Assignment 3

Sebastian Kaczmarczyk

# User interface prototype

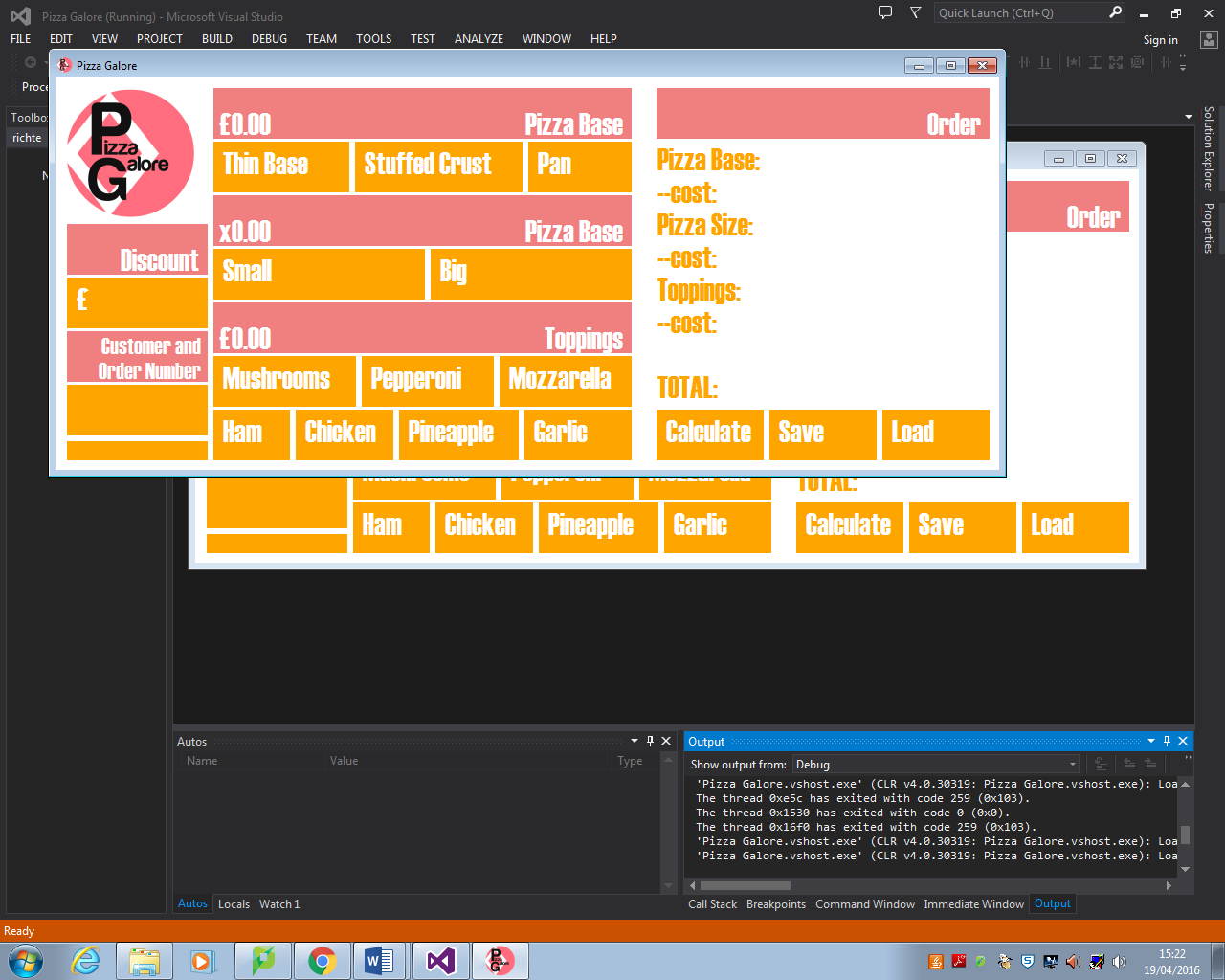
### USER INTERFACE (FORM1)

