**AS 91896: Use advanced programming techniques to develop a computer program.**

**Project Outline**

I have been asked to plan, create and test an advanced computer program for Rotorua Dream Pizza. Rotorua Dream Pizza are a local pizza takeaway business that wish to computerise their phone ordering system. Specifically, they want to be able to enter customer details, pizza(s) ordered and pick-up or delivery requirements into a computer and have it display the delivery details, itemised order, and total cost. Phone orders generally consist of several kinds of pizza.

The program must meet the following **specifications**:

* The program contains options for the phone operator to specify whether the pizza order is for pickup or delivery.
* If the order is for delivery:
  + the program should collect the customer’s name, address and phone number
  + a $3 delivery charge should be added to the total cost
* If the order is for pick up:
  + the program should ask the phone operator to enter the customer’s name and phone number
* The program should allow the phone operator to input how many pizzas the customer would like (maximum 5 for any pizza).
* A menu of at least 12 pizza names should be presented to the phone operator.
* Each pizza to be ordered should be selected from the choices available on the menu and the order information should be stored.
* The cost of the first seven (regular) pizzas on the menu is $8.50 and the rest are $5 more as they are gourmet pizzas.
* When the order is finished:
  + the names of ordered pizzas and their individual prices should be displayed
  + the total cost of the order, including any delivery charge, should be displayed
  + customer name and phone number should be displayed
  + if the pizza is for delivery the address should be displayed.
* The program should allow the operator to cancel the order.
* After the order summary information has been displayed the program should be ready to accept another order or exit.

Research

* + Not Pizza flavours (Trivial)
  + Alternatives
    - Paper
    - Database
    - Code
    - website

Informed choice – discuss and make choice

Potential Implications (define and explain which implications may be important)

Planning

* Trello
* Flowchart (program flow)
* Drawings of Form layouts
* Id of variables/constants

Development: (Iteration)

* Prove you developed the code in stages – match Trello to code
* Trialling – 2 methods that work, chose one and justify - repeat
* Testing (formative)

End Product

* Testing (Summative)
* Evaluation
  + Of the product
    - Does it do what it should? Fit for Purpose
    - Is it easy to use (Suitable for audience)
  + Your performance
* Revisit implications (Which were important and what did you do?)
* Code Control (Version control- Git/GitHub)
* Code listing
* Digital copy of code.

Research

We can accomplish the assigned task in multiple ways such as recording the data on paper, using a existing database, coding it from the ground up and using a website. I have decided to do it with code from the ground up for the following reasons:

* If we decided to just record the orders on paper, it is the simplest option but it is also the option with the most flaws. First off, whoever operates the phone has to have very neat handwriting and also has to be able to write via hand very quickly, and considering the average hand writing speed is 30wpm, it would be hard to keep up with the client on the phone. Also after the order has been completed, disposing of the paper may be difficult as it has someone’s full information on it. The company would also have to spend money on pens and paper, both of which are actively hurting the environment with their production.
* If we decided to record it using a database, it could become a issue when the end user does not have a internet connection when the phone call is received. This would be a pretty niche scenario because the phone would still be active, but it is possible if the computer itself just could not take the internet connection. It would also mean that the clients data would be stored after they ordered food, which could have some ethical problems in the age where data is worth more than gold to many companies.
* If I decided to use a website the same problems from the database would be replicated as well as the fact that the website would cost $15 a year minimum for the domain name assuming you don’t already have one, most likely more than $15 however as a good domain name can sell for much more. Also websites rely on the domain that is hosting the site on the servers, and that means that downtimes are out of the companies control and that could be a issue.

I decided to use code because we can customize the program as much as we want, like a website, but we also have the ability to run the program offline so that data isn’t stored and so the ordering system can still function when the internet is down. Furthermore, human input will always be the same as a keyboard produces set characters in a font compared to handwriting which may be hard to read. It is completely free of charge as the coding software

Potential Implications

End User:  
  
*What specific needs does your end user have?* This implication regards the end user experience in my program including the visual interface, ease of access, bug fixes, easy navigation and more. This implication should be the central focus of any good program and every good developer should strive for a good end user experience. This will be a issue in my program as I will have to work towards a very clean and easy to understand interface for the average user to pick up and understand. I can achieve this by clearly labeling all of my inputs/outputs, limiting user inputs where possible to prevent control issues and or bugs, making the program run in a sensible order so that the end user could intuitively use it without knowing the program and more. This will be one of the implications I will be looking into the most as it basically affects the entire program and will affect every design decision. I will also have to clarify the computer they must use, such as the minimum operating system and the screen resolution required (1920x1080)

# Functionality:

*Does it work well?* This implication regards how the program runs in terms of speed, bug control and device usage (device usage meaning it can be used on many different operating systems on different screen ratios). This implication will be more related to optimizing my program and eliminating as much excess code and doing things with the least amount of lines possible to prevent potential lag and load times on slower computers and or mobile devices. It will also be about eliminating as many bugs and useless features as possible to both keep the quality control good. This will most likely come down to eliminating as much user input as possible (e.g., using dropdown boxes rather than text inputs for as many as possible) and eliminating as many fault points as possible by eliminating features that the program does not necessarily need. I have also decided to make the buttons and use interface large so that it is touch screen friendly.

# Usability:

*Is it easy to use?* This implication regards how usable the program is to use for the user. It concerns about if the user can correct mistakes they may have made earlier in the program runtime and if they can transfer skills from other similar programs to this program easily. I can make my program usable by limiting the number of inputs that the end user inputs in order to make it more seamless and using a lot of the features from the functionality standpoint. I can also make buttons big with big fonts in order to make vision easier for the end user.

# Privacy

*Is data kept private?* This implication regards how data is kept private to the end user and or the computer it is used on. I can make my program very privacy friendly by making it in code and making the code delete all temporary files upon its closing, such as the receipt file which will be printed, the names and address of the client and whatever else is inputted into the program. This is why I chose to do this project with code, as it is extremely easy to control what happens with the program and what gets saved and what doesn’t get saved. I decided that when the program loops back to start for another order it will be factory reset.

# Continues Next Page -

Ordering a pizza – flow chart

START

Collect Customer Details

END

New Order?

