

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

Cashier Application (POS System) for CRISTY'S LOVE BURGER HUB	4
Introduction.....	4
Login page	4
Registration page	8
Edit Data page.....	11
Version Control.....	13
Conclusion.....	14

Table of Figures

Figure 1.....	5
Figure 2	5
Figure 3	6
Figure 4	8
Figure 5	9
Figure 6	10
Figure 7.....	11
Figure 8	12
Figure 9	12
Figure 10	13

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

Introduction

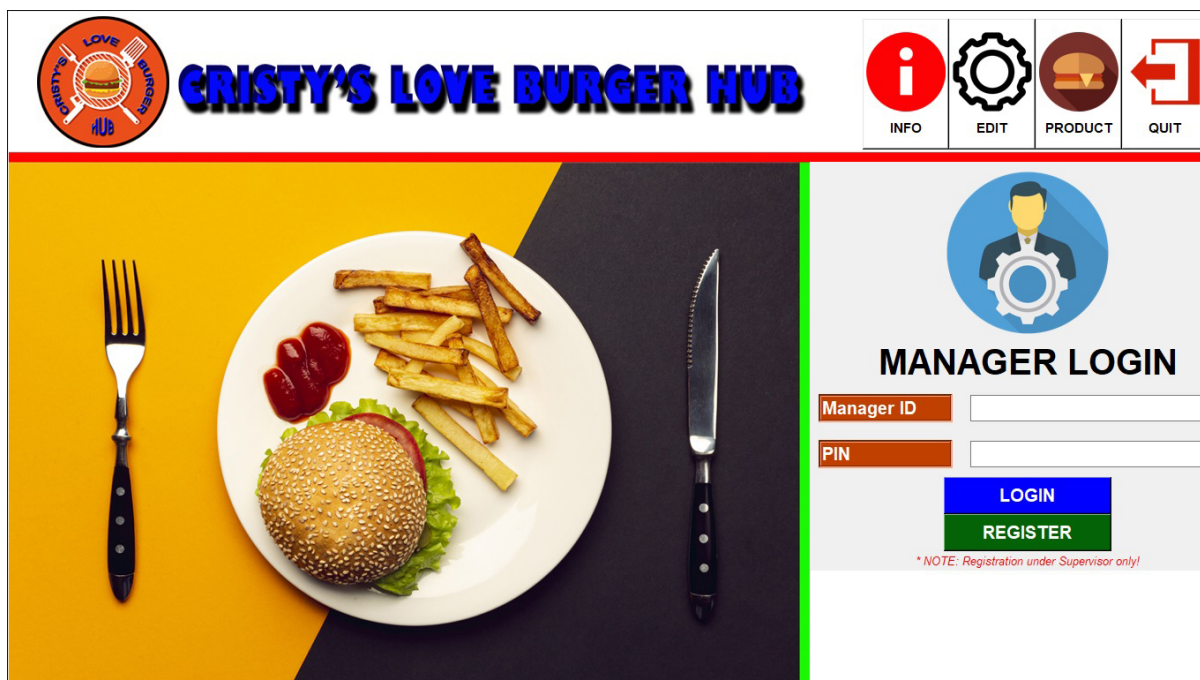
Objective of this frontend is to build a simple, user-friendly, and interactive '*Login-Registration-Modification*' GUI for employees in *Point of Sale (POS)* system using python's Tkinter tools. GUI adheres to prototypes from design phase of modern-waterfall SDLC.

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*

The interface for the Manager Login screen features a header with the 'CRISTY'S LOVE BURGER HUB' logo and a navigation bar with icons for INFO, EDIT, PRODUCT, and QUIT. The main content area is split into two sections: a large image of a burger and fries on the left, and a login form on the right. The login form includes fields for Manager ID and PIN, and buttons for LOGIN and REGISTER. A note at the bottom states: '* NOTE: Registration under Supervisor only!'.