LABEL

BUTTONS

LABELS

PICTURE



LABELS

BUTTONS

BUTTONS

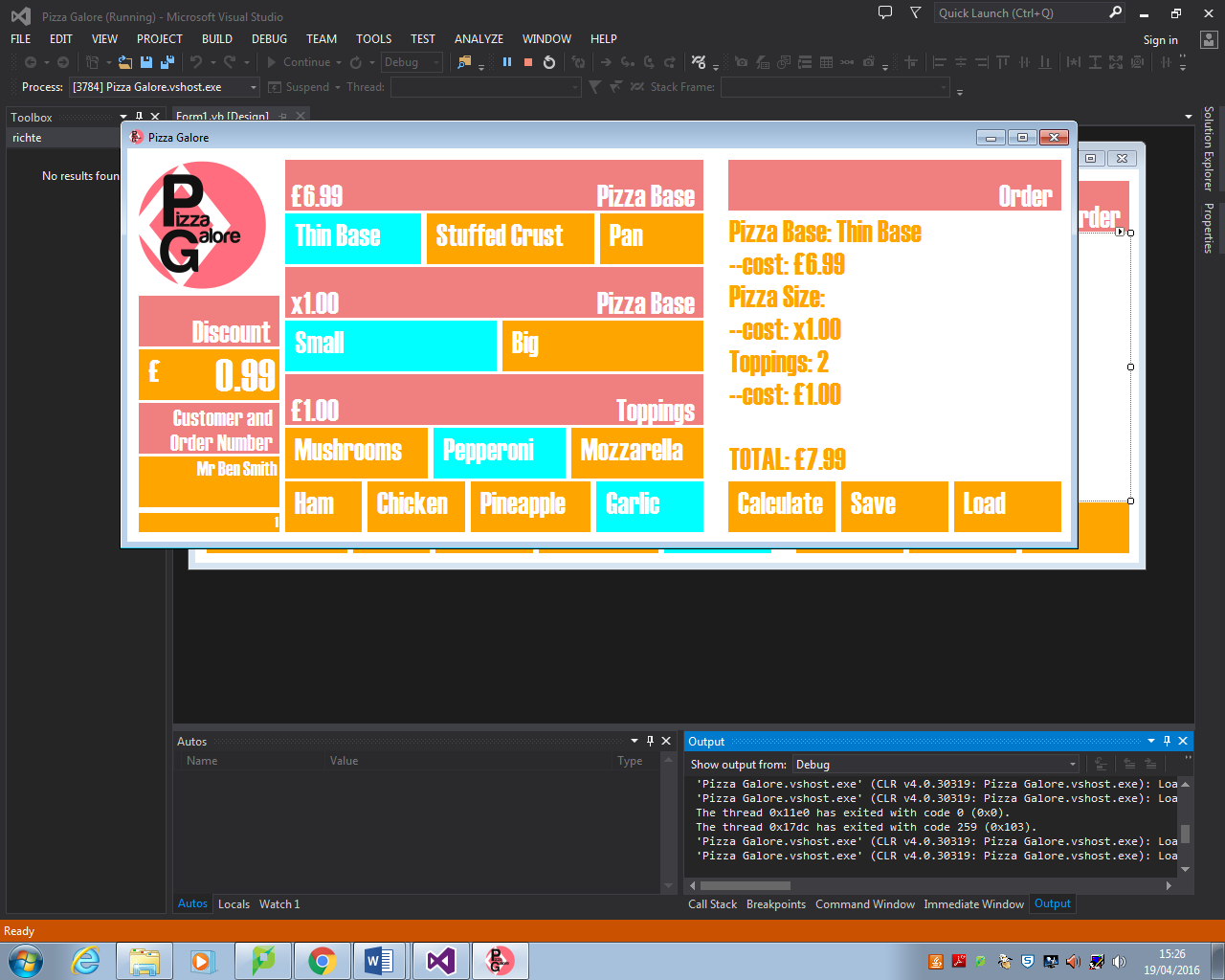
TEXTBOX

BUTTON

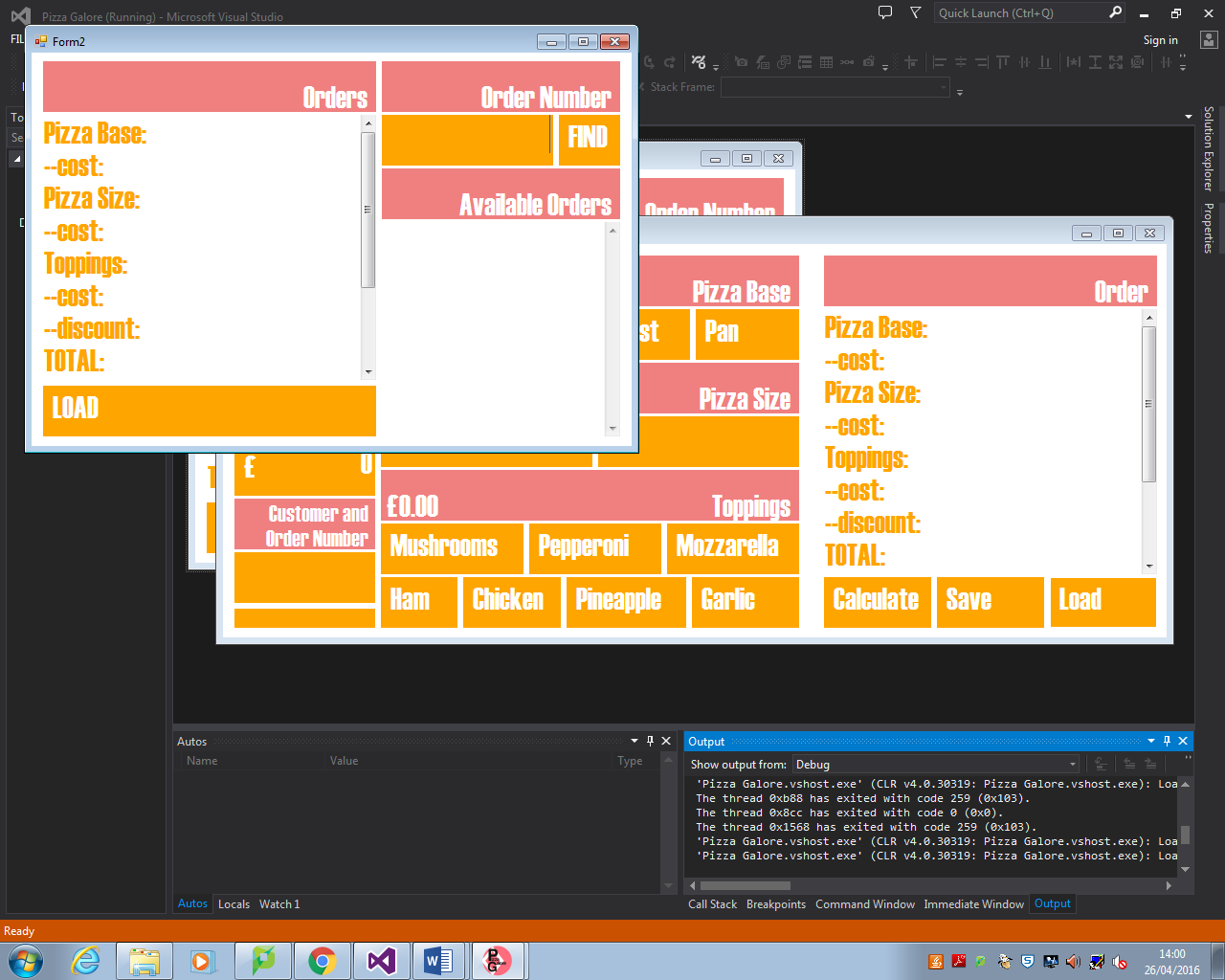
TEXTBOXES

LABELS

### USER INTERFACE WITH INPUTS AND OUTPUTS (FORM1)



### USER INTERFACE (FORM2)



TEXTBOX

BUTTON

LABELS

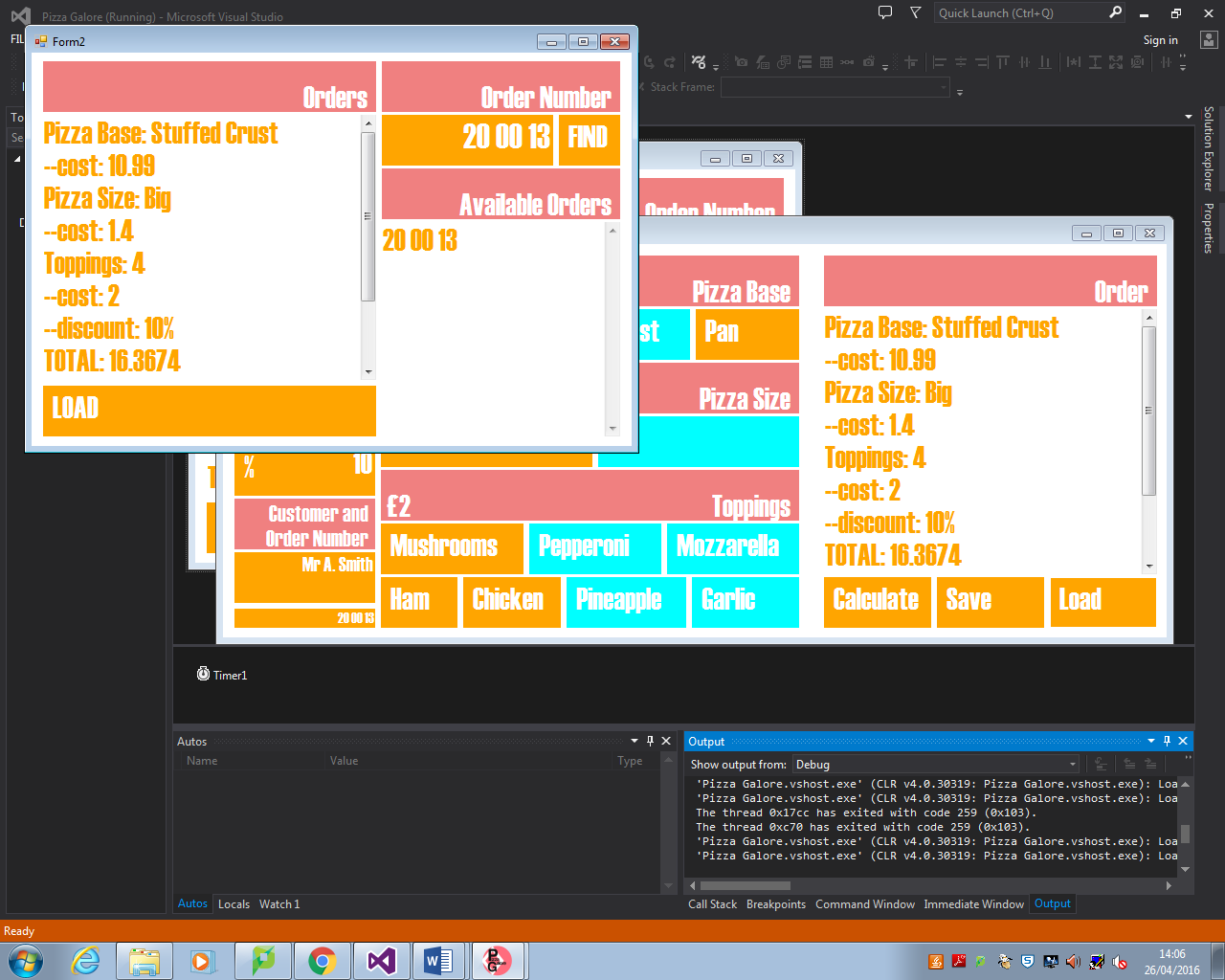
TEXTBOX

BUTTON

TEXTBOX

LABEL

### USER INTEFACE WITH INPUTS AND OUTPUTS (FORM2)



# Design Justification

I have designed the user interface for both of the forms in a specific way ensuring the program to be readable. Human eye is easily exhausted when affected by the movement or many different colours. To make sure that the program will not exhaust its users’ eyes I have decided to keep a colour palette very simple and to use only four colours; white, orange, coral (light red) and aqua (light intensified blue). I have selected three of them to be easily seen on the white background. Because orange and coral were intended to be put close together, these two colours were chosen to be similar to get other, and so they lie close on the colour palette, but not to blur / mend together, so users will easily see the difference. These hot, live colours make users to feel in a specific way. It makes them think that the program is easy to use. On the other hand I also used aqua, which is not a hot, but still a live colour. It is because it has to be easily adapted by the users’ eyes as different. The reason for that is this colour will show only when some choice is made and to show contrast between that choice and others available.

