**Project Documentation**

**Program:** Software Engineering

**Subject:** Visual Programming

**Worked by:** Klei Jahaj and Alban Hysaj

**Accepted by**: Eni Xhogu



**Prog. Languages and frameworks used: C# .NET**

**Project Topic: Multi-Purpose Management System**

1. **Introduction**

What’s better than collecting data? Managing them!

Our topic and goal for this project has been to create a multi-purpose management system, this system can be integrated/implemented in many types of business and be accessed by different types of users, as mentioned later in the draft.

Having had first-hand experiences in lots of businesses whose main focus is serving/selling products to individual customers, keeping track of expenses, budget and sales is the foundation of a growing business.

As seen in the next slides we will be having a closer look at the technologies used, how they’re used and how we have made everything much easier for both the end user and the any other entities who will manage or use our program.

Please note that this is an ongoing project where we aim to implement this in a real-life scenario, meaning that further changes may be necessary based on the way the business in question is run.

1. Description

Through our program we’ve made possible for lots of people to have use of the management system, starting from the Admin which is responsible and able to have an overview on everything in real time.

Other users can be the Finance Department and the most common type of user, a waiter/server.

Using the benefits of object-oriented programming we’ve made the exchange of data much simpler, where with a click of a button data can be added or retrieved to a remote, secure database.

To summarize in fewer words, our program is fast, secure, user-friendly and built taking any and every scenario in mind, thus having a solution to every occurring problem.

1. Methodology

On of our first tasks was to create a connection between our program and our database

For that we have used:

* MySQL
* MySQL Workbench 8.0 CE
* Xampp Control Panel
* MySql Entity Framework Core

IDE:

* Visual Studio 2022

Source control:

* GitHub

Packages:

* SQL Database Package
* BarCode.Slim Package

Frameworks

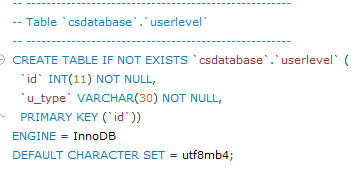
* C# .NET

1. Program Overview

Below we will provide some screenshots of the basis of our program, database, frontend and backend.

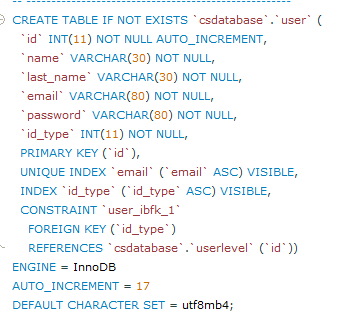
Starting with the database, up until now we have two tables and a helper table.

Starting with the helper table: UserLevel table:

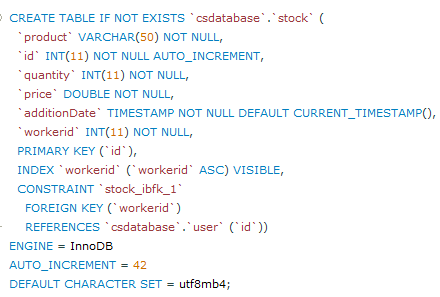


This table only contains an id and a u\_type attribute which contains the type of users which will be determinant in user login.