CRISTY'S LOVE BURGER HUB

INFO EDIT PRODUCT QUIT

MANAGER LOGIN

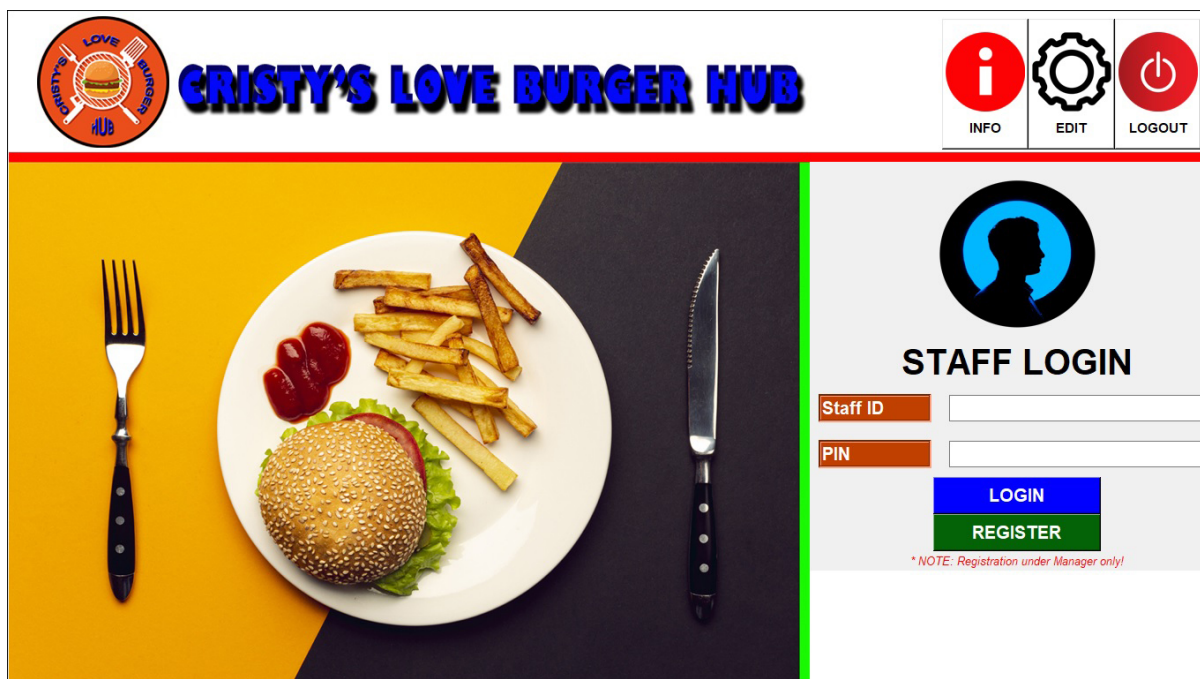
Manager ID

PIN

LOGIN

REGISTER

* NOTE: Registration under Supervisor only!

Figure 2*Staff login*

The interface for the Staff Login screen features a header with the 'CRISTY'S LOVE BURGER HUB' logo and a navigation bar with icons for INFO, EDIT, and LOGOUT. The main content area is split into two sections: a large image of a burger and fries on the left, and a login form on the right. The login form includes fields for Staff ID and PIN, and buttons for LOGIN and REGISTER. A note at the bottom states: '* NOTE: Registration under Manager only!'.

CRISTY'S LOVE BURGER HUB

INFO EDIT LOGOUT

STAFF LOGIN

Staff ID

PIN

LOGIN

REGISTER

* NOTE: Registration under Manager only!

Figure 3*Login code*

```

1 # import tkinter module to the program
2 from cgitb import text
3 from textwrap import fill
4 from tkinter import*
5 from tkinter import ttk
6 from tkinter import messagebox
7 from tkinter.font import BOLD
8 from PIL import Image, ImageTk
9 import os
10 # from click import style
11 import sqlite3
12
13 # create an application window
14 root= Tk()
15 #create the root title for the project
16 root.title("MANAGER LOGIN")
17
18 #default fullscreen
19 root.attributes('-fullscreen',True)
20
21 #default supervisor credentials
22 supervisor_ID=101
23 supervisor_pin=0000
24
25 #setting photo as background
26 def resize_image(event):
27     new_width = event.width
28     new_height = event.height
29     image = copy_of_image.resize((new_width, new_height))
30     photo = ImageTk.PhotoImage(image)
31     label.config(image = photo)
32     label.image = photo #avoid garbage collection
33
34 image = Image.open('img/login_bg.jpg')
35 copy_of_image = image.copy()
36 photo = ImageTk.PhotoImage(image)
37 label = ttk.Label(root, image = photo)
38 label.bind('<Configure>', resize_image)
39 label.pack(fill=BOTH, expand = YES)

