Cashier Application (POS System) for CRISTY'S LOVE BURGER HUB

Shadip Kumar Joshi

B.Sc. (Hons.) Computing, Softwarica College of IT and E-commerce, Coventry University

ST4008CEM Computing Activity Led Learning Project 1

Giriraj Rawat

July 28, 2022

Table of Contents

Table of Figures

Cashier Application (POS System) for CRISTY'S LOVE BURGER HUB

Introduction

Objective of this frontend is to build a simple, user-friendly, and interactive *'Login-Registration-Modification'* GUI for employees in *Point of Sale (POS)* system using python's Tkinter tools. GUI adheres to prototypes from design phase of modern-waterfall SDLC.

Login page

It facilitates login of employees to their respective accounts. 'MANAGER LOGIN' in figure 1 is for managers and 'STAFF LOGIN' in figure 2 is for staffs. New employees can be registered via "REGISTER" under superior's authorization in their respective page.

"INFO" button retrieves data of registered staffs through .treeview() in tabular format while "EDIT" permits modification of their respective details in database after authentication. Superiors can access database if staffs forget their pin and reset it. They also have power to delete record due to resignation of a staff. In "MANAGER LOGIN" page "PRODUCT" informs about available restaurant products while "QUIT" exits the whole program. Likewise, clicking "LOGOUT" sign-outs active manager, thereby returning to the default starting login page.

Figure 3 presents codes of "MANAGER LOGIN" page.

