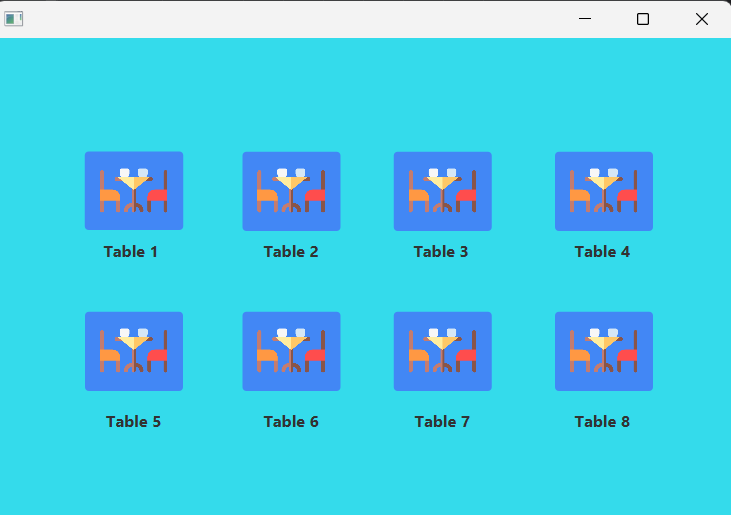
|  |  |
| --- | --- |
| **Project title** | **Restaurant interface** |
| **Author(s)** | **Muresan Ioan-Sebastian** |
| **Group** | **30424** |

# Task Description

My application represents an GUI used by waiters to make orders for the clients that come to restaurant. I did the project using JavaFX.

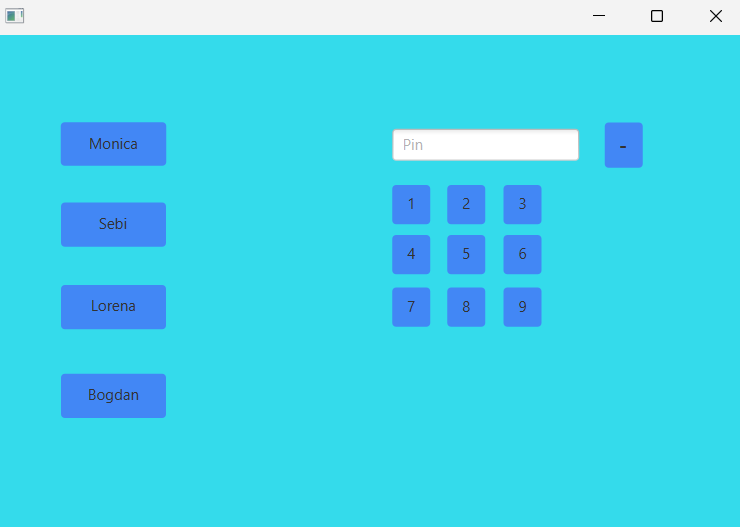
The waiter chooses the table that is free to put the order. The buttons are a combination of simple buttons and image views.



Then he should log in with a pin of 2 digits. If the pin is correct he will be redirect to the page with the menu of the restaurant. If the pin is incorrect he will be redirect towards the main page, the one with the tables.

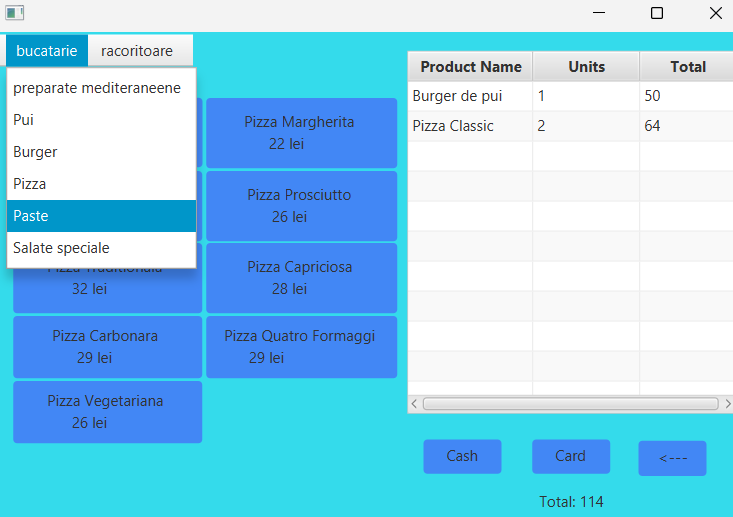
This scene contains in the left the buttons with all the waiters of the restaurant from the both shifts. Firstly the pin will be typed in then the waiter corresponding to the pin introduced.

The entire interface is made in such a way not to use a keyboard because the waiters have just a tactile screen.



The restaurant page contains the menuBar with the categories of the restaurant. Each category contains a subcategory shown below each menu of the menuBar.

For each subcategory exists a list of products which will be shown as buttons in order for this to be added to the list.



In the right is the list with all the products which have been added to that table. The arrow back button is used to return to the scene with the tables without destroying the elements of the list.

The cash and card buttons will empty the orders for that table and display a message corresponding with the method of payment chosen by the client.

If you add on a specific table an element that already exists just the units will be incremented in the database. In the right of each meal of the restaurant is the price of that meal multiplied with the quantity.

Moreover, if you double click a element the process of cancel of an element will appear.

If the units is more than 1 the units will be decremented, otherwise the element from the list will be deleted.

We have also a label that displays the total of the order of that table.

# Class Discovery

Here you should use CRC cards to find classes, responsibilities and collaborators (see Lecture 7). A sample CRC is shown below. Draw by hand and take snapshots with your camera/smartphone or edit directly in this document.

|  |  |
| --- | --- |
| **Connection\_db** | |
| Ensure the connectivity with the database used for storage the restaurant meals , the users and their pins for log-in | LoginController  Restaurant Controller |

|  |  |
| --- | --- |
| **HomeController** | |
| Is is the page with the 8 buttons representing the tables from the restaurant.  Every button which is pressed , we will be redirected towards login scene. | SharedController |

|  |  |
| --- | --- |
| **LoginController** | |
| It verify that the chosen pin is correct. Firstly, in the right the pin si typed in and then the user is chosen | SharedController |

|  |  |
| --- | --- |
| **RestaurantController** | |
| It displays the order in a friendly way such that the waiter to correctly introduce the meals. | SharedController  Product |

|  |  |
| --- | --- |
| **Product** | |
| It is a class declared to ease the displaying in the TableView the name of the product, units and the price calculated as units\*price of that meal in the menu. |  |