Program has been cut into two separate parts, so show a difference in functioning for both forms. First one is calculating, while second is managing information.

# DATA DICTIONARY

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NAME | DATA TYPE | VARIABLE OR CONSTANT | USE | EXPECTED VALUES | JUSTIFICATION |
| Val\_BASE | String | Variable | Stores pizza base, which is used in order information. | Thin Base / Stuffed Crust / Pan | This data is a variable string, because it has to store text, which will be continuously changed while using the program. |
| Val\_BASEcost | Decimal | Variable | Stores cost for specific pizza base, used in calculating total cost. | 6.99 / 10.99 / 8.99 | This data is a variable decimal, because it has to store one from three numbers with two decimal places, representing cost for pizza base. It is a variable, because user has to be able to change this value. |
| Val\_SIZE | String | Variable | Stores pizza size, which is used in order information. | Small / Big | This data is a variable string, because it has to store text. It could be a Boolean, as there are only two possible values, but it might make the code longer, what is not necessary. |
| Val\_SIZEcost | Decimal | Variable | Stores cost for specific pizza size, used in calculating total cost. | 1 / 1.4 | This data is a variable decimal, because it has to store one of two values, where one of them is a decimal. |
| Val\_TOPPINGS | Integer | Variable | Stores number of toppings, used in order information. | 2 to 7 | This data is a variable integer, because it has to store only whole numbers. It is not a constant, because number of toppings will depend on order, what cannot be predicted. |
| Val\_TOPPINGScost | Decimal | Variable | Stores total cost of toppings, used in calculating total cost. | 1 to 3.5 | This data is a variable decimal, because it must store values between 1 and 3.5, where some of them are decimals. It is a variable depending on the number of toppings. |
| Val\_DISCOUNT | Boolean | Variable | Used to check whether discount applies or not to the calculations. | True / False | This data is a variable Boolean, because it is best way show whether something is true or not, what was exactly needed for task for checking whether discount apply or not. |
| Val\_DICSOUNTcost | Decimal | Variable | Stores value of discount, which is used in calculating total cost. | Any positive number. E.g. 10.5 | This data is a variable decimal, because it has to store any number put by the user. |
| Val\_CUSTOMER | String | Variable | Stores information about customer, which is used in order information. | Any text with up to 100 characters. E.g. Mr A. Smith | This data is a variable string because it has to store any text put by the user. |
| Val\_ORDERNUMBER | String | Variable | Stores order number, which is used in order information but also to save or load order information. | Any positive number. E.g. 20 00 16 | This data is a variable string because it has to store any text but by the user. It is a string instead of long or integer, because some companies put letters among numbers for order numbers. |
| Val\_TOTAL | Decimal | Variable | Stores total cost. | Any positive number. E.G. 20 | This data is a variable decimal, because it is calculated my multiplying and adding numbers, including decimals. |
| Val\_VAT | Decimal. | Constant | Stores value of VAT, which is used in calculating total cost. | 1.2 | This data is a constant decimal, because its value will never change and is equal to 1.2. |

