**HIGHER CERTIFICATE IN ELECTRONIC ENGINEERING**

**ACADEMIC YEAR 2021/2022**



Department of Electronic Engineering

End of Semester Report

April 2022

**Subject:** GUI Development in Python

**Project title:** Centra Retail Store

**Student name:** Agert Berisha

**Student ID:** X00175572

**Submission date:** 27/04/2022

**Supervisor:** Mr. Andrew Donnellan

**Declaration**

This GUI project entitled “Centra Retail Store” is my original work and has not been submitted for any other purpose to any other institute.

Signed Agert Berisha

Full Name Agert Berisha

Student No X00175572

**Acknowledgements**

I would like to acknowledge Mr Andrew Donnellan for their support and assistance.

**­Executive Summary**

The main aim of this project is for the student to learn how to program in Python, with particular emphasis using the Tkinter GUI library.

In this report, the application that has been developed is called “ Centra Retail Store”.

Essentially, what the app does is it requires a user to log in with their account, where upon successful login they can get an idea what Centra stores are about, do a bit of shopping If they wish, and at the end they have the option to evaluate and give feedback about the experience they during the trial of the application/retail store. The user has the option to create a new account in case he does not already have one.

In this application the following topics have been included which are described into detail further on the report:

1. Data Management
2. Labels and Messages
3. Buttons
4. Entry
5. Screen Design – Layout management – primarily using grid
6. Radio buttons and CheckButtons
7. Canvas
8. Sliders
9. Menus
10. Frames
11. Event Management ( Key pressed )
12. “Best Thing I Did “

**[1] Data Management**

[A] The extracts of the working and fully commented code related to the topic or topics

# open 2 text files and read them, username and password

R01 **=** **open(**"Username.txt"**)**

Read\_Username **=** R01**.**read**()**

R02 **=** **open(**"Password.txt"**)**

Read\_Password **=** R02**.**read**()**

**def** Centra\_Login**():**

#executes when trying to log into the account

# opens text files in read mode

NewUsername **=** **open(**"NewUsername.txt"**,**"r"**)**

readUsername **=** NewUsername**.**read**()**

NewPassword **=** **open(**"NewPassword.txt"**,**"r"**)**

readPassword **=** NewPassword**.**read**()**

# if username and password entered in the entry boxes match the text in the text files than it will succesfully login

**if** **(**Username**.**get**()** **==** Read\_Username **or** Username**.**get**()** **==** readUsername**)** **and** **(**Password**.**get**()** **==** Read\_Password **or** Password**.**get**()** **==** readPassword**):**

# print text

**print(**"Succesful login"**)**

#removes grids if login is successful

Centra**.**grid\_remove**()**

Label01**.**grid\_remove**()**

UsernameLabel**.**grid\_remove**()**

UsernameEntry**.**grid\_remove**()**

PasswordLabel**.**grid\_remove**()**

PasswordEntry**.**grid\_remove**()**

LoginButton**.**grid\_remove**()**

# adding a menu

menuBar **=** tk**.**Menu**(**r**)**

r**.**config**(**menu**=**menuBar**)**

SubMenu1 **=** tk**.**Menu**(**menuBar**,**tearoff**=**0**)**

SubMenu2 **=** tk**.**Menu**(**menuBar**,**tearoff**=**0**)**

#add menus to the menu

# Shopping menu linked to SubMenu1 "Food Shopping" that when pressed command is Food\_Shopping

menuBar**.**add\_cascade**(**label**=**"Shopping"**,**

menu**=**SubMenu1**,**

command**=**Food\_Shopping**)**

# Contact menu linked to SubMenu2 "Contact us" that when pressed command is ContactSection

menuBar**.**add\_cascade**(**label**=**"Contact"**,**

menu**=**SubMenu2**)**

# add submenus of the menu

SubMenu1**.**add\_command**(**label**=**"Food Shopping"**,**

command**=**Food\_Shopping**)**

SubMenu2**.**add\_command**(**label**=**"Contact Us"**,**

command**=**ContactSection**)**

#reassign root window size

r**.**geometry**(**"1100x675"**)**

#remove grid

CreateNewAccount**.**grid\_remove**()**

# display canvas

LoginCanvas**.**grid**(**row**=**1**,**column**=**1**)**

#creates an image in the canvas with SouthEast anchor and CentraBG image and height and width

LoginCanvas**.**create\_image**(**1100**,**675**,**anchor**=**"se"**,**image**=**CentraBg**)**

# create a rectangle in the canvas being used to display a similar centra logo

LoginCanvas**.**create\_rectangle**(**10**,**30**,**300**,**140**,**fill**=**"blue"**)**

# create text in the canvas that is positioned inside the rectangle, filled with orange

# font Times 20 italic bold

LoginCanvas**.**create\_text**(**150**,**65**,**fill**=**"orange"**,**font**=**"Times 20 italic bold"**,**

text**=**"CENTRA"**)**

# create text in canvas positioned inside the rectangle under the previous text,

# filled with white and font is Times 20 italic bold underline

LoginCanvas**.**create\_text**(**150**,**100**,**fill**=**"white"**,**font**=**"Times 20 italic bold underline"**,**

text**=**"Brighten up your day"**)**

# created text in canvas than is on after the rectangle on the right stating a title with Bold font and size 17, filled with red color

LoginCanvas**.**create\_text**(**700**,**100**,**fill**=**"red"**,**text**=**CentraTitle**,**font**=(**"Bold"**,**17**))**

# centra mission text created in canvas with custom color and Helvetica font size 14

LoginCanvas**.**create\_text**(**260**,**400**,**fill**=**"#3A4F31"**,**text**=**CentraMission**,**font**=(**"Helvetica"**,**14**))**

**elif** Username**.**get**()** **!=** Read\_Username **or** Password**.**get**()** **!=** Read\_Password**:**

**print(**"Try again"**)**

**def** Back\_To\_Login**():**

# this is executed after creating a new account and going back to log-in area

# open text file in write mode

NewUsername **=** **open(**"NewUsername.txt"**,**"w"**)**

# writes the string username entered in the username entry box

NewUsername**.**write**(**Username**.**get**())**

#close file

NewUsername**.**close**()**

# open file in write mode

NewPassword **=** **open(**"NewPassword.txt"**,**"w"**)**

# write the news string password entered in the entry box

NewPassword**.**write**(**Password**.**get**())**

#close file

NewPassword**.**close**()**

#reconfigures label

Label01**.**configure**(**text**=**"Please enter details below to login"**)**

#reconfigures username label

UsernameLabel**.**configure**(**text**=**"Username\*"**)**

#reconfigures username label

PasswordLabel**.**configure**(**text**=**"Password\*"**)**

# removes create account button

CA**.**grid\_remove**()**

# displays a new loggin button with command Centra\_Login

LoginButton **=** tk**.**Button**(**r**,**text**=**"Login"**,**

bg**=**"blue"**,**

fg**=**"yellow"**,**

command**=**Centra\_Login**)**

#display button at specified row and column with sticky NORTH

LoginButton**.**grid**(**row**=**7**,**column**=**1**,**sticky**=**tk**.**N**)**

**def** text\_review**():**

# gets the full string input from the text area

**input** **=** text\_area**.**get**(**"1.0"**,** "end-1c"**)**

# opens a file in write mode and writes what was inputted in the text area to that file

review\_file **=** **open(**"Customer\_Review.txt"**,**'w'**)**

review\_file**.**write**(input)**

review\_file**.**close**()** # closes the file

#opens the file again this time in read mode

review\_file**=open(**"Customer\_Review.txt"**,**"r"**)**

#prints what was written in the text area in the console window

**print(**review\_file**.**read**())**

# display label in the SliderFrame

Review\_message **=** tk**.**Label**(**SliderFrame**,**text**=**"Thank you so much for the review!\nHave a great day!\nPress ESC to exit application"**,**

font**=(**"Arial"**,**10**))**

Review\_message**.**place**(**x**=**110**,**y**=**530**)**

[B] Explanation of how the code behaves

In the Data Management, I both read and wrote to files. The first instance is reading a file when the user is trying to log in. In my project folder, I already created 2 text files where I stored a username **“Agert2001”** and password **“Centra2022”.** If the user prompts this username and password he is allowed to log-in, otherwise he is told to try again. In case this happens, the user has the option to create a new account, where if he does so he will be prompted to a new window asked to do so. Upon new account creation, the new details will be stored in 2 new separate text files called *New Username* and ***New Password.***

In the last text\_review() function, whatever the user inputs in the text area is written in a file containing the user experience feedback

Files that are read in the project folder are called **Username.txt** and **Password.txt**

Files that I wrote and read to are called **NewUsername.txt**, **NewPassword.txt**

File that I only wrote to is called **Customer\_Review.txt**

**[2] Labels and Messages**

[A] The extracts of the working and fully commented code related to the topic or topics

#text label telling the user to log their details below

Label01 **=** tk**.**Label**(**r**,**

text**=**"Please enter details below to login"**,**

width**=**35**,**

pady**=**10**,**

background**=**"#C6F745"**)**

Label01**.**grid**(**row**=**1**,**column**=**1**,**sticky**=**tk**.**N**)**

#Username text Label, padding is 10 pixels

UsernameLabel **=** tk**.**Label**(**r**,**text**=**"Username\*"**,**

font**=**"arial"**,**

pady**=**10**)**

UsernameLabel**.**grid**(**row**=**2**,**column**=**1**)**

#Password text label , pady is extra space that is added above or below the text within the widget and assigned to 10

PasswordLabel **=** tk**.**Label**(**r**,**text**=**"Password\*"**,**

font**=**"arial"**,**

pady**=**10**)**

#display to respective row and column in the grid

PasswordLabel**.**grid**(**row**=**4**,**column**=**1**)**

#useless label just to occupy space in order to display the login button further down

UselessLabel**=**tk**.**Label**(**r**)**

UselessLabel**.**grid**(**row**=**6**)**

#another useless label to occupy space to display CreateNewAccount button further down

UselessLabel01 **=** tk**.**Label**(**r**)**

UselessLabel01**.**grid**(**row**=**8**)**

#Message widget initialized in main code for easier calls in between functions with main window RSFrame

M2 **=** tk**.**Message**(**RSFrame**)**

#Message widget with main window RSFrame in Shopping section, used later to display total items selected and their price

totalMessage **=** tk**.**Message**(**RSFrame**)**

#Label widget with main window RSFrame, yellow background, 250 width and size 10 with font type Open Sans

Search\_label **=** tk**.**Message**(**RSFrame**,**text**=**"Check if an item is in stock: "**,**

bg**=**"yellow"**,**width**=**250**,**font **=(**'Open Sans'**,**10**))**

#Message widget with parent window RSframe, and 350 width

m1 **=** tk**.**Message**(**RSFrame**,**width**=**350**)**

# label widget with parent window ContactFrame

# will be used later to display info of the centra staff in the contact section

SelectLabel **=** tk**.**Label**(**ContactFrame**)**

manager\_info **=** tk**.**Label**(**ContactFrame**)**

supervisor\_info **=** tk**.**Label**(**ContactFrame**)**

**def** text\_review**():**

# display label with parent window -> SliderFrame, displays text with Arial font, size 10

Review\_message **=** tk**.**Label**(**SliderFrame**,**text**=**"Thank you so much for the review!\nHave a great day!\nPress ESC to exit application"**,**

font**=(**"Arial"**,**10**))**

Review\_message**.**place**(**x**=**110**,**y**=**530**)**

**def** Submit\_Rating**():**

# creates a message widget in the sliderFrame with green background and width 200

RatingMessage **=** tk**.**Message**(**SliderFrame**,**text**=**"Thank you!\nHope to see you back!!"**,**

width**=**200**,**

bg**=**"green"**)**

RatingMessage**.**place**(**x**=**150**,**y**=**150**)**

**def** Finish**():**

# label widget with parent window SliderFrame, displayed as text in Arial font size 10

review **=** tk**.**Label**(**SliderFrame**,**text**=**"How happy were you from 0 to 100 with the service?"**,**

font**=(**"Arial"**,**10**))**

review**.**place**(**x**=**100**,**y**=**30**)** # will appear at these co-ordinates in the frame

# label widget in SliderFrame with parent window SliderFrame, displays text in Arial font size 10

textlabel**=**tk**.**Label**(**SliderFrame**,**

text**=**"Please, could you provide extra feedback on how we could improve our service?"**,**

font**=(**"Arial"**,**10**))**

textlabel**.**place**(**x**=**25**,**y**=**220**)** # will appear at these co-ordinates in the frame

**def** TotalPrice**():**

# reconfigures message widget to display all the names of items selected and their total price

#relief raised makes it stick out a bit more, looking more proffesional, custom background, width 450

# so even if the message is long it kinda stays in 1 line

totalMessage**.**config**(**text**=**"Total items added are:" **+** **str(**nameArray**)** **+** " with final price of "**+str(**summ**)+**" €"**,**

width**=**450**,**bg**=**"#B2C13A"**,**relief**=**"raised"**)** # raised makes the message stick out more

totalMessage**.**place**(**x**=**10**,**y**=**500**)**# will appear at these co-ordinates in the frame

**def** SearchItem**():**

# reconfigures message widget that displays text

m1**.**config**(**text **=** "Nothing has been searched and/or is not available yet"**)**

m1**.**place**(**x**=**40**,**y**=**50**)** # will appear at these co-ordinates in the frame

