder():
- loadOrderFromFile():
- getOrderFromTable():
- getCompletedOrders():
- stampAndDeleteOrder():
- saveOrderOnFile():
- getMaxNameLength():

3 Funzionalità

- **Creare e Modificare un Menù**

Il menù è contenuto in un file .txt con il seguente formato:

Listing 1: Menu.txt

```
[lastValidID]
ID;Scope_Name;Scope_Cost
ID;Scope_Name;Scope_Cost
...
...
```

dove [lastValidID] è il prossimo ID che verrà associato all'aggiunta di una nova portata. Questo viene incrementato ad ogni aggiunta di portata, e viene ricalcolato alla chiusura del programma, trovando l'ID più grande assegnato alle portate ed incrementandolo di 1. Successivamente è salvata una lista di portate con i propri ID, Nomi e Costi separati da un ";"

Tramite il menù dello chef, è possibile aggiungere una nuova portata inserendone il nome e il costo all'interno di un JDialog personalizzato.

Ed è anche possibile modificarne/eliminarne una già esistente, cliccando con il tasto destro del mouse e selezionando la voce "Edit"/"Remove" nel Pop-up menù aperto.

Il menù è letto da un file all'avvio del programma, tramite il metodo **loadMenuFromFile()** all'interno della classe **MenuUtil**.

Il file in questione è, di default il file "Menu.txt" all'interno della stessa cartella in cui è situato il programma, altrimenti è il file passato da riga di comando tramite l'opzione "-m/-menu [path]".

- **Creare ed inviare un Ordine**

Gli ordini sono contenuti all'interno della cartella "orders" all'interno della stessa cartella contenente il programma, e sono salvati in file .txt avendo nome: "OrderI.txt", dove I è il numero del tavolo. Il formato dei file di ordini è il seguente:

Listing 2: Order3.txt

```
[tableNumber]
ID;Scope_Quantity
ID;Scope_Quantity
...
...
==COOKED==
ID;Scope_Quantity
ID;Scope_Quantity
...
...
```

dove [tableNumber] è il numero del tavolo associato all'ordine, seguito da una lista di piatti ordinati ma non ancora cotti, salvati con il proprio ID e la propria quantità. La stringa di separazione "=="COOKED=="", che divide i piatti non cotti da quelli cotti e spediti al tavolo, con a seguire la lista delle portate cotte e spedite al tavolo con i propri ID e quantità. Il file è leggibile e convertibile in un oggetto di classe Order tramite il metodo **loadOrderFromFile()** della classe **OrderUtil**. Questo oggetto è utilizzato nel Menù del Cameriere, del Cuoco e della Cassa.

- **Cuocere Portate di un Ordine**

Questa operazione è svolta dal menù del Cuoco, il quale può visualizzare una delle portate non cucinate di un tavolo, ed in caso notificare l'avvenuta preparazione.

La classe, tramite il metodo **getActiveOrders()** della classe **OrderUtil**, inizializza la lista di tavoli con ordini in sospeso, e permette la loro visualizzazione e modifica tramite cottura delle portate.

Quando una portata viene cotta, la quantità della portata non cotta all'interno dell'ordine viene decrementata di 1, e viene aumentata di 1 la quantità della portata cotta nell'ordine.

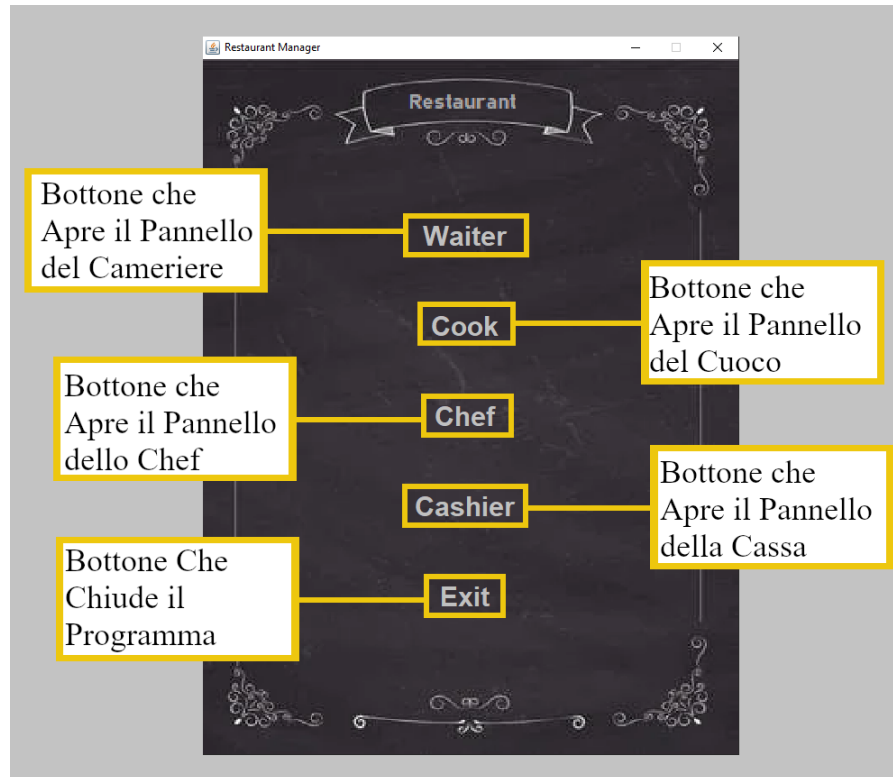
- **Pagare l'ordine alla cassa e Stampa Scontrino**

Questa operazione è svolta dal menù del Cassiere, il quale può visualizzare il conto di un tavolo che abbia consumato almeno 1 portata, e stamparne lo scontrino.