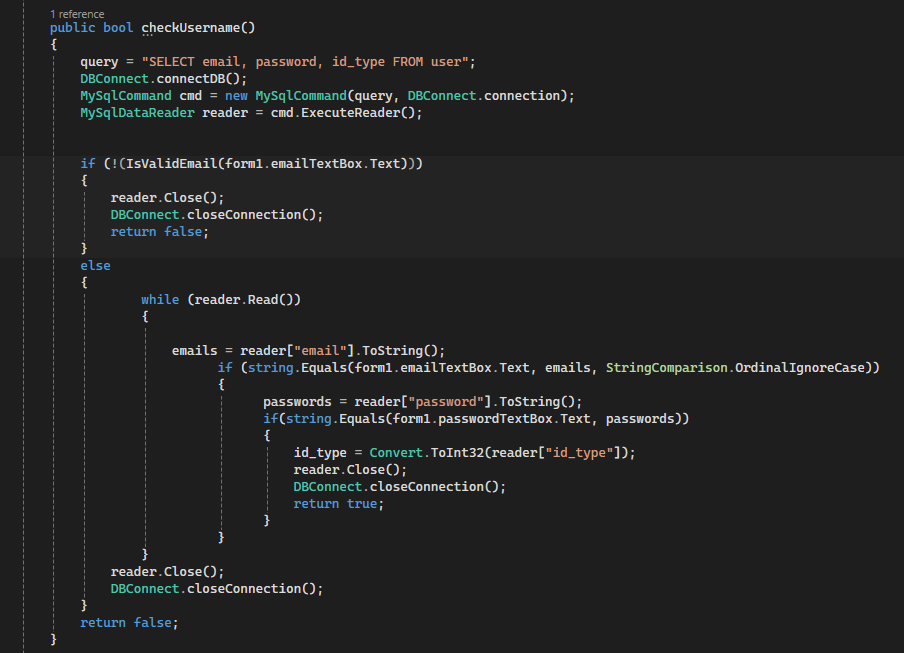
The user table, containing the following attributes: id, name, last\_name, email, password, u\_level (foreign key from the above table)



The last table is the stock table which stores all the items possessed

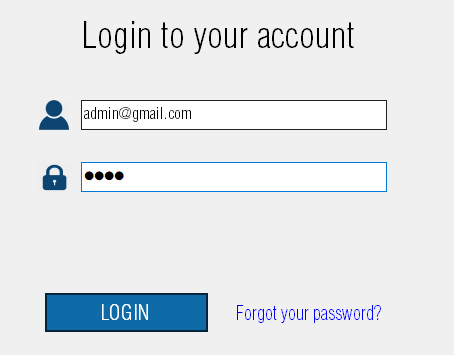


This table contains the date of creation and the id of the user that inserted in the stock.



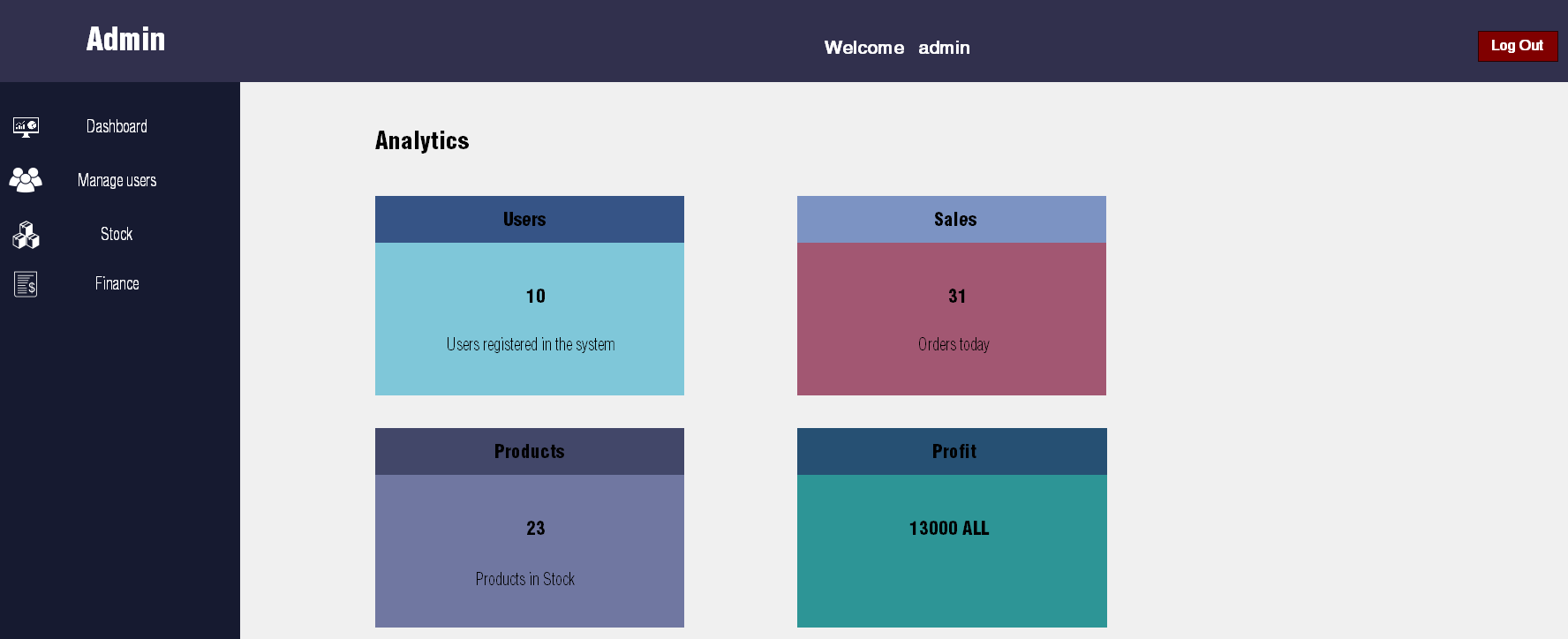
The function CheckUsername() validates if the input entered in the login screen matches an existing user in the database, based on the user type a specific form is loaded with the contents that this specific user type would expect to see.