# productName runs through the names, productPrice runs through the prices in the Items\_on\_Stock array

**for** productName**,**productPrice **in** Items\_on\_stock**:**

# if string inside search entry box is equal to a product name executes

**if** Search**.**get**()** **==** productName**:**

# reconfigures message widget to display product name and its price

m1**.**config**(**text**=**productName**+**" is available and it's price is: "**+str(**productPrice**)+**" euro"**)**

# moves the product price to tempPrice

tempPrice **=** productPrice

#add the product name that was searched into the nameArray, later used to display all the item names selected

nameArray**.**append**(**productName**)**

m1**.**place**(**x**=**40**,**y**=**50**)** # will appear at these co-ordinates in the frame

Add\_to\_Basket**.**place**(**x **=** 380**,**y**=**50**)** # will appear at these co-ordinates in the frame

**def** Back\_To\_Login**():**

#reconfigures label

Label01**.**configure**(**text**=**"Please enter details below to login"**)**

#reconfigures username label

UsernameLabel**.**configure**(**text**=**"Username\*"**)**

#reconfigures username label

PasswordLabel**.**configure**(**text**=**"Password\*"**)**

**def** NewAccount\_Display**():**

#executed when trying to create a new account to log-in

# reconfigure label

Label01**.**configure**(**text**=**"Please enter new account details"**)**

# reconfigure username and password label

UsernameLabel**.**configure**(**text**=**"Create New Username\*"**)**

PasswordLabel**.**configure**(**text**=**"Create New Password\*"**)**

**def** addPrice**():**

#x1 runs through names, y1 runs through prices

**for** x1 **,** y1 **in** Items\_on\_stock**:**

# if radiobutton value is equal to a price value y1 in item\_on\_stock array it executes

**if** **float(**v**.**get**())** **==** y1**:**

# reconfigures message to display name of item and its price, message has width of 400

M2**.**config**(**text **=** x1 **+**" has price of: "**+str(**y1**)+**" €" **+**" and it has been added to basket"**,**

width**=**400**)**

M2**.**place**(**x **=** 50**,**y **=**350**)** # will appear at these co-ordinates in the frame

# adds the name of this item in the nameArray

nameArray**.**append**(**x1**)**

#print the array of names in the console window

**print(**nameArray**)**

**def** Food\_Shopping**():**

# configures label with parent window root, silver background and width 20

space\_label**=**tk**.**Label**(**r**,**bg**=**"silver"**,**width**=**20**)**

# displays at assigned row and column with sticky NORTH

space\_label**.**grid**(**row**=**1**,**column**=**5**,**sticky**=**tk**.**N**)**

# will appear at these co-ordinates in the frame or root window

Search\_label**.**place**(**x**=**40**,**y**=**10**)**

**def** ContactSection**():**

# configure label with new text and font, size 13

SelectLabel**.**config**(**text**=**"Select which staff of our Centra would u like to contact"**,**

font **=(**"Italic Bold"**,**13**))**

SelectLabel**.**place**(**x**=**10**,**y**=**10**)** # will appear at these co-ordinates in the frame

[B] Explanation of how the code behaves

A lot of labels and messages were used in this application, as obviously they are very necessary and can probably always occur when developing applications.

When using labels, the parent window assigned was either the root or sometimes a frame. Options frequently used where pady which basically determines how much extra space u want added above and below the text within the widget. I set this particular option pady=10 to add space between the Username and password label with the entry boxes so it looks a bit prettier. I used different fonts for my labels and messages in this application but most frequent was Arial and size 10. I could of use relief but I felt it was unnecessary but I did use that option later on the entry boxes, which is discussed further down in the report. Text was the most frequent option used and it is obvious as that is the main job of a label to display text, or either images

In one instance, in the login gui, I used useless invisible labels to occupy a row so I could display another widget further down. This is useful when the root window doesn’t have a fixed geometry as even if u assign a higher row, the widget will only display at the last row or column in the grid. A way to fix this not having to create useless widgets is creating a frame and putting the widgets all in a frame or by using place(x,y) in the grid

**[2] Buttons**

[A] The extracts of the working and fully commented code related to the topic or topics

#Login Button with command Centra\_Login when pressed, blue background and yellow text foreground, when pressed executes command

LoginButton **=** tk**.**Button**(**r**,**text**=**"Login"**,**

bg**=**"blue"**,**

fg**=**"yellow"**,**

command**=**Centra\_Login**)**

#displays at respective row column in the grid sticking up NORTH

LoginButton**.**grid**(**row**=**7**,**column**=**1**,**sticky**=**tk**.**N**)**

#Button that when pressed has command NewAccount\_Display, where a new account can be created

#silver background and executes command when pressed

CreateNewAccount **=** tk**.**Button**(**r**,**text**=**"Create a new Account?"**,**

bg **=** "silver"**,**

command **=** NewAccount\_Display**)**

#displays at respective row column in the grid sticking up NORTH

CreateNewAccount**.**grid**(**row**=**9**,**column**=**1**,**sticky**=**tk**.**N**)**

#initialized button in main code, so it's easier to use later and call in between functions

CA **=** tk**.**Button**(**r**)**

#Search button with main window RSFrame, green background

#executes command if pressed

Search\_button **=** tk**.**Button**(**RSFrame**,**text**=**"SEARCH ITEM"**,**

bg **=** "green"**,**

command**=**SearchItem**)**

#Button with parent window RSFrame and command AddtoBasket if pressed

Add\_to\_Basket **=** tk**.**Button**(**RSFrame**,**text**=**"Add item to Basket"**,**

command**=**AddtoBasket**)**

# Button with parent window RSFRame and command TotalPrice, custom background

total\_price\_topSellers **=** tk**.**Button**(**RSFrame**,**text**=**"Calculate Total price"**,**

bg**=**"#641994"**,**

command**=**TotalPrice**)**

# finish shopping button with parent window RSFrame and command Finish, brown background

Finish **=** tk**.**Button**(**RSFrame**,**text**=**"Finish Shopping"**,**

bg **=** "brown"**,**

command**=**Finish**)**

# Button for the text area, when pressed executes command text\_review, parent window is SliderFrame

textArea\_Button **=** tk**.**Button**(**SliderFrame**,**text**=**"Submit Review"**,**

command**=** text\_review**)**

**def** Finish**():**

# Rating Button with yellow background and submit\_rating command when pressed, parent window is Slider Frame

# yellow background

RatingButton **=** tk**.**Button**(**SliderFrame**,**text**=**"Submit Rating"**,**

bg**=**"yellow"**,**

command**=**Submit\_Rating**)**

RatingButton**.**place**(**x**=**185**,**y**=**110**)** # will appear at these co-ordinates in the frame

**def** Back\_To\_Login**():**

# displays a new loggin button with command Centra\_Login if pressed

# blue background and yellow text foreground

LoginButton **=** tk**.**Button**(**r**,**text**=**"Login"**,**

bg**=**"blue"**,**

fg**=**"yellow"**,**

command**=**Centra\_Login**)**

#display button at specified row and column with sticky NORTH

LoginButton**.**grid**(**row**=**7**,**column**=**1**,**sticky**=**tk**.**N**)**

**def** NewAccount\_Display**():**

# reconfigure button with silver background and command that brings user back to log-in area when pressed

CA**.**configure**(**text**=**"Create Account"**,**

bg **=** "silver"**,**

command **=** Back\_To\_Login**)**

# displays button at specified row and grid with sticky NORTH

CA**.**grid**(**row**=**9**,**column**=**1**,**sticky**=**tk**.**N**)**

[B] Explanation of how the code behaves

Buttons are also a really important widget when developing GUI applications using Tkinter toolkit. Main options in the buttons in my application were bg = assigns the background pic or color of the button. Fg = assigns the color of the text in this widget

Parent windows was assigned the root, or a frame

In the main code there is a login button which if successful, will direct the user to the next step, if not successful there is a create new account button which directs the user to create new account NewAccount\_Display() and than if button pressed will be directed back to login area

Back\_To\_Login()

In the Finish() function, there is a button that displays text after the user gives a rating using the slider

For my Buttons I did not use any other options such as relief,pady etc as I did not feel it was necessary

**[4] Entry**

[A] The extracts of the working and fully commented code related to the topic or topics

#variable that is used to get the string entered in the entry box for username login

Username **=** tk**.**StringVar**()**

#Entry box to input the username string, light blue background, relief is solid so the entry box looks as name says " solid "

UsernameEntry **=** tk**.**Entry**(**r**,**textvariable**=**Username**,**

bg**=**"light blue"**,**

relief**=**"solid"**)**

UsernameEntry**.**grid**(**row**=**3**,**column**=**1**)**

#String variable used to get the string entered in the password entry box

Password**=**tk**.**StringVar**()**

#Password entry box, instead of plain characters it puts "\*" for security reasons using show option

#light blue background and solid relief making it look more solid

PasswordEntry **=** tk**.**Entry**(**r**,**textvariable**=**Password**,**

bg**=**"light blue"**,**

show**=**"\*"**,**

relief**=**"solid"**)**

PasswordEntry**.**grid**(**row**=**5**,**column**=**1**)**

# variable to get the string when searching an item in the Search\_item entry box, after the search\_button is pressed

Search **=** tk**.**StringVar**()**

#Search entry box with main window RSFrame and textvariable Search, width 30 and silver background,

#similar relief to previous entries

#initialized in the main code so easier to use in between functions

Search\_item **=** tk**.**Entry**(**RSFrame**,**textvariable**=**Search**,**width**=**30**,**

relief**=**"solid"**,**

bg **=** "silver"**)**

**def** Food\_Shopping**():**

# will appear at these co-ordinates in the frame

Search\_item**.**place**(**x**=**220**,**y**=**13**)**

[B] Explanation of how the code behaves

Entries in my application were used for the login gui, in order to check if username and password match what’s stored in the file, and later on for searching an item in the centra store

Parent window in the login gui was the root window, than the search item entry was used in a frame. The most common options used was **relief** to “ solid “ but could have used “ raised “ aswell to make it stick out more but I believe solid looks more professional for entry boxes. Other options used were **width**, **background** and **textvariable** ( variable that is used to get the string entered in the entry box ), and also **show** = “\*” that puts \*\*\*\*\*\* instead of the plain string password, making the user feel more secure if inserting their account details in a public place.

Entry Boxes are really important too and their main point of use is to retrieve data as strings/numbers or compare information ( used so in the login area gui) etc

[5] Screen Design – Layout Management – Grid()

[A] The extracts of the working and fully commented code related to the topic or topics

# display in respective row and column in the grid

Centra**.**grid**(**row**=**0**,**column**=**1**)**

Label01**.**grid**(**row**=**1**,**column**=**1**,**sticky**=**tk**.**N**)** # sticks NORTH ( middle top )

UsernameLabel**.**grid**(**row**=**2**,**column**=**1**)**

UsernameEntry**.**grid**(**row**=**3**,**column**=**1**)**

PasswordLabel**.**grid**(**row**=**4**,**column**=**1**)**

PasswordEntry**.**grid**(**row**=**5**,**column**=**1**)**

UselessLabel**.**grid**(**row**=**6**)**

LoginButton**.**grid**(**row**=**7**,**column**=**1**,**sticky**=**tk**.**N**)** # sticks NORTH ( middle top )

UselessLabel01**.**grid**(**row**=**8**)**

CreateNewAccount**.**grid**(**row**=**9**,**column**=**1**,**sticky**=**tk**.**N**)**# sticks NORTH ( middle top )

**def** text\_review**():**

# displays widget at the coordinates assigned in the root window or frame , x is horizontal offset in pixels, y is vertical offset in pixels

Review\_message**.**place**(**x**=**110**,**y**=**530**)**

RatingMessage**.**place**(**x**=**150**,**y**=**150**)**

text\_area**.**place**(**x**=**40**,**y**=**250**)**

textArea\_Button**.**place**(**x**=**170**,**y**=**500**)**

textlabel**.**place**(**x**=**25**,**y**=**220**)**

RatingButton**.**place**(**x**=**185**,**y**=**110**)**

**def** Finish**():**

review**.**place**(**x**=**100**,**y**=**30**)**

Slider**.**place**(**x**=**175**,**y**=**60**)**

**def** TotalPrice**():**

totalMessage**.**place**(**x**=**10**,**y**=**500**)**

**def** SearchItem**():**

m1**.**place**(**x**=**40**,**y**=**50**)**

m1**.**place**(**x**=**40**,**y**=**50**)**

Add\_to\_Basket**.**place**(**x **=** 380**,**y**=**50**)**

**def** Back\_To\_Login**():**

# display in respective row and column in the grid, sticking up middle top or NORTH

LoginButton**.**grid**(**row**=**7**,**column**=**1**,**sticky**=**tk**.**N**)**

**def** NewAccount\_Display**():**

# removes or hides grid but does not destroy it

LoginButton**.**grid\_remove**()**

CreateNewAccount**.**grid\_remove**()**

CA**.**grid**(**row**=**9**,**column**=**1**,**sticky**=**tk**.**N**)**

**def** addPrice**():**

M2**.**place**(**x **=** 50**,**y **=**350**)**

**def** Food\_Shopping**():**

# remove grids from tkinter window

ContactFrame**.**grid\_remove**()**

SliderFrame**.**grid\_remove**()**

LoginCanvas**.**grid\_remove**()**

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**1**)**

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**2**)**

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**3**)**

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**4**)**

Search\_label**.**place**(**x**=**40**,**y**=**10**)**

Search\_item**.**place**(**x**=**220**,**y**=**13**)**

Search\_button**.**place**(**x**=**410**,**y**=**10**)**

total\_price\_topSellers**.**place**(**x**=**350**,**y**=**630**)**

Finish**.**place**(**x**=**10**,**y**=**630**)**

# displays at assigned row and column

TopSellersFrame**.**grid**(**row**=**1**,**column**=**1**)**

RSFrame**.**grid**(**row**=**1**,**column**=**2**)**

**def** displayInfo**():**

manager\_info**.**place**(**x**=**10**,**y**=**130**)**

supervisor\_info**.**place**(**x**=**10**,**y**=**350**)**

**def** ContactSection**():**

#removes grids

LoginButton**.**grid\_remove**()**

LoginCanvas**.**grid\_remove**()**

TopSellersFrame**.**grid\_remove**()**

RSFrame**.**grid\_remove**()**

ContactFrame**.**grid**(**row**=**1**,**column**=**1**)**

SelectLabel**.**place**(**x**=**10**,**y**=**10**)**

Manager**.**place**(**x**=**10**,**y**=**60**)**

Supervisor**.**place**(**x**=**10**,**y**=**250**)**

**def** Centra\_Login**():**

#removes grids if login is successful

Centra**.**grid\_remove**()**

Label01**.**grid\_remove**()**

UsernameLabel**.**grid\_remove**()**

UsernameEntry**.**grid\_remove**()**

PasswordLabel**.**grid\_remove**()**

PasswordEntry**.**grid\_remove**()**

LoginButton**.**grid\_remove**()**

[B] Explanation of how the code behaves

For my application the only display method used is Grid() using rows and columns or place(x,y) where x is the horizontal offset in pixels and y is the vertical offset in pixels