Figure 1

Manager Login

Figure 2

Staff login

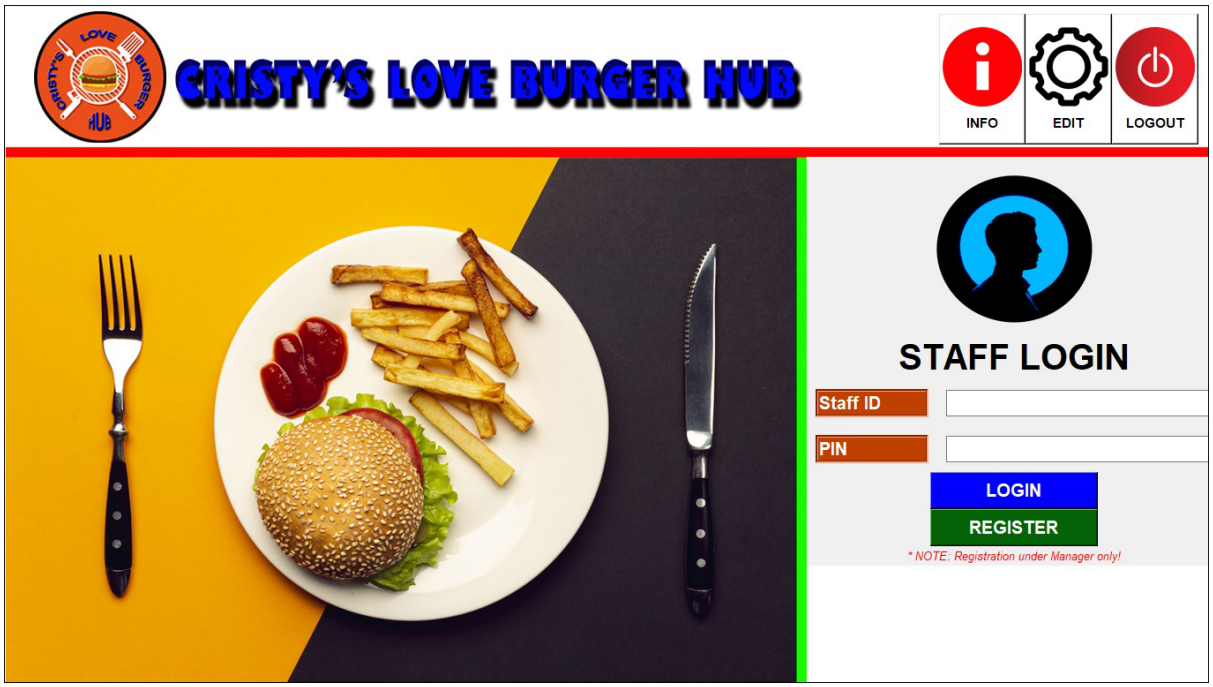


Figure 3

Login code

```
# import tkinter module to the program
from cgitb import text
from textwrap import fill
from tkinter import*
from tkinter import ttk
from tkinter import messagebox
from tkinter.font import BOLD
from PIL import Image, ImageTk
import os
# from click import style
import sqlite3

# create an application window
root= Tk()
#create the root title for the project
root.title("MANAGER LOGIN")

#default fullscreen
root.attributes('-fullscreen',True)

#default supervisor credentials
supervisor_ID=101
supervisor_pin=0000

#setting photo as background
def resize_image(event):
    new_width = event.width
    new_height = event.height
    image = copy_of_image.resize((new_width, new_height))
    photo = ImageTk.PhotoImage(image)
    label.config(image = photo)
    label.image = photo #avoid garbage collection

image = Image.open('img/login_bg.jpg')
copy_of_image = image.copy()
photo = ImageTk.PhotoImage(image)
label = ttk.Label(root, image = photo)
label.bind('<Configure>', resize_image)
label.pack(fill=BOTH, expand = YES)
```

```
# Create textbox labels

manager_profile=Image.open("img/manager.png")
resized_image=manager_profile.resize((200,200))
converted_image=ImageTk.PhotoImage(resized_image)
manager_profile_pic=Label(login_frame,image=converted_image, text="MANAGER LOGIN",
                          font=('Arial','30','bold'),compound='top')
manager_profile_pic.grid(row=0,column=1,columnspan=2)

username_label=Label(login_frame, borderwidth=3,relief=GROOVE,text="Manager ID",
                     font=('Arial','15','bold'),width=12,
anchor="w",bg='#C04000',fg='white')
username_label.grid(row=3,column=1, padx=10,pady=10)

pin_label=Label(login_frame, borderwidth=3,relief=GROOVE,text="PIN",
                font=('Arial','15','bold'),width=12,
anchor="w",bg='#C04000',fg='white')
pin_label.grid(row=4,column=1, padx=10,pady=10)

#Create entry boxes
username_entry=ttk.Entry(login_frame,font="arial 15 bold",width=27)
username_entry.grid(row=3,column=2,padx=10,pady=10)

pin_entry=ttk.Entry(login_frame,font="arial 15 bold",width=27,show="*")
pin_entry.grid(row=4,column=2,padx=10,pady=10)

# Create sign in button
sign_in_btn=Button(login_frame,text="LOGIN",font=('Arial','15','bold'),
                   anchor="c",bg='blue',fg='white',width=15,command=login)
sign_in_btn.grid(row=6,column=1,columnspan=2)

# Create sign up  button
sign_up_btn=Button(login_frame,text="REGISTER",font=('Arial','15','bold'),
                   anchor="c",bg='#046307',fg='white',width=
15,command=register_validate)
sign_up_btn.grid(row=7,column=1,columnspan=2)

notice_label=Label(login_frame,text="* NOTE: Registration under Supervisor only!",
                   font=('Arial','10','italic'),anchor="c",fg='red',width= 40)
notice_label.grid(row=8,column=0,columnspan=4)

# conn.commit()
```

Registration page

It provides interface to add new employees to database. If all fields are filled with valid data, “REGISTER” prompts “Success” message. It also has “BACK” button to return backward if no registration is needed. Figures 4, 5 and 6 represent registration of manager and staff, and their coding respectively.