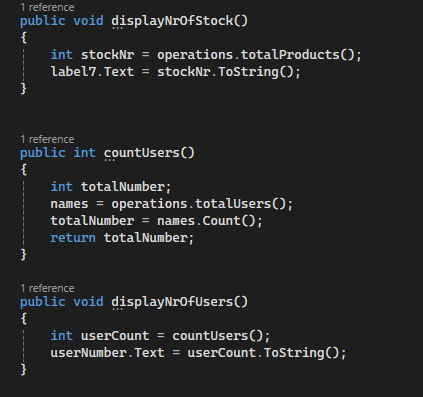
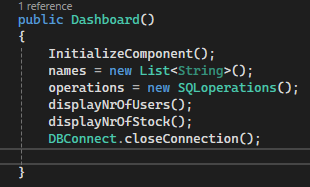
Here checkUsername() is called and the email and password is validated. As we can see in this snipped we also have the “Forgot your password” option.



The following snipped will show the admin dashboard which is a **User Control**:

*(From this point onwards every page with a different use will be a* ***User Control)***

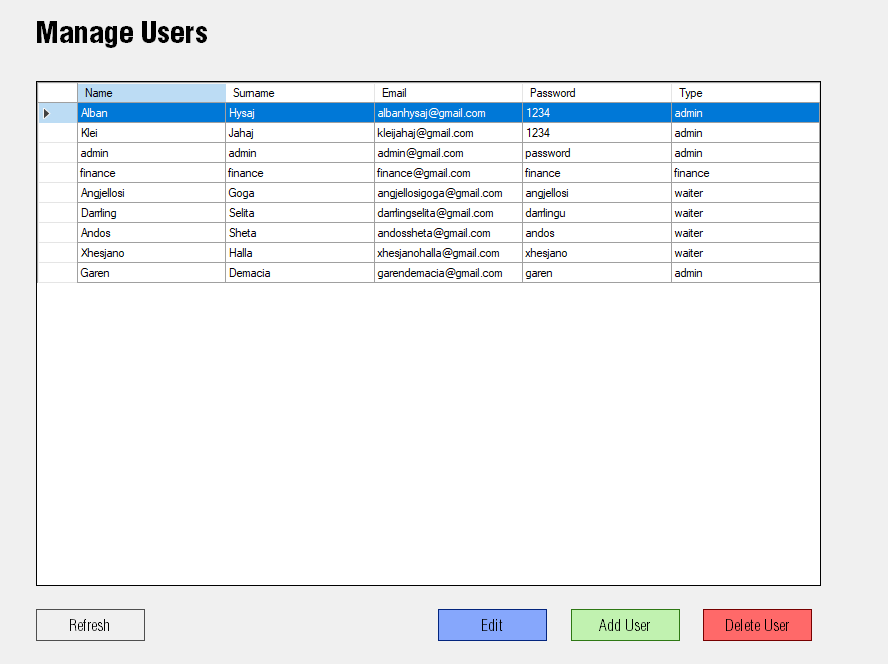


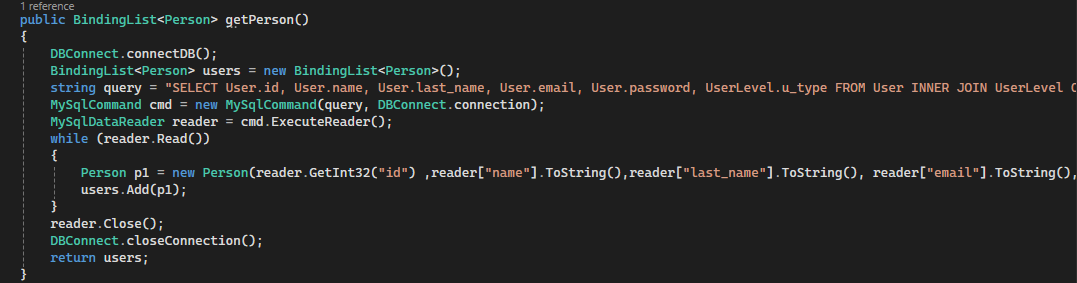


As we can see in the above snippet we have created an overview that displays the most important information an admin wants to see when he logs in. As well as implemented a **Log Out** button which sends you to the previous log in page.