The most frequent way I did it throughout this semester was using rows and columns but in this project, I used place a lot. I found place to be quite handy to use when trying to place widgets in different places with lots of space between them

Other geometry manager I could have used was pack() but I felt with the amount of widgets I had it would not be as efficient as grid() or place()

I mostly used Frames to hold my widgets especially after successful login. Frames are super useful when creating complex layouts as they group widgets together

[6] Radio Buttons and Checkbuttons

[A] The extracts of the working and fully commented code related to the topic or topics

# assigned all photoImages to variables that will be used to display items in the Shopping section as RadioButtons

# subsample was used to decrease height and width of the images so they relatively are all the same size

AppleImage **=** tk**.**PhotoImage**(**file**=**"Apple.gif"**)**

AppleImage **=** AppleImage**.**subsample**(**2**,**3**)**

SweetsImage **=** tk**.**PhotoImage**(**file**=**"Sweets.gif"**)**

SweetsImage **=** SweetsImage**.**subsample**(**5**,**5**)**

DonutImage **=** tk**.**PhotoImage**(**file**=**"Donut.gif"**)**

DonutImage **=** DonutImage**.**subsample**(**4**,**4**)**

CoffeeImage **=** tk**.**PhotoImage**(**file**=**"Coffee.gif"**)**

CoffeeImage **=** CoffeeImage**.**subsample**(**6**,**6**)**

CroissantImage **=** tk**.**PhotoImage**(**file**=**"Croissant.gif"**)**

CroissantImage **=** CroissantImage**.**subsample**(**3**,**4**)**

CremEggImage **=** tk**.**PhotoImage**(**file**=**"CremEgg.gif"**)**

CremEggImage **=** CremEggImage**.**subsample**(**7**,**7**)**

SlusheeImage **=** tk**.**PhotoImage**(**file**=**"Slushee.gif"**)**

SlusheeImage **=** SlusheeImage**.**subsample**(**2**,**3**)**

TeaImage **=** tk**.**PhotoImage**(**file**=**"Tea.gif"**)**

TeaImage **=** TeaImage**.**subsample**(**3**,**6**)**

BreadImage **=** tk**.**PhotoImage**(**file**=**"Bread.gif"**)**

BreadImage **=** BreadImage**.**subsample**(**10**,**10**)**

CakeImage **=** tk**.**PhotoImage**(**file**=**"Cake.gif"**)**

CakeImage **=** CakeImage**.**subsample**(**3**,**2**)**

AeroImage **=** tk**.**PhotoImage**(**file**=**"Aero.gif"**)**

AeroImage **=** AeroImage**.**subsample**(**2**,**2**)**

HunkyDoryImage **=** tk**.**PhotoImage**(**file**=**"HunkyDory.gif"**)**

HunkyDoryImage **=** HunkyDoryImage**.**subsample**(**2**,**3**)**

TaytoImage **=** tk**.**PhotoImage**(**file**=**"Tayto.gif"**)**

TaytoImage **=** TaytoImage**.**subsample**(**2**,**3**)**

StampImage **=** tk**.**PhotoImage**(**file**=**"Stamp.gif"**)**

StampImage **=** StampImage**.**subsample**(**10**,**10**)**

ChickenImage **=** tk**.**PhotoImage**(**file**=**"Chicken.gif"**)**

ChickenImage **=** ChickenImage**.**subsample**(**2**,**3**)**

CheeseImage **=** tk**.**PhotoImage**(**file**=**"Cheese.gif"**)**

CheeseImage **=** CheeseImage**.**subsample**(**5**,**5**)**

DogFoodImage **=** tk**.**PhotoImage**(**file**=**"DogFood.gif"**)**

DogFoodImage **=** DogFoodImage**.**subsample**(**15**,**17**)**

NutsImage **=** tk**.**PhotoImage**(**file**=**"Nuts.gif"**)**

NutsImage **=** NutsImage**.**subsample**(**4**,**5**)**

PenImage **=** tk**.**PhotoImage**(**file**=**"Pen.gif"**)**

PenImage **=** PenImage**.**subsample**(**6**,**7**)**

MuffinImage **=** tk**.**PhotoImage**(**file**=**"Mufin.gif"**)**

MuffinImage **=** MuffinImage**.**subsample**(**4**,**4**)**

#4 arrays that contain a counter, PhotoImage, name of Item and its price

# used to display the radiobuttons as items in the shopping section

#in order to display them in separate columns had to divide them into 4 different arrays

Items\_left **=** **[(**1**,** AppleImage**,** "Apple"**,** 0.7**),**

**(**2**,** SweetsImage**,** "Sweets"**,** 1.2**),**

**(**3**,** DonutImage**,** "Donut"**,** 1.1**),**

**(**4**,** CoffeeImage**,** "Coffee"**,** 2.8**),**

**(**5**,** CroissantImage**,** "Croissant"**,** 0.8**)]**

Items\_left01 **=** **[(**1**,** CremEggImage**,** "CremEgg"**,** 0.9**),**

**(**2**,** SlusheeImage**,** "Slushee"**,** 2.5**),**

**(**3**,** TeaImage**,** "Tea"**,** 2.3**),**

**(**4**,** BreadImage**,** "Bread"**,** 2**),**

**(**5**,** CakeImage**,** "Cake"**,** 5.5**)]**

Items\_right **=** **[(**1**,** AeroImage**,** "Aero"**,** 1.4**),**

**(**2**,** HunkyDoryImage**,** "HunkyDory"**,** 2.7**),**

**(**3**,** TaytoImage**,** "Tayto"**,** 1.8**),**

**(**4**,** StampImage**,** "Stamp"**,** 1**),**

**(**5**,** ChickenImage**,** "Chicken"**,** 5.2**)]**

Items\_right01 **=** **[(**1**,** CheeseImage**,** "Cheese"**,** 1.6**),**

**(**2**,** DogFoodImage**,** "DogFood"**,** 6.5**),**

**(**3**,** NutsImage**,** "Nuts"**,** 0.54**),**

**(**4**,** PenImage**,** "Pens"**,** 0.35**),**

**(**5**,** MuffinImage**,** "Muffin"**,** 1.5**)]**

# array with all the item names on stock and their prices

Items\_on\_stock **=** **[(**"Apple"**,** 0.7**),**

**(**"Sweets"**,** 1.2**),**

**(**"Donut"**,** 1.1**),**

**(**"Coffee"**,** 2.8**),**

**(**"Croissant"**,** 0.8**),**

**(**"CremEgg"**,** 0.9**),**

**(**"Slushee"**,** 2.5**),**

**(**"Tea"**,** 2.3**),**

**(**"Bread"**,** 2**),**

**(**"Cake"**,** 5.5**),**

**(**"Aero"**,** 1.4**),**

**(**"HunkyDory"**,** 2.7**),**

**(**"Tayto"**,** 1.8**),**

**(**"Stamp"**,** 1**),**

**(**"Chicken"**,** 5.2**),**

**(**"Cheese"**,** 1.6**),**

**(**"DogFood"**,** 6.5**),**

**(**"Nuts"**,** 0.54**),**

**(**"Pens"**,** 0.35**),**

**(**"Muffin"**,** 1.5**)]**

# checkbuttons and labels with parent windows ContactFrame

# will be used later to display info of the centra staff in the contact section

Manager **=** tk**.**Checkbutton**(**ContactFrame**)**

Supervisor **=** tk**.**Checkbutton**(**ContactFrame**)**

**def** Food\_Shopping**():**

# calls global variables

**global** ImageCounter

**global** FoodImage

**global** name

**global** price

#for loops with 4 variables to access 4-dimensional arrays

#needed 4 variables otherwise I would get error that it wouldn't unpack

# loop creates 5 Radiobuttons on the left Side column 1

# more efficient than writing every radiobutton one by one

# Radiobutton has parrent window TopSellersFrame, text is the name variable

# image is FoodImage , value is price and command is addPrice

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_left**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

# column 1 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**1**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_left01**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 2 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**2**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_right**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 3 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**3**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_right01**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 4 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**4**)**

**def** ContactSection**():**

#removes grids

LoginButton**.**grid\_remove**()**

LoginCanvas**.**grid\_remove**()**

TopSellersFrame**.**grid\_remove**()**

RSFrame**.**grid\_remove**()**

#reassign root window size

r**.**geometry**(**"450x600"**)**

# configures frame with new height and width and green background

ContactFrame**.**config**(**height**=**600**,**width**=**500**,**bg**=**"green"**)**

#display at row 1 and column 2

ContactFrame**.**grid**(**row**=**1**,**column**=**1**)**

# configure label with new text and font, size 13

SelectLabel**.**config**(**text**=**"Select which staff of our Centra would u like to contact"**,**

font **=(**"Italic Bold"**,**13**))**

SelectLabel**.**place**(**x**=**10**,**y**=**10**)** # will appear at these co-ordinates in the frame

# configures manager checkbutton with variable t, and onvalue=1,offvalue=0

# yellow background and Arial font with size 20

# command is displayInfo when pressed

Manager**.**config**(**text**=**"Manager: David Cruise"**,**

variable **=** t**,**

onvalue**=**1**,**

offvalue**=**0**,**

command**=**displayInfo**,**

bg**=**"yellow"**,**

font**=(**"Arial"**,**20**))**

# will appear at these co-ordinates in the frame

Manager**.**place**(**x**=**10**,**y**=**60**)**

# confugres supervisor checkbuttons with variable s, onvalue=1,offvalue=0

# yellow background and ariall font with size 20

# command is displayInfo when pressed

Supervisor**.**config**(**text**=**"Supervisor: Agert Berisha"**,**

variable **=** s**,**

onvalue**=**1**,**

offvalue**=**0**,**

command**=**displayInfo**,**

bg**=**"yellow"**,**

font**=(**"Arial"**,**20**))**

# will appear at these co-ordinates in the frame

Supervisor**.**place**(**x**=**10**,**y**=**250**)**

**def** displayInfo**():**

# if t value is 1 display manager info

**if** **(**t**.**get**()** **==**1**):**

# displays text details with Roman times font size 10 in a light blue background

manager\_info**.**config**(**text**=**"Phone: 083 333 3333\nEmail: davidcruise01@gmail.com\nAddress: Centra Greenpark, Clondalkin,Dublin"**,**

font **=(**"Roman Times"**,**10**),**

bg**=**"light blue"**)**

manager\_info**.**place**(**x**=**10**,**y**=**130**)**# will appear at these co-ordinates in the frame

# if s value is 1, displays supervisor info

**if** **(**s**.**get**()==**1**):**

# displays text details with Roman Times font size 10 in a light blue background

supervisor\_info**.**config**(**text**=**"Phone: 083 123 3211\nEmail: agertb01@gmail.com\nAddress: Centra Greenpark, Clondalkin,Dublin"**,**

font **=(**"Roman Times"**,**10**),**

bg**=**"light blue"**)**

supervisor\_info**.**place**(**x**=**10**,**y**=**350**)** # will appear at these co-ordinates in the frame

**def** Centra\_Login**():**

# adding a menu

menuBar **=** tk**.**Menu**(**r**)**

r**.**config**(**menu**=**menuBar**)**

SubMenu1 **=** tk**.**Menu**(**menuBar**,**tearoff**=**0**)**

SubMenu2 **=** tk**.**Menu**(**menuBar**,**tearoff**=**0**)**

#add menus to the menu

# Shopping menu linked to SubMenu1 "Food Shopping" that when pressed command is Food\_Shopping

menuBar**.**add\_cascade**(**label**=**"Shopping"**,**

menu**=**SubMenu1**,**

command**=**Food\_Shopping**)**

# add submenus of the menu

SubMenu1**.**add\_command**(**label**=**"Food Shopping"**,**

command**=**Food\_Shopping**)**

[B] Explanation of how the code behaves

In my application, Radio Buttons were used to display items in the Food\_Shopping() function with images and a value set for each item.