```

```

446 # Create textbox labels
447
448 manager_profile=Image.open("img/manager.png")
449 resized_image=manager_profile.resize((200,200))
450 converted_image=ImageTk.PhotoImage(resized_image)
451 manager_profile_pic=Label(login_frame,image=converted_image, text="MANAGER LOGIN",
452                             font=('Arial', '30', 'bold'),compound='top')
453 manager_profile_pic.grid(row=0,column=1,columnspan=2)
454
455 username_label=Label(login_frame, borderwidth=3,relief=GROOVE,text="Manager ID",
456                        font=('Arial', '15', 'bold'),width=12,
457                        anchor="w",bg='#C04000',fg='white')
458 username_label.grid(row=3,column=1, padx=10,pady=10)
459
460 pin_label=Label(login_frame, borderwidth=3,relief=GROOVE,text="PIN",
461                 font=('Arial', '15', 'bold'),width=12,
462                 anchor="w",bg='#C04000',fg='white')
463 pin_label.grid(row=4,column=1, padx=10,pady=10)
464
465 #Create entry boxes
466 username_entry=ttk.Entry(login_frame,font="arial 15 bold",width=27)
467 username_entry.grid(row=3,column=2,padx=10,pady=10)
468
469 pin_entry=ttk.Entry(login_frame,font="arial 15 bold",width=27,show="*")
470 pin_entry.grid(row=4,column=2,padx=10,pady=10)
471
472 # Create sign in button
473 sign_in_btn=Button(login_frame,text="LOGIN",font=('Arial', '15', 'bold'),
474                    anchor="c",bg='blue',fg='white',width=15,command=login)
475 sign_in_btn.grid(row=6,column=1,columnspan=2)
476
477 # Create sign up button
478 sign_up_btn=Button(login_frame,text="REGISTER",font=('Arial', '15', 'bold'),
479                    anchor="c",bg='#046307',fg='white',width=
480                    15,command=register_validate)
481 sign_up_btn.grid(row=7,column=1,columnspan=2)
482
483 notice_label=Label(login_frame,text="* NOTE: Registration under Supervisor only!",
484                    font=('Arial', '10', 'italic'),anchor="c",fg='red',width= 40)
485 notice_label.grid(row=8,column=0,columnspan=4)
486
487 # 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.

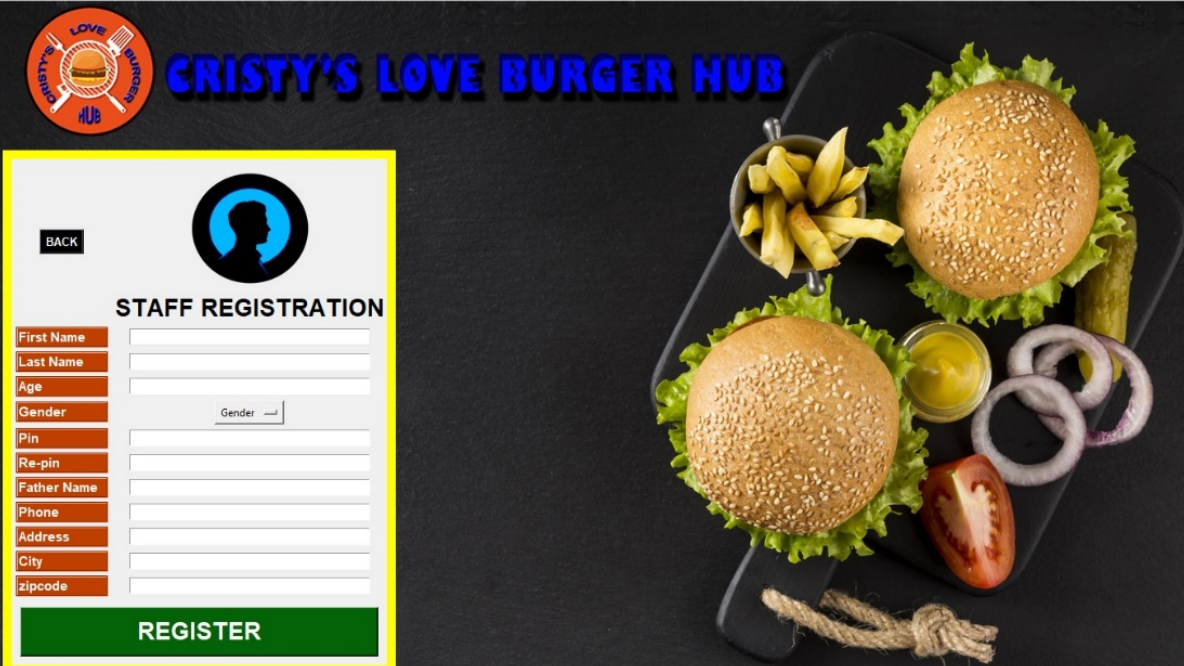
Figure 4

Manager Registration



The image displays a web-based registration form for a manager at "CRISTY'S LOVE BURGER HUB". The form is overlaid on a background image of two burgers, fries, and onion rings. The form itself has a yellow border and a grey header area. In the header, there is a "BACK" button, a circular icon of a person with a gear, and the text "MANAGER REGISTRATION". The form fields are organized into two columns. The left column contains labels for "First Name", "Last Name", "Age", "Gender", "Pin", "Re-pin", "Father Name", "Phone", "Address", "City", and "zipcode". The right column contains corresponding input fields. The "Gender" field is a dropdown menu. At the bottom of the form is a large green "REGISTER" button. The background image shows two sesame seed burgers on lettuce, a small bowl of fries, a slice of tomato, and onion rings on a black tray.

MANAGER REGISTRATION	
First Name	<input type="text"/>
Last Name	<input type="text"/>
Age	<input type="text"/>
Gender	<input type="text" value="Gender"/>
Pin	<input type="text"/>
Re-pin	<input type="text"/>
Father Name	<input type="text"/>
Phone	<input type="text"/>
Address	<input type="text"/>
City	<input type="text"/>
zipcode	<input type="text"/>
<input type="button" value="REGISTER"/>	

Figure 5*Staff Registration*

The image displays a staff registration form for 'CRISTY'S LOVE BURGER HUB'. The form is set against a background featuring a circular logo with a burger and the text 'CRISTY'S LOVE BURGER HUB', the business name in large blue letters, and a photograph of two burgers with fries and toppings on a black tray. The form itself has a yellow border and includes a 'BACK' button, a profile icon placeholder, and a 'STAFF REGISTRATION' title. The registration fields are: First Name, Last Name, Age, Gender (a dropdown menu), Pin, Re-pin, Father Name, Phone, Address, City, and zipcode. A green 'REGISTER' button is at the bottom.

CRISTY'S LOVE BURGER HUB

STAFF REGISTRATION

First Name

Last Name

Age

Gender

Pin

Re-pin

Father Name

Phone

Address

City

zipcode

REGISTER

Figure 6*Registration code*