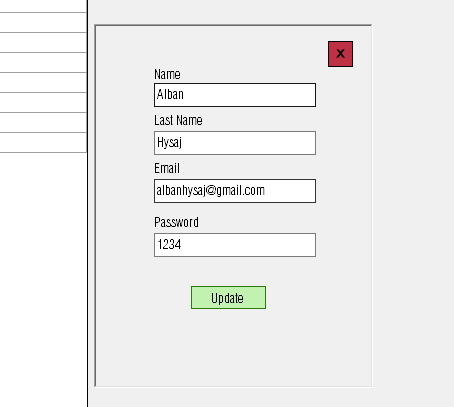
The below snippet is the **Manage Users**  user control. We have made use of the datagrid view

tool which stores all the information from the users. As we can see there are 4 different buttons. The **Reload** button simply reloads the data in the table to obtain the latest changes, meanwhile the **DeleteUser** button removes the selected user from the database

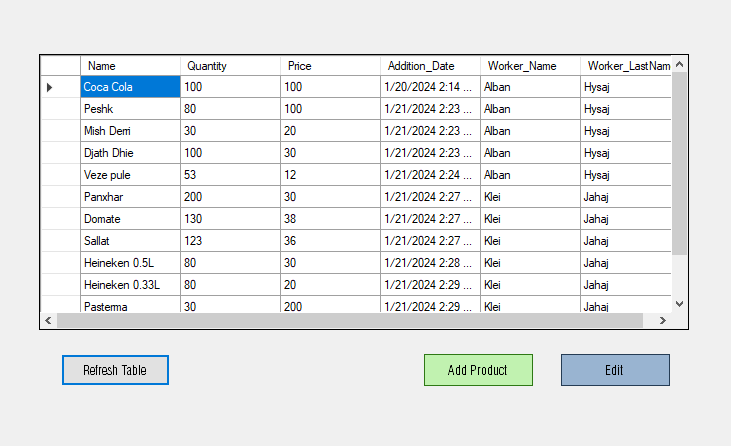




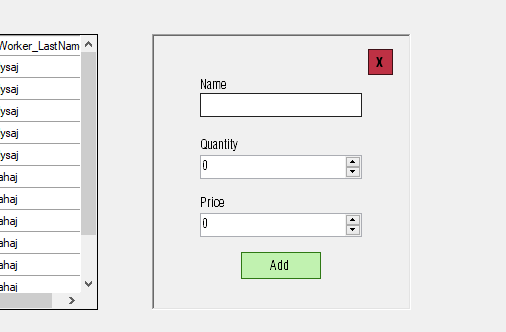
The **Edit** button works as follows. A popup is shown and the user should prompt the changes that he wants to take place, then he just presses **Update** and everything is updated real time.