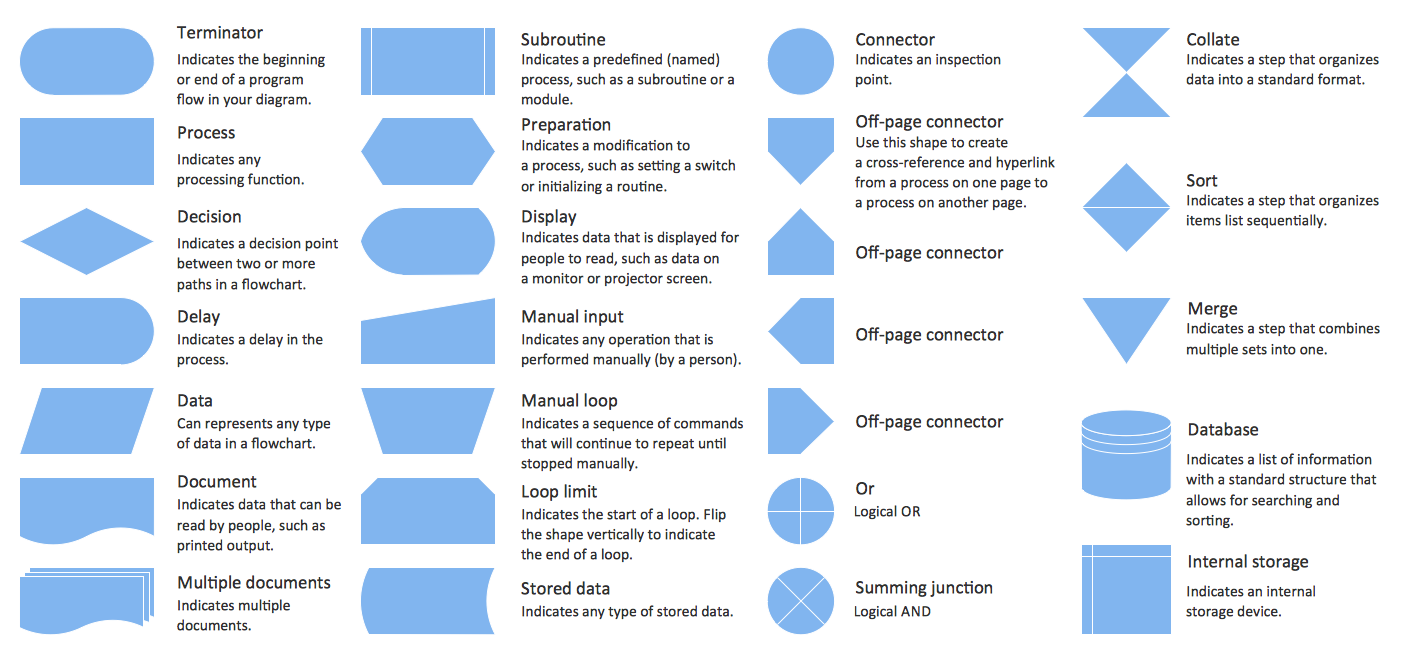
Summary of Order

Order Details

Get Delivery Address

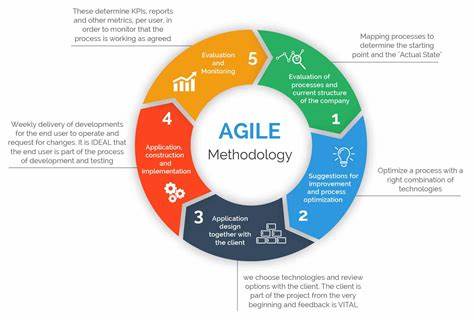
Collection Or Delivery?

This flowchart demonstrates how my program should function. It may change later into the process as it was developed very early into the development stages.

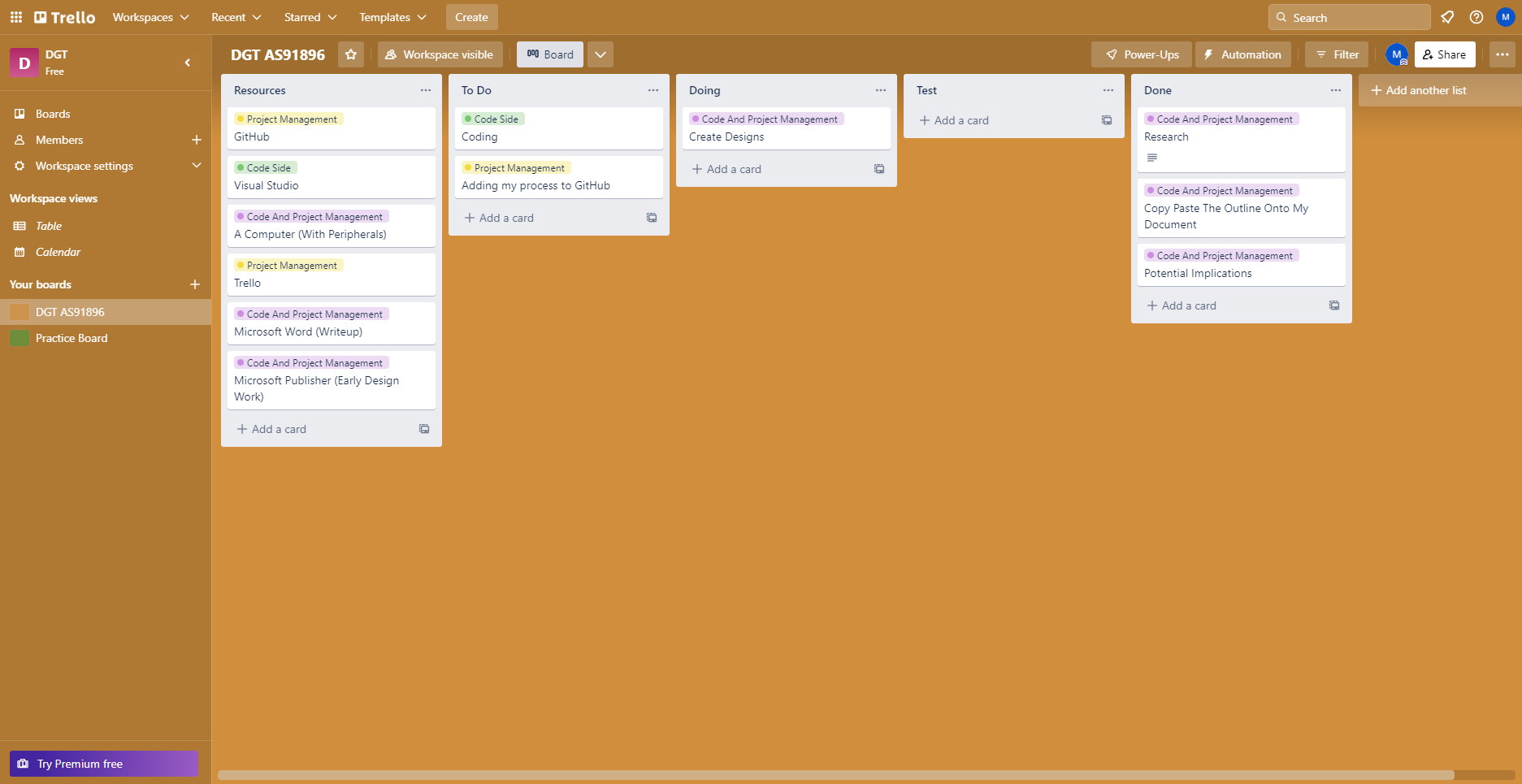


Planning:

I have decided to use an AGILE methodology as my project management tool in the form of Trello. Trello is an implementation of a Kanban board, with the purpose being to deliver working software as soon as possible compared to other methodologies. It is not normally used to create a fully working product and it is a program that helps create a series of iterations with increased functionality. I will use it to break down my task into smaller parts, and then assemble a final product. I will also be using GitHub as my form of Git to store old versions of my code and potentially need to restore to a previous version just in case things go wrong. Using GitHub rather than saving files locally is better because if the computer being used breaks down, then you can still access the code. It is also good practice to use GitHub so if you work with other people in the future, you will have a much easier time working together as GitHub is best used with other developers to make a large project. It is also easier because the end user can create feature requests and pull requests much easier than if the file was saved locally, where the end user would have to contact the developer for a change to occur. I have also chosen to use Visual BASIC as my coding language, as it is what I was taught in my class time, and I am not confident enough in another language to be able to code for an assessment in it easily, especially considering that this assessment will require a lot of bug fixing and troubleshooting, both of which are easier with a language you are more comfortable on.