4 arrays were created with each array being 4 – dimensional basically meaning its holding 4 different types of data types, first being the counter ( 1,2,3,4,5 ) and than the name of the Image ( PhotoImage ), name of the product and it’s price

What happens during the for loop is that ImageCounter runs through the numbers in the array (0,1,2,3,4,5) and assigns that to the row for each radiobutton. The FoodImage variable runs through the PhotoImage Names of the products that we already downloaded into our file and assigns that photoImage to each radiobutton. Name variable runs through the string names and assigns each radiobutton it’s name. Lastly the price variable runs through the prices and this is set to the value option so during the loop each radiobutton is set a price.

I felt that using a for loop this way was the most efficient way as creating radio buttons 1 by 1 would take a lot of time and lots of extra lines of code.

As for the check buttons, this were used in the contact section to display the information of the manager or supervisor if a customer would like to contact them personally. By pressing the check button the information Is displayed

**[7] Canvas**

[A] The extracts of the working and fully commented code related to the topic or topics

#String variable with centra title message

CentraTitle **=** "Centra is Ireland's leading convenience retail group,\nwith 480 bright accessible stores in communities throughout the country"

#Centra mission string text containing centra mission

CentraMission **=** "We have a reputation for quality, value and friendly service\nand the fact that stores are independently owned\nand operated by local people means that shoppers \nget the best of both worlds:\n-Commitment to the traditional values of good fresh foods\n-Good fresh foods and excellent service\n\nCentra Stores provide you with convenient shopping\nsolutions, have all the items you need and are good\nvalue for money\n\nHappy to continue shopping?\n -Go to \*Shopping\* section\n\nWant to contact us?\n -Go to \*Contact\* section (please leave a review if u can!)"

**def** Centra\_Login**():**

# display canvas

LoginCanvas**.**grid**(**row**=**1**,**column**=**1**)**

#creates an image in the canvas with SouthEast anchor and CentraBG image and height and width

LoginCanvas**.**create\_image**(**1100**,**675**,**anchor**=**"se"**,**image**=**CentraBg**)**

# create a rectangle in the canvas being used to display a similar centra logo

LoginCanvas**.**create\_rectangle**(**10**,**30**,**300**,**140**,**fill**=**"blue"**)**

# create text in the canvas that is positioned inside the rectangle, filled with orange

# font Times 20 italic bold

LoginCanvas**.**create\_text**(**150**,**65**,**fill**=**"orange"**,**font**=**"Times 20 italic bold"**,**

text**=**"CENTRA"**)**

# create text in canvas positioned inside the rectangle under the previous text,

# filled with white and font is Times 20 italic bold underline

LoginCanvas**.**create\_text**(**150**,**100**,**fill**=**"white"**,**font**=**"Times 20 italic bold underline"**,**

text**=**"Brighten up your day"**)**

# created text in canvas than is on after the rectangle on the right stating a title with Bold font and size 17, filled with red color

LoginCanvas**.**create\_text**(**700**,**100**,**fill**=**"red"**,**text**=**CentraTitle**,**font**=(**"Bold"**,**17**))**

# centra mission text created in canvas with custom color and Helvetica font size 14

LoginCanvas**.**create\_text**(**260**,**400**,**fill**=**"#3A4F31"**,**text**=**CentraMission**,**font**=(**"Helvetica"**,**14**))**

[B] Explanation of how the code behaves

In my application, the canvas is shown after user log-in is successful. On the Top-left there is a rectangle with a Centra text inside of it, and on the bottom right of the canvas there is an image of a centra store. Text is created in the middle of the canvas mainly describing the mission of centra and how to navigate the application in case the customer wants to shop around.

Anchor in the image of the canvas was put to SE(South East ) but the image was subsampled to reduce its size.

**[8] Sliders**

[A] The extracts of the working and fully commented code related to the topic or topics

#Slider frame intialized in main code with parent window root tkinter

#will be used after finishing shopping button has been pressed

SliderFrame **=** tk**.**Frame**(**r**)**

# slider scale with values from 0 to 100 and displays horizontally, parent window is SliderFrame

Slider **=** tk**.**Scale**(**SliderFrame**,**from\_**=**0**,**to**=**100**,** orient**=**"horizontal"**)**

**def** Finish**():**

#configures SliderFrame with height, width and light blue background

SliderFrame**.**config**(**height**=**600**,**width**=**500**,**bg**=**"light blue"**)**

SliderFrame**.**grid**(**row**=**1**,**column**=**1**)**

Slider**.**place**(**x**=**175**,**y**=**60**)** # will appear at these co-ordinates in the frame

# label widget with parent window SliderFrame, displayed as text in Arial font size 10

# label widget with parent window SliderFrame, displayed as text in Arial font size 10

review **=** tk**.**Label**(**SliderFrame**,**text**=**"How happy were you from 0 to 100 with the service?"**,**

font**=(**"Arial"**,**10**))**

review**.**place**(**x**=**100**,**y**=**30**)** # will appear at these co-ordinates in the frame

# Rating Button with yellow background and submit\_rating command when pressed, parent window is Slider Frame

# yellow background

RatingButton **=** tk**.**Button**(**SliderFrame**,**text**=**"Submit Rating"**,**

bg**=**"yellow"**,**

command**=**Submit\_Rating**)**

RatingButton**.**place**(**x**=**185**,**y**=**110**)** # will appear at these co-ordinates in the frame

# label widget in SliderFrame with parent window SliderFrame, displays text in Arial font size 10

textlabel**=**tk**.**Label**(**SliderFrame**,**

text**=**"Please, could you provide extra feedback on how we could improve our service?"**,**

font**=(**"Arial"**,**10**))**

textlabel**.**place**(**x**=**25**,**y**=**220**)** # will appear at these co-ordinates in the frame

# reconfigures text area with white background and height and width

text\_area**.**config**(**bg**=**"white"**,**height**=**15**,**width**=**50**)**

text\_area**.**place**(**x**=**40**,**y**=**250**)** # will appear at these co-ordinates in the frame

textArea\_Button**.**place**(**x**=**170**,**y**=**500**)** # will appear at these co-ordinates in the frame

**def** Submit\_Rating**():**

# creates a message widget in the sliderFrame with green background and width 200

RatingMessage **=** tk**.**Message**(**SliderFrame**,**text**=**"Thank you!\nHope to see you back!!"**,**

width**=**200**,**

bg**=**"green"**)**

RatingMessage**.**place**(**x**=**150**,**y**=**150**)**

**def** text\_review**():**

# gets the full string input from the text area

**input** **=** text\_area**.**get**(**"1.0"**,** "end-1c"**)**

# opens a file in write mode and writes what was inputted in the text area to that file

review\_file **=** **open(**"Customer\_Review.txt"**,**'w'**)**

review\_file**.**write**(input)**

review\_file**.**close**()** # closes the file

#opens the file again this time in read mode

review\_file**=open(**"Customer\_Review.txt"**,**"r"**)**

#prints what was written in the text area in the console window

**print(**review\_file**.**read**())**

# display label with parent window -> SliderFrame, displays text with Arial font, size 10

Review\_message **=** tk**.**Label**(**SliderFrame**,**text**=**"Thank you so much for the review!\nHave a great day!\nPress ESC to exit application"**,**

font**=(**"Arial"**,**10**))**

Review\_message**.**place**(**x**=**110**,**y**=**530**)**

[B] Explanation of how the code behaves

Slider was created and initialized in the main code so its easier to call out in between functions if needed. The slider’s parent window was the frame where the customer could reach after pressing the finish shopping button after at the Food\_Shopping() frame.

I noticed that when creating a slider and assigning the range option u have to be careful to add ‘\_’ at from\_ = 0, to =100 as you’d get an error. I didn’t notice that at first and it could me several minutes until I found out why

When the customer slides the slider to determine the satisfaction of his experience he can press the button, which than displays a text label thanking the customer for the feedback.

After that, the customer is kindly asked to provide written feedback in a text area of how we can improve our service. This feedback is than written into a file in our project folder

[9] Menus

[A] The extracts of the working and fully commented code related to the topic or topics

**def** Centra\_Login**():**

#executes when trying to log into the account

# opens text files in read mode

NewUsername **=** **open(**"NewUsername.txt"**,**"r"**)**

readUsername **=** NewUsername**.**read**()**

NewPassword **=** **open(**"NewPassword.txt"**,**"r"**)**

readPassword **=** NewPassword**.**read**()**

# if username and password entered in the entry boxes match the text in the text files than it will succesfully login

**if** **(**Username**.**get**()** **==** Read\_Username **or** Username**.**get**()** **==** readUsername**)** **and** **(**Password**.**get**()** **==** Read\_Password **or** Password**.**get**()** **==** readPassword**):**

# print text

**print(**"Succesful login"**)**

#removes grids if login is successful

Centra**.**grid\_remove**()**

Label01**.**grid\_remove**()**

UsernameLabel**.**grid\_remove**()**

UsernameEntry**.**grid\_remove**()**

PasswordLabel**.**grid\_remove**()**

PasswordEntry**.**grid\_remove**()**

LoginButton**.**grid\_remove**()**

# adding a menu

menuBar **=** tk**.**Menu**(**r**)**

r**.**config**(**menu**=**menuBar**)**

SubMenu1 **=** tk**.**Menu**(**menuBar**,**tearoff**=**0**)**

SubMenu2 **=** tk**.**Menu**(**menuBar**,**tearoff**=**0**)**

#add menus to the menu

# Shopping menu linked to SubMenu1 "Food Shopping" that when pressed command is Food\_Shopping

menuBar**.**add\_cascade**(**label**=**"Shopping"**,**

menu**=**SubMenu1**,**

command**=**Food\_Shopping**)**

# Contact menu linked to SubMenu2 "Contact us" that when pressed command is ContactSection

menuBar**.**add\_cascade**(**label**=**"Contact"**,**

menu**=**SubMenu2**)**

# add submenus of the menu

SubMenu1**.**add\_command**(**label**=**"Food Shopping"**,**

command**=**Food\_Shopping**)**

SubMenu2**.**add\_command**(**label**=**"Contact Us"**,**

command**=**ContactSection**)**

ef ContactSection**():**

#removes grids

LoginButton**.**grid\_remove**()**

LoginCanvas**.**grid\_remove**()**

TopSellersFrame**.**grid\_remove**()**

RSFrame**.**grid\_remove**()**

#reassign root window size

r**.**geometry**(**"450x600"**)**

# configures frame with new height and width and green background

ContactFrame**.**config**(**height**=**600**,**width**=**500**,**bg**=**"green"**)**

#display at row 1 and column 2

ContactFrame**.**grid**(**row**=**1**,**column**=**1**)**

# configure label with new text and font, size 13

SelectLabel**.**config**(**text**=**"Select which staff of our Centra would u like to contact"**,**

font **=(**"Italic Bold"**,**13**))**

SelectLabel**.**place**(**x**=**10**,**y**=**10**)** # will appear at these co-ordinates in the frame

# configures manager checkbutton with variable t, and onvalue=1,offvalue=0

# yellow background and Arial font with size 20

# command is displayInfo when pressed

Manager**.**config**(**text**=**"Manager: David Cruise"**,**

variable **=** t**,**

onvalue**=**1**,**

offvalue**=**0**,**

command**=**displayInfo**,**

bg**=**"yellow"**,**

font**=(**"Arial"**,**20**))**

# will appear at these co-ordinates in the frame

Manager**.**place**(**x**=**10**,**y**=**60**)**

# confugres supervisor checkbuttons with variable s, onvalue=1,offvalue=0

# yellow background and ariall font with size 20

# command is displayInfo when pressed

Supervisor**.**config**(**text**=**"Supervisor: Agert Berisha"**,**

variable **=** s**,**

onvalue**=**1**,**

offvalue**=**0**,**

command**=**displayInfo**,**

bg**=**"yellow"**,**

font**=(**"Arial"**,**20**))**

# will appear at these co-ordinates in the frame

Supervisor**.**place**(**x**=**10**,**y**=**250**)**

**def** Food\_Shopping**():**

# remove grids from tkinter window

ContactFrame**.**grid\_remove**()**

SliderFrame**.**grid\_remove**()**

LoginCanvas**.**grid\_remove**()**

#reassign root window size

r**.**geometry**(**"1100x675"**)**

# reconfigure background of root window

r**.**configure**(**bg**=**"silver"**)**

# reconfigures frame with green background

TopSellersFrame**.**configure**(**bg**=**"green"**)**

# calls global variables

**global** ImageCounter

**global** FoodImage

**global** name

**global** price

#for loops with 4 variables to access 4-dimensional arrays

#needed 4 variables otherwise I would get error that it wouldn't unpack

# loop creates 5 Radiobuttons on the left Side column 1

# more efficient than writing every radiobutton one by one

# Radiobutton has parrent window TopSellersFrame, text is the name variable

# image is FoodImage , value is price and command is addPrice

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_left**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

# column 1 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**1**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_left01**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 2 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**2**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_right**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 3 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**3**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_right01**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 4 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**4**)**

# reconfigures frame with light blue background and height,width

RSFrame**.**config**(**bg**=**"light blue"**,**height**=**675**,**width**=**600**)**

# configures label with parent window root, silver background and width 20

space\_label**=**tk**.**Label**(**r**,**bg**=**"silver"**,**width**=**20**)**

# displays at assigned row and column with sticky NORTH

space\_label**.**grid**(**row**=**1**,**column**=**5**,**sticky**=**tk**.**N**)**

# will appear at these co-ordinates in the frame or root window

Search\_label**.**place**(**x**=**40**,**y**=**10**)**

Search\_item**.**place**(**x**=**220**,**y**=**13**)**

Search\_button**.**place**(**x**=**410**,**y**=**10**)**

total\_price\_topSellers**.**place**(**x**=**350**,**y**=**630**)**

Finish**.**place**(**x**=**10**,**y**=**630**)**

# displays at assigned row and column

TopSellersFrame**.**grid**(**row**=**1**,**column**=**1**)**

RSFrame**.**grid**(**row**=**1**,**column**=**2**)**

[B] Explanation of how the code behaves

There is 2 menu bars in my application, Shopping and Contact sections

The Shopping section has a submenu called “Food Shopping “ which also when pressed executes Food\_Shopping() function.