La classe, tramite il metodo **stampAndDeleteOrder()** della classe **OrderUtil**, inizializza la lista di tavoli con ordini che hanno almeno una pietanza sotto a sezione delle pietanze cotte e spedite. Oltre alla visualizzazione, permette di stampare lo scontrino tramite un pulsante, il quale richiama il metodo della classe, che salva lo scontrino dell'ordine, ed elimina l'ordine corrente dalla cartella "orders".

4 Manuale della GUI

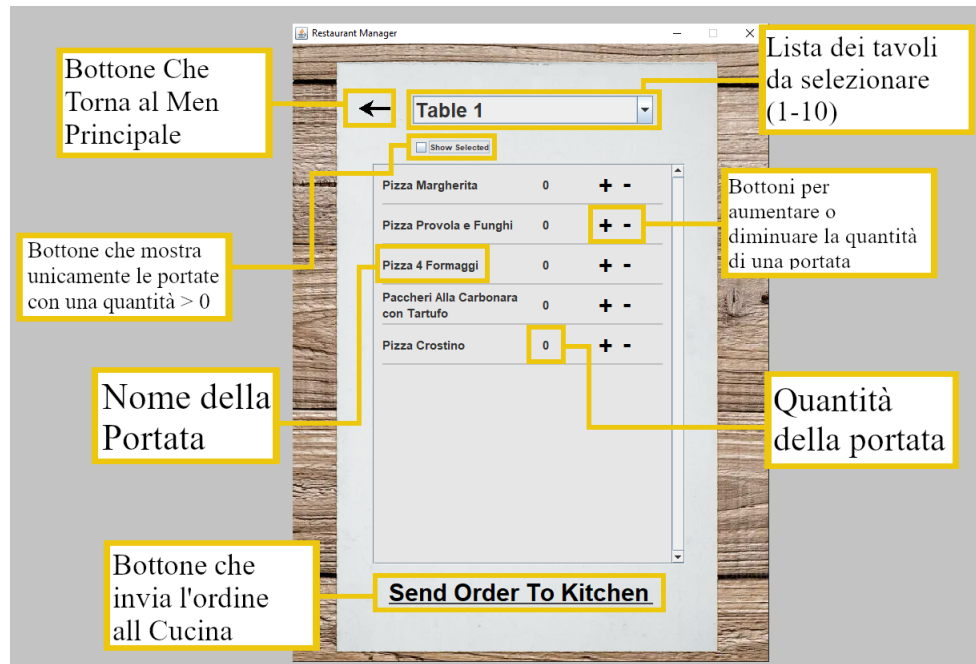
4.1 Menù Principale



Il menù principale è la schermata visualizzata all'apertura del programma. Da qui è possibile accedere alle diverse schermate del: Cameriere, Cuoco, Chef e Cassa.

Inoltre è possibile chiudere il programma tramite l'apposito bottone "Exit" posizionato in fondo.

4.2 Menù del Cameriere



La schermata del Cameriere.

Da qui è possibile selezionare un tavolo, creare un ordine ed inviarlo alla cucina tramite l'apposito bottone posizionato in fondo "Send Order To Kitchen"

4.3 Menù del Cuoco



La schermata del Cuoco.

Da qui è possibile modificare, eliminare e aggiungere portate al menù del ristorante. Cliccando con il tasto destro del mouse su una portata, è possibile aprire un menù pop-up grazie al quale si può modificare o eliminare una portata presente nel menù.