```

149 '''
150 Labels
151 '''
152
153 # Create textbox labels
154 f_name_label=Label(register_frame,text="First Name",borderwidth=2,relief=GROOVE,
155                    font=('Arial','11','bold'),width=11,
156                    anchor="w",bg='#C04000',fg='white')
157 f_name_label.grid(row=1,column=0,padx=5,pady=2)
158
159 l_name_label=Label(register_frame,text="Last Name",borderwidth=2,relief=GROOVE,
160                    font=('Arial','11','bold'),width=11,
161                    anchor="w",bg='#C04000',fg='white')
162 l_name_label.grid(row=2,column=0,padx=5,pady=2)
163
164 age_label=Label(register_frame,text="Age",borderwidth=2,relief=GROOVE,
165                 font=('Arial','11','bold'),width=11,
166                 anchor="w",bg='#C04000',fg='white')
167 age_label.grid(row=3,column=0,padx=5,pady=2)
168
169 '''
170 ENTRY
171 '''
172
173 # Create text entries
174 f_name=Entry(register_frame,width=45,bg='white')
175 f_name.grid(row=1,column=1,padx=5)
176
177 l_name=Entry(register_frame,width=45)
178 l_name.grid(row=2,column=1,padx=5)
179
180 age=Entry(register_frame,width=45)
181 age.grid(row=3,column=1,padx=5)
182
183 #set the Menu initially
184 gender=StringVar()
185 gender.set("Gender")
186
187 #creating dropdown menu
188 drop=OptionMenu(register_frame,gender,"Male","Female","Other")
189 drop.grid(row=4,column=1,padx=5)
190
191 # Create register button
192 register_btn=Button(register_frame,text="REGISTER",font=
193                    ('Arial','20','bold'),bg='#046307',
194                    fg='white',command=manager_register)
195 register_btn.grid(row=13,column=0,padx=10,pady=10,columnspan=2,ipadx=120)
196
197 # commit change
198 conn.commit()
199
200 # # close connection
201 conn.close()
202
203 #running the project till it is closed
204.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

CRISTY'S LOVE BURGER HUB

EDIT MANAGER DATA

First Name	<input type="text"/>
Last Name	<input type="text"/>
Age	<input type="text"/>
Gender	<input type="text" value="Gender"/>
Pin	<input type="text"/>
Re-pin	<input type="text"/>
Father Name	<input type="text"/>
Phone	<input type="text"/>
Address	<input type="text"/>
City	<input type="text"/>
zipcode	<input type="text"/>