The Contact Section has a submenu called “Contact Us” which when pressed executes ContactSection() function

The menu code is quite basic but I don’t feel it was necessary to add anything else that would affect the effectiveness. To make it look better I could have probably added a background color and an image for the menus

Menu can normally be torn off, but I chose to leave that as 0 so the menu will not have the tear-off feature and all the submenu choices will be added starting at position 0

I used a few methods in my menus:

Add\_cascade in order to create a new menu bar

Add\_command in order to create submenus which when pressed call a function

For example the “Food shopping” submenus’ parent menu is Shopping menu.

“Contact us” submenu’s parent menu is Contact menu

[10] Event Management

[A] The extracts of the working and fully commented code related to the topic or topics

**def** Finish**():**

# binds the root to escape button, when this button pressed the CloseApplication command is executed

r**.**bind**(**"<Escape>"**,**CloseApplication**)**

**def** CloseApplication**(**event**):** #event is passed on to our function

r**.**destroy**()** # destroys the root window of tkinter application

[B] Explanation of how the code behaves

Event management using Key Press ( activated when pressing the specified key )

Event is the representative string that contains information on what to listen for.

After clicking the finish shopping button, the user is redirected to another frame where he is asked to give feedback about his experience using the shopping application.

For the event management, when user reaches this point and gives feedback, he can press the <ESC> key in the keyboard to exit the application

To do that, I had to use a bind() function that takes 2 arguments, in our case the key that will be pressed, and the function which it will be called.

Event object is passed on to the next function CloseApplication() in order to execute the key bind.

This was the most convenient way I thought that would apply to my application.

There is other options that I could have used such as:

Button Release ( activated when mouse button is released )

Mouse Wheel ( activated when the mouse wheel is scrolled )

And many more…

**[11] Best Thing I Did**

[A] The extracts of the working and fully commented code related to the topic or topics

#Created a frame for the radiobutton items in the shopping section

TopSellersFrame **=** tk**.**Frame**(**r**,**width**=**1200**,**height**=**650**)**

#Right side frame in the shopping section

RSFrame **=** tk**.**Frame**(**r**)**

Items\_left **=** **[(**1**,** AppleImage**,** "Apple"**,** 0.7**),**

**(**2**,** SweetsImage**,** "Sweets"**,** 1.2**),**

**(**3**,** DonutImage**,** "Donut"**,** 1.1**),**

**(**4**,** CoffeeImage**,** "Coffee"**,** 2.8**),**

**(**5**,** CroissantImage**,** "Croissant"**,** 0.8**)]**

Items\_left01 **=** **[(**1**,** CremEggImage**,** "CremEgg"**,** 0.9**),**

**(**2**,** SlusheeImage**,** "Slushee"**,** 2.5**),**

**(**3**,** TeaImage**,** "Tea"**,** 2.3**),**

**(**4**,** BreadImage**,** "Bread"**,** 2**),**

**(**5**,** CakeImage**,** "Cake"**,** 5.5**)]**

Items\_right **=** **[(**1**,** AeroImage**,** "Aero"**,** 1.4**),**

**(**2**,** HunkyDoryImage**,** "HunkyDory"**,** 2.7**),**

**(**3**,** TaytoImage**,** "Tayto"**,** 1.8**),**

**(**4**,** StampImage**,** "Stamp"**,** 1**),**

**(**5**,** ChickenImage**,** "Chicken"**,** 5.2**)]**

Items\_right01 **=** **[(**1**,** CheeseImage**,** "Cheese"**,** 1.6**),**

**(**2**,** DogFoodImage**,** "DogFood"**,** 6.5**),**

**(**3**,** NutsImage**,** "Nuts"**,** 0.54**),**

**(**4**,** PenImage**,** "Pens"**,** 0.35**),**

**(**5**,** MuffinImage**,** "Muffin"**,** 1.5**)]**

# array with all the item names on stock and their prices

Items\_on\_stock **=** **[(**"Apple"**,** 0.7**),**

**(**"Sweets"**,** 1.2**),**

**(**"Donut"**,** 1.1**),**

**(**"Coffee"**,** 2.8**),**

**(**"Croissant"**,** 0.8**),**

**(**"CremEgg"**,** 0.9**),**

**(**"Slushee"**,** 2.5**),**

**(**"Tea"**,** 2.3**),**

**(**"Bread"**,** 2**),**

**(**"Cake"**,** 5.5**),**

**(**"Aero"**,** 1.4**),**

**(**"HunkyDory"**,** 2.7**),**

**(**"Tayto"**,** 1.8**),**

**(**"Stamp"**,** 1**),**

**(**"Chicken"**,** 5.2**),**

**(**"Cheese"**,** 1.6**),**

**(**"DogFood"**,** 6.5**),**

**(**"Nuts"**,** 0.54**),**

**(**"Pens"**,** 0.35**),**

**(**"Muffin"**,** 1.5**)]**

**def** Food\_Shopping**():**

# remove grids from tkinter window

ContactFrame**.**grid\_remove**()**

SliderFrame**.**grid\_remove**()**

LoginCanvas**.**grid\_remove**()**

#reassign root window size

r**.**geometry**(**"1100x675"**)**

# reconfigure background of root window

r**.**configure**(**bg**=**"silver"**)**

# reconfigures frame with green background

TopSellersFrame**.**configure**(**bg**=**"green"**)**

# loop creates 5 Radiobuttons on the left Side column 1

# more efficient than writing every radiobutton one by one

# Radiobutton has parrent window TopSellersFrame, text is the name variable

# image is FoodImage , value is price and command is addPrice

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_left**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

# column 1 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**1**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_left01**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 2 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**2**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_right**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 3 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**3**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_right01**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 4 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**4**)**

**def** TotalPrice**():**

#call global variables

**global** tempPrice

**global** Sum\_of\_Prices

**global** x1

**global** nameArray

#runs through the AllPrices array

**for** x **in** AllPrices**:**

# adds each element of the array into Sum\_of\_Prices variable one at a time

Sum\_of\_Prices **=** x **+** Sum\_of\_Prices

# created a new variable that stores the Sum\_Of\_prices float value with a max of 2 decimal values

summ **=** **float(**"{0:.2f}"**.format(**Sum\_of\_Prices**))**

# reconfigures message widget to display all the names of items selected and their total price

#relief raised makes it stick out a bit more, looking more proffesional, custom background, width 450

# so even if the message is long it kinda stays in 1 line

totalMessage**.**config**(**text**=**"Total items added are:" **+** **str(**nameArray**)** **+** " with final price of "**+str(**summ**)+**" €"**,**

width**=**450**,**bg**=**"#B2C13A"**,**relief**=**"raised"**)** # raised makes the message stick out more

totalMessage**.**place**(**x**=**10**,**y**=**500**)**# will appear at these co-ordinates in the frame

# refreshes these variables

Sum\_of\_Prices **=** 0

tempPrice **=** 0 #

**def** SearchItem**():**

#calls global variable

**global** tempPrice

# reconfigures message widget that displays text

m1**.**config**(**text **=** "Nothing has been searched and/or is not available yet"**)**

m1**.**place**(**x**=**40**,**y**=**50**)** # will appear at these co-ordinates in the frame

# productName runs through the names, productPrice runs through the prices in the Items\_on\_Stock array

**for** productName**,**productPrice **in** Items\_on\_stock**:**

# if string inside search entry box is equal to a product name executes

**if** Search**.**get**()** **==** productName**:**

# reconfigures message widget to display product name and its price

m1**.**config**(**text**=**productName**+**" is available and it's price is: "**+str(**productPrice**)+**" euro"**)**

# moves the product price to tempPrice

tempPrice **=** productPrice

#add the product name that was searched into the nameArray, later used to display all the item names selected

nameArray**.**append**(**productName**)**

m1**.**place**(**x**=**40**,**y**=**50**)** # will appear at these co-ordinates in the frame

Add\_to\_Basket**.**place**(**x **=** 380**,**y**=**50**)** # will appear at these co-ordinates in the frame

# prints the temporary price in console window

**print(**tempPrice**)**

**def** AddtoBasket**():**

# add the tempPrice value in the AllPrices array

AllPrices**.**append**(**tempPrice**)**

#print in console window

**print(**AllPrices**)**

**def** addPrice**():**

#executed when selecting radiobutton items in the shopping section

# call global variables

**global** x1

**global** y1

**global** nameArray

#x1 runs through names, y1 runs through prices

**for** x1 **,** y1 **in** Items\_on\_stock**:**

# if radiobutton value is equal to a price value y1 in item\_on\_stock array it executes

**if** **float(**v**.**get**())** **==** y1**:**

# reconfigures message to display name of item and its price, message has width of 400

M2**.**config**(**text **=** x1 **+**" has price of: "**+str(**y1**)+**" €" **+**" and it has been added to basket"**,**

width**=**400**)**

M2**.**place**(**x **=** 50**,**y **=**350**)** # will appear at these co-ordinates in the frame

# adds the name of this item in the nameArray

nameArray**.**append**(**x1**)**

#print the array of names in the console window

**print(**nameArray**)**

# adds the price of that item that was selected in the AllPrices array

AllPrices**.**append**(**v**.**get**())**

#prints total price in console window

**print(**"Current prices in basket are: " **+** **str(**AllPrices**))**

[B] Explanation of how the code behaves

At the beginning when creating my application, I was running into issues on how I was going to display my items as radiobuttons as efficiently and as less code as possible.

Using a for loop with 4 variables to access 4 dimensional array was probably one of the best bits of my application. Creating different frames and grouping widgets in these frames was also very handy, as when it also came to remove the widgets, all I had to do was remove the grid of that frame and the whole lot is gone.

When searching for an item I had to create another array which had similar items to the previous ones with same prices. The radiobutton item would display values when pressed but not a name. I decided I was going to compare the price of the radiobuttons with the price of the items in the new search\_item array and if the price matches than it assigns the radiobutton that specific name. So if value was 0.7 , the name would be Apple.

I used the method .append() to add all of these into a new names array that in the end if show total price button was pressed, it would show all the names of the items selected and their total price

**ABSTRACT**

"""

Name: Agert Berisha

Student: X00175572

Date: 21/04/2022

Description:

Python GUI Development Project - Centra Retail Store

"""

**import** tkinter **as** tk

**def** Centra\_Login**():**

#executes when trying to log into the account

# opens text files in read mode

NewUsername **=** **open(**"NewUsername.txt"**,**"r"**)**

readUsername **=** NewUsername**.**read**()**

NewPassword **=** **open(**"NewPassword.txt"**,**"r"**)**

readPassword **=** NewPassword**.**read**()**

# if username and password entered in the entry boxes match the text in the text files than it will succesfully login

**if** **(**Username**.**get**()** **==** Read\_Username **or** Username**.**get**()** **==** readUsername**)** **and** **(**Password**.**get**()** **==** Read\_Password **or** Password**.**get**()** **==** readPassword**):**

# print text

**print(**"Succesful login"**)**

#removes grids if login is successful

Centra**.**grid\_remove**()**

Label01**.**grid\_remove**()**

UsernameLabel**.**grid\_remove**()**

UsernameEntry**.**grid\_remove**()**

PasswordLabel**.**grid\_remove**()**

PasswordEntry**.**grid\_remove**()**

LoginButton**.**grid\_remove**()**

# adding a menu

menuBar **=** tk**.**Menu**(**r**)**

r**.**config**(**menu**=**menuBar**)**

SubMenu1 **=** tk**.**Menu**(**menuBar**,**tearoff**=**0**)**

SubMenu2 **=** tk**.**Menu**(**menuBar**,**tearoff**=**0**)**

#add menus to the menu

# Shopping menu linked to SubMenu1 "Food Shopping" that when pressed command is Food\_Shopping

menuBar**.**add\_cascade**(**label**=**"Shopping"**,**

menu**=**SubMenu1**,**

command**=**Food\_Shopping**)**

# Contact menu linked to SubMenu2 "Contact us" that when pressed command is ContactSection

menuBar**.**add\_cascade**(**label**=**"Contact"**,**

menu**=**SubMenu2**)**

# add submenus of the menu

SubMenu1**.**add\_command**(**label**=**"Food Shopping"**,**

command**=**Food\_Shopping**)**

SubMenu2**.**add\_command**(**label**=**"Contact Us"**,**

command**=**ContactSection**)**

#reassign root window size

r**.**geometry**(**"1100x675"**)**

#remove grid

CreateNewAccount**.**grid\_remove**()**

# display canvas

LoginCanvas**.**grid**(**row**=**1**,**column**=**1**)**

#creates an image in the canvas with SouthEast anchor and CentraBG image and height and width

LoginCanvas**.**create\_image**(**1100**,**675**,**anchor**=**"se"**,**image**=**CentraBg**)**

# create a rectangle in the canvas being used to display a similar centra logo

LoginCanvas**.**create\_rectangle**(**10**,**30**,**300**,**140**,**fill**=**"blue"**)**

