*OOP Teamwork –*

*Project documentation*

Team: “Aventurine”

# Team members list

1. Станислав Илиев (Flystar)
2. Цонко Генов (tsonko\_genov)
3. Спас Христов (bgpa4o)
4. Тодор Маринов (todmarinov)
5. Светослав Бозов(svetoslavbozov)

Project

Restaurant Aventurine

Restaurant “Aventurine” is an object-oriented business application – restaurant menu, created to run on WPF project type. It shows how with C# language, object-oriented programming and XAML (UI) can be created a restaurant menu with colors, text, selection options, sliding options, images, database logic (input and output for menu orders).

# *Introducing the application*

Restaurant “Aventurine” simulates a restaurant application menu, where the user (waiter/waitress) can create a client order from the restaurant menu with prices and quantity of the items, check for available free tables and items from the menu. The main window consists of three tabs and several buttons. The access to the Main Window menu gives a Login window, which requires the user username and password (in this case, the restaurant personal access validation). Usage can be simple, for checking if a table is free or reserved, if a menu dish of given type is still orderable or just check item image to give the client information about it. At the right corner of the Main Window menu is a clock and when adding a new order gives information about the order timeline, and also gives the restaurant owner the ability to check when a personal is working.

The main window (tab Orders) holds a datagrid with orders, which are implemented through the database in a Microsoft Access file(.mdb), and contains information about the order number, table, date, status of order, client(optional implementation), and personal executing the order. The second tab (Tables) gives a table view for checking and reserve/free a table, and button for checking price and quantity of item. The last tab (Menu Items) shows an image view of the menu order items, providing personal with information about the Menu items (what can the personal offer the client at the moment as types of items). The visualization has a green type of color with animation as a background.

# *Buttons and second windows*

## *Show Order*

Button for implementation to call the selected order.

## *New Order*

Button that calls a second window – “NewOrder”, which holds operation buttons and fields(text) for table number, category of item, item, quantity, price, person, discount, Add Order(add selection to order database), Add Item (adds selected item from menu category), Cancel Item (cancels selected item), Cancel(cancels the current window and brings user back to Main Window)**.** In the upper right corner is a calendar button function, showing the date of the order (for further implementation to make orders at a different date (type of reservation)).

## *Find Now*

Button for implementation to find a specific order from the database**.**

## *Clear*

Button for implementation to clear an order from the database.

## *Log Off*

Button which logs off the user from the window.

# *Code structure*

## Restaurant

Main folder, holds the StructNameStruct.cs, Restaurant.cs, Corporate.cs, Table.cs, Zone.cs, with Name, Address, Zones, Personnel and Reservation fields, and folders for Items (types of dishes, drinks, desserts classes and enumerations), Menu (Menu and Speciality classes), Orders (classes for type of orders), People (inheriting Person class for Clients and Personnel classes types) and Services( classes for different types of services).

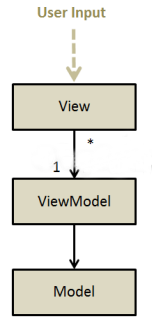
## Database

Folder that contains the Login.mdb, Menu.mdb and Orders.mdb database files for data connection and usage.

## Images

Folder that contains the image files used in the project.

## UI

Folder that holds the Login.xaml and .cs; CheckItem.xaml and .cs; NewOrder.xaml and .cs; CustomException.cs, MenuCategory.cs and ViewModelbase.cs which holds the MVVM Design Pattern usage:  


which connects CheckItem button to a set of options:

* Opening CheckItem window
* Pulling the data from Dessert class
* Choosing type of Desserts in first Combobox
* Bind selected data to second Combobox
* Get data options like price and quantity

The class ViewModelbase.cs holds the MVVM logic and MenuCategory.cs implements the model, with an EventHandler .

The Login.xamland Login.csholds the user login access usage and implement connection to the database.

## MainWindow

Main Window consists of MainWindow.xaml and MainWindow.cs and holds the base binding of all other buttons/windows and classes implementation. In the MainWindow.xaml are used WPF controls like:

Canvas, StackPanel, Grid, DataGrid, Combobox, ToggleButton and others,   
Data Binding for connecting the xaml with the MainWindow.cs content as resource, different text and colors usage.

The MainWindow.cs file holds the database connection string implementation, also user Login access, data reading, Exception class usage, image selection and initializes the components in the xaml file.

Aditional usage:

## Clock

Method for implementing a time function for the MainWindow, usage is as in the explanation above in the introduction.

# *SVN Repository*

*URL*:<https://code.google.com/p/store-menu/source/list>

*Username*: Terminja@gmail.com

***Additional information:***  
*For more information about the scripts and UI visualization please check source code folder and PowerPoint presentation demo in the added archive.*

**Note:**  *Thisdocumentationreflectsthecurrentstate of theRestaurant Aventurinemenu sampleatthetime of writing.*