Figure 8*Edit Staff Data***Figure 9***Edit Employee Data code*

```

265 # Create delete button
266 delete_box_label = Label(editor_frame, text="ID-delete/update",
267                           width=15, anchor="w", bg="red", fg='black')
268 delete_box_label.grid(row=13, column=0, pady=2)
269
270 delete_box = Entry(editor_frame, width=32, bg='grey', fg='white')
271 delete_box.grid(row=13, column=1, pady=2)
272
273 delete_box_btn = Button(editor_frame, text="DELETE", font=(
274     'Arial', '20', 'bold'), bg='red', command=delete)
275 delete_box_btn.grid(row=14, column=0, columnspan=2, pady=1, padx=0, ipadx=120)
276
277
278 # Create update button
279 edit_box_btn = Button(editor_frame, text="UPDATE", font=(
280     'Arial', '20', 'bold'), bg='#046307', fg='white', command=update_func)
281 edit_box_btn.grid(row=15, column=0, columnspan=2, pady=1, padx=10, ipadx=120)
282

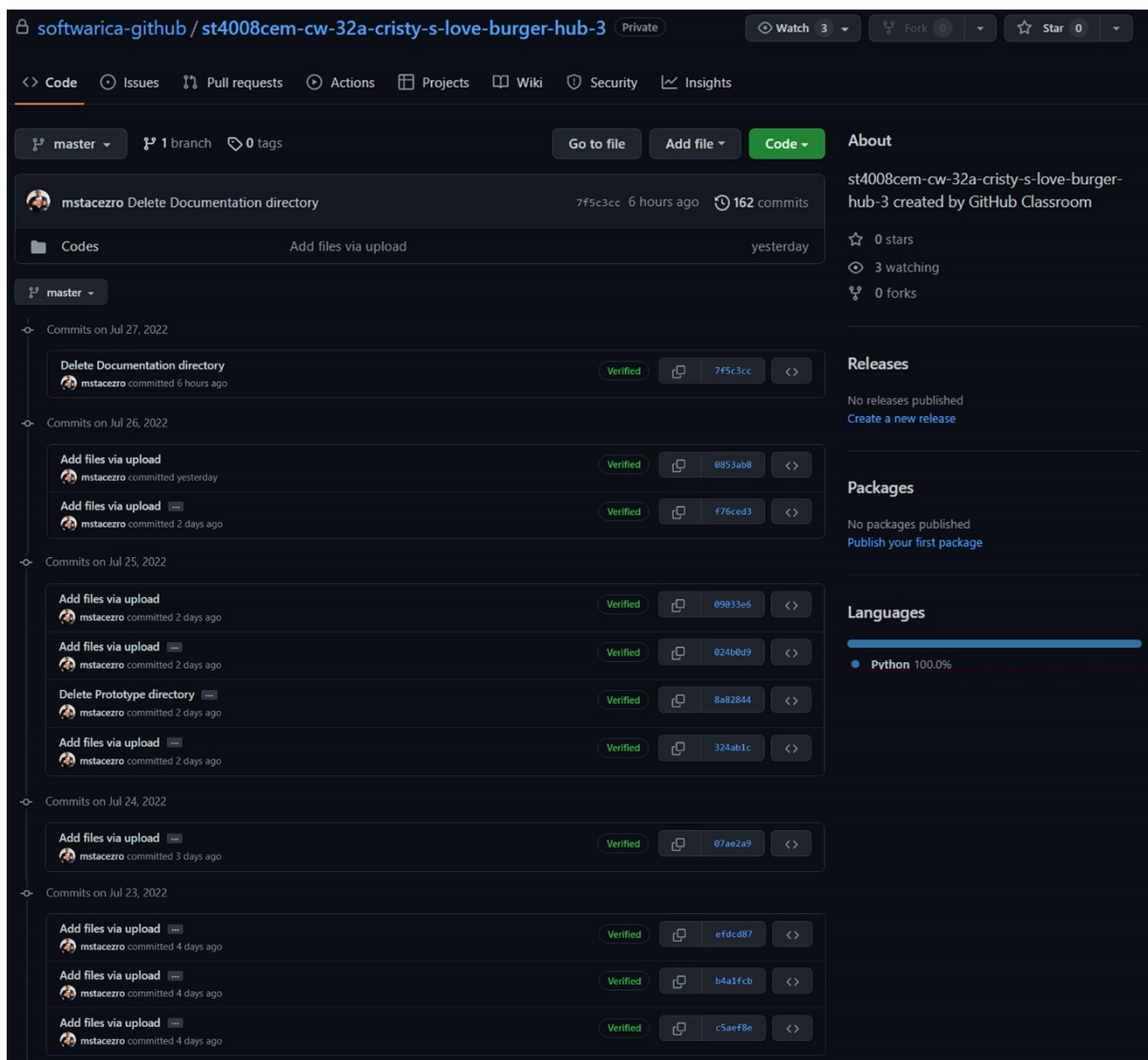
```


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.