# create text in the canvas that is positioned inside the rectangle, filled with orange

# font Times 20 italic bold

LoginCanvas**.**create\_text**(**150**,**65**,**fill**=**"orange"**,**font**=**"Times 20 italic bold"**,**

text**=**"CENTRA"**)**

# create text in canvas positioned inside the rectangle under the previous text,

# filled with white and font is Times 20 italic bold underline

LoginCanvas**.**create\_text**(**150**,**100**,**fill**=**"white"**,**font**=**"Times 20 italic bold underline"**,**

text**=**"Brighten up your day"**)**

# created text in canvas than is on after the rectangle on the right stating a title with Bold font and size 17, filled with red color

LoginCanvas**.**create\_text**(**700**,**100**,**fill**=**"red"**,**text**=**CentraTitle**,**font**=(**"Bold"**,**17**))**

# centra mission text created in canvas with custom color and Helvetica font size 14

LoginCanvas**.**create\_text**(**260**,**400**,**fill**=**"#3A4F31"**,**text**=**CentraMission**,**font**=(**"Helvetica"**,**14**))**

**elif** Username**.**get**()** **!=** Read\_Username **or** Password**.**get**()** **!=** Read\_Password**:**

**print(**"Try again"**)**

**def** ContactSection**():**

#removes grids

LoginButton**.**grid\_remove**()**

LoginCanvas**.**grid\_remove**()**

TopSellersFrame**.**grid\_remove**()**

RSFrame**.**grid\_remove**()**

#reassign root window size

r**.**geometry**(**"450x600"**)**

# configures frame with new height and width and green background

ContactFrame**.**config**(**height**=**600**,**width**=**500**,**bg**=**"green"**)**

#display at row 1 and column 2

ContactFrame**.**grid**(**row**=**1**,**column**=**1**)**

# configure label with new text and font, size 13

SelectLabel**.**config**(**text**=**"Select which staff of our Centra would u like to contact"**,**

font **=(**"Italic Bold"**,**13**))**

SelectLabel**.**place**(**x**=**10**,**y**=**10**)** # will appear at these co-ordinates in the frame

# configures manager checkbutton with variable t, and onvalue=1,offvalue=0

# yellow background and Arial font with size 20

# command is displayInfo when pressed

Manager**.**config**(**text**=**"Manager: David Cruise"**,**

variable **=** t**,**

onvalue**=**1**,**

offvalue**=**0**,**

command**=**displayInfo**,**

bg**=**"yellow"**,**

font**=(**"Arial"**,**20**))**

# will appear at these co-ordinates in the frame

Manager**.**place**(**x**=**10**,**y**=**60**)**

# confugres supervisor checkbuttons with variable s, onvalue=1,offvalue=0

# yellow background and ariall font with size 20

# command is displayInfo when pressed

Supervisor**.**config**(**text**=**"Supervisor: Agert Berisha"**,**

variable **=** s**,**

onvalue**=**1**,**

offvalue**=**0**,**

command**=**displayInfo**,**

bg**=**"yellow"**,**

font**=(**"Arial"**,**20**))**

# will appear at these co-ordinates in the frame

Supervisor**.**place**(**x**=**10**,**y**=**250**)**

**def** displayInfo**():**

# if t value is 1 display manager info

**if** **(**t**.**get**()** **==**1**):**

# displays text details with Roman times font size 10 in a light blue background

manager\_info**.**config**(**text**=**"Phone: 083 333 3333\nEmail: davidcruise01@gmail.com\nAddress: Centra Greenpark, Clondalkin,Dublin"**,**

font **=(**"Roman Times"**,**10**),**

bg**=**"light blue"**)**

manager\_info**.**place**(**x**=**10**,**y**=**130**)**# will appear at these co-ordinates in the frame

# if s value is 1, displays supervisor info

**if** **(**s**.**get**()==**1**):**

# displays text details with Roman Times font size 10 in a light blue background

supervisor\_info**.**config**(**text**=**"Phone: 083 123 3211\nEmail: agertb01@gmail.com\nAddress: Centra Greenpark, Clondalkin,Dublin"**,**

font **=(**"Roman Times"**,**10**),**

bg**=**"light blue"**)**

supervisor\_info**.**place**(**x**=**10**,**y**=**350**)** # will appear at these co-ordinates in the frame

**def** Food\_Shopping**():**

# remove grids from tkinter window

ContactFrame**.**grid\_remove**()**

SliderFrame**.**grid\_remove**()**

LoginCanvas**.**grid\_remove**()**

#reassign root window size

r**.**geometry**(**"1100x675"**)**

# reconfigure background of root window

r**.**configure**(**bg**=**"silver"**)**

# reconfigures frame with green background

TopSellersFrame**.**configure**(**bg**=**"green"**)**

# calls global variables

**global** ImageCounter

**global** FoodImage

**global** name

**global** price

#for loops with 4 variables to access 4-dimensional arrays

#needed 4 variables otherwise I would get error that it wouldn't unpack

# loop creates 5 Radiobuttons on the left Side column 1

# more efficient than writing every radiobutton one by one

# Radiobutton has parrent window TopSellersFrame, text is the name variable

# image is FoodImage , value is price and command is addPrice

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_left**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

# column 1 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**1**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_left01**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 2 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**2**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_right**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 3 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**3**)**

**for** ImageCounter**,** FoodImage**,** name**,** price **in** Items\_right01**:**

FoodImage **=** tk**.**Radiobutton**(**TopSellersFrame**,**text**=**name**,**

compound**=**"top"**,**

image**=**FoodImage**,**

variable **=** v**,**

value **=** price**,**

indicatoron**=**"false"**,**

command**=**addPrice**)**

#column 4 radiobuttons

FoodImage**.**grid**(**row**=**ImageCounter**,**column**=**4**)**

# reconfigures frame with light blue background and height,width

RSFrame**.**config**(**bg**=**"light blue"**,**height**=**675**,**width**=**600**)**

# configures label with parent window root, silver background and width 20

space\_label**=**tk**.**Label**(**r**,**bg**=**"silver"**,**width**=**20**)**

# displays at assigned row and column with sticky NORTH

space\_label**.**grid**(**row**=**1**,**column**=**5**,**sticky**=**tk**.**N**)**

# will appear at these co-ordinates in the frame or root window

Search\_label**.**place**(**x**=**40**,**y**=**10**)**

Search\_item**.**place**(**x**=**220**,**y**=**13**)**

Search\_button**.**place**(**x**=**410**,**y**=**10**)**

total\_price\_topSellers**.**place**(**x**=**350**,**y**=**630**)**

Finish**.**place**(**x**=**10**,**y**=**630**)**

# displays at assigned row and column

TopSellersFrame**.**grid**(**row**=**1**,**column**=**1**)**

RSFrame**.**grid**(**row**=**1**,**column**=**2**)**

**def** AddtoBasket**():**

# add the tempPrice value in the AllPrices array

AllPrices**.**append**(**tempPrice**)**

#print in console window

**print(**AllPrices**)**

**def** addPrice**():**

#executed when selecting radiobutton items in the shopping section

# call global variables

**global** x1

**global** y1

**global** nameArray

#x1 runs through names, y1 runs through prices

**for** x1 **,** y1 **in** Items\_on\_stock**:**

# if radiobutton value is equal to a price value y1 in item\_on\_stock array it executes

**if** **float(**v**.**get**())** **==** y1**:**

# reconfigures message to display name of item and its price, message has width of 400

M2**.**config**(**text **=** x1 **+**" has price of: "**+str(**y1**)+**" €" **+**" and it has been added to basket"**,**

width**=**400**)**

M2**.**place**(**x **=** 50**,**y **=**350**)** # will appear at these co-ordinates in the frame

# adds the name of this item in the nameArray

nameArray**.**append**(**x1**)**

#print the array of names in the console window

**print(**nameArray**)**

# adds the price of that item that was selected in the AllPrices array

AllPrices**.**append**(**v**.**get**())**

#prints total price in console window

**print(**"Current prices in basket are: " **+** **str(**AllPrices**))**

**def** NewAccount\_Display**():**

#executed when trying to create a new account to log-in

# reconfigure label

Label01**.**configure**(**text**=**"Please enter new account details"**)**

# reconfigure username and password label

UsernameLabel**.**configure**(**text**=**"Create New Username\*"**)**

PasswordLabel**.**configure**(**text**=**"Create New Password\*"**)**

#remove login and createnewaccount button grid

LoginButton**.**grid\_remove**()**

CreateNewAccount**.**grid\_remove**()**

# reconfigure button with silver background and command that brings user back to log-in area when pressed

CA**.**configure**(**text**=**"Create Account"**,**

bg **=** "silver"**,**

command **=** Back\_To\_Login**)**

# displays button at specified row and grid with sticky NORTH

CA**.**grid**(**row**=**9**,**column**=**1**,**sticky**=**tk**.**N**)**

**def** Back\_To\_Login**():**

# this is executed after creating a new account and going back to log-in area

# open text file in write mode

NewUsername **=** **open(**"NewUsername.txt"**,**"w"**)**

# writes the string username entered in the username entry box

NewUsername**.**write**(**Username**.**get**())**

#close file

NewUsername**.**close**()**

# open file in write mode

NewPassword **=** **open(**"NewPassword.txt"**,**"w"**)**

# write the news string password entered in the entry box

NewPassword**.**write**(**Password**.**get**())**

#close file

NewPassword**.**close**()**

#reconfigures label

Label01**.**configure**(**text**=**"Please enter details below to login"**)**

#reconfigures username label

UsernameLabel**.**configure**(**text**=**"Username\*"**)**

#reconfigures username label

PasswordLabel**.**configure**(**text**=**"Password\*"**)**

# removes create account button

CA**.**grid\_remove**()**

# displays a new loggin button with command Centra\_Login if pressed

# blue background and yellow text foreground

LoginButton **=** tk**.**Button**(**r**,**text**=**"Login"**,**

bg**=**"blue"**,**

fg**=**"yellow"**,**

command**=**Centra\_Login**)**

#display button at specified row and column with sticky NORTH

LoginButton**.**grid**(**row**=**7**,**column**=**1**,**sticky**=**tk**.**N**)**

**def** SearchItem**():**

#calls global variable

**global** tempPrice

# reconfigures message widget that displays text

m1**.**config**(**text **=** "Nothing has been searched and/or is not available yet"**)**

m1**.**place**(**x**=**40**,**y**=**50**)** # will appear at these co-ordinates in the frame

# productName runs through the names, productPrice runs through the prices in the Items\_on\_Stock array

**for** productName**,**productPrice **in** Items\_on\_stock**:**

# if string inside search entry box is equal to a product name executes

**if** Search**.**get**()** **==** productName**:**

# reconfigures message widget to display product name and its price

m1**.**config**(**text**=**productName**+**" is available and it's price is: "**+str(**productPrice**)+**" euro"**)**

# moves the product price to tempPrice

tempPrice **=** productPrice

#add the product name that was searched into the nameArray, later used to display all the item names selected

nameArray**.**append**(**productName**)**

m1**.**place**(**x**=**40**,**y**=**50**)** # will appear at these co-ordinates in the frame

Add\_to\_Basket**.**place**(**x **=** 380**,**y**=**50**)** # will appear at these co-ordinates in the frame

# prints the temporary price in console window

**print(**tempPrice**)**

**def** TotalPrice**():**

#call global variables

**global** tempPrice

**global** Sum\_of\_Prices

**global** x1

**global** nameArray

#runs through the AllPrices array

**for** x **in** AllPrices**:**

# adds each element of the array into Sum\_of\_Prices variable one at a time

Sum\_of\_Prices **=** x **+** Sum\_of\_Prices

# created a new variable that stores the Sum\_Of\_prices float value with a max of 2 decimal values

summ **=** **float(**"{0:.2f}"**.format(**Sum\_of\_Prices**))**

# reconfigures message widget to display all the names of items selected and their total price

#relief raised makes it stick out a bit more, looking more proffesional, custom background, width 450

# so even if the message is long it kinda stays in 1 line

totalMessage**.**config**(**text**=**"Total items added are:" **+** **str(**nameArray**)** **+** " with final price of "**+str(**summ**)+**" €"**,**

width**=**450**,**bg**=**"#B2C13A"**,**relief**=**"raised"**)** # raised makes the message stick out more

totalMessage**.**place**(**x**=**10**,**y**=**500**)**# will appear at these co-ordinates in the frame

# refreshes these variables

Sum\_of\_Prices **=** 0

tempPrice **=** 0

**def** Finish**():**

# removes these frames

TopSellersFrame**.**grid\_remove**()**

RSFrame**.**grid\_remove**()**

# new root window size

r**.**geometry**(**"500x600"**)**

#configures SliderFrame with height, width and light blue background

SliderFrame**.**config**(**height**=**600**,**width**=**500**,**bg**=**"light blue"**)**

SliderFrame**.**grid**(**row**=**1**,**column**=**1**)**

Slider**.**place**(**x**=**175**,**y**=**60**)** # will appear at these co-ordinates in the frame

# label widget with parent window SliderFrame, displayed as text in Arial font size 10

review **=** tk**.**Label**(**SliderFrame**,**text**=**"How happy were you from 0 to 100 with the service?"**,**

font**=(**"Arial"**,**10**))**

review**.**place**(**x**=**100**,**y**=**30**)** # will appear at these co-ordinates in the frame

# Rating Button with yellow background and submit\_rating command when pressed, parent window is Slider Frame

# yellow background

RatingButton **=** tk**.**Button**(**SliderFrame**,**text**=**"Submit Rating"**,**

bg**=**"yellow"**,**

command**=**Submit\_Rating**)**

RatingButton**.**place**(**x**=**185**,**y**=**110**)** # will appear at these co-ordinates in the frame