Figure 4

Manager Registration

Figure 5

Staff Registration

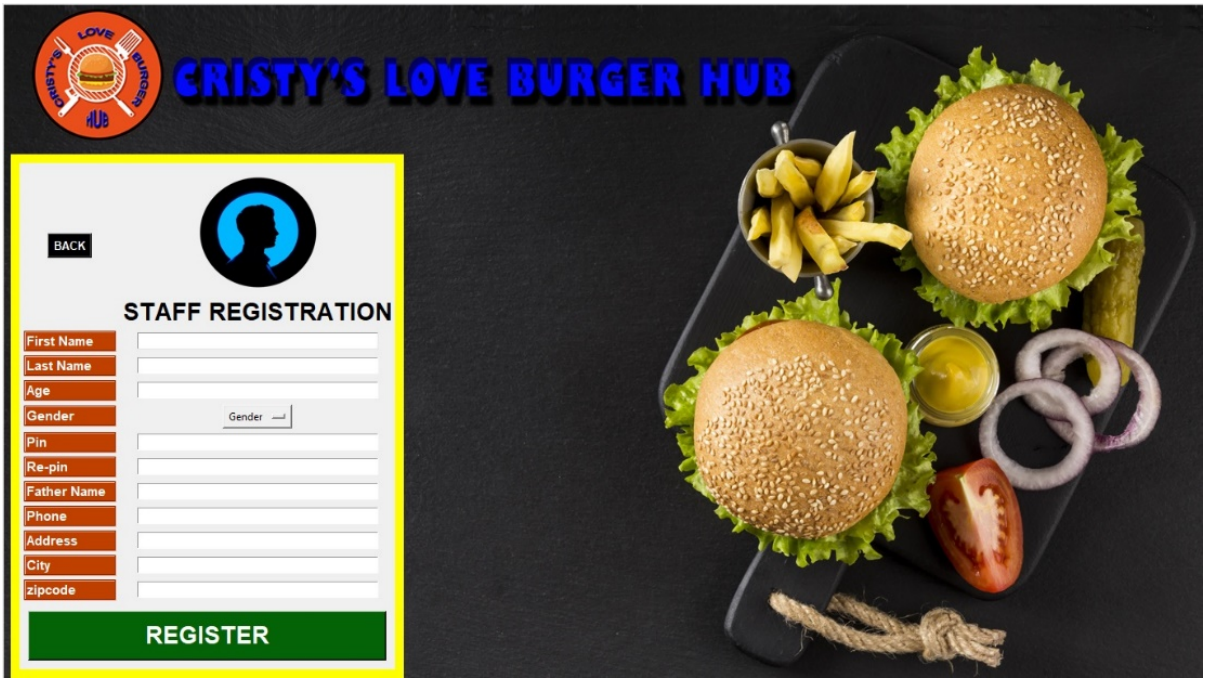


Figure 6

Registration code

```
'''
Labels
'''

# Create textbox labels
f_name_label=Label(register_frame,text="First Name",borderwidth=2,relief=GROOVE,
                   font=('Arial','11','bold'),width=11,
anchor="w",bg='#C04000',fg='white')
f_name_label.grid(row=1,column=0,padx=5,pady=2)

l_name_label=Label(register_frame,text="Last Name",borderwidth=2,relief=GROOVE,
                   font=('Arial','11','bold'),width=11,
anchor="w",bg='#C04000',fg='white')
l_name_label.grid(row=2,column=0,padx=5,pady=2)

age_label=Label(register_frame,text="Age",borderwidth=2,relief=GROOVE,
                font=('Arial','11','bold'),width=11,
anchor="w",bg='#C04000',fg='white')
age_label.grid(row=3,column=0,padx=5,pady=2)
'''
ENTRY
'''
# Create text entries
f_name=Entry(register_frame ,width=45,bg='white')
f_name.grid(row=1,column=1,padx=5)

l_name=Entry(register_frame,width=45)
l_name.grid(row=2,column=1,padx=5)

age=Entry(register_frame,width=45)
age.grid(row=3,column=1,padx=5)

#set the Menu initially
gender=StringVar()
gender.set("Gender")

#creating dropdown menu
drop=OptionMenu(register_frame,gender,"Male","Female","Other")
drop.grid(row=4,column=1,padx=5)

# Create register button
register_btn=Button(register_frame,text="REGISTER",font=
('Arial','20','bold'),bg='#046307',
                    fg='white',command=manager_register)
register_btn.grid(row=13,column=0,padx=10,pady=10,columnspan=2,ipadx=120)

# commit change
conn.commit()

# # close connection
conn.close()

#running the project till it is closed
mainloop()
```

Edit Data page

It provides UI for employees to update their individual information and save it in system by clicking “UPDATE” button. Furthermore, superior can also edit or delete staff-data using “DELETE” if necessary. It provides “BACK” button to return to the previous page. Figures 7, 8 and 9 represent edit page of manager and staff, and their source-code respectively.