In Trello, I will use a 5-column approach to help me visualise my tasks and break them down into easier sections

[DGT AS91896 | Trello](https://trello.com/b/NRPDvWka/dgt-as91896) My Trello Board  
  
Early Version of my Trello board  


Constants And Variables

For my project I will likely use these following variables:  
  
Const REGULARPRICE as Decimal = 8.50

Const GOURMETPRICE as Decimal = REGULARPRICE + PREMIUM

Const PREMIUM as Decimal = 5

Dim CustomerDetails as Array

Planned Iterations

I will create my design following these planned iterations, it will also be pushed to github in this order

1. Created Form
2. Stopped typing letters into number box
3. Gets name, PH etc and stores them
4. Hidden address and credit card via checkbox
5. Pull numbers from pizza boxes as values
6. Make review order function by outputting values
7. Allowing print order function to work by printing values

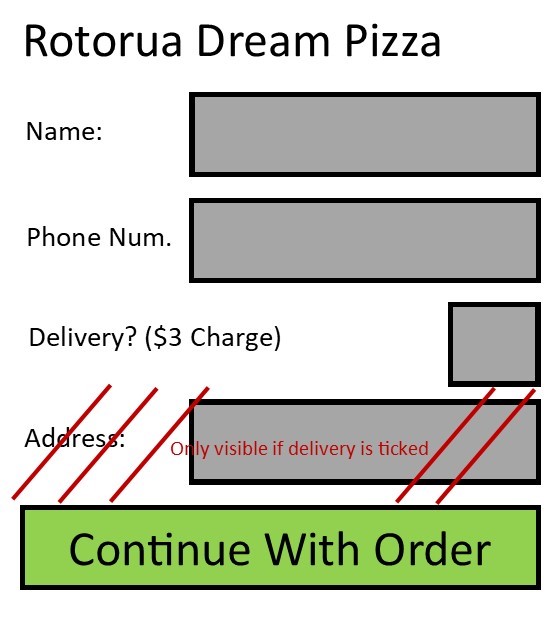
IF TIME PERMITS:

1. Create link to password txt to allow password changes
2. Implement password system

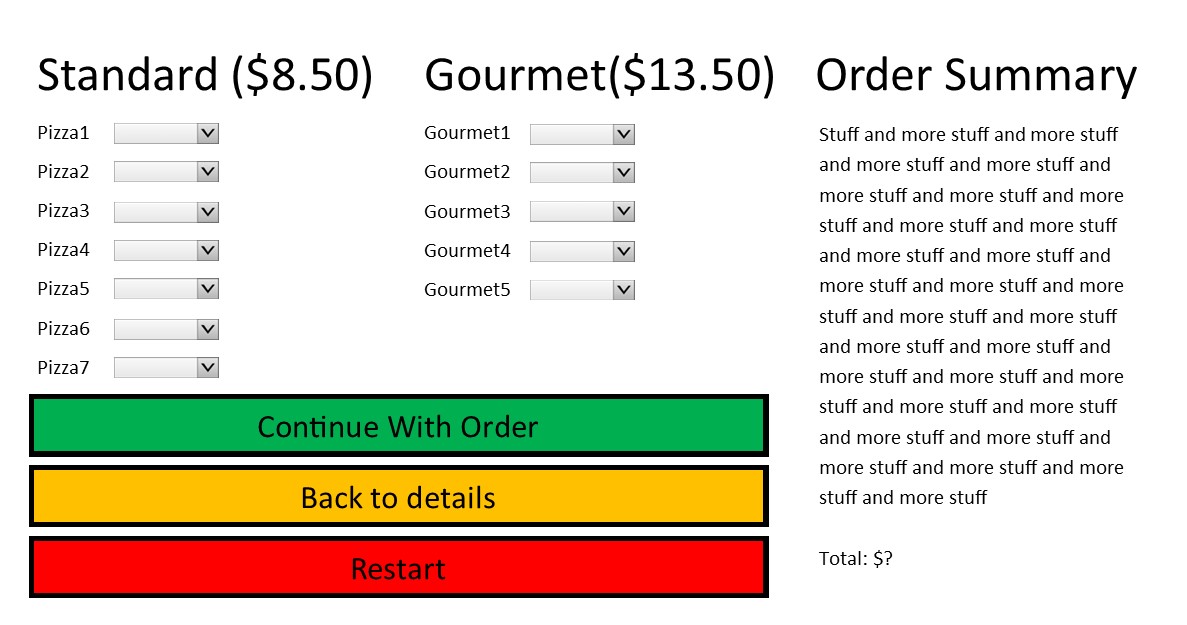
(Added to Trello)  
Creating Designs

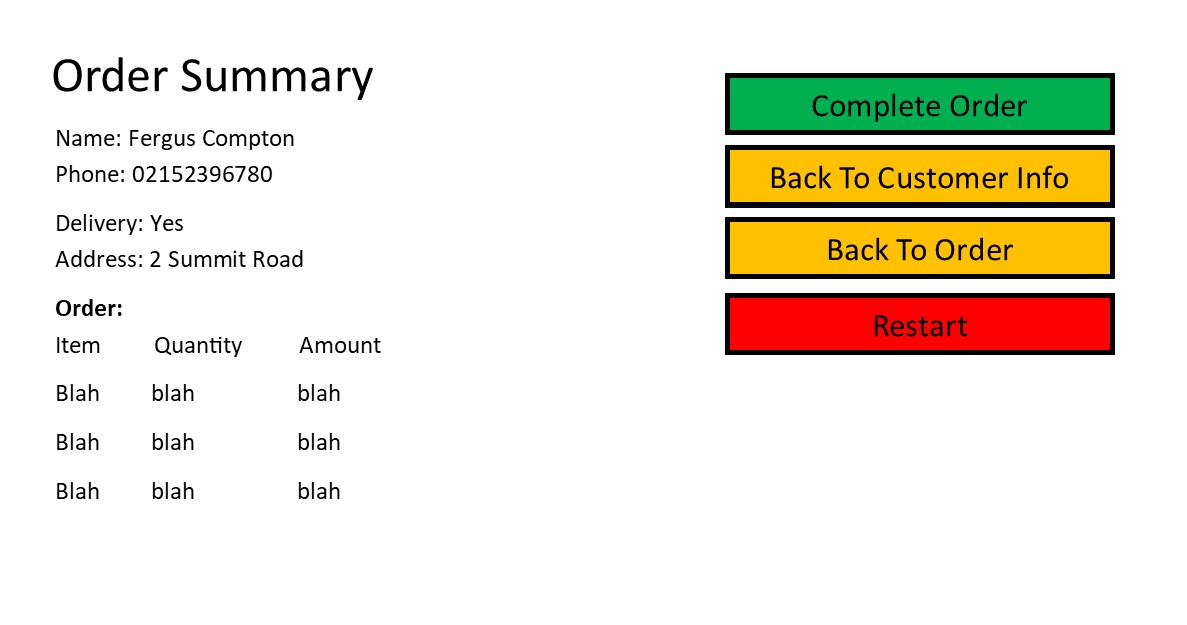
Design 1, A design with multiple forms

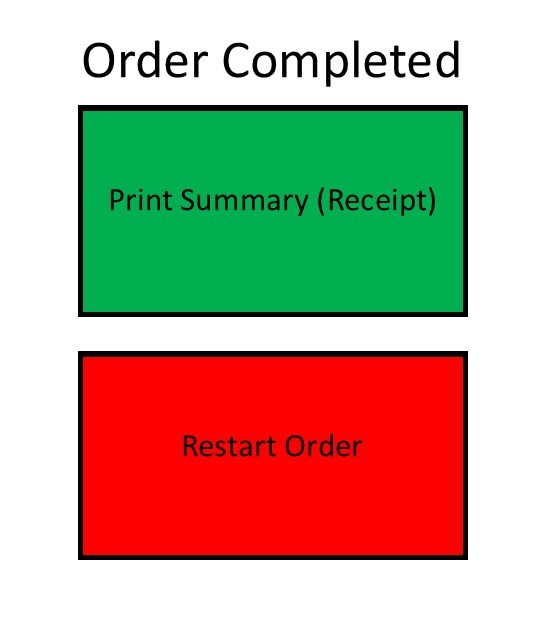
Form 1: Customer Detail Inputs



Form 2: Pizza Order

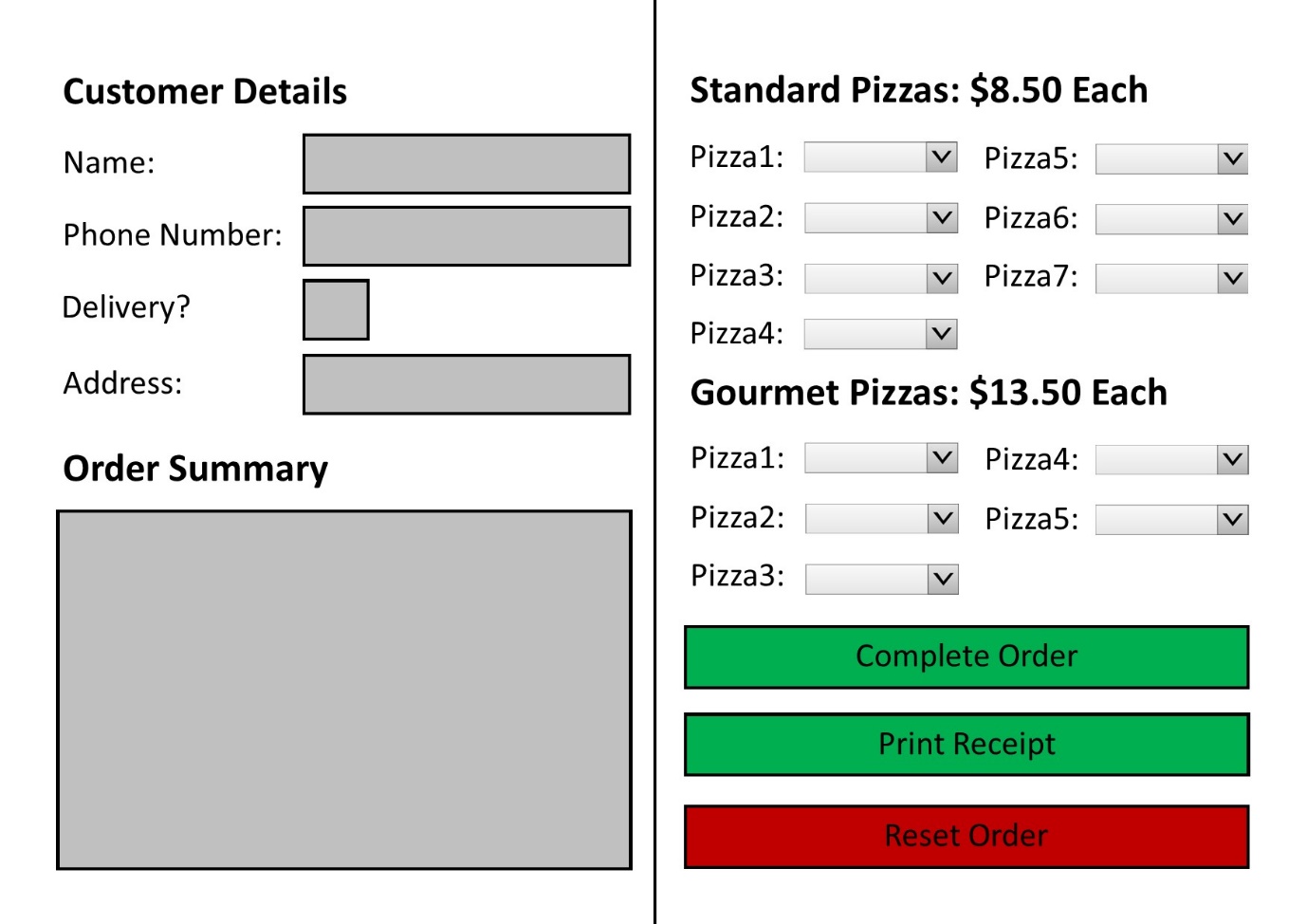


Form 3: Order SummaryForm 4: Order Complete

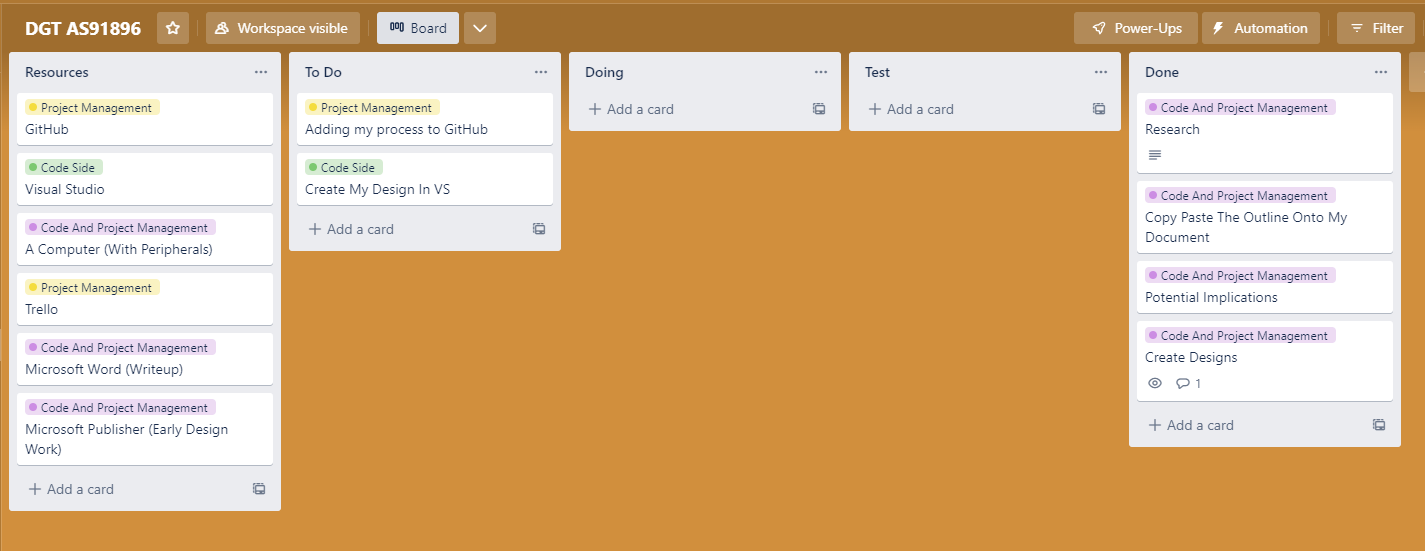


Design 2, All on a single form

NEXT PAGE

  
  
I have decided I will follow the single form design as it seems much simpler to design and code and may also be more intuitive to the end user, however I will make the single form design into a multi form design by making the order summary appear on a second form and have both forms visible at the same time. This is to make the program more modular and move the order summary away from where it currently sits, as it looks slightly out of place under the customer details. Also having multiple forms is considered a “advanced feature” in the NCEA standard, and because it is easy to implement it is an extremely easy way to create an advanced feature.

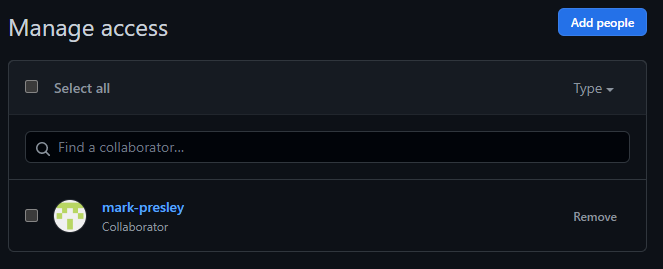
My Trello board after designing has been finished:

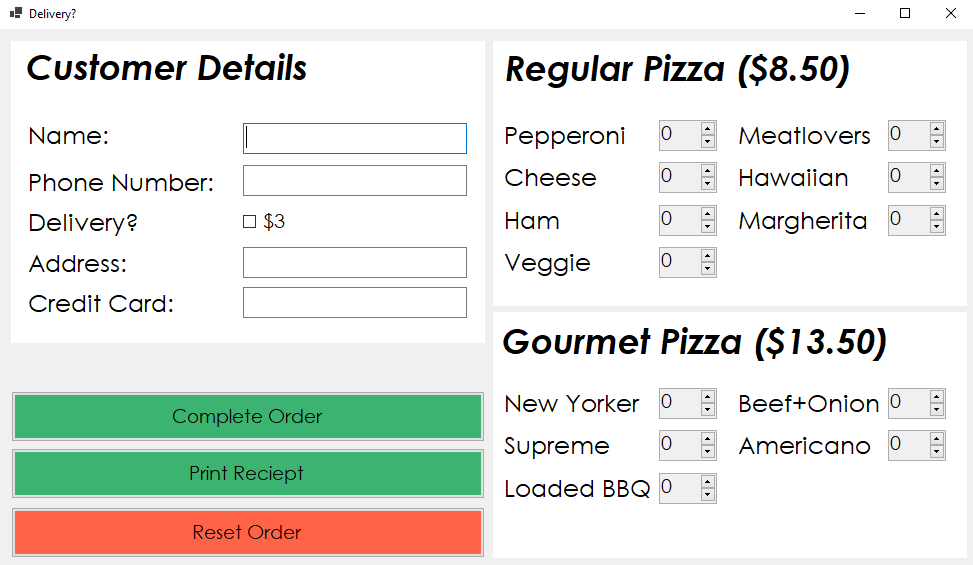


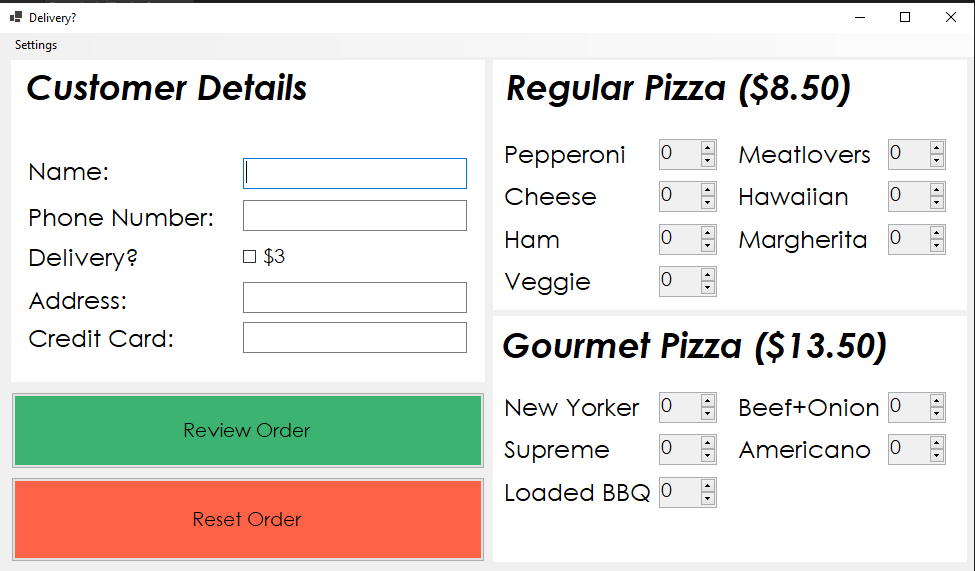
GitHub

I created my GitHub repository so I can store my code and have past versions or “iterations” so if things go south, I can always revert my program back to its previous state. This is better than saving the files locally because I will only lose my files on GitHub if GitHub shuts down, compared to if the files just corrupt locally on my computer. I will be committing a new feature every time I make a change in my program.  
  
[miyudt/AS91896-AS91897 (github.com)](https://github.com/miyudt/AS91896-AS91897/tree/master)

After creating the GitHub repository, the first thing I did was add Mark Presley as an editor in the repository to prove that I can theoretically work in a team, thus filling out the project management side of the standard



After making my project, I created the first form in Visual Studio, when the program is run, the interface looks like this:  
  
  
I decided to move the order summary onto a different form as stated before, and I also decided to make the process buttons have a more washed out colour as I decided that lime green and traffic light red were too harsh to look at for longer periods of time. I also prefer the look of the washed-out colour anyway, so it works in my favour. I also added invisible text and named it PANELEXTENDER as it keeps the customer details panel extended. This will be important later, as I wish to hide the address and credit card options if delivery is ticked. I also added a menu box so that access to more admin settings like password changes and quitting the program is possible. I have also decided to make a Review Order form, and for that to have the print receipt option instead.

After all revisions, this is what my form1 looks like:  
  


The first thing I decided to do was to limit the inputs in “phone number” and “credit card” by limiting numbers. I decided to do it with a KeyChar command like so:  
  
Public Class Form1

Private Sub txtphone\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtphone.KeyPress

If Asc(e.KeyChar) <> 8 Then 'keeps backspace key enabled as you may need to delete numbers

If Asc(e.KeyChar) < 48 Or Asc(e.KeyChar) > 57 Then 'disables all keys that are not 0-9 or backspace

e.Handled = True 'makes sure all extra key presses are declined that are not 0-9 or backspace

End If

End If

End Sub

Private Sub txtcreditcard\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtcreditcard.KeyPress

If Asc(e.KeyChar) <> 8 Then 'keeps backspace key enabled as you may need to delete numbers

If Asc(e.KeyChar) < 48 Or Asc(e.KeyChar) > 57 Then 'disables all keys that are not 0-9 or backspace

e.Handled = True 'makes sure all extra key presses are declined that are not 0-9 or backspace

End If

End If

End Sub

End Class

It disabled every key that is less that 48 and more than 57, leaving 0-9 available. It also leaves key 8 available which is the backspace key. This is good because it also stops the user from putting in symbols such as - + / \* ect. I used [Windows Key Codes - Boost Robotics](https://boostrobotics.eu/windows-key-codes/) in order to understand keycodes better as I feel I will have to use key codes again at some point in the future. This code worked on its first attempt so no revisions will be needed, this has been pushed as an iteration.

Next, I made the code that hides boxes that are not needed for my order if delivery is not ticked. I wrote the following code:

Private Sub chkdelivery\_CheckedChanged(sender As Object, e As EventArgs) Handles chkdelivery.CheckedChanged

If chkdelivery.Checked = True Then 'if the tickbox is not ticked, completes the following:

txtaddress.Show() 'shows address

txtcreditcard.Show() 'shows creditcard

lbladdress.Show() 'shows address

lblcreditcard.Show() 'shows creditcard

Else

txtaddress.Hide() 'hides address

txtcreditcard.Hide() 'hides creditcard

lbladdress.Hide() 'hides address

lblcreditcard.Hide() 'hides creditcard

End If

End Sub

End Class

After testing this, I encountered a small problem, the issue being that on the first cycle of the delivery box being ticked, the boxes would not change. They would only change on the second or third cycle of clicks, which is extremely impractical. I edited the code so that it would detect on false rather than true, as this seemed to solve the issue. I also tested that all functionality from previous iterations still functioned, and it did.

Private Sub chkdelivery\_CheckedChanged(sender As Object, e As EventArgs) Handles chkdelivery.CheckedChanged

If chkdelivery.Checked = False Then 'if the tickbox is not ticked, completes the following:

txtaddress.Hide() 'hides address

txtcreditcard.Hide() 'hides creditcard

lbladdress.Hide() 'hides address

lblcreditcard.Hide() 'hides creditcard

Else

txtaddress.Show() 'shows address

txtcreditcard.Show() 'shows creditcard

lbladdress.Show() 'shows address

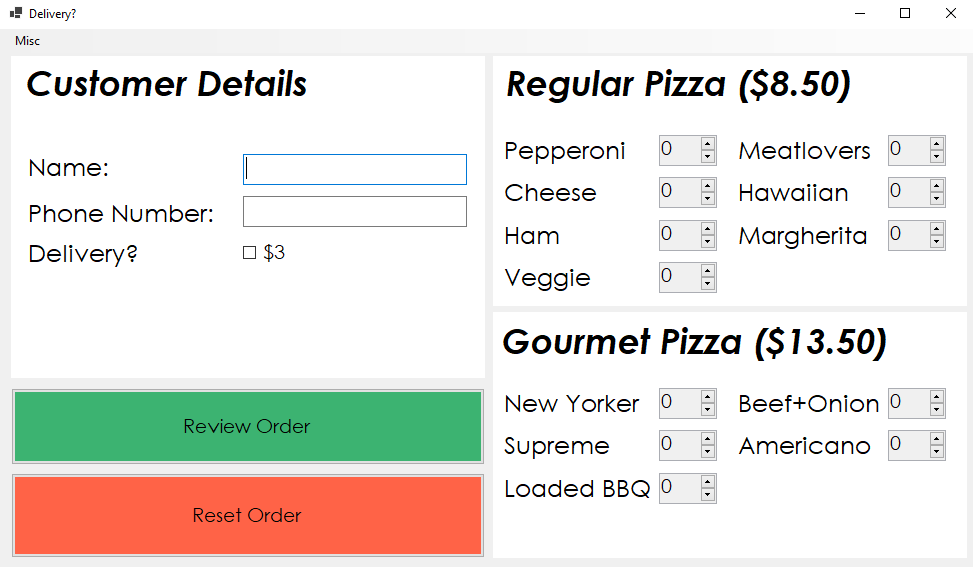
lblcreditcard.Show() 'shows creditcard

End If

End Sub

End Class

Box unticked

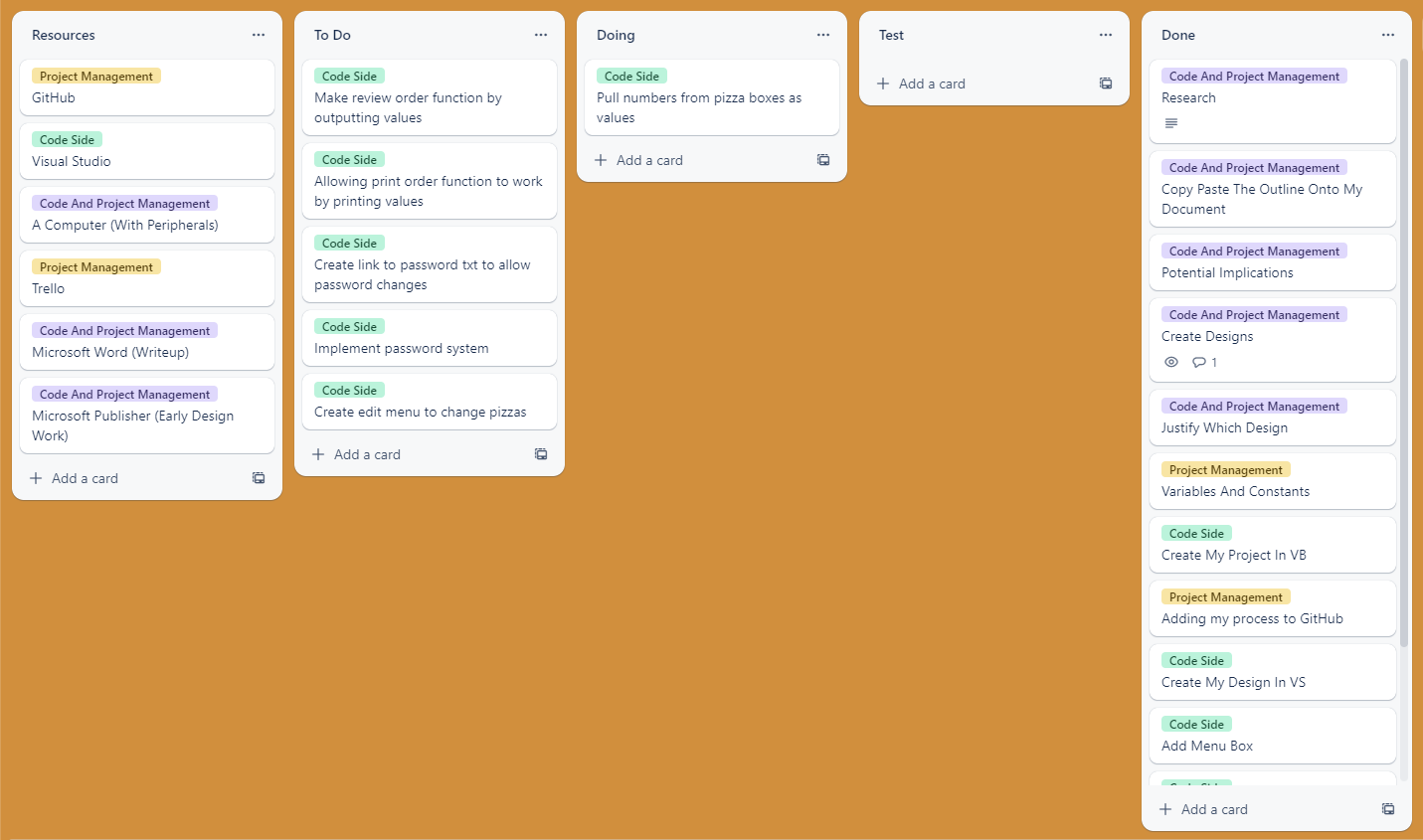


Box ticked

Graphical user interface, table

Description automatically generated

Trello Board:



Full code so far:

Public Class Form1

Private Sub txtphone\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtphone.KeyPress