# STRUCTURE DIAGRAM

\*

# FORM PROPERTIES

### Form1

|  |  |  |
| --- | --- | --- |
| OBJECT | PROPERTY | SETTING |
| Pct\_logo | Image |  |
| Lbl\_base | Text | Pizza Base |
| Lbl\_cost\_base | Text | £0.00 |
| Btn\_base\_thin | Text | Thin Base |
| Btn\_base\_stuffed | Text | Stuffed Crust |
| Btn\_base\_pan | Text | Pan |
| Lbl\_size | Text | Pizza Size |
| Lbl\_cost\_size | Text | X0.00 |
| Btn\_size\_big | Text | Big |
| Btn\_size\_small | Text | Small |
| Lbl\_toppings | Text | Toppings |
| Lbl\_cost\_toppings | Text | £0.00 |
| Btn\_mushrooms | Text | Mushrooms |
| Btn\_pepperoni | Text | Pepperoni |
| Btn\_mozzarella | Text | Mozzarella |
| Btn\_ham | Text | Ham |
| Btn\_chicken | Text | Chicken |
| Btn\_pineapple | Text | Pineapple |
| Btn\_garlic | Text | Garlic |
| Lbl\_discount | Text | Discount |
| Btn\_discount | Text | £ |
| Txt\_discount | Text | 0 |
| Lbl\_customer | Text | Customer and Order Number |
| Txt\_customer | Text |  |
| Txt\_ordernumber | Text |  |
| Lbl\_order | Text | Order |
| Txt\_calculate | Text | Pizza Base:  --cost:  Pizza Size:  --cost:  Toppings:  --cost:  --discount:  TOTAL:  Customer  Order Number: |
| Btn\_calculate | Text | Calculate |
| Btn\_save | Text | Save |
| Btn\_load | Text | Load |

### Form2

|  |  |  |
| --- | --- | --- |
| OBJECT | PROPERTY | SETTING |
| Lbl\_orders | Text | Orders |
| Txt\_order | Text | Pizza Base:  --cost:  Pizza Size:  --cost:  Toppings:  --cost:  --discount:  TOTAL:  Customer  Order Number: |
| Btn\_load | Text | Load |
| Lbl\_ordernumber | Text | Order Number |
| Txt\_number | Text |  |
| Btn\_find | Text | Find |
| Lbl\_available | Text | Available Orders |
| Txt\_available | Text |  |

# Pseudo Code

## Start program.

## Open Form1.

Ask user to select Pizza Base.

Store val\_BASE as Thin and val\_BASEcost as 6.99 if btn\_base\_thin was clicked.

Show result as lbl\_cost\_base text.

Store val\_BASE as Stuffed Crust and val\_BASEcost as 10.99 if btn\_base\_stuffed was clicked.

Show result as lbl\_cost\_base text.

Store val\_BASE as Pan and val\_BASEcost as 8.99 if btn\_base\_pan was clicked.

Show result as lbl\_cost\_base text.

Ask user to select Pizza Size.

Store val\_SIZE as Small and val\_SIZEcost as 1 if btn\_size\_small was clicked.

Show result as lbl\_cost\_size text.

Store val\_SIZE as Big and val\_SIZEcost as 1.4 if btn\_size\_big was clicked.

Show result as lbl\_cost\_size text.

Ask user to select Toppings.

For btn\_mushrooms, btn\_pepperoni, btn\_mozzarella, btn\_ham, btn\_chicken, btn\_pineapple, btn\_garlic:

Add 1 to val\_TOPPINGS and 0.5 to val\_TOPPINGScost if button was clicked once.

Subtract 1 from val\_TOPPINGS and 0.5 from val\_TOPPINGScost if button was clicked twice or not clicked.

Show btn\_calculate if val\_TOPPINGS is greater than 1.

Hide btn\_calculate if val\_TOPPINGS is smaller than 2.

Store val\_TOPPINGS and val\_TOPPINGScost.

Show result as lbl\_cost\_toppings text.

Ask user to select Discount.

Store number written in txt\_discount as val\_DISCOUNTcost.

Store discount type selected by clicking on btn\_discount.

Store results.

Ask user to select Customer Name.

Store text written in txt\_customer as val\_CUSTOMER.

Ask user to select Order Number.

Store text written in txt\_ordernumber as val\_ORDERNUMBER.

Calculate Total Cost.

Add val\_BASEcost to val\_TOPPINGS cost and multiply that number by val\_SIZEcost. Subtract val\_DISCOUNTcost from received number and then multiply it by val\_VAT.

Show result as text in txt\_calculate.

Ask user to save results.

Store results.

Ask user to load results.

## Open Form2.

Show available results as text in txt\_available.

Ask user for Order Number.

As a text in txt\_number.

Show specific order.

After clicking on btn\_find show order with a Order Number written in txt\_number, as text in txt\_load.

Ask user to load results.

After clicking on btn\_load change text written in txt\_calculate to the text in txt\_order.

## Close program