Figure 7

Edit manager Data

Figure 8

Edit Staff Data

Figure 9

Edit Employee Data code

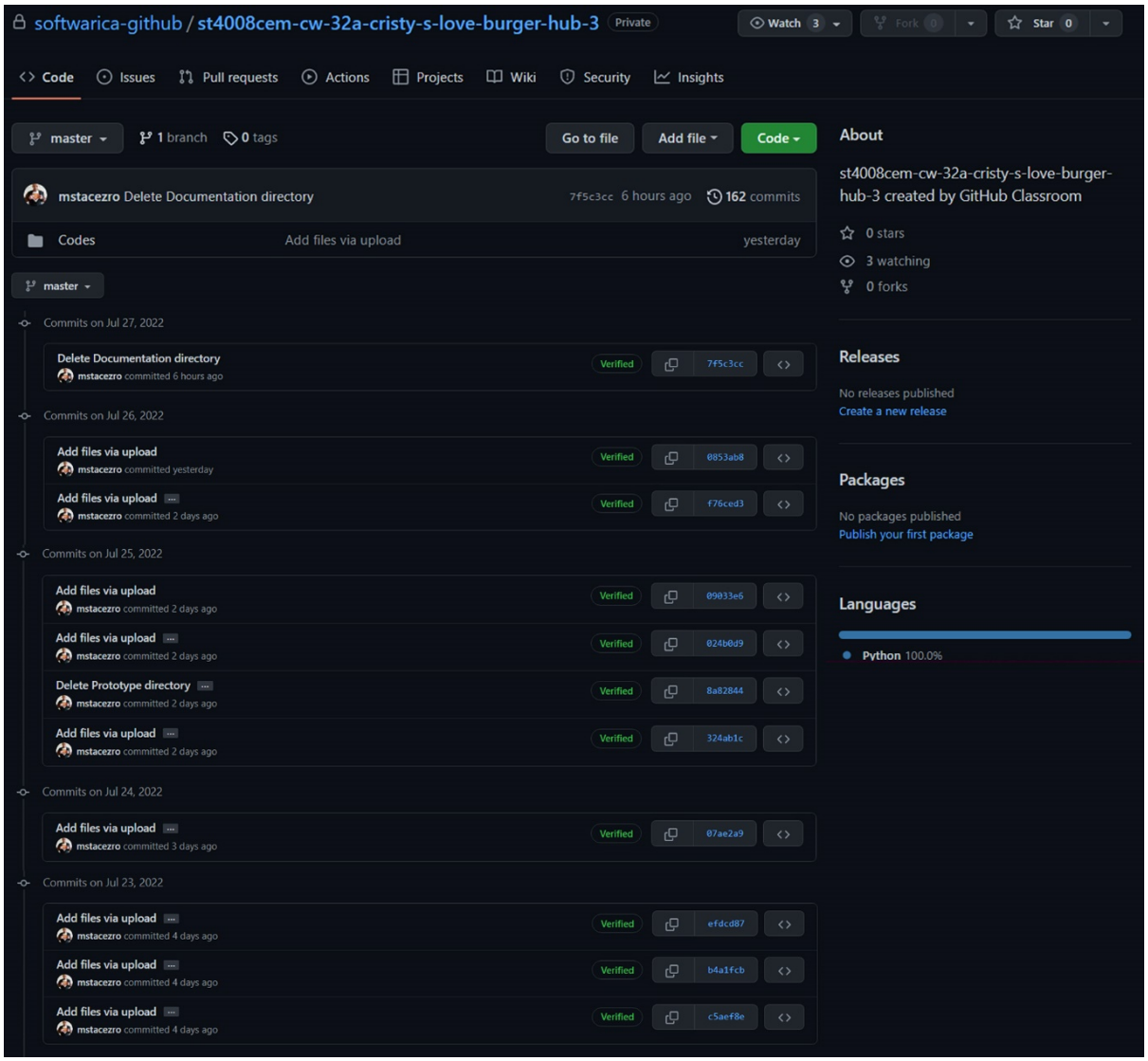
```
# Create delete button
delete_box_label = Label(editor_frame, text="ID-delete/update",
                         width=15, anchor="w", bg="red", fg='black')
delete_box_label.grid(row=13, column=0, pady=2)

delete_box = Entry(editor_frame, width=32, bg='grey', fg='white')
delete_box.grid(row=13, column=1, pady=2)

delete_box_btn = Button(editor_frame, text="DELETE", font=(
    'Arial', '20', 'bold'), bg='red', command=delete)
delete_box_btn.grid(row=14, column=0, columnspan=2, pady=1, padx=0, ipadx=120)


# Create update button
edit_box_btn = Button(editor_frame, text="UPDATE", font=(
    'Arial', '20', 'bold'), bg='#046307', fg='white', command=update_func)
edit_box_btn.grid(row=15, column=0, columnspan=2, pady=1, padx=10, ipadx=120)
```

Version Control

Github:https://github.com/softwarica-github/st4008cem-cw-32a-cristy-s-love-burger-hub-3.git as in figure 10.

Figure 10

Github commits



Conclusion

Concepts of Tkinter tools taught in classroom were fully utilized to design an attractive GUI for data-entry according to SRS document and prototype. Science has proved that colours influence mood. I want my users to be in good mood while using this POS, so I focused on combining simplicity with aesthetics, and popping bright-warm colours to counter outdated classic POS-designs in market. The GUI is simple enough for connection to backends. I chose GUI because of my expertise in graphics.

Due to inexperience as first-timer in coding, many individual UI(s) for secure login expended lots of my time plus effort. Though tiring, I recognized my shortcomings; consulted internet, instructors and concerned books for help. In future projects, research on functionalities integration in single interface will be prioritized. I will amass experience by coding more GUI in my daily life. This endeavour to learn by continuous practice without being discouraged is my strength. This project taught me: *A working software is a good software, however a working software focusing user is a better software.*

Creation of this software enriched my experience, knowledge, teamwork, and critical thinking. It has validated practicality of theoretical knowledge in IT field, enlightening me that problem solving means thinking smarter to find alternative solutions when one method doesn't work.