If Asc(e.KeyChar) <> 8 Then 'keeps backspace key enabled as you may need to delete numbers

If Asc(e.KeyChar) < 48 Or Asc(e.KeyChar) > 57 Then 'disables all keys that are not 0-9 or backspace

e.Handled = True 'makes sure all extra key presses are declined that are not 0-9 or backspace

End If

End If

End Sub

Private Sub txtcreditcard\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtcreditcard.KeyPress

If Asc(e.KeyChar) <> 8 Then 'keeps backspace key enabled as you may need to delete numbers

If Asc(e.KeyChar) < 48 Or Asc(e.KeyChar) > 57 Then 'disables all keys that are not 0-9 or backspace

e.Handled = True 'makes sure all extra key presses are declined that are not 0-9 or backspace

End If

End If

End Sub

Private Sub chkdelivery\_CheckedChanged(sender As Object, e As EventArgs) Handles chkdelivery.CheckedChanged

If chkdelivery.Checked = False Then 'if the tickbox is not ticked, completes the following:

txtaddress.Hide() 'hides address

txtcreditcard.Hide() 'hides creditcard

lbladdress.Hide() 'hides address

lblcreditcard.Hide() 'hides creditcard

Else

txtaddress.Show() 'shows address

txtcreditcard.Show() 'shows creditcard

lbladdress.Show() 'shows address

lblcreditcard.Show() 'shows creditcard

End If

End Sub

End Class

Next, pulling all the customer data from the boxes after it is typed. In class, we have done work where we pull data from text boxes, and since this is practically the same application I will be, uhhh borrowing code from my previous project.

Private Sub btnrevieworder\_Click(sender As Object, e As EventArgs) Handles btnrevieworder.Click

Dim TrimmedName As String = txtname.Text.Trim 'trims spaces off text

Dim TrimmedPH As String = Val(txtphone.Text.Trim) 'trims spaces off text

Dim TrimmedCredit As String = Val(txtcreditcard.Text.Trim) 'trims spaces off text

Dim TrimmedAddress As String = txtaddress.Text.Trim 'trims spaces off text

End Sub

End Class

After writing this code, to quickly check that it works I wrote this extra piece of code (message box) and got the following successful result

Private Sub btnrevieworder\_Click(sender As Object, e As EventArgs) Handles btnrevieworder.Click

Dim TrimmedName As String = txtname.Text.Trim 'trims spaces off text

Dim TrimmedPH As String = Val(txtphone.Text.Trim) 'trims spaces off text

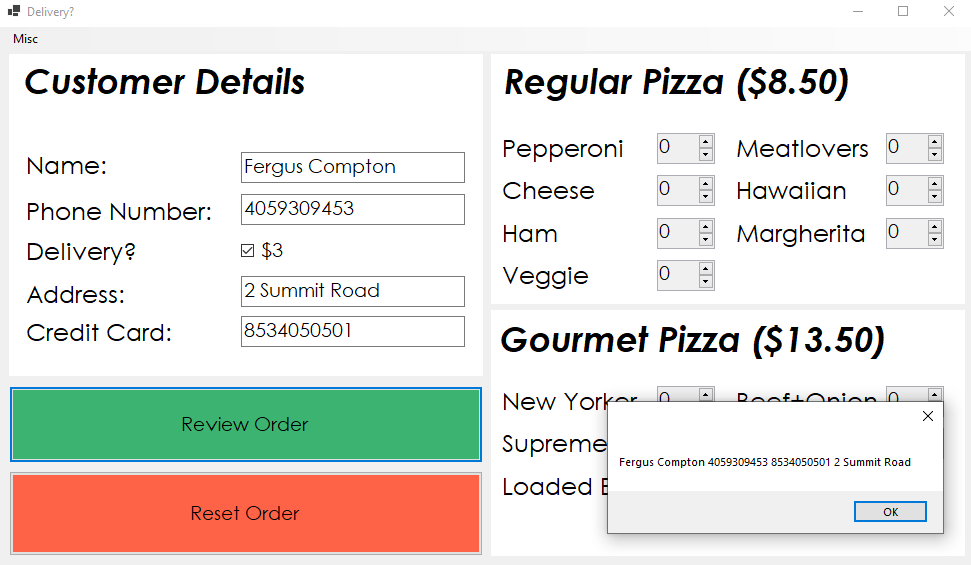
Dim TrimmedCredit As String = Val(txtcreditcard.Text.Trim) 'trims spaces off text

Dim TrimmedAddress As String = txtaddress.Text.Trim 'trims spaces off text

MessageBox.Show(TrimmedName & " " & TrimmedPH & " " & TrimmedCredit & " " & TrimmedAddress)

End Sub

End Class



Because of me locking the inputs of the phone number and credit card boxes, it means that I will never have to deal with issues of x/0 to crash the program as / can never be typed

I then set up a second form in my program that looks like the following (it will change)

Graphical user interface

Description automatically generated with medium confidence

And then wrote the following code to push all the form info to form2. This code also hides the address and credit card info if delivery is not ticked, it also refuses continuation to form2 if all the required fields to fill out are not filled

Public Sub btnrevieworder\_Click(sender As Object, e As EventArgs) Handles btnrevieworder.Click

Dim TrimmedName As String = txtname.Text.Trim 'trims spaces off text

Dim TrimmedPH As String = Val(txtphone.Text.Trim) 'trims spaces off text

Dim TrimmedCredit As String = Val(txtcreditcard.Text.Trim) 'trims spaces off text

Dim TrimmedAddress As String = txtaddress.Text.Trim 'trims spaces off text

If chkdelivery.Checked = True Then

If TrimmedName = "" Or TrimmedPH = 0 Or TrimmedAddress = "" Or TrimmedCredit = 0 Then

MessageBox.Show("Fill In All Fields For Delivery") 'fails if all requirements are not met

Else

Form2.txtname.Text = "Name: " & TrimmedName 'pushes name to form2

Form2.txtaddress.Text = "Address: " & TrimmedAddress 'pushes address to form2

Form2.txtph.Text = "Phone Number: " & TrimmedPH 'pushes phone number to form2

Form2.txtcredit.Text = "Credit Card: " & TrimmedCredit 'pushes credit card to form2

Form2.Show() 'show form 2

Me.Hide() 'hide form 1

Form2.txtcredit.Show() 'shows credit card text if hidden

Form2.txtaddress.Show() 'shows address text if hidden

End If

Else

If TrimmedName = "" Or TrimmedPH = 0 Then

MessageBox.Show("Fill In All Fields For Pickup") 'fails if all requirements are not met

Else

Form2.txtph.Text = "Phone Number: " & TrimmedPH 'pushes phone number to form2

Form2.txtname.Text = "Name: " & TrimmedName 'pushes name to form2

Form2.txtcredit.Hide() 'hides credit text if shown

Form2.txtaddress.Hide() 'hides address text if shown

Form2.Show() 'shows form 2

Me.Hide() 'hides form 1

End If

End If

End Sub

End Class

Also added an input limit into the phone number box and credit card boxes, phone number box accepts a max input of 11 and credit card accepts a maximum of 16. This was done through the designer. I also decided to not store the names in a proper variable as I felt it is not necessary to complete the given task.

After testing this code, I once again encountered no bugs, I also tested that all functionality from previous iterations still functioned, and it did.

Code as of now:

Public Class Form1

Private Sub txtphone\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtphone.KeyPress

If Asc(e.KeyChar) <> 8 Then 'keeps backspace key enabled as you may need to delete numbers

If Asc(e.KeyChar) < 48 Or Asc(e.KeyChar) > 57 Then 'disables all keys that are not 0-9 or backspace

e.Handled = True 'makes sure all extra key presses are declined that are not 0-9 or backspace

End If

End If

End Sub

Private Sub txtcreditcard\_KeyPress(ByVal sender As Object, ByVal e As System.Windows.Forms.KeyPressEventArgs) Handles txtcreditcard.KeyPress

If Asc(e.KeyChar) <> 8 Then 'keeps backspace key enabled as you may need to delete numbers

If Asc(e.KeyChar) < 48 Or Asc(e.KeyChar) > 57 Then 'disables all keys that are not 0-9 or backspace

e.Handled = True 'makes sure all extra key presses are declined that are not 0-9 or backspace

End If

End If

End Sub

Private Sub chkdelivery\_CheckedChanged(sender As Object, e As EventArgs) Handles chkdelivery.CheckedChanged

If chkdelivery.Checked = False Then 'if the tickbox is not ticked, completes the following:

txtaddress.Hide() 'hides address

txtcreditcard.Hide() 'hides creditcard

lbladdress.Hide() 'hides address

lblcreditcard.Hide() 'hides creditcard

Else

txtaddress.Show() 'shows address

txtcreditcard.Show() 'shows creditcard

lbladdress.Show() 'shows address

lblcreditcard.Show() 'shows creditcard

End If

End Sub

Public Sub btnrevieworder\_Click(sender As Object, e As EventArgs) Handles btnrevieworder.Click

Dim TrimmedName As String = txtname.Text.Trim 'trims spaces off text

Dim TrimmedPH As String = Val(txtphone.Text.Trim) 'trims spaces off text

Dim TrimmedCredit As String = Val(txtcreditcard.Text.Trim) 'trims spaces off text

Dim TrimmedAddress As String = txtaddress.Text.Trim 'trims spaces off text

If chkdelivery.Checked = True Then

If TrimmedName = "" Or TrimmedPH = 0 Or TrimmedAddress = "" Or TrimmedCredit = 0 Then

MessageBox.Show("Fill In All Fields For Delivery") 'fails if all requirements are not met

Else

Form2.txtname.Text = "Name: " & TrimmedName 'pushes name to form2

Form2.txtaddress.Text = "Address: " & TrimmedAddress 'pushes address to form2

Form2.txtph.Text = "Phone Number: " & TrimmedPH 'pushes phone number to form2

Form2.txtcredit.Text = "Credit Card: " & TrimmedCredit 'pushes credit card to form2

Form2.Show() 'show form 2

Me.Hide() 'hide form 1

Form2.txtcredit.Show() 'shows credit card text if hidden

Form2.txtaddress.Show() 'shows address text if hidden

End If

Else

If TrimmedName = "" Or TrimmedPH = 0 Then

MessageBox.Show("Fill In All Fields For Pickup") 'fails if all requirements are not met

Else

Form2.txtph.Text = "Phone Number: " & TrimmedPH 'pushes phone number to form2

Form2.txtname.Text = "Name: " & TrimmedName 'pushes name to form2

Form2.txtcredit.Hide() 'hides credit text if shown

Form2.txtaddress.Hide() 'hides address text if shown

Form2.Show() 'shows form 2

Me.Hide() 'hides form 1

End If

End If

End Sub

End Class

Public Class Form2

Private Sub btnreset\_Click(sender As Object, e As EventArgs) Handles btnback.Click

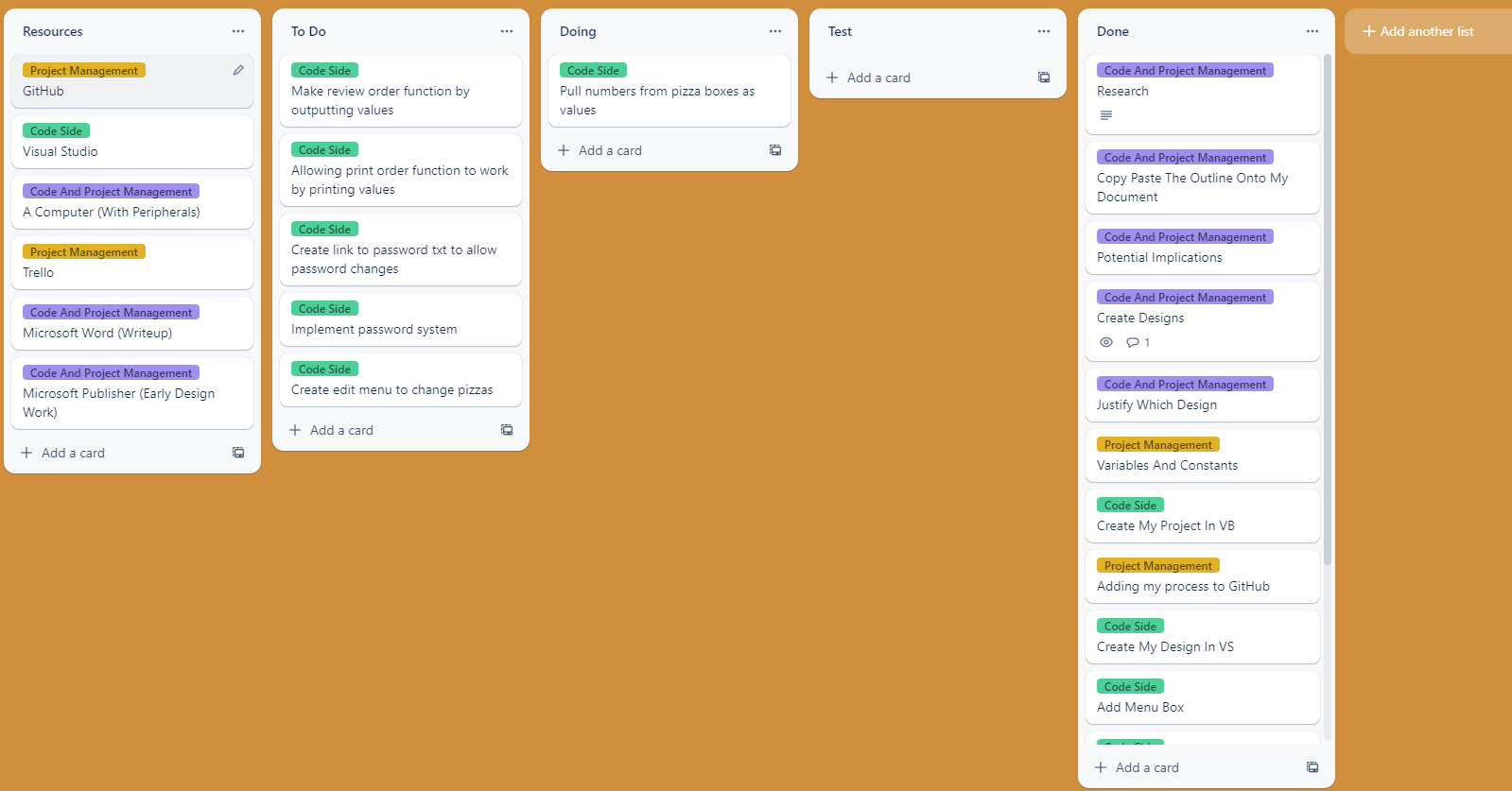
Form1.Show()

Me.Hide()

End Sub

End Class

Trello:



Next