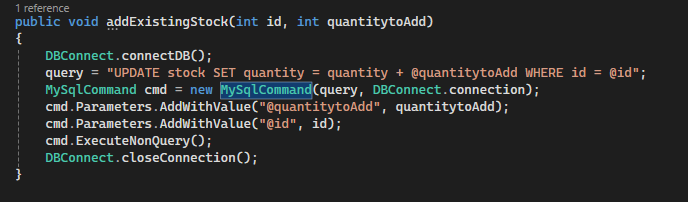


The **Stock** user control displays the stock in a similar manner as the **User** user control:

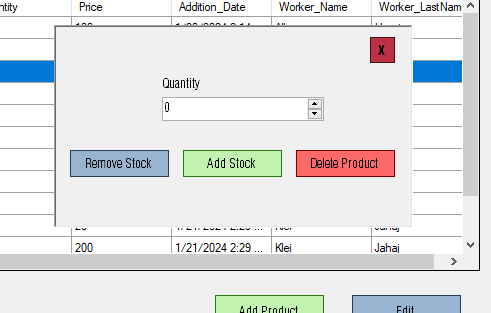
**Reload** button serves the same purpose as above.

The add product shown in the following snippet shows a popup where the user is asked for the name of the product, quantity and the price, Date is added automatically as well as the user who added the product.





The edit button creates a popup containing other options such as: removing it from the table, adding more quantity, or removing the quantity.

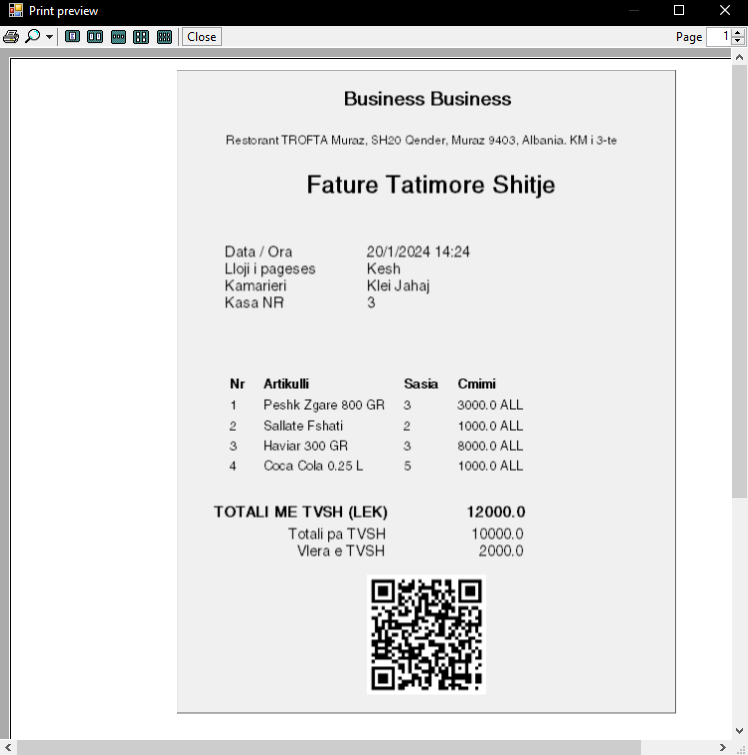


This is the finance tab:  
*(NOTE: This part is not a final product considering constrains from our stakeholders)*

In this tab are all the receipts created by the ”Waiter” user. It shows basic information about the receipt and once clicked it displays the whole receipt



The print button takes you to the printing settings and options:



1. Conclusion

In conclusion this are the main functionalities of our project. The project is open for changes to any specific scenario or business requests.

Thank you very much for your attention!