# label widget in SliderFrame with parent window SliderFrame, displays text in Arial font size 10

textlabel**=**tk**.**Label**(**SliderFrame**,**

text**=**"Please, could you provide extra feedback on how we could improve our service?"**,**

font**=(**"Arial"**,**10**))**

textlabel**.**place**(**x**=**25**,**y**=**220**)** # will appear at these co-ordinates in the frame

# reconfigures text area with white background and height and width

text\_area**.**config**(**bg**=**"white"**,**height**=**15**,**width**=**50**)**

text\_area**.**place**(**x**=**40**,**y**=**250**)** # will appear at these co-ordinates in the frame

textArea\_Button**.**place**(**x**=**170**,**y**=**500**)** # will appear at these co-ordinates in the frame

# binds the root to escape button, when this button pressed the CloseApplication command is executed

r**.**bind**(**"<Escape>"**,**CloseApplication**)**

**def** CloseApplication**(**event**):** #event is passed on to our function

r**.**destroy**()** # destroys the root window of tkinter application

**def** Submit\_Rating**():**

# creates a message widget in the sliderFrame with green background and width 200

RatingMessage **=** tk**.**Message**(**SliderFrame**,**text**=**"Thank you!\nHope to see you back!!"**,**

width**=**200**,**

bg**=**"green"**)**

RatingMessage**.**place**(**x**=**150**,**y**=**150**)**

**def** text\_review**():**

# gets the full string input from the text area

**input** **=** text\_area**.**get**(**"1.0"**,** "end-1c"**)**

# opens a file in write mode and writes what was inputted in the text area to that file

review\_file **=** **open(**"Customer\_Review.txt"**,**'w'**)**

review\_file**.**write**(input)**

review\_file**.**close**()** # closes the file

#opens the file again this time in read mode

review\_file**=open(**"Customer\_Review.txt"**,**"r"**)**

#prints what was written in the text area in the console window

**print(**review\_file**.**read**())**

# display label with parent window -> SliderFrame, displays text with Arial font, size 10

Review\_message **=** tk**.**Label**(**SliderFrame**,**text**=**"Thank you so much for the review!\nHave a great day!\nPress ESC to exit application"**,**

font**=(**"Arial"**,**10**))**

Review\_message**.**place**(**x**=**110**,**y**=**530**)**

#create tkinter window

r **=** tk**.**Tk**()**

#make tkinter window not resizable

r**.**resizable**(False,False)**

#assign a title to tkinter window

r**.**title**(**"Centra Greenpark"**)**

#create a variable with a centra png

CentraBg **=** tk**.**PhotoImage**(**file**=**"CentraBG.png"**)**

#decrease size of the image

CentraBg **=** CentraBg**.**subsample**(**2**,**2**)**

#created canvas that will be displayed after login is successful

LoginCanvas **=** tk**.**Canvas**(**r**,**height**=**668**,**width**=**1085**)** # 1100x675

#initizialise string variables

#will be used for the login section

readUsername **=** ""

usernameWord **=** ""

readPassword **=** ""

passwordWord **=** ""

#String variable with centra title message

CentraTitle **=** "Centra is Ireland's leading convenience retail group,\nwith 480 bright accessible stores in communities throughout the country"

#Centra mission string text containing centra mission

CentraMission **=** "We have a reputation for quality, value and friendly service\nand the fact that stores are independently owned\nand operated by local people means that shoppers \nget the best of both worlds:\n-Commitment to the traditional values of good fresh foods\n-Good fresh foods and excellent service\n\nCentra Stores provide you with convenient shopping\nsolutions, have all the items you need and are good\nvalue for money\n\nHappy to continue shopping?\n -Go to \*Shopping\* section\n\nWant to contact us?\n -Go to \*Contact\* section (please leave a review if u can!)"

# assigned all photoImages to variables that will be used to display items in the Shopping section as RadioButtons

# subsample was used to decrease height and width of the images so they relatively are all the same size

AppleImage **=** tk**.**PhotoImage**(**file**=**"Apple.gif"**)**

AppleImage **=** AppleImage**.**subsample**(**2**,**3**)**

SweetsImage **=** tk**.**PhotoImage**(**file**=**"Sweets.gif"**)**

SweetsImage **=** SweetsImage**.**subsample**(**5**,**5**)**

DonutImage **=** tk**.**PhotoImage**(**file**=**"Donut.gif"**)**

DonutImage **=** DonutImage**.**subsample**(**4**,**4**)**

CoffeeImage **=** tk**.**PhotoImage**(**file**=**"Coffee.gif"**)**

CoffeeImage **=** CoffeeImage**.**subsample**(**6**,**6**)**

CroissantImage **=** tk**.**PhotoImage**(**file**=**"Croissant.gif"**)**

CroissantImage **=** CroissantImage**.**subsample**(**3**,**4**)**

CremEggImage **=** tk**.**PhotoImage**(**file**=**"CremEgg.gif"**)**

CremEggImage **=** CremEggImage**.**subsample**(**7**,**7**)**

SlusheeImage **=** tk**.**PhotoImage**(**file**=**"Slushee.gif"**)**

SlusheeImage **=** SlusheeImage**.**subsample**(**2**,**3**)**

TeaImage **=** tk**.**PhotoImage**(**file**=**"Tea.gif"**)**

TeaImage **=** TeaImage**.**subsample**(**3**,**6**)**

BreadImage **=** tk**.**PhotoImage**(**file**=**"Bread.gif"**)**

BreadImage **=** BreadImage**.**subsample**(**10**,**10**)**

CakeImage **=** tk**.**PhotoImage**(**file**=**"Cake.gif"**)**

CakeImage **=** CakeImage**.**subsample**(**3**,**2**)**

AeroImage **=** tk**.**PhotoImage**(**file**=**"Aero.gif"**)**

AeroImage **=** AeroImage**.**subsample**(**2**,**2**)**

HunkyDoryImage **=** tk**.**PhotoImage**(**file**=**"HunkyDory.gif"**)**

HunkyDoryImage **=** HunkyDoryImage**.**subsample**(**2**,**3**)**

TaytoImage **=** tk**.**PhotoImage**(**file**=**"Tayto.gif"**)**

TaytoImage **=** TaytoImage**.**subsample**(**2**,**3**)**

StampImage **=** tk**.**PhotoImage**(**file**=**"Stamp.gif"**)**

StampImage **=** StampImage**.**subsample**(**10**,**10**)**

ChickenImage **=** tk**.**PhotoImage**(**file**=**"Chicken.gif"**)**

ChickenImage **=** ChickenImage**.**subsample**(**2**,**3**)**

CheeseImage **=** tk**.**PhotoImage**(**file**=**"Cheese.gif"**)**

CheeseImage **=** CheeseImage**.**subsample**(**5**,**5**)**

DogFoodImage **=** tk**.**PhotoImage**(**file**=**"DogFood.gif"**)**

DogFoodImage **=** DogFoodImage**.**subsample**(**15**,**17**)**

NutsImage **=** tk**.**PhotoImage**(**file**=**"Nuts.gif"**)**

NutsImage **=** NutsImage**.**subsample**(**4**,**5**)**

PenImage **=** tk**.**PhotoImage**(**file**=**"Pen.gif"**)**

PenImage **=** PenImage**.**subsample**(**6**,**7**)**

MuffinImage **=** tk**.**PhotoImage**(**file**=**"Mufin.gif"**)**

MuffinImage **=** MuffinImage**.**subsample**(**4**,**4**)**

#4 arrays that contain a counter, PhotoImage, name of Item and its price

# used to display the radiobuttons as items in the shopping section

#in order to display them in separate columns had to divide them into 4 different arrays

Items\_left **=** **[(**1**,** AppleImage**,** "Apple"**,** 0.7**),**

**(**2**,** SweetsImage**,** "Sweets"**,** 1.2**),**

**(**3**,** DonutImage**,** "Donut"**,** 1.1**),**

**(**4**,** CoffeeImage**,** "Coffee"**,** 2.8**),**

**(**5**,** CroissantImage**,** "Croissant"**,** 0.8**)]**

Items\_left01 **=** **[(**1**,** CremEggImage**,** "CremEgg"**,** 0.9**),**

**(**2**,** SlusheeImage**,** "Slushee"**,** 2.5**),**

**(**3**,** TeaImage**,** "Tea"**,** 2.3**),**

**(**4**,** BreadImage**,** "Bread"**,** 2**),**

**(**5**,** CakeImage**,** "Cake"**,** 5.5**)]**

Items\_right **=** **[(**1**,** AeroImage**,** "Aero"**,** 1.4**),**

**(**2**,** HunkyDoryImage**,** "HunkyDory"**,** 2.7**),**

**(**3**,** TaytoImage**,** "Tayto"**,** 1.8**),**

**(**4**,** StampImage**,** "Stamp"**,** 1**),**

**(**5**,** ChickenImage**,** "Chicken"**,** 5.2**)]**

Items\_right01 **=** **[(**1**,** CheeseImage**,** "Cheese"**,** 1.6**),**

**(**2**,** DogFoodImage**,** "DogFood"**,** 6.5**),**

**(**3**,** NutsImage**,** "Nuts"**,** 0.54**),**

**(**4**,** PenImage**,** "Pens"**,** 0.35**),**

**(**5**,** MuffinImage**,** "Muffin"**,** 1.5**)]**

#initialize variables

x1 **=** 0

y1 **=** 0

#initialize array that will store the names of Items selected in the Shopping section

nameArray **=** **[]**

# variable used to get the price of items in the Shopping section

v **=** tk**.**DoubleVar**()**

#initialized variable, will be used to get sum of all prices of items selected in Shopping section

Sum\_of\_Prices **=** 0

#array that will store all the prices of items selected

AllPrices **=** **[]**

#centra logo

CentraImage **=** tk**.**PhotoImage**(**file**=**"Centra.gif"**)**

# login gui label with Centra image on top

Centra **=** tk**.**Label**(**r**,**image**=**CentraImage**,**

width**=**390**)**

# display in row 0 column 1 in the grid

Centra**.**grid**(**row**=**0**,**column**=**1**)**

#text label telling the user to log their details below

#width 35 and pady 10 with custom background

Label01 **=** tk**.**Label**(**r**,**

text**=**"Please enter details below to login"**,**

width**=**35**,**

pady**=**10**,**

background**=**"#C6F745"**)**

Label01**.**grid**(**row**=**1**,**column**=**1**,**sticky**=**tk**.**N**)**

#Username text Label, padding is 10 pixels

UsernameLabel **=** tk**.**Label**(**r**,**text**=**"Username\*"**,**

font**=**"arial"**,**

pady**=**10**)**

UsernameLabel**.**grid**(**row**=**2**,**column**=**1**)**

#variable that is used to get the string entered in the entry box for username login

Username **=** tk**.**StringVar**()**

#Entry box to input the username string, light blue background, relief is solid so the entry box looks as name says " solid "

UsernameEntry **=** tk**.**Entry**(**r**,**textvariable**=**Username**,**

bg**=**"light blue"**,**

relief**=**"solid"**)**

UsernameEntry**.**grid**(**row**=**3**,**column**=**1**)**

#Password text label , pady is extra space that is added above or below the text within the widget and assigned to 10

PasswordLabel **=** tk**.**Label**(**r**,**text**=**"Password\*"**,**

font**=**"arial"**,**

pady**=**10**)**

#display to respective row and column in the grid

PasswordLabel**.**grid**(**row**=**4**,**column**=**1**)**

#String variable used to get the string entered in the password entry box

Password**=**tk**.**StringVar**()**

#Password entry box, instead of plain characters it puts "\*" for security reasons using show option

#light blue background and solid relief making it look more solid

PasswordEntry **=** tk**.**Entry**(**r**,**textvariable**=**Password**,**

bg**=**"light blue"**,**

show**=**"\*"**,**

relief**=**"solid"**)**

PasswordEntry**.**grid**(**row**=**5**,**column**=**1**)**

#useless label just to occupy space in order to display the login button further down

UselessLabel**=**tk**.**Label**(**r**)**

UselessLabel**.**grid**(**row**=**6**)**

#Login Button with command Centra\_Login when pressed, blue background and yellow text foreground, when pressed executes command

LoginButton **=** tk**.**Button**(**r**,**text**=**"Login"**,**

bg**=**"blue"**,**

fg**=**"yellow"**,**

command**=**Centra\_Login**)**

#displays at respective row column in the grid sticking up NORTH

LoginButton**.**grid**(**row**=**7**,**column**=**1**,**sticky**=**tk**.**N**)**

#another useless label to occupy space to display CreateNewAccount button further down

UselessLabel01 **=** tk**.**Label**(**r**)**

UselessLabel01**.**grid**(**row**=**8**)**

#Button that when pressed has command NewAccount\_Display, where a new account can be created

#silver background and executes command when pressed

CreateNewAccount **=** tk**.**Button**(**r**,**text**=**"Create a new Account?"**,**

bg **=** "silver"**,**

command **=** NewAccount\_Display**)**

#displays at respective row column in the grid sticking up NORTH

CreateNewAccount**.**grid**(**row**=**9**,**column**=**1**,**sticky**=**tk**.**N**)**

#initialized button in main code, so it's easier to use later and call in between functions

CA **=** tk**.**Button**(**r**)**

# open 2 text files and read them, username and password

R01 **=** **open(**"Username.txt"**)**

Read\_Username **=** R01**.**read**()**

R02 **=** **open(**"Password.txt"**)**

Read\_Password **=** R02**.**read**()**

#Created a frame for the radiobutton items in the shopping section

TopSellersFrame **=** tk**.**Frame**(**r**,**width**=**1200**,**height**=**650**)**