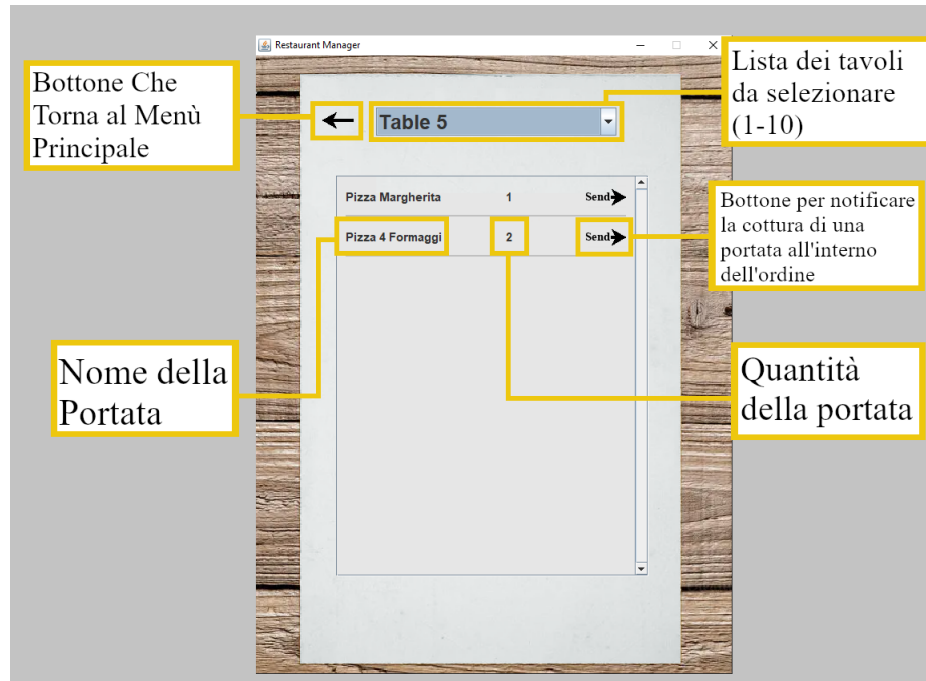
4.3.1 Dialogo di Creazione/Modifica di una portata



Questo menù, accessibile dalla schermata del cuoco, permette di modificare il

nome e il costo di una portata, o crearne una da zero. Tramite il bottone "Save", è possibile salvare la portata e le sue modifiche all'interno del menù. Tramite il bottone "Cancel", è possibile annullare le modifiche/la creazione della portata.

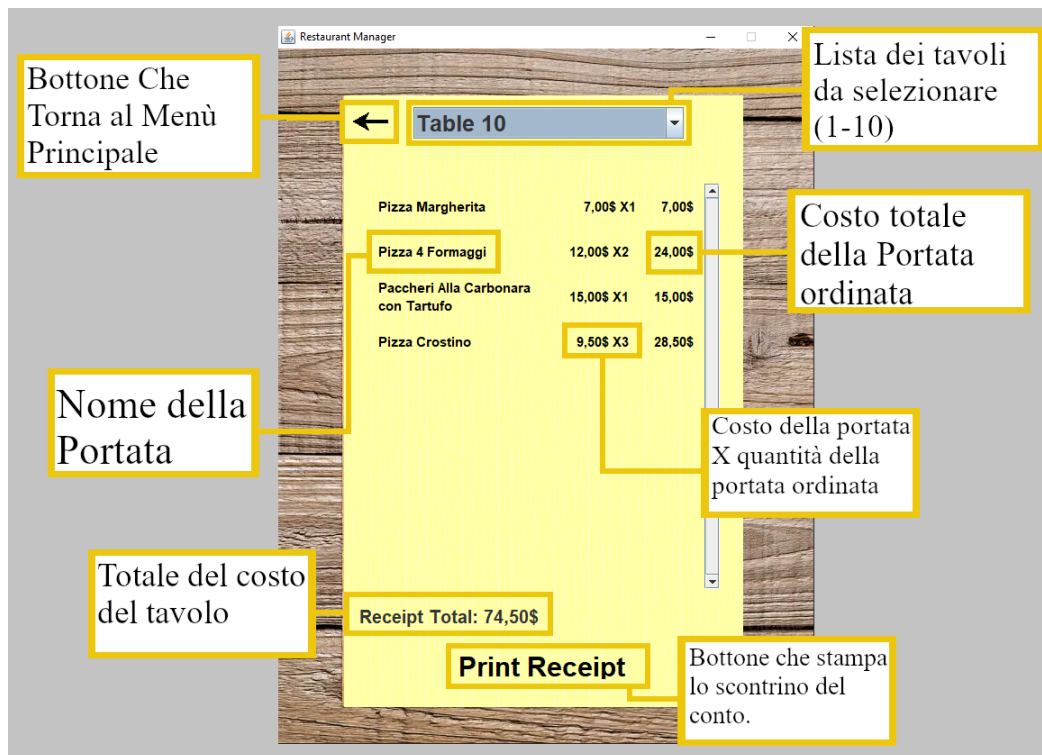
4.4 Menù della Cucina



La schermata della Cucina.

Da qui è possibile visualizzare l'ordine di un tavolo, selezionabile dalla lista posizionata in alto, e notificare la preparazione dei piatti tramite il bottone "Send" posto alla destra delle portate.

4.5 Menù della Cassa



La Schermata della Cassa.

Da qui è possibile visualizzare il conto di un tavolo, selezionabile dalla lista posizionata in alto, e stampare lo scontrino tramite il bottone "Print Receipt" in fondo alla schermata.

Quando viene stampato lo scontrino, l'ordine del tavolo viene cancellato. Dando la possibilità di riutilizzare il tavolo e creare un nuovo ordine.

5 Referenti di sviluppo

L'intero progetto è stato sviluppato da: Simone Di Cesare (1938649)