

Project outline/specifications:

GUI Design:

- Use tkinter to create the main window and separate frames for input fields, menu selection, and order summary.
- Implement input validation for the first name, last name, phone number, and email address fields to ensure correct formats.
- Disable the menu selection button until all required fields are filled in.
- Display selected items, quantities, and calculated prices in the order summary section.

Classes:

- Define classes for Pizza, Side, Drink, and Order to represent the different items and the overall order.
- Implement methods within the classes for item selection, price calculation, and order details handling.

Input Validation:

- Use regular expressions or built-in Python functions to validate the user input for name, phone number, and email address formats.
- Set minimum and maximum order quantity limits for items and enforce them during selection.
- Check that at least one item is selected before allowing the order to be placed.
- Calculate Total Cost:
- Utilise the selected items and their respective prices to calculate the subtotal cost.
- Apply the GST (Goods and Services Tax) and a \$10 delivery fee to the subtotal to obtain the total cost.

Generate Receipt:

- Create a receipt with customer information, selected items, quantities, prices, total cost, and any additional charges.
- Display the receipt on the GUI or save it to a file using file I/O operations.

Testing and Debugging:

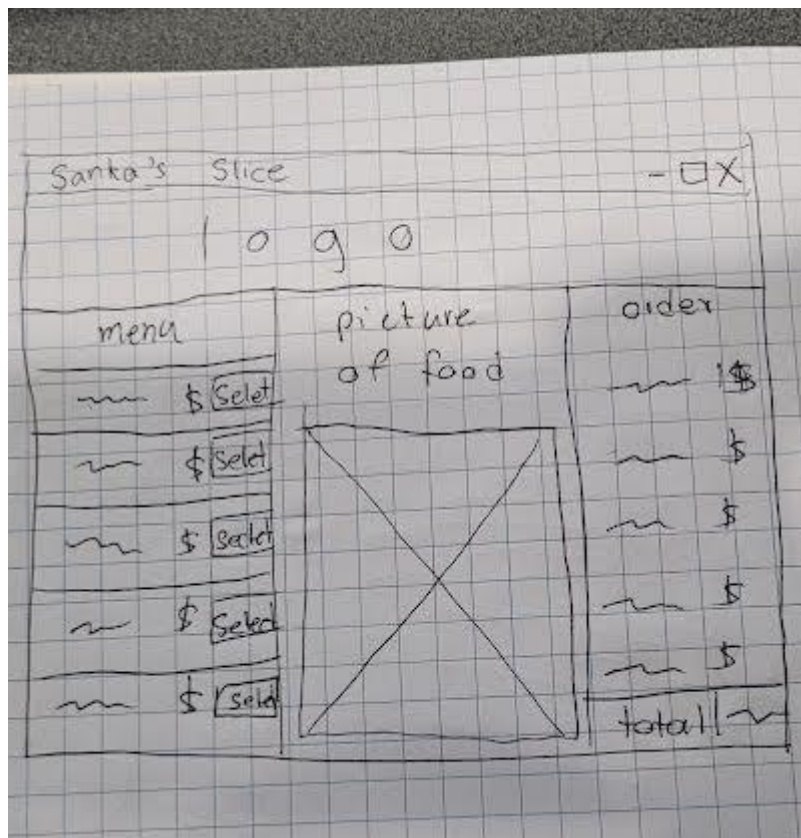
- Conduct thorough testing with various scenarios, including expected and boundary cases, to ensure the program functions correctly.
- Debug any issues or errors encountered during testing to ensure a smooth user

experience.

Daily log:

8/5/23:

- Cloned github repository and made first commit which was just the python file
- Drew a initial wireframe of what i want my gui to look like



10/5/23:

- Finished creating working form gui class with first name, last name and phone number input fields

Form Window

First Name:

Last Name:

Phone Number:

10/5/23:

- Added a submit button that doesn't really do anything yet

Form Window

First Name:

Last Name:

Phone Number:

Submit

10/5/23:

-added input validation to all form fields and if the user enters incorrect information then a message box pops up saying what/where the error is. It makes sure that in the first and last name input field that the input is text not numbers or characters and for the phone number it is sure it's only numbers and is exactly 10 digits long.

Form Window

First Name: tt5

Last Name: tr

Phone Number: 0000000000

Submit

Error

Invalid first name. Use alphabets only

OK

Form Window

First Name: tt

Last Name: 5

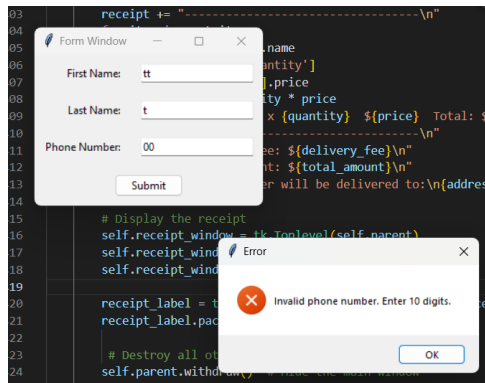
Phone Number: 0000000000

Submit

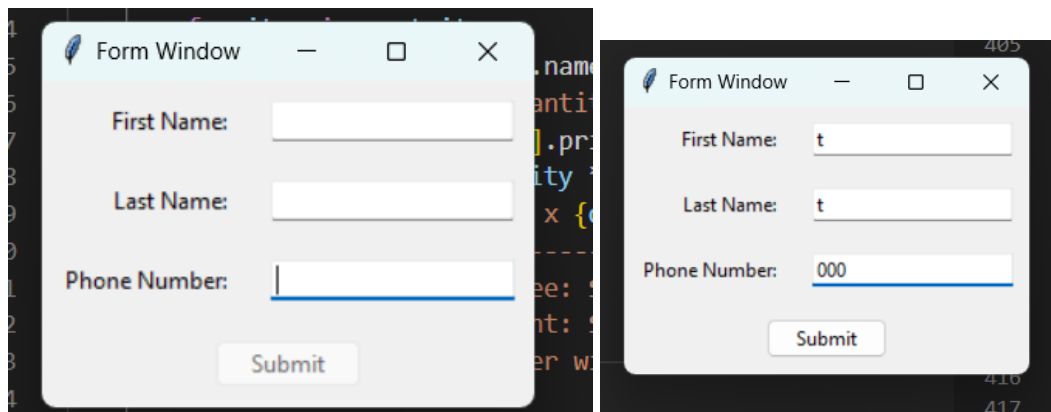
Error

Invalid last name. Use alphabets only

OK



I disabled the submit button until all the fields were filled out.



14/5/23:

Created a category class which will store the pizza, sides and drink categories which will allow the user to easily find

The items class represent a singular item on the menu such as a pepperoni pizza and it also represents the price of the item for example \$10

```

class Category:
    def __init__(self, name):
        self.name = name
        self.items = []

    def add_item(self, item):
        self.items.append(item)

class Item:
    def __init__(self, name, price):
        self.name = name
        self.price = price

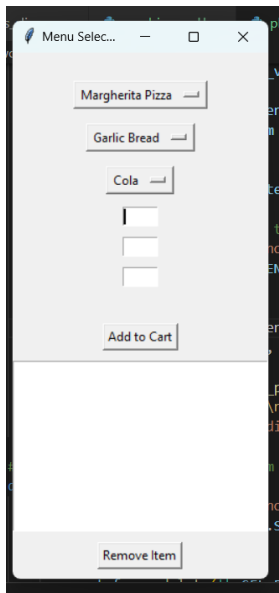
```

14/5/23:

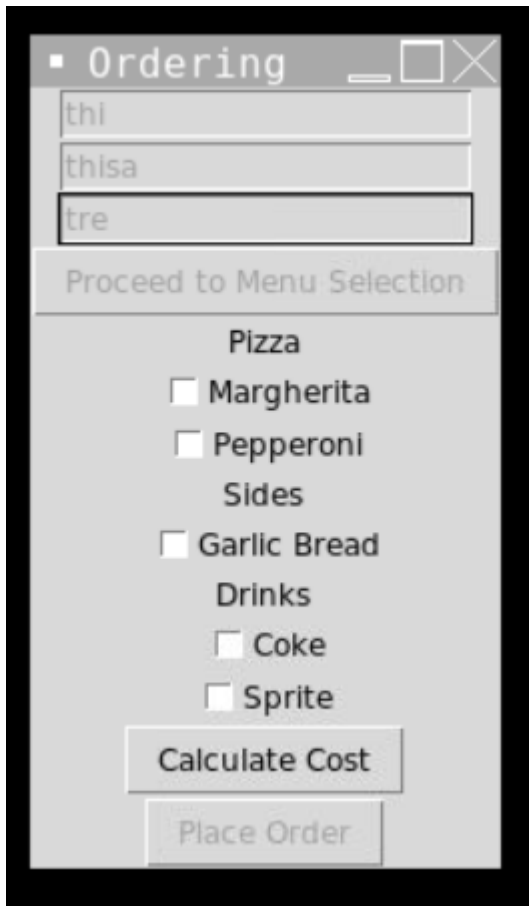
Try to create/Thinking of a design for my menu selection gui

16/5/23:

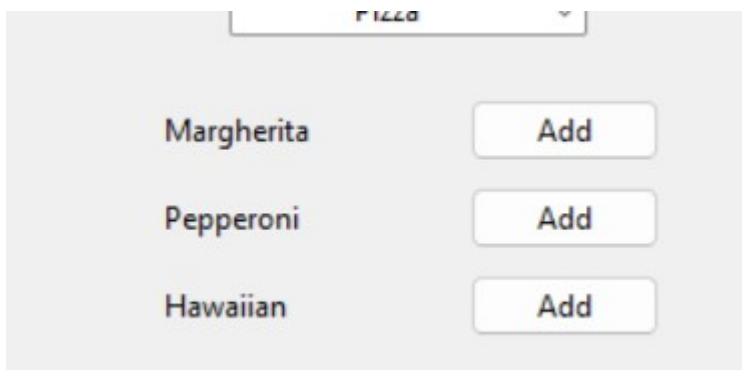
Created the menu selection class and create the items that are going to be available on my menu



Don't like the design that I created above. The new design that does everything on the same window



This is my final menu design which has an add to cart button to add items into the “cart” - i haven't made a cart yet.

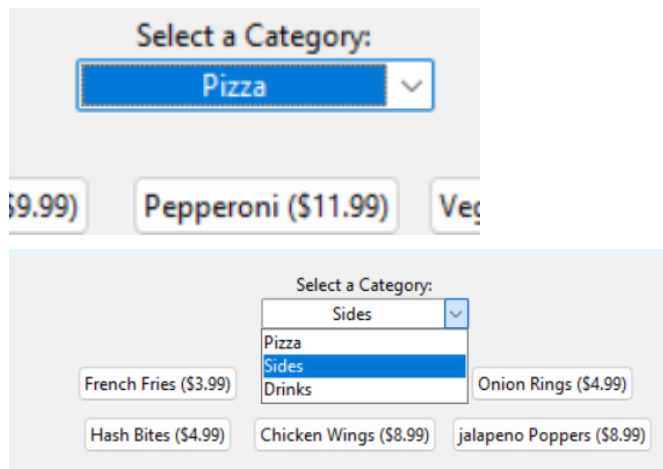


I watched a tkinter tutorial video by Bro Code, where I learned about button grids. I thought that the button grid was a better menu design, so I asked for feedback from Juanye and my family about their preferred method for creating an intuitive menu using tkinter. After showing both designs to them, they all preferred the button grid design. With this feedback in mind, I concluded that the button grid approach would be the best way to design an intuitive menu using tkinter. Instead of just having an item label and an "Add to Cart" button next to it, the menu items will be represented by buttons organised in a grid layout. This layout not only enhances the visual appeal of the menu but also provides an intuitive and interactive way for

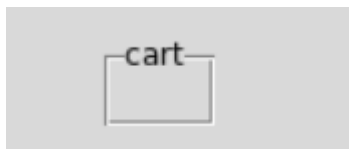
users to make their selections.



I have added a dropdown menu that allows the user to choose from different categories. This enhancement provides a more organised and convenient way for users to navigate through the menu options, I have also centred everything in the menu layout to create a visually pleasing appearance.



Then i tired to create a cart area that when an item button is clicked it shows the item in the cart area. But i wasn't working the way i want it to.



17/5/23

When i was testing juanyes program i saw that he had used treeview so i watched a video on how to use it because that exactly what i was trying to make.

Discovering treeview:

[Treeview - Python Tkinter GUI Tutorial #116 - YouTube](#)

And then i implemented into my code:

I made it so if nothing is in the treeview then all three buttons are disabled

Select a Category:
Sides

French Fries (\$3.99)

Garlic Bread (\$4.99)

Onion Rings (\$4.99)

Hash Bites (\$4.99)

Chicken Wings (\$8.99)

jalapeno Poppers (\$8.99)

Item	Quantity	Price
Total	0	\$0.00

Delete All

Complete Order

Delete Item

When an item is added the delete all and complete order buttons become active

Select a Category:
Sides

French Fries (\$3.99)

Garlic Bread (\$4.99)

Onion Rings (\$4.99)

Hash Bites (\$4.99)

Chicken Wings (\$8.99)

jalapeno Poppers (\$8.99)

Item	Quantity	Price
French Fries	1	\$3.99
Total	1	\$3.99

Delete All

Complete Order

Delete Item

When an item is selected is when the delete item button becomes active

Item	Quantity	Price
French Fries	1	\$3.99
Total	1	\$3.99

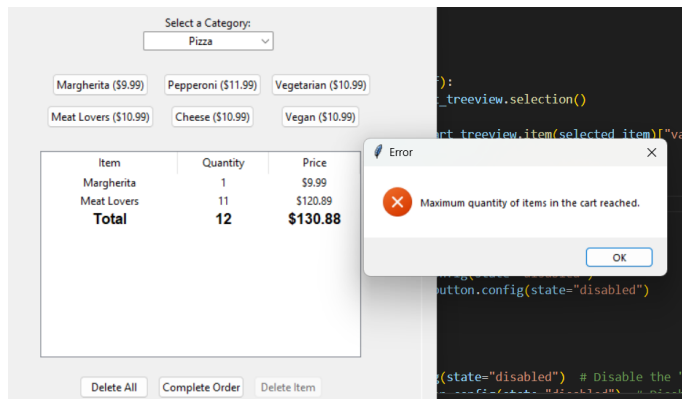
Delete All

Complete Order

Delete Item

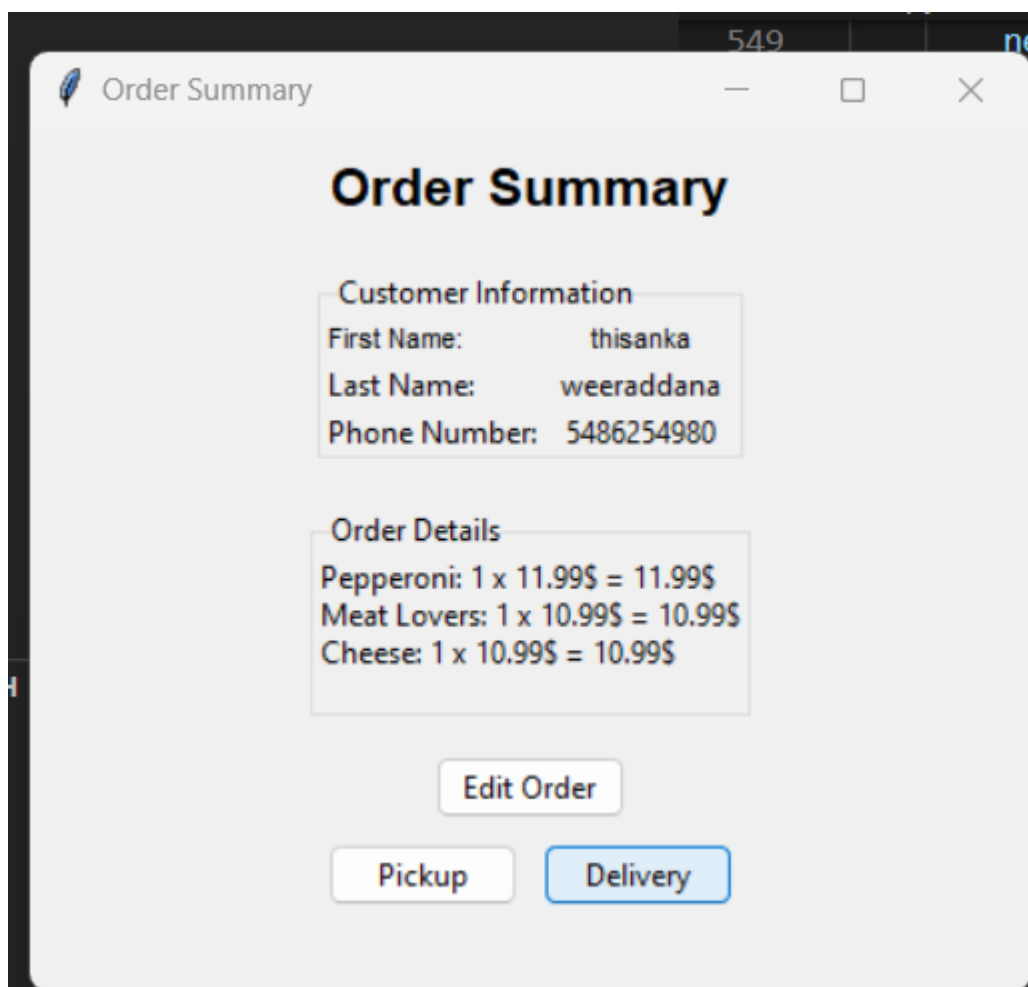
When the order complete button is clicked it takes you to a new window

Added a limit to the amount of items allowed to be in the cart to no more than 12 if it exceeds this limit a error will appear.



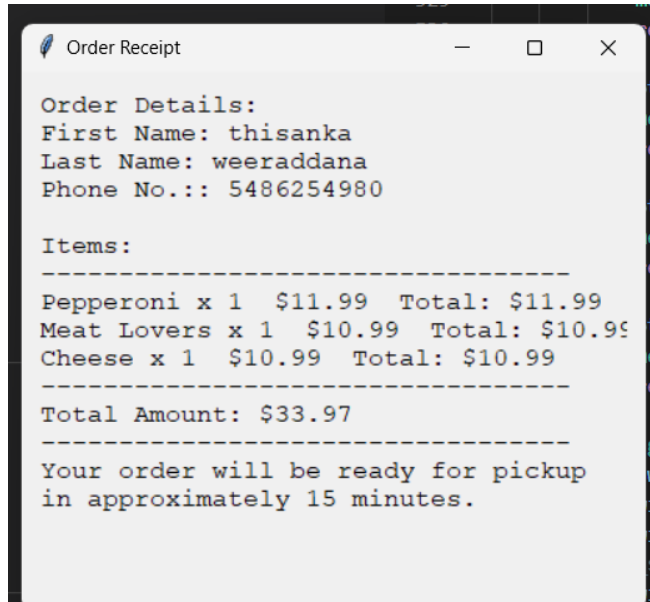
18/5/23

Created an order summary window that displays the customers information and the items in there order it also has three buttons at the bottom which shows

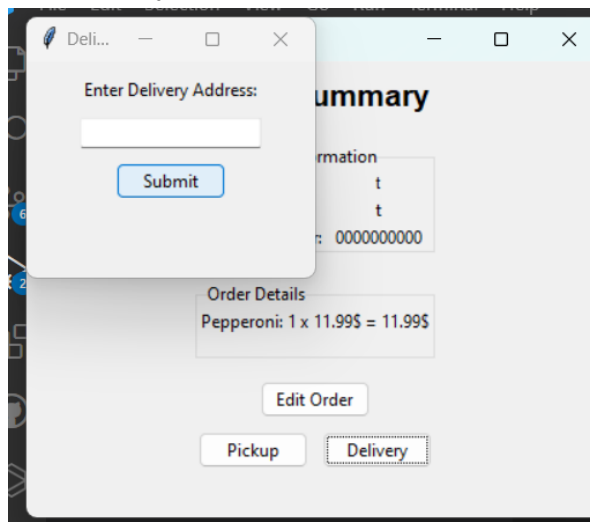


When edit order is clicked it takes you back to the menu selection gui.

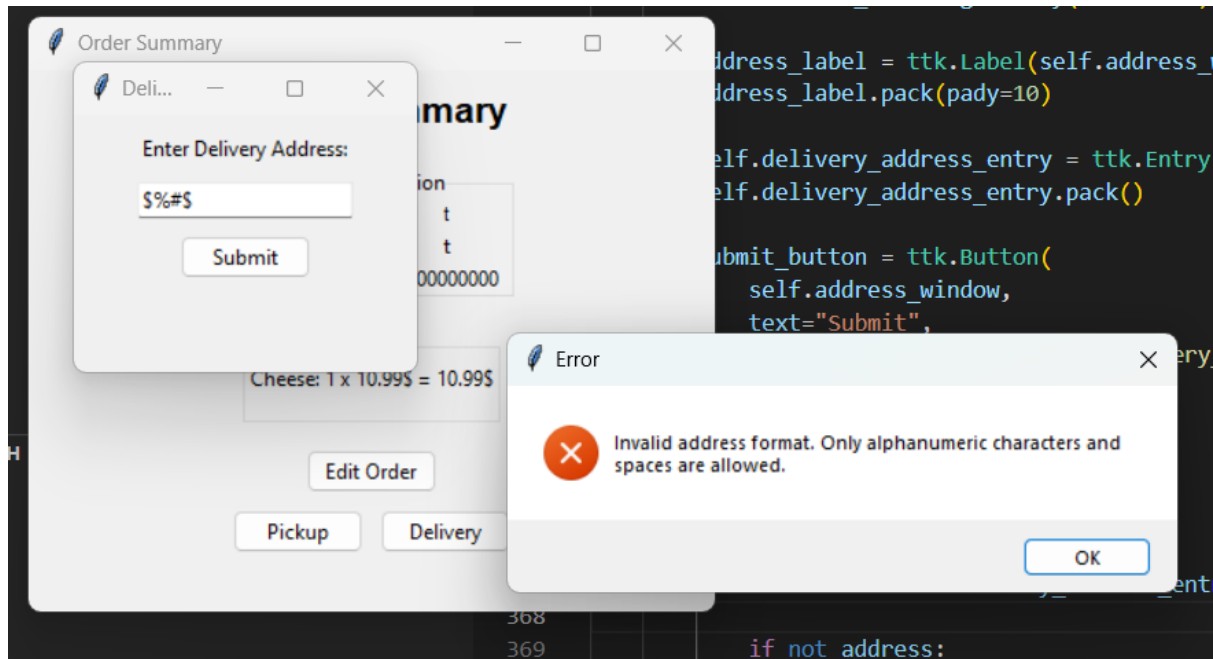
When pickup is clicked it closes all windows and generates a receipt that shows customer information and order details



When the delivery button is clicked a popup window appears asking for the user to enter their delivery address.

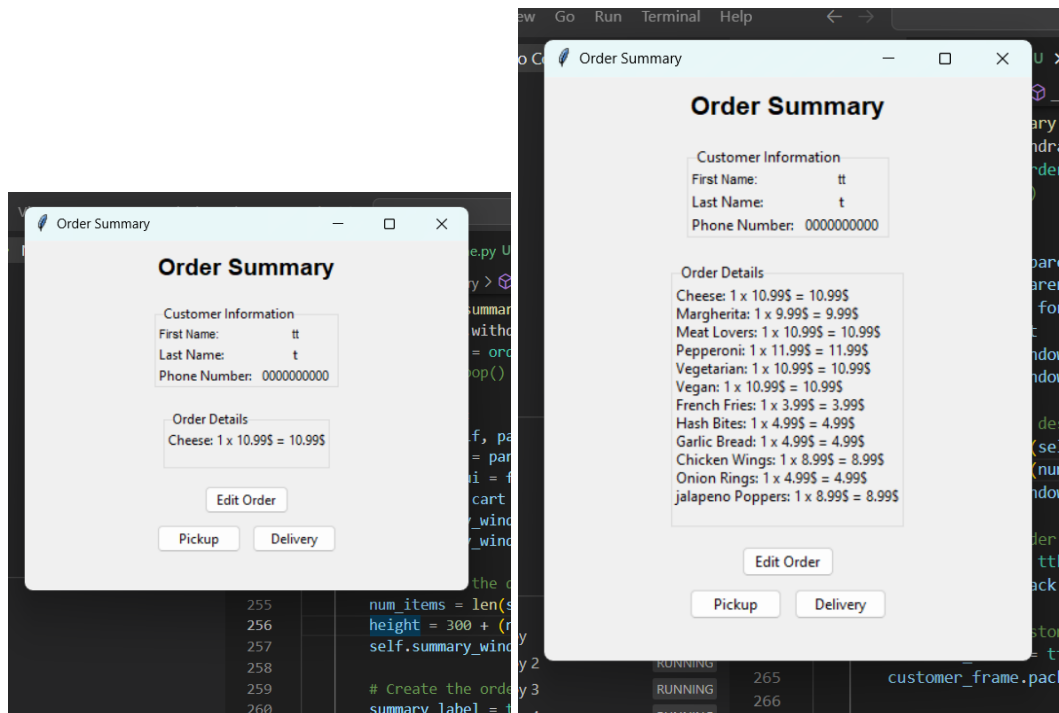


I've now added input validation to the delivery address field so it only expects alphanumeric characters.

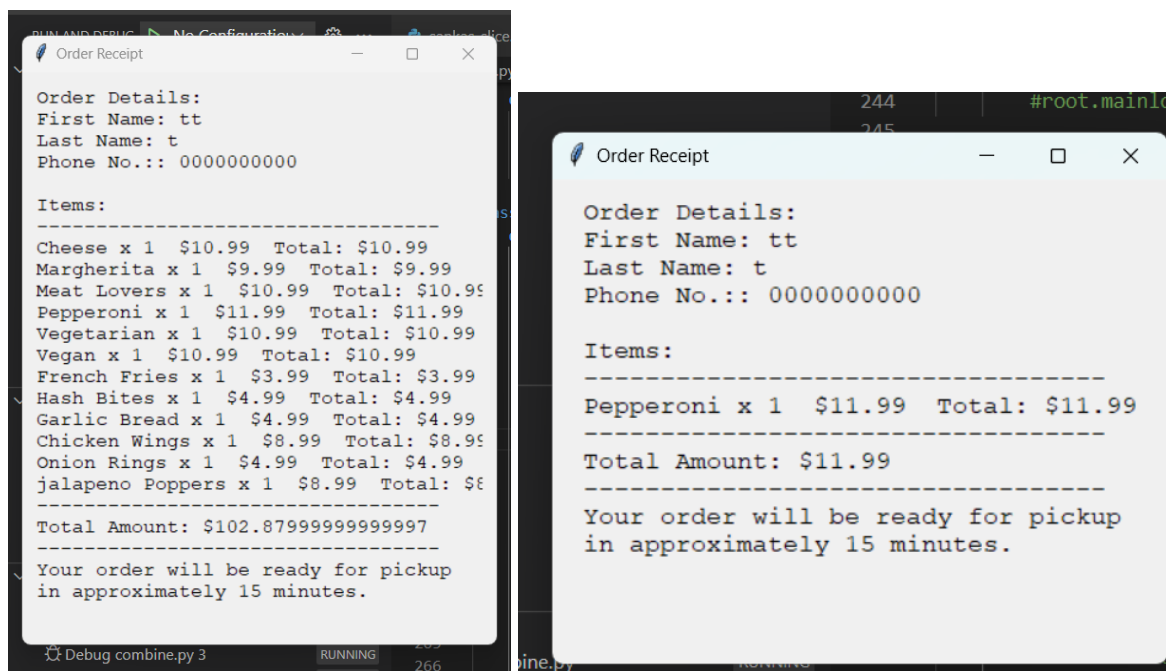


When i order the maximum number of items allowed in the cart the height of the widget is too small to show all of the information and i can't just make it big enough to fit everything because if the user orders only 1 item then the window would be too large which is what i've added the feature below which calculates the height of the window based off the amount of items in the cart so that the window will always be the perfect size

```
# Calculate the desired height based on the number
num_items = len(self.cart)
height = 300 + (num_items * 15)
self.summary_window.geometry(f"400x{height}")
```



I've also added this feature to both of the receipts so that they get bigger depending on the amount of items in the order.



After running my code through a python PEP 8 checker i got a lot of errors so i fixed them

47 style issues found

Click a style issue to see where it occurs in your code.

I have reduced the style issues significantly and the only issues left are some “line is longer than 75 charters” and “docsting is required before a class”

28 style issues found

Click a style issue to see where it occurs in your code.

19/5/23:
Due Date

Testing plan:

Input to test	Expected Input			Invalid Input		
	Input	Desired outcome	Test feedback	Input	Desired outcome	Test feedback
Enter first name	Letters from a-z	Form successfully completed	When the name “ben” is entered the form successful message box appears	Special characters and numbers	Error box saying invalid input	The error message box appears as expected
Last name	Letters from a-z	Form successfully completed	When the name “johnson” is entered the form	Special characters and numbers	Error box saying invalid input	The error message box appears as expected

			successful message box appears			
Phone number	Numbers from 0 to 9	Form successfully completed	When the numbers "02148 45620" are entered the form successful message box appears	Letters from a-z and special characters	Error box saying invalid input	The error message box appears as expected
Enter delivery address	Alphanumeric character	The delivery receipt appears with all the correct information	When "43 wakati pu" is entered the delivery receipt appears with the correct information	Anything other than alphanumeric character	When "^&&%*" is entered an error box should appear saying to only enter alphanumeric character	The error message box appears as expected

I have implemented robust code validation to ensure that the input forms in the GUI only accept accurate and valid information. This guarantees that the form data remains consistent and reliable. Furthermore, to enhance the user experience and streamline the process, I have implemented a feature that disables the submit button until all the required fields have been filled out. This not only promotes usability but also prevents any incomplete or erroneous data from being submitted.

In the menu selection GUI, I have taken usability a step further by implementing intelligent button disabling. Initially, the "Delete All," "Complete Order," and "Delete" buttons below the

treeview are disabled. However, as soon as an item is added to the cart/treeview, these buttons become active, allowing the user to perform the respective actions. Additionally, to prevent any confusion or frustration, I have disabled the delete button until the user selects an item from the cart/treeview. This ensures that the user cannot click the delete button unless it serves a purpose, thus preventing any ineffective or unintended actions from occurring.

Overall, these enhancements not only validate the user input and improve the system's reliability but also make the GUI more user-friendly and intuitive, leading to a smoother and more efficient user experience.