#Right side frame in the shopping section

RSFrame **=** tk**.**Frame**(**r**)**

#Message widget initialized in main code for easier calls in between functions with main window RSFrame

M2 **=** tk**.**Message**(**RSFrame**)**

#Message widget with main window RSFrame in Shopping section, used later to display total items selected and their price

totalMessage **=** tk**.**Message**(**RSFrame**)**

#Label widget with main window RSFrame, yellow background, 250 width and size 10 with font type Open Sans

Search\_label **=** tk**.**Message**(**RSFrame**,**text**=**"Check if an item is in stock: "**,**

bg**=**"yellow"**,**width**=**250**,**font **=(**'Open Sans'**,**10**))**

# variable to get the string when searching an item in the Search\_item entry box, after the search\_button is pressed

Search **=** tk**.**StringVar**()**

#Search entry box with main window RSFrame and textvariable Search, width 30 and silver background,

#similar relief to previous entries

#initialized in the main code so easier to use in between functions

Search\_item **=** tk**.**Entry**(**RSFrame**,**textvariable**=**Search**,**width**=**30**,**

relief**=**"solid"**,**

bg **=** "silver"**)**

#Search button with main window RSFrame, green background

#executes command if pressed

Search\_button **=** tk**.**Button**(**RSFrame**,**text**=**"SEARCH ITEM"**,**

bg **=** "green"**,**

command**=**SearchItem**)**

#Button with parent window RSFrame and command AddtoBasket if pressed

Add\_to\_Basket **=** tk**.**Button**(**RSFrame**,**text**=**"Add item to Basket"**,**

command**=**AddtoBasket**)**

# Button with parent window RSFRame and command TotalPrice, custom background

total\_price\_topSellers **=** tk**.**Button**(**RSFrame**,**text**=**"Calculate Total price"**,**

bg**=**"#641994"**,**

command**=**TotalPrice**)**

# initialized vaiables that will be used to run through Items\_on\_Stock array

productName **=** ""

productPrice **=** 0

#Message widget with parent window RSframe, and 350 width

m1 **=** tk**.**Message**(**RSFrame**,**width**=**350**)**

# initialized variable

tempPrice **=** 0

# array with all the item names on stock and their prices

Items\_on\_stock **=** **[(**"Apple"**,** 0.7**),**

**(**"Sweets"**,** 1.2**),**

**(**"Donut"**,** 1.1**),**

**(**"Coffee"**,** 2.8**),**

**(**"Croissant"**,** 0.8**),**

**(**"CremEgg"**,** 0.9**),**

**(**"Slushee"**,** 2.5**),**

**(**"Tea"**,** 2.3**),**

**(**"Bread"**,** 2**),**

**(**"Cake"**,** 5.5**),**

**(**"Aero"**,** 1.4**),**

**(**"HunkyDory"**,** 2.7**),**

**(**"Tayto"**,** 1.8**),**

**(**"Stamp"**,** 1**),**

**(**"Chicken"**,** 5.2**),**

**(**"Cheese"**,** 1.6**),**

**(**"DogFood"**,** 6.5**),**

**(**"Nuts"**,** 0.54**),**

**(**"Pens"**,** 0.35**),**

**(**"Muffin"**,** 1.5**)]**

# finish shopping button with parent window RSFrame and command Finish, brown background

Finish **=** tk**.**Button**(**RSFrame**,**text**=**"Finish Shopping"**,**

bg **=** "brown"**,**

command**=**Finish**)**

#Slider frame intialized in main code with parent window root tkinter

#will be used after finishing shopping button has been pressed

SliderFrame **=** tk**.**Frame**(**r**)**

# slider scale with values from 0 to 100 and displays horizontally, parent window is SliderFrame

Slider **=** tk**.**Scale**(**SliderFrame**,**from\_**=**0**,**to**=**100**,** orient**=**"horizontal"**)**

#created a text area to be used as a way for the user to write feedback, parent window is SliderFrame

text\_area **=** tk**.**Text**(**SliderFrame**)**

# Button for the text area, when pressed executes command text\_review, parent window is SliderFrame

textArea\_Button **=** tk**.**Button**(**SliderFrame**,**text**=**"Submit Review"**,**

command**=** text\_review**)**

#created a frame with parent window root, will be used in the contact section

ContactFrame **=** tk**.**Frame**(**r**)**

# label widget with parent window ContactFrame

SelectLabel **=** tk**.**Label**(**ContactFrame**)**

# two variables that will be used to get the value of two checkbuttons in the contact section

t **=** tk**.**IntVar**()**

s **=** tk**.**IntVar**()**

# checkbuttons and labels with parent windows ContactFrame

# will be used later to display info of the centra staff in the contact section

Manager **=** tk**.**Checkbutton**(**ContactFrame**)**

manager\_info **=** tk**.**Label**(**ContactFrame**)**

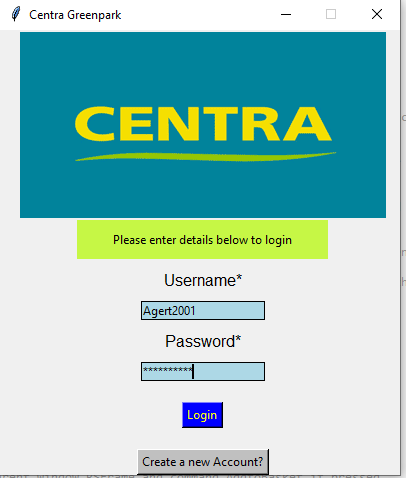
Supervisor **=** tk**.**Checkbutton**(**ContactFrame**)**

supervisor\_info **=** tk**.**Label**(**ContactFrame**)**

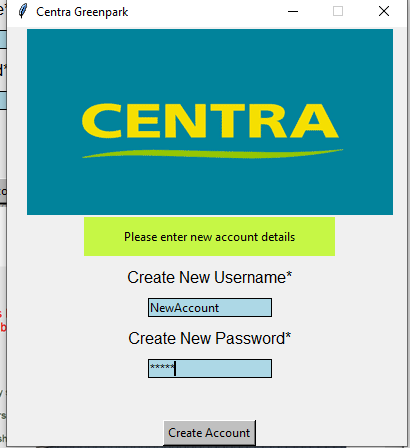
# infinite loop of tkinter window

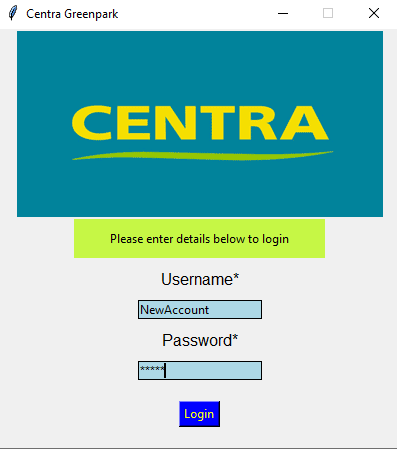
r**.**mainloop**()**

Logging in with an already existing account 🡪 Username:( Agert2001 ) Pass(Centra2022)

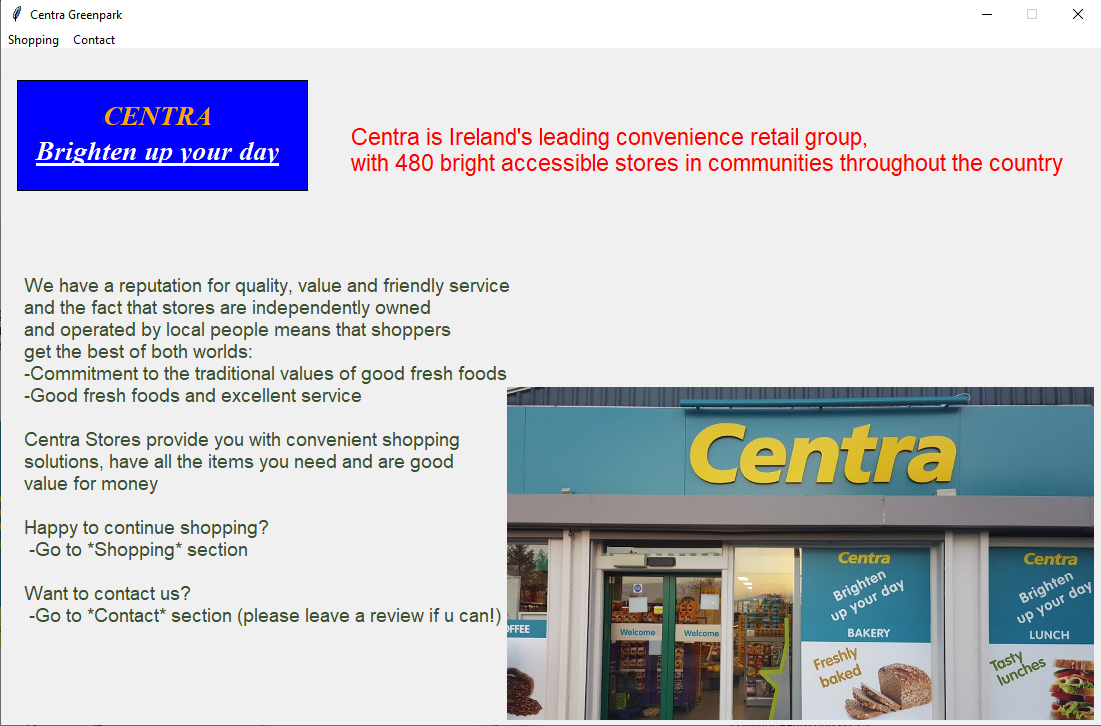


Creating a new account

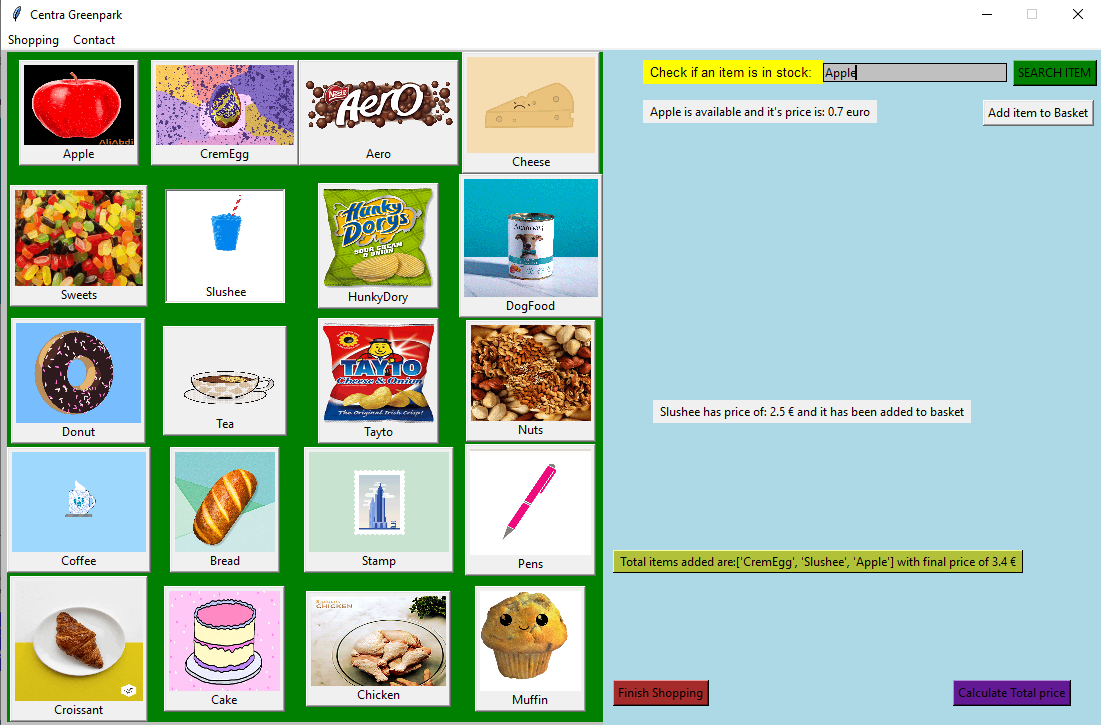




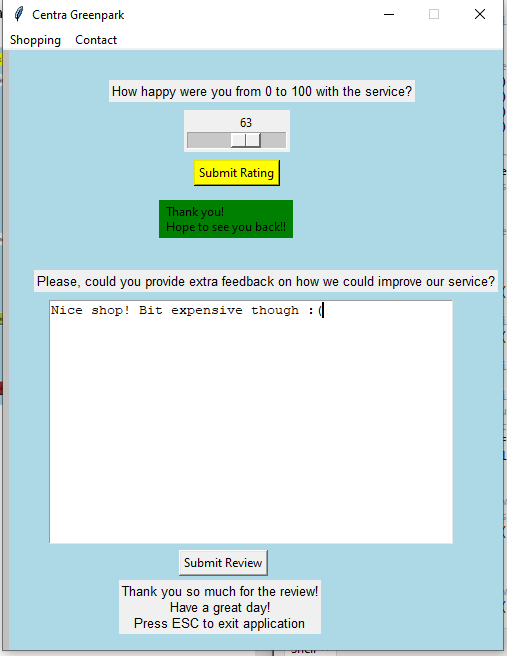
If Login is succesful, this is what shows up:



When pressing the “Food Shopping “ submenu in the Shopping menu



If finish shopping button is pressed:



If “Contact us” submenu in Contact menu is pressed shows this:

