

การเขียนโปรแกรมตัวจำหน่ายขนมอัตโนมัติ จากบอร์ด Raspberry Pi



“การต่อวงจรและอุปกรณ์”

1. **SD Card** เสียบ SD Card ขนาด 64 GB ที่ติดตั้งระบบปฏิบัติการ Raspbian เรียบร้อยแล้ว เข้ากับช่องเสียบ SD Card ที่อยู่ด้านล่างของบอร์ด Raspberry Pi



2. **สาย Adapter** เสียบสาย Adapter เพื่อจ่ายไฟเลี้ยงให้กับบอร์ด Raspberry Pi



3. **จอทัชสกรีน**

3.1 ต่อสาย Micro HDMI to HDMI เข้ากับพอร์ต HDMI ของบอร์ด Raspberry Pi



3.2 ต่อสายทัชสกรีนจากจอทัชสกรีนเข้ากับพอร์ต USB ของบอร์ด Raspberry Pi

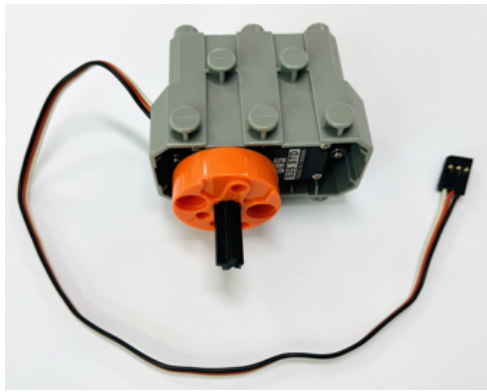


4. กล้องเว็บแคม ต่อสายกล้องเว็บแคมเข้ากับพอร์ต USB ของบอร์ด Raspberry Pi



5. มอเตอร์ 6 ตัว

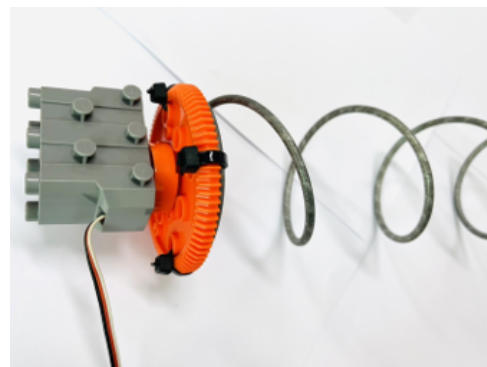
5.1 นำแกนเฟืองมาติดกับมอเตอร์ทั้ง 6 ตัว ดังรูป



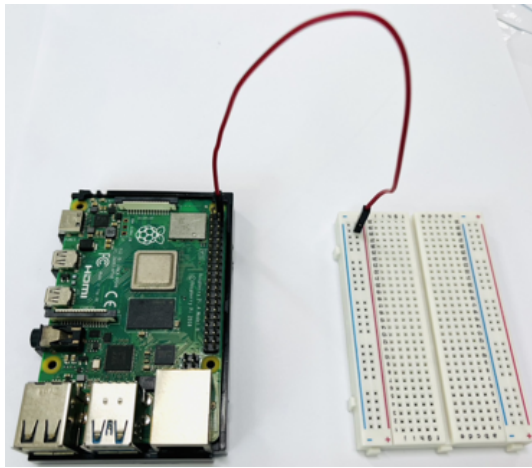
5.2 นำเฟืองยึดติดกับปลายเหล็กสปริงทั้ง 6 ชิ้น ดังรูป



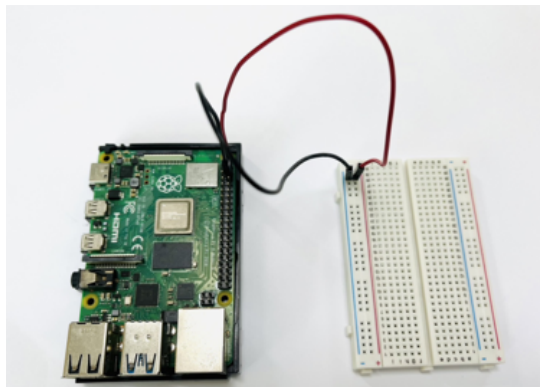
5.3 นำมอเตอร์แต่ละตัวมาติดกับเหล็กสปริง ดังรูป



5.4 เสียบสายแพเส้นสีแดงเข้ากับ PIN 2 (5V) ของบอร์ด Raspberry Pi เชื่อมต่อกับแถว + บน Breadboard

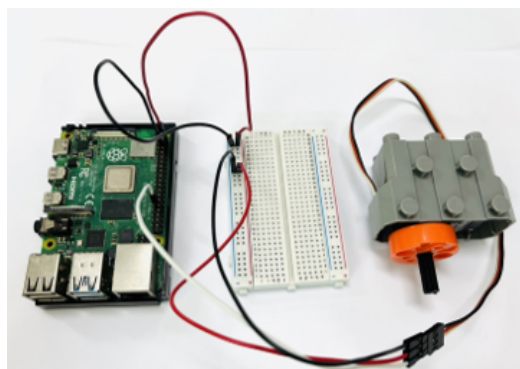


5.5 เชียบสายแพเส้นสีดำเข้ากับ PIN 6 (GND) ของบอร์ด Raspberry Pi เชื่อมต่อกับแถว - บน Breadboard



5.6 เชื่อมต่อบอร์ด Raspberry Pi กับมอเตอร์ตัวที่ 1 (ขนมฮานามิ)

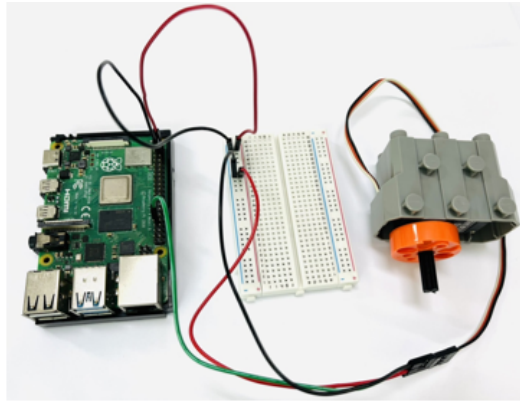
- เส้นสีดำของมอเตอร์เชื่อมต่อกับแถว - ของบอร์ด Raspberry Pi
- เส้นสีแดงของมอเตอร์เชื่อมต่อกับแถว + ของบอร์ด Raspberry Pi
- เส้นสีขาวของมอเตอร์เชื่อมต่อกับขา PIN 29 ของบอร์ด Raspberry Pi



5.7 เชื่อมต่อบอร์ด Raspberry Pi กับมอเตอร์ตัวที่ 2 (ขนมสแน็คแจ็ค)

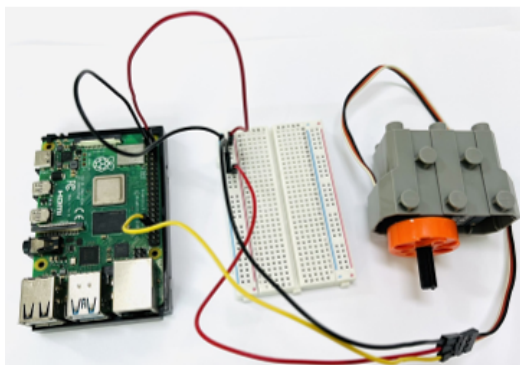
- เส้นสีดำของมอเตอร์เชื่อมต่อกับแถว - ของบอร์ด Raspberry Pi

- เส้นสีแดงของมอเตอร์เชื่อมต่อกับแถว + ของบอร์ด Raspberry Pi
- เส้นสีขาวของมอเตอร์เชื่อมต่อกับขา PIN 31 ของบอร์ด Raspberry Pi



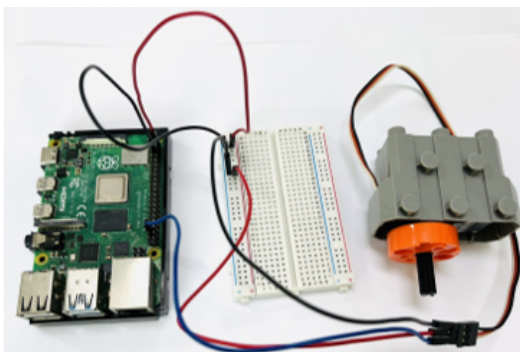
5.8 เชื่อมต่อบอร์ด Raspberry Pi กับมอเตอร์ตัวที่ 3 (ขนมมเลย์)

- เส้นสีดำของมอเตอร์เชื่อมต่อกับแถว - ของบอร์ด Raspberry Pi
- เส้นสีแดงของมอเตอร์เชื่อมต่อกับแถว + ของบอร์ด Raspberry Pi
- เส้นสีขาวของมอเตอร์เชื่อมต่อกับขา PIN 33 ของบอร์ด Raspberry Pi



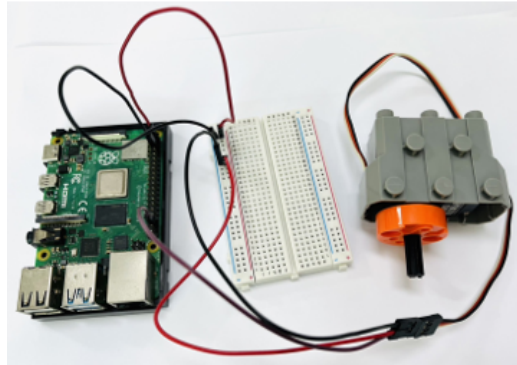
5.9 เชื่อมต่อบอร์ด Raspberry Pi กับมอเตอร์ตัวที่ 4 (ขนมปปริก้า)

- เส้นสีดำของมอเตอร์เชื่อมต่อกับแถว - ของบอร์ด Raspberry Pi
- เส้นสีแดงของมอเตอร์เชื่อมต่อกับแถว + ของบอร์ด Raspberry Pi
- เส้นสีขาวของมอเตอร์เชื่อมต่อกับขา PIN 35 ของบอร์ด Raspberry Pi



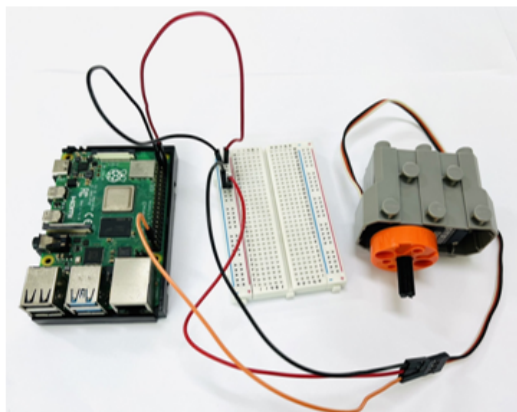
5.10 เชื่อมต่อบอร์ด Raspberry Pi กับมอเตอร์ตัวที่ 5 (ขนมปาร์ตี้)

- เส้นสีดำของมอเตอร์เชื่อมต่อกับแถว - ของบอร์ด Raspberry Pi
- เส้นสีแดงของมอเตอร์เชื่อมต่อกับแถว + ของบอร์ด Raspberry Pi
- เส้นสีขาวของมอเตอร์เชื่อมต่อกับขา PIN 36 ของบอร์ด Raspberry Pi



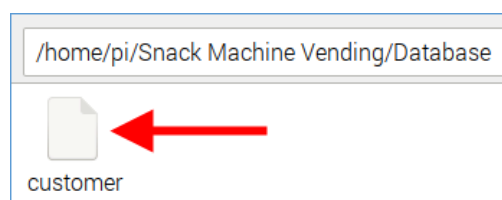
5.11 เชื่อมต่อบอร์ด Raspberry Pi กับมอเตอร์ตัวที่ 6 (ขนมแจ้กซ์)

- เส้นสีดำของมอเตอร์เชื่อมต่อกับแถว - ของบอร์ด Raspberry Pi
- เส้นสีแดงของมอเตอร์เชื่อมต่อกับแถว + ของบอร์ด Raspberry Pi
- เส้นสีขาวของมอเตอร์เชื่อมต่อกับขา PIN 37 ของบอร์ด Raspberry Pi

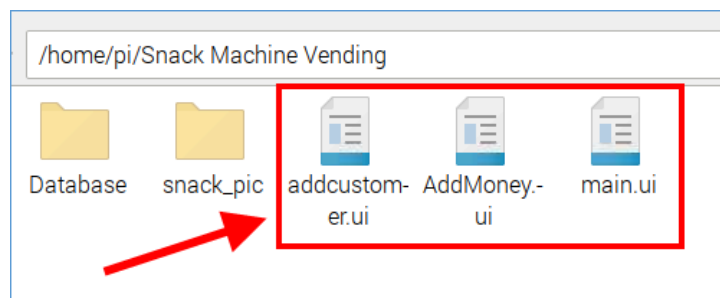


“ การเตรียมไฟล์ ”

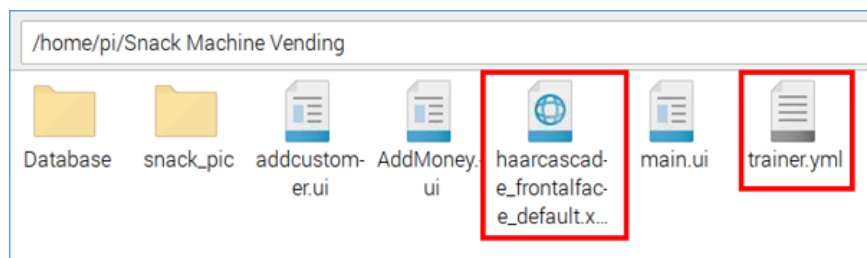
1. เตรียมไฟล์ฐานข้อมูล 1 ไฟล์ (ซึ่งสร้างไว้ในหน่วยการเรียนรู้ที่ 2) ดังรูป



2. เตรียมไฟล์ GUI 3 ไฟล์ (ซึ่งสร้างไว้ในหน่วยการเรียนรู้ที่ 3) ดังรูป



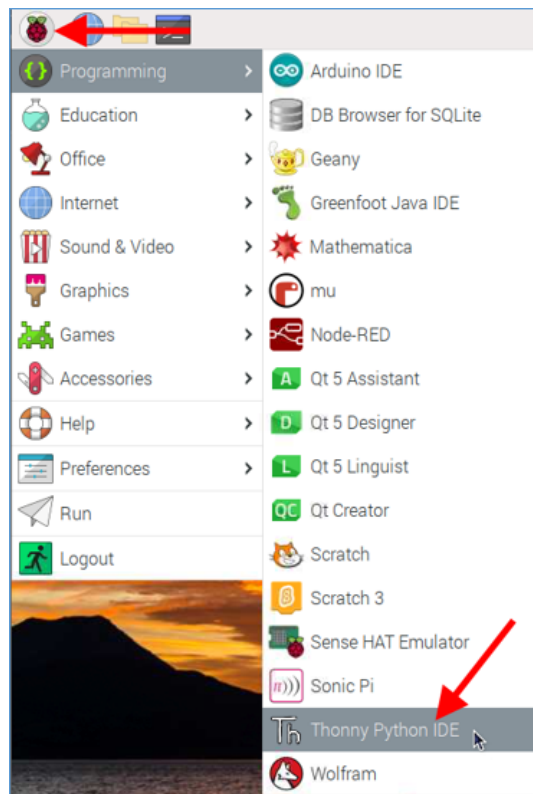
3. คัดลอกไฟล์ haarcascade_frontalface_default.xml และ trainer.yml จากโฟลเดอร์ Face_recognition มาไว้ในโฟลเดอร์ Snack Machine Vending เพื่อใช้ในกระบวนการห้ใบหน้า



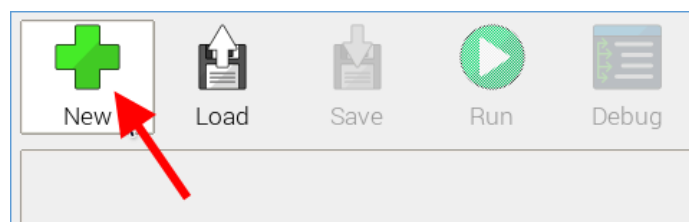
“การเขียนโค้ดคำสั่ง”

การเขียนโค้ดคำสั่งสำหรับเปิดหน้าจอ **main.ui** (หน้าจอหลัก)

1. เปิดโปรแกรม Thonny Python IDE โดยคลิกสัญลักษณ์  จากนั้นเลือกเมนู Programming à Thonny Python IDE



2. เมื่อเปิดโปรแกรม Thonny Python IDE ขึ้นมาแล้วให้คลิกปุ่ม New เพื่อสร้างไฟล์ใหม่



3. เขียนโค้ดคำสั่ง ดังนี้

```

1 import sys
2 from PyQt5.QtCore import pyqtSlot #โมดูลสำหรับสร้างหน้าจอ GUI
3 from PyQt5 import QtWidgets, QtGui
4 from PyQt5.QtWidgets import QApplication, QMainWindow, QDialog, QMessageBox
5 from PyQt5.uic import loadUi
6
7 #หน้าต่างเมนูหลัก คือ ฟอรัม main.ui
8 class MainWindow(QDialog):
9     def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
10         super(MainWindow, self).__init__()
11         loadUi('main.ui',self) #เรียกหน้าต่าง main.ui ขึ้นมา
12
13 app=QApplication(sys.argv) #เริ่มต้นสร้าง GUI
14 widget = QtWidgets.QStackedWidget() #สร้างหน้าต่าง GUI
15 widget.addWidget(MainWindow()) #เรียกหน้าต่างหลักขึ้นมาก่อนลำดับแรก
16 widget.setFixedHeight(800) #กำหนดความสูงหน้าจอ
17 widget.setFixedWidth(1280) #กำหนดความกว้างหน้าจอ
18 widget.show() #แสดงหน้าจอ
19 sys.exit(app.exec_())

```

4. บันทึกไฟล์ชื่อ snack_machine.py และกดปุ่ม Run ผลลัพธ์ที่ได้คือ หน้าต่าง main.ui จะถูกเปิดขึ้นมาแสดง ดังรูป



การเขียนโค้ดคำสั่งในการคลิกปุ่มบนหน้าจอหลัก

1. เขียนโค้ดคำสั่งด้านล่างโค้ด loadUi('main.ui', self) ที่อยู่ใน class MainWindow() ดังนี้

```

8 class MainWindow(QDialog):
9     def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
10         super(MainWindow, self).__init__()
11         loadUi('main.ui',self) #เรียกหน้าต่าง main.ui ขึ้นมา
12         self.btnHanami.clicked.connect(self.selectHanami) #ปุ่มขนมฮานามิ
13         self.btnJaxx.clicked.connect(self.selectJaxx) #ปุ่มขนมแจ๊คซี่
14         self.btnLays.clicked.connect(self.selectLays) #ปุ่มขนมเลย์
15         self.btnPaprika.clicked.connect(self.selectPaprika) #ปุ่มขนมปาปริก้า
16         self.btnParty.clicked.connect(self.selectParty) #ปุ่มขนมปาร์ตี้
17         self.btnSnackjack.clicked.connect(self.selectSnackjack) #ปุ่มขนมสแน็คแจ๊ค
18         self.btnOK.clicked.connect(self.selectOK) #ปุ่มยืนยัน
19         self.btnClear.clicked.connect(self.selectClear) #ปุ่มยกเลิก
20         self.btnPay.clicked.connect(self.selectPay) #ปุ่มชำระเงิน
21         self.btnCancelPay.clicked.connect(self.SelectCancelPay) #ปุ่มยกเลิกการชำระเงิน
22         self.btnAddCust.clicked.connect(self.gotoAddCustomer) #ปุ่มเพิ่มข้อมูลลูกค้า
23         self.btnAddMoney.clicked.connect(self.gotoAddMoney) #ปุ่มเติมเงิน
24         self.btnClose.clicked.connect(self.gotoClose) #ปุ่มปิดหน้าจอ
25
26     @pyqtSlot() #ฟังก์ชันปุ่มในหน้าต่าง GUI ของฟอร์มชื่อขนม
27     def gotoAddCustomer(self): #ฟังก์ชันไปยังหน้าต่าง "เพิ่มข้อมูลลูกค้า"
28         print("Add Customer")
29
30     def gotoAddMoney(self): #ฟังก์ชันไปยังหน้าต่าง "เติมเงิน"
31         print("Add Money")
32
33     def selectHanami(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Hanami"
34         print("Select Hanami")
35
36     def selectSnackjack(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Snackjack"
37         print("Select Snackjack")
38
39     def selectLays(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Lays"
40         print("Select Lays")
41
42     def selectPaprika(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Paprika"
43         print("Select Paprika")
44
45     def selectParty(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Party"
46         print("Select Party")
47
48     def selectJaxx(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Jaxx"
49         print("Select Jaxx")
50
51     def selectClear(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ยกเลิก" จะเคลียร์การเลือกขนม
52         print("Clear")
53
54     def SelectCancelPay(self): #ฟังก์ชันเมื่อกดปุ่ม "ยกเลิกการชำระเงิน" จะเคลียร์หน้าจอทั้งหมด
55         print("Cancel Pay")
56
57     def gotoClose(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ปิดหน้าจอ" จะปิดหน้าต่าง GUI
58         print("Close")
59
60     def selectOK(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ยืนยัน"
61         print("OK")
62
63     def selectPay(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ชำระเงิน"
64         print("Pay")

```

2. คลิกปุ่ม Run แล้วทดลองคลิกทุกปุ่มที่อยู่บนหน้าจอ GUI แล้วสังเกตข้อความในช่อง Shell ว่าแสดงผลถูกต้องหรือไม่ ซึ่งเมื่อกดปุ่มจะต้องแสดงข้อความในช่อง Shell ด้วยคำสั่ง print() ดังรูป


```
Shell
>>> %Run snack.py
Select Hanami
Select Snackjack
Select Lays
Select Paprika
Select Party
Select Jaxx
OK
Clear
Add Customer
Add Money
Close
Pay
Cancel Pay
```

การเขียนโค้ดคำสั่งในฟังก์ชันการเลือกซื้อขนม

หลักการทำงาน คือ เมื่อคลิกปุ่มขนมชนิดใด จะแสดงชื่อและราคาของขนมชนิดนั้นขึ้นมา และหากคลิกปุ่ม “ยกเลิก” จะทำการรีเซ็ตหน้าจอการเลือกขนม โดยมีขั้นตอนดังนี้

1. กำหนดฟังก์ชันปุ่มขนมฮานามิ selectHanami()

1.1 กำหนดราคาสินค้าแต่ละชนิด ไว้ด้านบนคลาส MainWindow(QDialog)

```
7 #ตัวแปรเก็บราคาสินค้าแต่ละชนิด
8 price_Hanami = 15
9 price_Jaxx = 10
10 price_Lays = 20
11 price_Paprika = 15
12 price_Party = 10
13 price_Snackjack = 20
14
15 #หน้าต่างเมนูหลัก คือ ฟอรัม main.ui
16 class MainWindow(QDialog):
17     def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
18         super(MainWindow, self).__init__()
19         loadUi('main.ui',self) #เรียกหน้าต่าง main.ui ขึ้นมา
```

1.2 ลบคำสั่ง print() ในฟังก์ชัน selectHanami() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
48 def selectHanami(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Hanami"
49     self.lblPrice.setText(str(price_Hanami)) #แสดงราคาสินค้า
50     self.lblSnack.setText('ขนมฮานามิ') #แสดงชื่อขนม
```

1.3 คลิกปุ่ม Run แล้วทดลองคลิกปุ่มขนมฮานามิ



จะปรากฏชื่อขนมและราคาขนม ดังรูป



2. กำหนดฟังก์ชันปุ่มขนมสแน็คแจ็ค selectSnackJack()

2.1 ลบคำสั่ง print() ในฟังก์ชัน selectSnackJack() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
52 def selectSnackjack(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Snackjack"
53 self.lblPrice.setText(str(price_Snackjack)) #แสดงราคาขนม
54 self.lblSnack.setText('ขนมสแน็คแจ็ค') #แสดงชื่อขนม
```

2.2 คลิกปุ่ม Run แล้วทดลองคลิกปุ่มขนมสแน็คแจ็ค



จะปรากฏชื่อขนมและราคาขนม ดังรูป



3. กำหนดฟังก์ชันปุ่มขนม selectLays()

3.1 ลบคำสั่ง print() ในฟังก์ชัน selectLays() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
56 def selectLays(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Lays"
57     self.lblPrice.setText(str(price_Lays)) #แสดงราคาขนม
58     self.lblSnack.setText('ขนมเลย์') #แสดงชื่อขนม
```

3.2 คลิกปุ่ม Run แล้วทดลองคลิกปุ่มขนมเลย์



จะปรากฏชื่อขนมและราคาขนม ดังรูป



4. กำหนดฟังก์ชันปุ่มขนมปาปริก้า selectPaprika()

4.1 ลบคำสั่ง print() ในฟังก์ชัน selectPaprika() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
60 def selectPaprika(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Paprika"
61     self.lblPrice.setText(str(price_Paprika)) #แสดงราคาขนม
62     self.lblSnack.setText('ขนมปาปริก้า') #แสดงชื่อขนม
```

4.2 คลิกปุ่ม Run แล้วทดลองคลิกปุ่มขนมปาปริก้า



จะปรากฏชื่อขนมและราคาขนม ดังรูป



5. กำหนดฟังก์ชันปุ่มขนมปาร์ตี้ selectParty()

5.1 ลบคำสั่ง print() ในฟังก์ชัน selectParty() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
64 def selectParty(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Party"
65     self.lblPrice.setText(str(price_Party)) #แสดงราคาขนม
66     self.lblSnack.setText('ขนมปาร์ตี้') #แสดงชื่อขนม
```

5.2 คลิกปุ่ม Run แล้วทดลองคลิกปุ่มขนมปาร์ตี้



จะปรากฏชื่อขนมและราคาราขนม ดังรูป



6. กำหนดฟังก์ชันปุ่มขนมแฉีก selectJaxx()

6.1 ลบคำสั่ง print() ในฟังก์ชัน selectJaxx() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
68 def selectJaxx(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Jaxx"
69     self.lblPrice.setText(str(price_Jaxx)) #แสดงราคาราขนม
70     self.lblSnack.setText('ขนมแฉีก') #แสดงชื่อขนม
```

6.2 คลิกปุ่ม Run แล้วทดลองคลิกปุ่มขนมแฉีก



จะปรากฏชื่อขนมและราคาราขนม ดังรูป



7. กำหนดฟังก์ชันปุ่ม “ยกเลิก” selectClear()

7.1 ลบคำสั่ง print() ในฟังก์ชัน selectClear() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
72 def selectClear(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ยกเลิก" จะเคลียร์ชื่อขนมจอ
73 self.lblSnack.setText('และเลือกขนม')
74 self.lblPrice.setText('0')
```

7.2 คลิกปุ่ม Run แล้วทดลองคลิกเลือกขนม จากนั้นคลิกปุ่ม “ยกเลิก”



จะรีเซ็ตชื่อขนมและราคาขนม ดังรูป



การเขียนโค้ดคำสั่งในการเพิ่มข้อมูลลูกค้า

หลักการทำงาน คือ เมื่อคลิกปุ่ม “บันทึกข้อมูลลูกค้า” จะเปิดฟอร์มกรอกชื่อและจำนวนเงินที่ต้องการเติมเพื่อบันทึกเข้าฐานข้อมูล จากนั้นเมื่อคลิกปุ่ม “ถ่ายรูป” จะปรากฏป๊อปอัพขึ้นมาเพื่อแจ้งให้ลูกค้ามองกล้อง เมื่อคลิกปุ่ม “OK” กล้องจะถูกเปิดขึ้นมาและทำการถ่ายรูปใบหน้าจำนวน 30 รูปเก็บไว้ในโฟลเดอร์ dataset จากนั้นก็คลิกปุ่ม “กลับหน้าหลัก” เพื่อกลับมายังหน้าหลักของโปรแกรม โดยมีขั้นตอนดังนี้

1. แก้ไขฟังก์ชันปุ่มข้อมูลลูกค้า gotoAddCustomer()

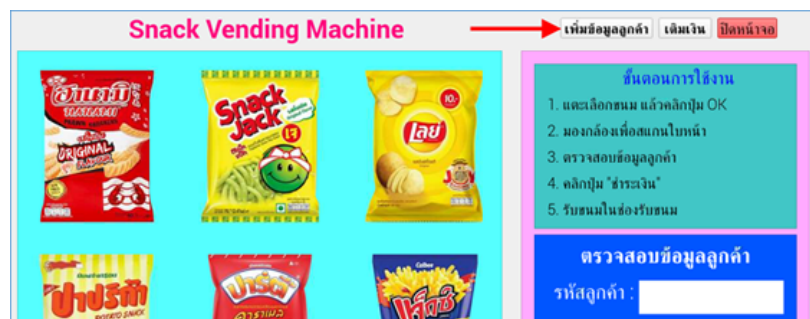
1.1 ลบคำสั่ง print() ในฟังก์ชัน gotoAddCustomer() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
27 def gotoAddCustomer(self):
28     #กดปุ่มเพิ่มข้อมูลลูกค้า แล้วเปิดหน้าจอ GUI เพิ่มข้อมูลลูกค้า
29     widget.addWidget(AddCustomer())
30     widget.setFixedHeight(400)
31     widget.setFixedWidth(600)
32     widget.setCurrentIndex(widget.currentIndex()+1)
```

1.2 สร้างคลาสหน้าต่าง AddCustomer() ขึ้นมา เนื่องจากมีการอ้างอิงถึงคลาส AddCustomer() โดยให้พิมพ์โค้ดคำสั่งไว้ด้านบนคำสั่ง app=QApplication(sys.argv) ดังนี้

```
70 #หน้าต่าง GUI ของฟอร์มเพิ่มข้อมูลลูกค้า
71 class AddCustomer(QDialog):
72     def __init__(self):
73         super(AddCustomer, self).__init__()
74         loadUi("addcustomer.ui", self) #โหลดหน้าต่างออกแบบฟอร์มเพิ่มข้อมูลลูกค้าขึ้นมา
75
76 app=QApplication(sys.argv) #เริ่มต้นสร้าง GUI
77 widget = QtWidgets.QStackedWidget() #สร้างหน้าต่าง GUI
78 widget.addWidget(MainWindow()) #เรียกหน้าต่างหลักขึ้นมาก่อนลำดับแรก
79 widget.setFixedHeight(800) #กำหนดความสูงหน้าจอ
80 widget.setFixedWidth(1280) #กำหนดความกว้างหน้าจอ
81 widget.show() #แสดงหน้าจอ
```

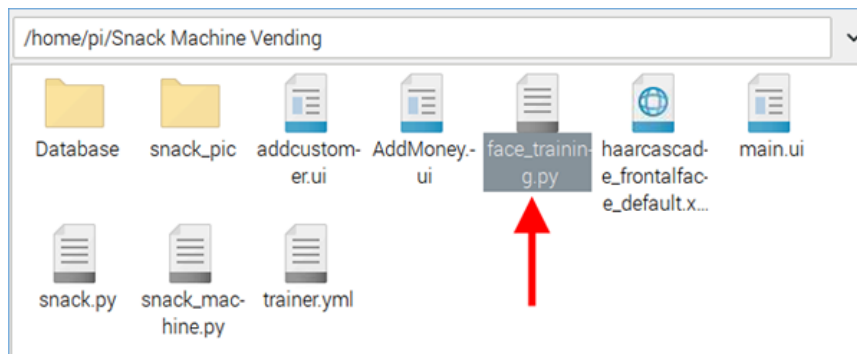
1.3 คลิกปุ่ม Run แล้วทดลองคลิกปุ่ม “เพิ่มข้อมูลลูกค้า”



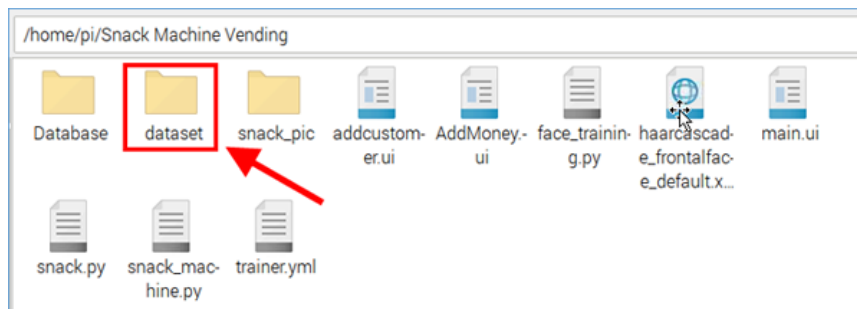
จะเห็นว่าหน้าจอ GUI จะเปลี่ยนเป็นหน้าจอฟอร์ม addcustomer.ui ดังรูป

2. เขียนโค้ดคำสั่ง save และ บันทึก ในหน้าในคลาส AddCustomer()

2.1 คัดลอกไฟล์ face_training.py จากโฟลเดอร์ Face_recognition ที่เขียนโค้ดคำสั่งไว้ในหน่วยการเรียนรู้ที่ 4 มาไว้ในโฟลเดอร์ Snack Machine Vending เพื่อใช้ในการเรียนรู้ใบหน้า



2.2 สร้างโฟลเดอร์ dataset เพื่อเก็บไฟล์รูปภาพใบหน้าลูกค้า



2.3 นำเข้าโมดูลสำหรับฐานข้อมูลและโมดูลที่ใช้สำหรับการวิเคราะห์ใบหน้า

```

1 import sys
2 from PyQt5.QtCore import pyqtSlot #โมดูลสำหรับสร้างหน้าจอ GUI
3 from PyQt5 import QtWidgets, QtGui
4 from PyQt5.QtWidgets import QApplication, QMainWindow, QDialog, QMessageBox
5 from PyQt5.uic import loadUi
6
7 import os
8 import os.path
9 import cv2 #เปิดใช้งาน OpenCV เพื่อใช้งานระบบวิเคราะห์ใบหน้า
10 import numpy as np #โมดูลที่ใช้ในการคำนวณทางคณิตศาสตร์
11 import sqlite3 #โมดูลที่ใช้ในการจัดการฐานข้อมูล

```

2.4 พิมพ์โค้ดคำสั่งในการเชื่อมต่อฐานข้อมูลและเรียกใช้ไฟล์สำหรับการวิเคราะห์ใบหน้า

```

7 import os
8 import os.path
9 import cv2 #เปิดใช้งาน OpenCV เพื่อใช้งานระบบวิเคราะห์ใบหน้า
10 import numpy as np #โมดูลที่ใช้ในการคำนวณทางคณิตศาสตร์
11 import sqlite3 #โมดูลที่ใช้ในการจัดการฐานข้อมูล
12
13 cn = sqlite3.connect('Database/customer')#ระบุไปยังไฟล์ฐานข้อมูล
14 cur = cn.cursor() #สร้างออบเจกต์เพื่อชี้ไปยังฐานข้อมูล
15
16 #โหลดไฟล์ cascade สำหรับตรวจจับใบหน้า
17 face_detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
18 recognizer = cv2.face.LBPHFaceRecognizer_create()
19 recognizer.read('trainer.yml')

```

2.5 เขียนโค้ดคำสั่งในคลาส AddCustomer() ดังนี้

```
107 #หน้าต่าง GUI ของฟอร์มเพิ่มข้อมูลลูกค้า
108 class AddCustomer(QDialog):
109     def __init__(self):
110         super(AddCustomer, self).__init__()
111         loadUi("addcustomer.ui", self) #โหลดหน้าต่างฟอร์มเพิ่มข้อมูลลูกค้าขึ้นมา
112         #เมื่อกดปุ่มถ่ายรูป ให้ไปทำคำสั่งในฟังก์ชัน selectCapture
113         self.btnCapture.clicked.connect(self.selectCapture)
114
115         #กำหนดให้ช่องรหัสลูกค้าแสดง ID ถัดไป โดยไม่ต้องกรอกเอง
116         ncust = cur.execute("SELECT * FROM tbCustomer")
117         nextID = (len(ncust.fetchall())) + 1
118         self.txtID.setText(str(nextID))
119
120         #เมื่อกดปุ่มย้อนกลับ ให้ไปทำคำสั่งในฟังก์ชัน gotoMainWindow
121         self.btnBack.clicked.connect(self.gotoMainWindow)
122
123 @pyqtSlot() #ฟังก์ชันปุ่มในหน้าต่าง GUI ของฟอร์มเพิ่มข้อมูลลูกค้า
124 def gotoMainWindow(self): #ไปหน้า GUI ฟอร์มซื้อสินค้า
125     widget.addWidget(MainWindow())
126     widget.setFixedHeight(800)
127     widget.setFixedWidth(1280)
128     widget.setCurrentIndex(widget.currentIndex()+1)
129
130 def selectCapture(self): #เมื่อกดปุ่มถ่ายรูป
131     face_id = self.txtID.text() #แสดงค่า ID ถัดไป
132     face_thainame = self.txtThainame.text() #รับค่าชื่อภาษาไทย
133     face_engname = self.txtEngname.text() #รับค่าชื่อภาษาอังกฤษ
134
135     #ตรวจสอบการคลิกเลือกจำนวนเงินจาก Radio Button
136     if self.rBtn_20.isChecked():
137         face_money = 20
138     elif self.rBtn_50.isChecked():
139         face_money = 50
140     elif self.rBtn_100.isChecked():
141         face_money = 100
142     elif self.rBtn_200.isChecked():
143         face_money = 200
144
145     msgBox = QMessageBox() #เปิดหน้าต่างป๊อปอัพขึ้นมาให้มองกล้อง
146     msgBox.setIcon(QMessageBox.Information)
147     msgBox.setText("กรุณามองกล้องเพื่อสแกนใบหน้าค่ะ")
148     msgBox.setWindowTitle("บันทึกใบหน้า")
149     msgBox.setStandardButtons(QMessageBox.Ok)
150     returnValue = msgBox.exec()
151     if returnValue == QMessageBox.Ok:
152         pass
153
154     count = 0 #เปิดกล้องเพื่อสแกนใบหน้า
155     cam = cv2.VideoCapture(0)
156     cam.set(3, 640) # set video width
157     cam.set(4, 480) # set video height
158
```

```

159 while True:
160     ret, img = cam.read()
161     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
162     faces = face_detector.detectMultiScale(img, 1.3, 5)
163     for (x, y, w, h) in faces:
164         cv2.rectangle(img, (x,y), (x+w, y+h), (255,0,0),2)
165         count += 1
166         #บันทึกรูปภาพในโฟลเดอร์ face_image
167         cv2.imwrite("dataset/user." + str(face_id) + '.'
168                     + str(count) + ".jpg", gray[y:y+h,x:x+w])
169         cv2.imshow('image', img)
170
171     #ถ้ามีการกดปุ่ม Esc หรือวนถ่ายภาพจนครบ 30 ภาพ
172     k = cv2.waitKey(100) & 0xff
173     if k == 27 or count >= 30:
174         cam.release() #ปิดกล้อง
175         cv2.destroyAllWindows()
176         break #ออกจากการวนลูป
177
178     cur.execute("INSERT INTO tbCustomer Values(?,?,?,?)",
179               (face_id,face_thainame,face_engname,face_money))
180     cn.commit()
181     import face_training #เรียนรู้ใบหน้า โดยเรียกใช้ไฟล์ face_training.py
182     self.btnCapture.hide()#ซ่อนปุ่มถ่ายรูป

```

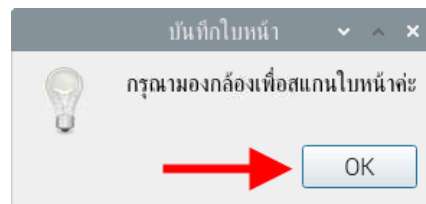
2.6 คลิกปุ่ม Run จากนั้นคลิกปุ่ม “เพิ่มข้อมูลลูกค้า”



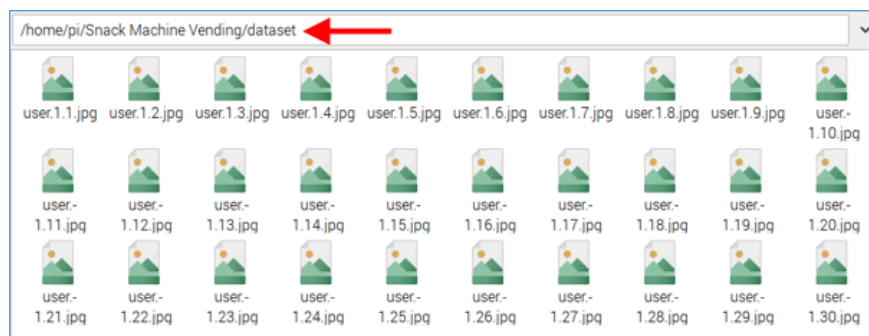
2.7 กรอกข้อมูลลูกค้า จากนั้นคลิกปุ่ม “ถ่ายรูป”



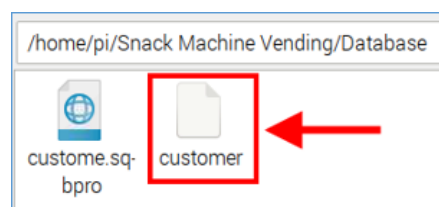
2.8 จะปรากฏหน้าต่างป๊อปอัพขึ้นมา ให้คลิกปุ่ม OK



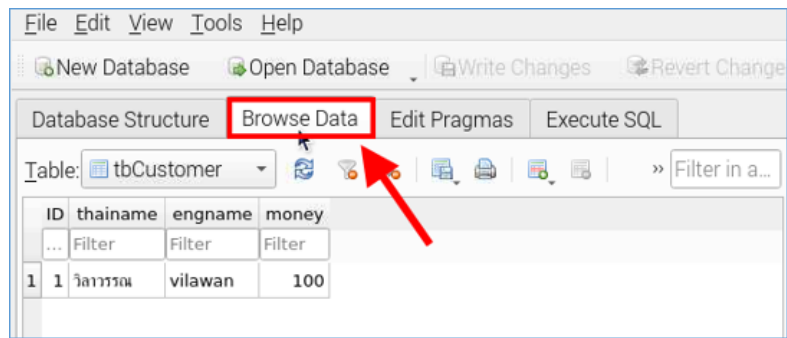
2.9 กล้องจะถูกเปิดขึ้นมา เพื่อวนลูปบันทึกใบหน้า 30 รูป เก็บไว้ในโฟลเดอร์ dataset



2.10 ข้อมูลที่กรอกเข้าไป จะถูกบันทึกเข้าไปในฐานข้อมูลที่อยู่ในโฟลเดอร์ Database ชื่อไฟล์ว่า customer สามารถดับเบิลคลิกที่ไฟล์ customer เพื่อเปิดดูเรคอร์ดข้อมูลได้



2.11 ไฟล์ฐานข้อมูลจะถูกเปิดด้วยโปรแกรม DB Browser for SQLite ให้คลิกที่เมนู Browse Data เพื่อดูเรคอร์ดข้อมูลลูกค้าที่บันทึกเข้ามาในระบบ



2.12 กลับมาที่หน้าจอโปรแกรม คลิกปุ่ม “กลับหน้าหลัก” เพื่อไปยังหน้าหลักของโปรแกรม



การเขียนโค้ดคำสั่งระบบเติมเงิน

หลักการทำงาน คือ เมื่อคลิกปุ่ม “เติมเงิน” จะเปิดกล่องขึ้นมาเพื่อวิเคราะห์ใบหน้าว่ามีในฐานข้อมูลหรือไม่ ถ้ามีในฐานข้อมูลจะเปิดหน้าต่างฟอร์มการเติมเงินขึ้นมา โดยจะแสดงยอดเงินคงเหลือในระบบ และให้เลือกจำนวนเงินที่ต้องการเติม จากนั้นคลิกปุ่ม “เติมเงิน” ก็ จะอัปเดตยอดเงินในฐานข้อมูล

1. แก้ไขฟังก์ชันปุ่มเติมเงิน gotoAddMoney()

1.1 ลบคำสั่ง print() ในฟังก์ชัน gotoAddMoney() ที่อยู่ในคลาส MainWindow() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```

56 def gotoAddMoney(self): #ฟังก์ชันไปยังหน้าต่าง "เติมเงิน"
57     widget.addWidget(AddMoney())
58     widget.setFixedHeight(400)
59     widget.setFixedWidth(600)
60     widget.setCurrentIndex(widget.currentIndex()+1)

```

1.2 สร้างคลาสหน้าต่าง AddMoney() ขึ้นมา เนื่องจากการอ้างอิงชื่อคลาส โดยให้พิมพ์คำสั่งไว้ด้านบนโค้ดคำสั่ง
app=QApplication(sys.argv) ดังนี้

```

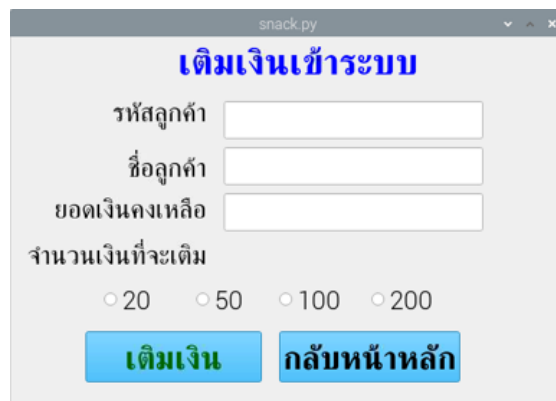
73 #หน้าต่าง GUI ของฟอร์มเพิ่มข้อมูลลูกค้า
74 class AddCustomer(QDialog):
75     def __init__(self):
76         super(AddCustomer, self).__init__()
77         loadUi("addcustomer.ui", self) #โหลดหน้าต่างออกแบบฟอร์มเพิ่มข้อมูลลูกค้าขึ้นมา
78
79 #หน้าต่าง GUI ของฟอร์มเติมเงิน
80 class AddMoney(QDialog):
81     def __init__(self):
82         super(AddMoney, self).__init__()
83         loadUi("AddMoney.ui", self) #โหลดหน้าต่างออกแบบฟอร์มฟอร์มเติมเงินขึ้นมา
84
85 app=QApplication(sys.argv) #เริ่มต้นสร้าง GUI

```

1.3 คลิกปุ่ม Run แล้วทดลองคลิกปุ่ม “เติมเงิน”



จะเห็นว่าหน้าจอ GUI จะเปลี่ยนเป็นหน้าจอฟอร์ม AddMoney.ui ดังรูป



2. เขียนโค้ดคำสั่งวิเคราะห์ใบหน้าและเติมเงินเข้าระบบในคลาส AddMoney()

2.1 เขียนโค้ดคำสั่งในคลาส AddMoney() ดังนี้

```

184 #หน้าต่าง GUI ของฟอร์มเติมเงิน
185 class AddMoney(QDialog):
186     def __init__(self):
187         super(AddMoney, self).__init__()
188         loadUi("AddMoney.ui", self) #โหลดหน้าต่างออกแบบฟอร์มฟอร์มเติมเงินขึ้นมา
189         self.btnAddMoney.clicked.connect(self.selectAddMoney)
190         self.btnBack.clicked.connect(self.gotoMainWindow)
191
192         #เปิดหน้าต่างป๊อปอัพขึ้นมาเพื่อบอกให้มองกล้อง
193         msgBox = QMessageBox()
194         msgBox.setIcon(QMessageBox.Information)
195         msgBox.setText("กรุณามองกล้องเพื่อสแกนใบหน้าในการเติมเงินค่ะ")
196         msgBox.setWindowTitle("ระบบเติมเงิน")
197         msgBox.setStandardButtons(QMessageBox.Ok)
198         returnValue = msgBox.exec()
199         if returnValue == QMessageBox.Ok:
200             pass
201
202         #วิเคราะห์ใบหน้า เพื่อตรวจสอบว่าตรงกับในฐานข้อมูลหรือไม่
203         import face_training #เรียนรู้ใบหน้า โดยทำคำสั่งในไฟล์ training.py
204         path = 'dataset' #ไฟล์เดอรัที่เก็บรูปใบหน้า
205         font = cv2.FONT_HERSHEY_SIMPLEX
206         capture = cv2.VideoCapture(0)
207         x = 0 #ตัวแปรในการวนลูป
208         id = 0
209         while True:
210             x += 1
211             ret, imgface = capture.read()
212             gray = cv2.cvtColor(imgface, cv2.COLOR_BGR2GRAY)
213             faces = face_detector.detectMultiScale(gray, 1.3, 5)
214             for (x,y,w,h) in faces:
215                 id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
216                 cv2.rectangle(imgface, (x,y), (x+w,y+h), (0,255,0), 2)
217                 #ค้นหาชื่อในฐานข้อมูล
218                 cmd = cur.execute("SELECT * FROM tbCustomer WHERE ID = "
219                                   + str(id))
220                 profile = None
221                 for row in cmd:
222                     profile = row
223
224                 #ถ้าใบหน้าตรงกับรูปภาพใบหน้าที่มีในระบบ ให้แสดงชื่อและค่าความแม่นยำ
225                 if profile != None:
226                     confidence = " {0}%".format(round(100 - confidence))
227                     cv2.putText(imgface, str(profile[1]) + str(confidence),
228                                 (x+5,y+h-5), font, 1, (255,255,0), 1)

```

```

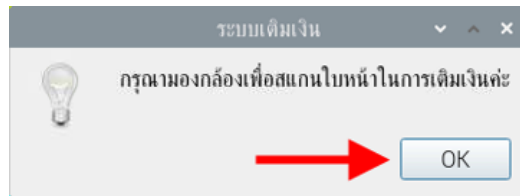
229         #ถ้าใบหน้าไม่ตรงให้ขึ้นว่า Unknown
230     else:
231         cv2.putText(imgface, 'Unknown', (x+5,y-5), font, 1,
232                     (255,255,255), 2)
233         moneyupdate = 0 #กำหนดตัวแปรเก็บเงินที่จะเติมเป็น 0
234
235     cv2.imshow('camera',imgface) #แสดงหน้าจอกล้อง
236
237     k = cv2.waitKey(10) & 0xff #ถ้ากดปุ่ม Esc หรือวนครบ 30 รอบก็จะออกจากลูป
238     if k == 27 or x >= 30:
239         break
240
241     capture.release() #ปิดกล้อง
242     cv2.destroyAllWindows() #ปิดหน้าต่าง
243
244     #แสดงข้อมูลลูกค้าใน Textbox
245     self.txtUpdateID.setText(str(profile[0]))
246     self.txtUpdateName.setText(str(profile[1]))
247     self.txtUpdateMoney.setText(str(profile[3]))
248
249 @pyqtSlot()
250 def gotoMainWindow(self): #ฟังก์ชันปุ่มย้อนกลับไปหน้าจอหลักของโปรแกรม
251     widget.addWidget(MainWindow())
252     widget.setFixedHeight(720)
253     widget.setFixedWidth(1280)
254     widget.setCurrentIndex(widget.currentIndex()+1)
255
256 def selectAddMoney(self): #ฟังก์ชันปุ่ม Radio Button สำหรับเลือกจำนวนเงิน
257     if self.rBtn_20.isChecked():
258         n = 20
259
260     elif self.rBtn_50.isChecked():
261         n = 50
262
263     elif self.rBtn_100.isChecked():
264         n = 100
265
266     elif self.rBtn_200.isChecked():
267         n = 200
268
269     #อัปเดตจำนวนเงินที่เติมเข้าไปในฐานข้อมูล
270     cur.execute("UPDATE tbCustomer SET money = "
271               + str(int(self.txtUpdateMoney.text()) + n)
272               + " WHERE ID = "
273               + str(self.txtUpdateID.text()))
274
275     cn.commit()
276     self.txtUpdateMoney.setText(str(int(self.txtUpdateMoney.text()) + n))
277     self.btnAddMoney.hide()

```

2.2 คลิกปุ่ม Run จากนั้นคลิกปุ่ม “เติมเงิน”



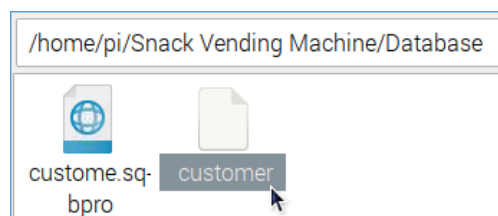
2.3 จะปรากฏหน้าต่างป๊อปอัพขึ้นมาให้มองกล้องเพื่อสแกนหน้า ให้คลิกปุ่ม OK



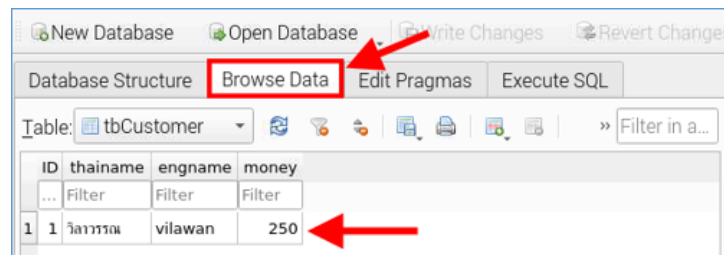
2.4 กล้องจะเปิดขึ้นมาเพื่อวนลูปสแกนใบหน้า 30 รอบและวิเคราะห์ใบหน้า จากนั้นจะดึงข้อมูลในฐานข้อมูลมาแสดงในฟอร์มเติมเงิน

2.5 คลิกเลือกจำนวนเงินที่จะเติม จากนั้นคลิกปุ่ม “เติมเงิน”

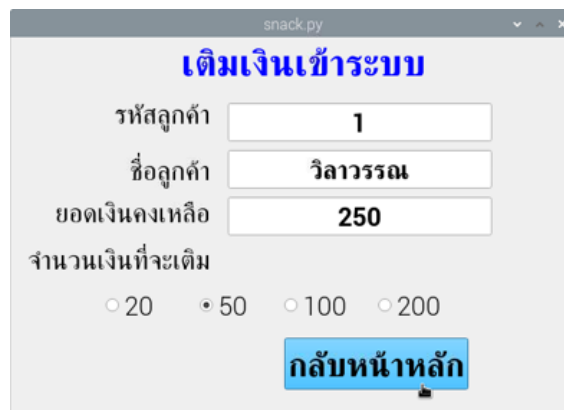
2.6 ระบบจะอัปเดตจำนวนเงินที่เติมเข้าไปในฐานข้อมูล โดยสามารถดับเบิลคลิกเปิดไฟล์ customer ที่อยู่ในโฟลเดอร์ Database ดูได้



2.7 ฐานข้อมูลจะถูกเปิดด้วยโปรแกรม DB Browser for SQLite จากนั้นคลิกแท็บ Browse Data จะเห็นว่าฟิลด์ money ได้ทำการอัปเดตยอดเงินเรียบร้อยแล้ว



2.8 หน้าต่างฟอร์มเติมเงินจะแสดงยอดเงินที่อัปเดต ปุ่มเติมเงินจะถูกซ่อน ให้เหลือแค่ปุ่มกลับหน้าหลัก ให้คลิกปุ่ม “กลับหน้าหลัก” เพื่อกลับไปยังหน้าหลักของโปรแกรม



การเขียนโค้ดคำสั่งระบบชำระเงิน

หลักการทำงาน คือ เมื่อเลือกขนมและคลิกปุ่ม “ยืนยัน” จะปรากฏหน้าต่างป๊อปอัพขึ้นมาเพื่อแจ้งให้ลูกค้ามองกล้อง จากนั้นคลิกปุ่ม “OK” กล้องจะเปิดขึ้นมาเพื่อสแกนใบหน้าว่าตรงกับฐานข้อมูลหรือไม่ ถ้าตรวจสอบแล้วมีในฐานข้อมูล จะแสดงข้อมูลลูกค้าขึ้นมาในช่องตรวจสอบข้อมูลลูกค้า หากถูกต้องให้คลิกปุ่ม “ชำระเงิน” หากไม่ถูกต้องให้คลิกปุ่ม “ยกเลิกการชำระเงิน” จะทำการรีเซ็ตหน้าจอทั้งหมดมีขั้นตอนดังนี้

1. กำหนดฟังก์ชันปุ่ม “ยืนยัน” selectOK()

1.1 ลบคำสั่ง print() ในฟังก์ชัน SelectOK() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```

101 def selectOK(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ยืนยัน"
102     #ตรวจสอบการเลือกขนม ถ้ายังไม่ได้เลือกขนม ให้เคลียร์ชื่อและราคายกขนม
103     if self.lblPrice.text() == "" or self.lblPrice.text() == "0":
104         self.lblSnack.setText('เลือกขนม')
105         self.lblPrice.setText('0')
106
107     else: #ถ้าเลือกขนมแล้ว
108         #สแกนและวิเคราะห์ใบหน้า
109         count = 0 #สร้างตัวแปรนับจำนวนรอบการวนลูปในสแกนใบหน้า
110         path = 'dataset' #โฟลเดอร์ที่เก็บรูปภาพใบหน้า
111         font = cv2.FONT_HERSHEY_SIMPLEX #ฟอนต์ที่จะแสดงในหน้าต่างกล้อง
112         cap = cv2.VideoCapture(0) #เรียกใช้งานกล้อง
113         cap.set(3, 640) # set video width
114         cap.set(4, 480) # set video height
115         # Define min window size to be recognized as a face
116         minW = 0.1*cap.get(3)
117         minH = 0.1*cap.get(4)
118
119         #เปิดหน้าต่างบ๊อบอัพ
120         msgBox = QMessageBox()
121         msgBox.setIcon(QMessageBox.Information)
122         msgBox.setText("กรุณามองกล้องเพื่อสแกนใบหน้าค่ะ")
123         msgBox.setWindowTitle("สแกนใบหน้า")
124         msgBox.setStandardButtons(QMessageBox.Ok)
125         returnValue = msgBox.exec()
126         if returnValue == QMessageBox.Ok: #ถ้าคลิกปุ่ม OK ให้ทำคำสั่งถัดไป
127             pass
128
129         while True: #วนสแกนใบหน้า 30 รอบ
130             ret, img = cap.read()
131             gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #แปลงรูปภาพเป็นขาวดำ
132             faces = face_detector.detectMultiScale(gray, 1.3, 5)
133             for (x,y,w,h) in faces:
134                 id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
135                 #สร้างกรอบสี่เหลี่ยมล้อมรอบใบหน้า
136                 cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
137                 #เปรียบเทียบใบหน้าและค้นหาชื่อในฐานข้อมูล
138                 cmd = cur.execute("SELECT * FROM tbCustomer WHERE ID = "
139                                 + str(id))
140                 #วนหาทีละแถว
141                 for row in cmd:
142                     profile = row #ใส่ข้อมูลที่ค้นพบลงในตัวแปร
143                     #ถ้าค่าความแม่นยำอยู่ระหว่าง 21 - 100%
144                     if (confidence < 100 and confidence > 20):
145                         id = profile[1] #เก็บค่าชื่อลูกค้าใส่ในตัวแปร id
146                         confidence = " {0}%".format(round(100 - confidence))
147                         #แสดงชื่อและค่าความแม่นยำบนหน้าจอกล้อง
148                         cv2.putText(img, str(profile[2]) + str(confidence)
149                                   , (x+5,y+h-5), font, 1, (255,255,0), 1)
150                         money = profile[3] #เก็บค่าจำนวนเงินของลูกค้าใส่ในตัวแปร money
151                     else:
152                         #id = "unknown" #เก็บค่าชื่อ "unknown" ใส่ในตัวแปร id
153                         confidence = " {0}%".format(round(100 - confidence))
154                         cv2.putText(img, 'Unknown', (x+5,y-5),
155                                   font, 1, (255,255,255), 2)
156                         money = 0 #ตัวแปร money = 0
157                 count += 1 #เพิ่มค่าตัวแปร count ทีละ 1 เพื่อวนลูปใหม่
158                 cv2.imshow('camera',img) #โชว์กล้อง
159
160                 k = cv2.waitKey(10) & 0xff #เมื่อกดปุ่ม 'ESC' หรือวนลูปครบ 30 รอบ
161                 if k == 27 or count >= 30:
162                     cap.release() #ปิดกล้อง
163                     cv2.destroyAllWindows()
164
165                 #แสดงข้อมูลลูกค้าให้ตรวจสอบก่อนชำระเงิน
166                 self.lblID.setText(str(profile[0]))
167                 self.lblName.setText(profile[1])
168                 self.lblMoney.setText(str(profile[3]))
169                 break #ออกจากลูป

```

2. กำหนดฟังก์ชันปุ่ม “ชำระเงิน” selectPay()

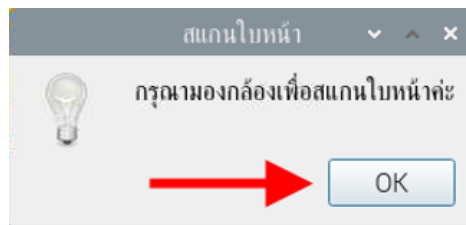
2.1 ลบคำสั่ง print() ในฟังก์ชัน selectPay() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
171 #ฟังก์ชันเมื่อคลิกปุ่ม "ชำระเงิน"
172 def selectPay(self):
173     #ถ้าเงินไม่พอจ่าย ให้โชว์ป๊อปอัพแจ้งว่าเงินไม่เพียงพอ
174     if int(self.lblMoney.text()) < int(self.lblPrice.text()):
175         msgBox = QMessageBox()
176         msgBox.setIcon(QMessageBox.Information)
177         msgBox.setText("คุณ" + profile[1] + " มีเงิน " + self.lblMoney.text()
178             + " บาท ไม่เพียงพอสำหรับซื้อขนม กรุณาเติมเงินก่อนค่ะ")
179         msgBox.setWindowTitle("Information's " + profile[1])
180         msgBox.setStandardButtons(QMessageBox.Ok)
181         returnValue = msgBox.exec()
182         #ถ้าคลิกปุ่ม OK ให้เคลียร์ชื่อและราคาขนม
183         if returnValue == QMessageBox.Ok:
184             self.lblSnack.setText('')
185             self.lblPrice.setText('0')
186
187     #ถ้าเงินเพียงพอ ให้ตรวจสอบเงื่อนไขว่าเลือกขนมอะไรและเงินต้องมากกว่าราคาขนม
188     else:
189         if (self.lblSnack.text() == 'ขนมฮานามิ') and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
190             print("คุณได้รับขนมฮานามิ")
191         elif self.lblSnack.text() == 'ขนมสแน็คแจ็ค' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
192             print("คุณได้รับขนมสแน็คแจ็ค")
193         elif self.lblSnack.text() == 'ขนมเลย์' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
194             print("คุณได้รับขนมเลย์")
195         elif self.lblSnack.text() == 'ขนมป๊อปคอร์น' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
196             print("คุณได้รับขนมป๊อปคอร์น")
197         elif self.lblSnack.text() == 'ขนมปาร์ตี้' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
198             print("คุณได้รับขนมปาร์ตี้")
199         elif self.lblSnack.text() == 'ขนมเบเกอรี่' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
200             print("คุณได้รับขนมเบเกอรี่")
201
202     #อัปเดตจำนวนเงินคงเหลือในฐานข้อมูล
203     cur.execute("UPDATE tbCustomer SET money = " + str(int(self.lblMoney.text()) - int(self.lblPrice.text()))
204         + " WHERE ID = " + self.lblID.text())
205     cn.commit()
206
207     #รีเซ็ตหน้าต่างชื่อขนม
208     self.lblSnack.setText('และเลือกขนม')
209     self.lblPrice.setText('0')
210     self.lblID.setText('')
211     self.lblName.setText('')
212     self.lblMoney.setText('')
```

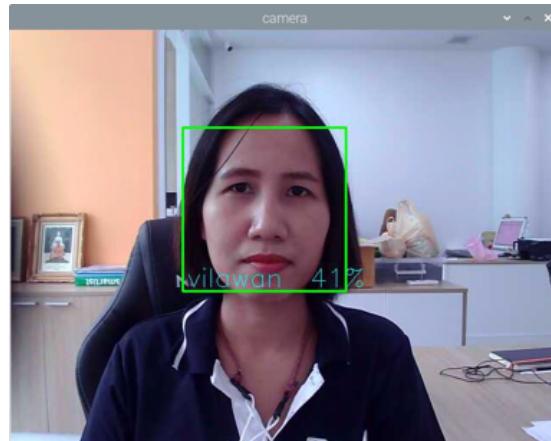
2.2 คลิกปุ่ม Run จากนั้นเลือกขนม 1 ชนิด แล้วคลิกปุ่ม “ยืนยัน”



2.3 จะปรากฏป๊อปอัพขึ้นมา ให้คลิกปุ่ม “OK”



2.4 กดปุ่มจะเปิดขึ้นมาสแกนและวิเคราะห์ใบหน้า โดยจะแสดงชื่อลูกค้าขึ้นมา



2.5 ข้อมูลของลูกค้าที่อยู่พื้นฐานข้อมูลจะถูกดึงมาแสดงในหน้าจอ ดังรูป

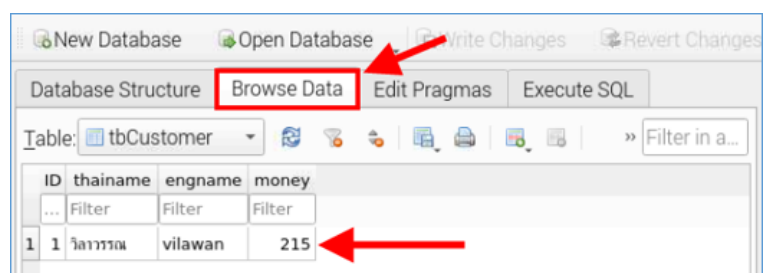


2.6 กดปุ่ม “ชำระเงิน” จะแสดงข้อความในช่อง Shell ว่าได้รับขนมเรียบร้อยแล้ว และจะรีเซ็ตหน้าจอทั้งหมด



```
Shell
Python 3.9.2 (/usr/bin/python3)
>>> %Run snack.py
คุณได้รับขนมฮานามิ
```

2.7 คลิกเข้าไปดูในฐานข้อมูล จะเห็นว่าจำนวนเงินของลูกค้าจะถูกหักเงินออกไปตามราคาขนม



3. กำหนดฟังก์ชันปุ่ม “ยกเลิกการชำระเงิน” SelectCancelPay()

3.1 ลบคำสั่ง print() ในฟังก์ชัน SelectCancelPay() แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
76 #ฟังก์ชันเมื่อกดปุ่ม "ยกเลิกการชำระเงิน" จะเคลียร์หน้าจอทั้งหมด
77 def SelectCancelPay(self):
78     self.lblSnack.setText('แต่ละเลือกขนม')
79     self.lblPrice.setText('0')
80     self.lblID.setText('')
81     self.lblName.setText('')
82     self.lblMoney.setText('')
```

3.2 คลิกปุ่ม Run แล้วทดลองเลือกสินค้าและสแกนใบหน้า จากนั้นลองคลิกปุ่ม “ยกเลิกการชำระเงิน” ข้อมูลบนหน้าจอจะถูกรีเซ็ตทั้งหมด ดังรูป



การเขียนโค้ดคำสั่งให้ขนมหล่นลงมายังช่องรับขนม

1. เขียนคำสั่งเรียกใช้ขา GPIO ของบอร์ด Raspberry Pi

```

1 import sys
2 from PyQt5.QtCore import pyqtSlot #โมดูลสำหรับสร้างหน้าจอ GUI
3 from PyQt5 import QtWidgets, QtGui
4 from PyQt5.QtWidgets import QApplication, QMainWindow, QDialog, QMessageBox
5 from PyQt5.uic import loadUi
6
7 import os
8 import os.path
9 import cv2 #เปิดใช้งาน OpenCV เพื่อใช้งานระบบวิเคราะห์ใบหน้า
10 import numpy as np #โมดูลที่ใช้ในการคำนวณทางคณิตศาสตร์
11 import sqlite3 #โมดูลที่ใช้ในการจัดการฐานข้อมูล
12
13 import RPi.GPIO as GPIO #โมดูลสำหรับเรียกใช้ขา GPIO
14 from time import sleep #โมดูลจัดการเวลา
15 GPIO.setwarnings(False) #ปิดการแจ้งเตือนในหน้าต่าง Shell

```

2. กำหนดขา GPIO ที่เชื่อมต่อกับมอเตอร์แต่ละตัว ก่อนคำสั่ง class MainWindow() ดังรูป


```

33 GPIO.setmode(GPIO.BOARD) #เรียกขา GPIO ตามลำดับขา
34 GPIO.setup(29,GPIO.OUT) #Servo 1 Hanami
35 GPIO.setup(31,GPIO.OUT) #Servo 2 SnackJack
36 GPIO.setup(33,GPIO.OUT) #Servo 3 Lays
37 GPIO.setup(35,GPIO.OUT) #Servo 4 Paprika
38 GPIO.setup(36,GPIO.OUT) #Servo 5 Party
39 GPIO.setup(37,GPIO.OUT) #Servo 6 Jaxx
40 pwm1 = GPIO.PWM(29,50) #กำหนดความถี่ของ Servo แต่ละตัว
41 pwm2 = GPIO.PWM(31,50)
42 pwm3 = GPIO.PWM(33,50)
43 pwm4 = GPIO.PWM(35,50)
44 pwm5 = GPIO.PWM(36,50)
45 pwm6 = GPIO.PWM(37,50)
46 pwm1.start(0) #สั่งให้ Servo ทุกตัวเริ่มต้นการทำงาน
47 pwm2.start(0)
48 pwm3.start(0)
49 pwm4.start(0)
50 pwm5.start(0)
51 pwm6.start(0)
52
53 #หน้าต่างเมนูหลัก คือ ฟอรัม main.ui
54 class MainWindow(QDialog):
55     def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI

```

3. แก้ไขโค้ดคำสั่งในฟังก์ชัน selectPay() โดยลบคำสั่ง print() ในแต่ละเงื่อนไขออก แล้วใส่โค้ดคำสั่งให้มอเตอร์หมุน ดังรูป

```

195 #ฟังก์ชันเมื่อคลิกปุ่ม "ชำระเงิน"
196 def selectPay(self):
197     #ถ้าเงินไม่พอจ่าย ให้โชว์ป๊อปอัพแจ้งว่าเงินไม่เพียงพอ
198     if int(self.lblMoney.text()) < int(self.lblPrice.text()):
199         msgBox = QMessageBox()
200         msgBox.setIcon(QMessageBox.Information)
201         msgBox.setText("คุณ" + profile[1] + " มีเงิน " + self.lblMoney.text()
202             + " บาท ไม่เพียงพอสำหรับซื้อขนม กรุณาเติมเงินก่อนค่ะ")
203         msgBox.setWindowTitle("Information's " + profile[1])
204         msgBox.setStandardButtons(QMessageBox.Ok)
205         returnValue = msgBox.exec()
206         #ถ้าคลิกปุ่ม OK ให้เคลียร์ชื่อและราคาสินค้า
207         if returnValue == QMessageBox.Ok:
208             self.lblSnack.setText('')
209             self.lblPrice.setText('0')
210
211     #ถ้าเงินเพียงพอ ให้ตรวจสอบเงื่อนไขว่าเลือกขนมอะไรและเงินต้องมากกว่าราคาสินค้า
212     else:
213         if (self.lblSnack.text() == 'ขนมฮานามิ' and int(self.lblMoney.text()) >= int(self.lblPrice.text())):
214             pwm1.ChangeDutyCycle(10) #หมุนไปทางขวา 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหมุน
215             sleep(2)
216             pwm1.ChangeDutyCycle(0)
217         elif self.lblSnack.text() == 'ขนมสแน็คแจ็ค' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
218             pwm2.ChangeDutyCycle(10) #หมุนไปทางขวา 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหมุน
219             sleep(2)
220             pwm2.ChangeDutyCycle(0)
221         elif self.lblSnack.text() == 'ขนมเลย์' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
222             pwm3.ChangeDutyCycle(10) #หมุนไปทางขวา 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหมุน
223             sleep(2)
224             pwm3.ChangeDutyCycle(0)
225         elif self.lblSnack.text() == 'ขนมป๊อปริค้า' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
226             pwm4.ChangeDutyCycle(5) ##หมุนไปทางซ้าย 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหมุน
227             sleep(2)
228             pwm4.ChangeDutyCycle(0)
229         elif self.lblSnack.text() == 'ขนมปาร์ตี' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
230             pwm5.ChangeDutyCycle(5) ##หมุนไปทางซ้าย 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหมุน
231             sleep(2)
232             pwm5.ChangeDutyCycle(0)
233         elif self.lblSnack.text() == 'ขนมแจ็กซ์' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
234             pwm6.ChangeDutyCycle(10) #หมุนไปทางขวา 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหมุน
235             sleep(2)
236             pwm6.ChangeDutyCycle(0)

```

การเขียนโค้ดคำสั่งปิดหน้าจอโปรแกรม

1. ลบคำสั่ง `print()` ในฟังก์ชัน `gotoClose()` ที่อยู่ในคลาส `MainWindow()` แล้วเปลี่ยนเป็นคำสั่งต่อไปนี้

```
84 def gotoClose(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ปิดหน้าจอ" จะปิดหน้าต่าง GUI
85     widget.close()
```

2. เมื่อคลิกปุ่ม Run แล้วทดลองคลิกปุ่ม “ปิดหน้าจอ” หน้าจอ GUI จะถูกปิด



โค้ดคำสั่งทั้งหมดของโปรแกรมตู้จำหน่ายขนมอัตโนมัติ

```

1 import sys
2 from PyQt5.QtCore import pyqtSlot #โมดูลสำหรับสร้างหน้าจอ GUI
3 from PyQt5 import QtWidgets, QtGui
4 from PyQt5.QtWidgets import QApplication, QMainWindow, QDialog, QMessageBox
5 from PyQt5.uic import loadUi
6
7 import os
8 import os.path
9 import cv2 #เปิดใช้งาน OpenCV เพื่อใช้งานระบบวีเคราะห์ใบหน้า
10 import numpy as np #โมดูลที่ใช้ในการคำนวณทางคณิตศาสตร์
11 import sqlite3 #โมดูลที่ใช้ในการจัดการฐานข้อมูล
12
13 import RPi.GPIO as GPIO #โมดูลสำหรับเรียกใช้ขา GPIO
14 from time import sleep #โมดูลจัดการเวลา
15 GPIO.setwarnings(False) #ปิดการแจ้งเตือนในหน้าต่าง Shell
16
17 cn = sqlite3.connect('Database/customer')#ระบุไปยังไฟล์ฐานข้อมูล
18 cur = cn.cursor() #สร้างออบเจกต์เพื่อไปยังฐานข้อมูล
19
20 #โหลดไฟล์ cascade สำหรับตรวจจับใบหน้า
21 face_detector = cv2.CascadeClassifier("haarcascade_frontalface_default.xml")
22 recognizer = cv2.face.LBPHFaceRecognizer_create()
23 recognizer.read('trainer.yml')
24
25 #ตัวแปรเก็บราคาสินค้าละชนิด
26 price_Hanami = 15
27 price_Jaxx = 10
28 price_Lays = 20
29 price_Paprika = 15
30 price_Party = 10
31 price_Snackjack = 20
32
33 GPIO.setmode(GPIO.BOARD) #เรียกขา GPIO ตามลำดับขา
34 GPIO.setup(29,GPIO.OUT) #Servo 1 Hanami
35 GPIO.setup(31,GPIO.OUT) #Servo 2 SnackJack
36 GPIO.setup(33,GPIO.OUT) #Servo 3 Lays
37 GPIO.setup(35,GPIO.OUT) #Servo 4 Paprika
38 GPIO.setup(36,GPIO.OUT) #Servo 5 Party
39 GPIO.setup(37,GPIO.OUT) #Servo 6 Jaxx
40 pwm1 = GPIO.PWM(29,50) #กำหนดความถี่ของ Servo แต่ละตัว
41 pwm2 = GPIO.PWM(31,50)
42 pwm3 = GPIO.PWM(33,50)
43 pwm4 = GPIO.PWM(35,50)
44 pwm5 = GPIO.PWM(36,50)
45 pwm6 = GPIO.PWM(37,50)
46 pwm1.start(0) #สั่งให้ Servo ทุกตัวเริ่มเดินการทำงาน
47 pwm2.start(0)
48 pwm3.start(0)
49 pwm4.start(0)
50 pwm5.start(0)
51 pwm6.start(0)
52
53 #หน้าต่างเมนูหลัก คือ ฟอรัม main.ui
54 class MainWindow(QDialog):
55     def __init__(self): #ฟังก์ชันหลักของหน้าต่าง GUI
56         super(MainWindow, self).__init__()
57         loadUi('main.ui',self) #เรียกหน้าต่าง main.ui ขึ้นมา
58         self.btnHanami.clicked.connect(self.selectHanami) #ปุ่มขนมฮานามิ
59         self.btnJaxx.clicked.connect(self.selectJaxx) #ปุ่มขนมแจ๊คซ์
60         self.btnLays.clicked.connect(self.selectLays) #ปุ่มขนมเลย์
61         self.btnPaprika.clicked.connect(self.selectPaprika) #ปุ่มขนมปาปริก้า
62         self.btnParty.clicked.connect(self.selectParty) #ปุ่มขนมปาร์ตี้
63         self.btnSnackjack.clicked.connect(self.selectSnackjack) #ปุ่มขนมสแน็คแจ๊ค
64         self.btnOK.clicked.connect(self.selectOK) #ปุ่มยืนยัน
65         self.btnClear.clicked.connect(self.selectClear) #ปุ่มยกเลิก
66         self.btnPay.clicked.connect(self.selectPay) #ปุ่มชำระเงิน
67         self.btnCancelPay.clicked.connect(self.SelectCancelPay) #ปุ่มยกเลิกการชำระเงิน
68         self.btnAddCust.clicked.connect(self.gotoAddCustomer) #ปุ่มเพิ่มข้อมูลลูกค้า
69         self.btnAddMoney.clicked.connect(self.gotoAddMoney) #ปุ่มเติมเงิน
70         self.btnClose.clicked.connect(self.gotoClose) #ปุ่มปิดหน้าจอ
71
72     @pyqtSlot() #ฟังก์ชันปุ่มในหน้าต่าง GUI ของฟอรัมชื่อขนม
73     def gotoAddCustomer(self):
74         #กดปุ่มเพิ่มข้อมูลลูกค้า แล้วเปิดหน้าจอ GUI เพิ่มข้อมูลลูกค้า
75         widget.addWidget(AddCustomer())
76         widget.setFixedHeight(400)
77         widget.setFixedWidth(600)
78         widget.setCurrentIndex(widget.currentIndex()+1)
79
80     def gotoAddMoney(self): #ฟังก์ชันไปยังหน้าต่าง "เติมเงิน"

```

```

81 widget.addWidget(AddMoney())
82 widget.setFixedHeight(400)
83 widget.setFixedWidth(600)
84 widget.setCurrentIndex(widget.currentIndex()+1)
85
86 def selectHanami(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Hanami"
87     self.lblPrice.setText(str(price_Hanami)) #แสดงราคาขนม
88     self.lblSnack.setText('ขนมฮานามิ') #แสดงชื่อขนม
89
90 def selectSnackjack(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Snackjack"
91     self.lblPrice.setText(str(price_Snackjack)) #แสดงราคาขนม
92     self.lblSnack.setText('ขนมสแน็คแจ็ค') #แสดงชื่อขนม
93
94 def selectLays(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Lays"
95     self.lblPrice.setText(str(price_Lays)) #แสดงราคาขนม
96     self.lblSnack.setText('ขนมเลย์') #แสดงชื่อขนม
97
98 def selectPaprika(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Paprika"
99     self.lblPrice.setText(str(price_Paprika)) #แสดงราคาขนม
100    self.lblSnack.setText('ขนมปาปริก้า') #แสดงชื่อขนม
101
102 def selectParty(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Party"
103    self.lblPrice.setText(str(price_Party)) #แสดงราคาขนม
104    self.lblSnack.setText('ขนมปาร์ตี้') #แสดงชื่อขนม
105
106 def selectJaxx(self): #ฟังก์ชันเมื่อคลิกเลือกขนม "Jaxx"
107    self.lblPrice.setText(str(price_Jaxx)) #แสดงราคาขนม
108    self.lblSnack.setText('ขนมแจ็กซ์') #แสดงชื่อขนม
109
110 def selectClear(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ยกเลิก" จะเคลียร์รีเซ็ตหน้าจอ
111    self.lblSnack.setText('แต่เลือกขนม')
112    self.lblPrice.setText('0')
113
114 #ฟังก์ชันเมื่อกดปุ่ม "ยกเลิกการชำระเงิน" จะเคลียร์หน้าจอทั้งหมด
115 def SelectCancelPay(self):
116    self.lblSnack.setText('แต่เลือกขนม')
117    self.lblPrice.setText('0')
118    self.lblID.setText('')
119    self.lblName.setText('')
120    self.lblMoney.setText('')
121
122 def gotoClose(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ปิดหน้าจอ" จะปิดหน้าต่าง GUI
123    widget.close()
124
125 def selectOK(self): #ฟังก์ชันเมื่อคลิกปุ่ม "ยืนยัน"
126    #ตรวจสอบการเลือกขนม ถ้ายังไม่ได้เลือกขนม ให้เคลียร์ชื่อและราคาขนม
127    if self.lblPrice.text() == "" or self.lblPrice.text() == "0":
128        self.lblSnack.setText('แต่เลือกขนม')
129        self.lblPrice.setText('0')
130
131    else: #ถ้าเลือกขนมแล้ว
132        #สแกนและวิเคราะห์ใบหน้า
133        count = 0 #สร้างตัวแปรนับจำนวนรอบการวนลูปในสแกนใบหน้า
134        path = 'dataset' #โฟลเดอร์ที่เก็บรูปภาพใบหน้า
135        font = cv2.FONT_HERSHEY_SIMPLEX #ฟอนต์ที่จะแสดงในหน้าต่างกล่อง
136        cap = cv2.VideoCapture(0) #เรียกใช้จากกล้อง
137        cap.set(3, 640) # set video width
138        cap.set(4, 480) # set video height
139        # Define min window size to be recognized as a face
140        minW = 0.1*cap.get(3)
141        minH = 0.1*cap.get(4)
142
143        #เปิดหน้าต่างป๊อปอัพ
144        msgBox = QMessageBox()
145        msgBox.setIcon(QMessageBox.Information)
146        msgBox.setText("กรุณามองกล้องเพื่อสแกนใบหน้าค่ะ")
147        msgBox.setWindowTitle("สแกนใบหน้า")
148        msgBox.setStandardButtons(QMessageBox.Ok)
149        returnValue = msgBox.exec()
150        if returnValue == QMessageBox.Ok: #ถ้าคลิกปุ่ม OK ให้ทำคำสั่งถัดไป
151            pass
152
153        while True: #วนสแกนใบหน้า 30 รอบ
154            ret, img = cap.read()
155            gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) #แปลงรูปภาพเป็นขาวดำ
156            faces = face_detector.detectMultiScale(gray, 1.3, 5)
157            for (x,y,w,h) in faces:
158                id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
159                #สร้างกรอบสี่เหลี่ยมล้อมรอบใบหน้า
160                cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)

```

```

161 #เปรียบเทียบใบหน้าและค้นหาชื่อในฐานข้อมูล
162 cmd = cur.execute("SELECT * FROM tbCustomer WHERE ID = "
163                  + str(id))
164
165 #วนหาทีละแถว
166 for row in cmd:
167     profile = row #ใส่ข้อมูลที่ค้นพบลงในตัวแปร
168     #ถ้าค่าความแม่นยำอยู่ระหว่าง 21 - 100%
169     if (confidence < 100 and confidence > 20):
170         id = profile[1] #เก็บค่าชื่อลูกค้าไว้ในตัวแปร id
171         confidence = " {0}%".format(round(100 - confidence))
172         #แสดงชื่อและค่าความแม่นยำบนหน้าจอกล้อง
173         cv2.putText(img, str(profile[2]) + str(confidence)
174                    , (x+5,y+h-5), font, 1, (255,255,0), 1)
175         money = profile[3] #เก็บค่าจำนวนเงินของลูกค้าไว้ในตัวแปร money
176     else:
177         #id = "unknown" #เก็บค่าชื่อ "unknown" ไว้ในตัวแปร id
178         confidence = " {0}%".format(round(100 - confidence))
179         cv2.putText(img, 'Unknown', (x+5,y-5),
180                    font, 1, (255,255,255), 2)
181         money = 0 #ตัวแปร money = 0
182
183 count += 1 #เพิ่มค่าตัวแปร count ทีละ 1 เพื่อวนลูปใหม่
184 cv2.imshow('camera',img) #โชว์กล้อง
185
186 k = cv2.waitKey(10) & 0xff #เมื่อกดปุ่ม 'ESC' หรือวนลูปครบ 30 รอบ
187 if k == 27 or count >= 30:
188     cap.release() #ปิดกล้อง
189     cv2.destroyAllWindows()
190
191 #แสดงข้อมูลลูกค้าให้ตรวจสอบก่อนชำระเงิน
192 self.lblID.setText(str(profile[0]))
193 self.lblName.setText(str(profile[1]))
194 self.lblMoney.setText(str(profile[3]))
195 break #ออกจากลูป
196
197 #ฟังก์ชันเมื่อคลิกปุ่ม "ชำระเงิน"
198 def selectPay(self):
199     #ถ้าเงินไม่พอจ่าย ให้โชว์ป๊อปอัพแจ้งว่าเงินไม่เพียงพอ
200     if int(self.lblMoney.text()) < int(self.lblPrice.text()):
201         msgBox = QMessageBox()
202         msgBox.setIcon(QMessageBox.Information)
203         msgBox.setText("คุณ " + profile[1] + " มีเงิน " + self.lblMoney.text()
204                        + " บาท ไม่เพียงพอสำหรับซื้อขนม กรุณาเติมเงินก่อนค่ะ")
205         msgBox.setWindowTitle("Information's " + profile[1])
206         msgBox.setStandardButtons(QMessageBox.Ok)
207         returnValue = msgBox.exec()
208         #ถ้าคลิกปุ่ม OK ให้เคลียร์ชื่อและราคาขนม
209         if returnValue == QMessageBox.Ok:
210             self.lblSnack.setText('')
211             self.lblPrice.setText('0')
212
213     #ถ้าเงินเพียงพอ ให้ตรวจสอบเงื่อนไขว่าเลือกขนมอะไรและเงินต้องมากกว่าราคาขนม
214     else:
215         if (self.lblSnack.text() == 'ขนมฮานามิ') and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
216             pwm1.ChangeDutyCycle(10) #หมุนไปทางขวา 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหล่น
217             sleep(2)
218             pwm1.ChangeDutyCycle(0)
219         elif self.lblSnack.text() == 'ขนมสเน็คแจ็ค' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
220             pwm2.ChangeDutyCycle(10) #หมุนไปทางขวา 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหล่น
221             sleep(2)
222             pwm2.ChangeDutyCycle(0)
223         elif self.lblSnack.text() == 'ขนมเลอ' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
224             pwm3.ChangeDutyCycle(10) #หมุนไปทางขวา 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหล่น
225             sleep(2)
226             pwm3.ChangeDutyCycle(0)
227         elif self.lblSnack.text() == 'ขนมป๊อปรั๊ก' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
228             pwm4.ChangeDutyCycle(5) ##หมุนไปทางซ้าย 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหล่น
229             sleep(2)
230             pwm4.ChangeDutyCycle(0)
231         elif self.lblSnack.text() == 'ขนมปาร์ตี้' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
232             pwm5.ChangeDutyCycle(5) ##หมุนไปทางซ้าย 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหล่น
233             sleep(2)
234             pwm5.ChangeDutyCycle(0)
235         elif self.lblSnack.text() == 'ขนมแจ๊คช' and int(self.lblMoney.text()) >= int(self.lblPrice.text()):
236             pwm6.ChangeDutyCycle(10) ##หมุนไปทางขวา 90 องศาเป็นเวลา 2 วินาที ขนมก็จะหล่น
237             sleep(2)
238             pwm6.ChangeDutyCycle(0)
239
240     #อัพเดทจำนวนเงินคงเหลือในฐานข้อมูล
241     cur.execute("UPDATE tbCustomer SET money = " + str(int(self.lblMoney.text()) - int(self.lblPrice.text()))
242                + " WHERE ID = " + self.lblID.text())

```



```

241         cn.commit()
242
243         #รีเซ็ตหน้าตาข้อมูล
244         self.lblSnack.setText('แต่เลือกขนม')
245         self.lblPrice.setText('0')
246         self.lblID.setText('')
247         self.lblName.setText('')
248         self.lblMoney.setText('')
249
250 #หน้าต่าง GUI ของฟอร์มเพิ่มข้อมูลลูกค้า
251 class AddCustomer(QDialog):
252     def __init__(self):
253         super(AddCustomer, self).__init__()
254         loadUi("addcustomer.ui", self) #โหลดหน้าต่างฟอร์มเพิ่มข้อมูลลูกค้าขึ้นมา
255         #เมื่อกดปุ่มถ่ายรูป ให้ไปทำคำสั่งในฟังก์ชัน selectCapture
256         self.btnCapture.clicked.connect(self.selectCapture)
257
258         #กำหนดให้ช่องรหัสลูกค้าแสดง ID ถัดไป โดยไม่ต้องกรอกเอง
259         ncust = cur.execute("SELECT * FROM tbCustomer")
260         nextID = (len(ncust.fetchall())) + 1
261         self.txtID.setText(str(nextID))
262
263         #เมื่อกดปุ่มย้อนกลับ ให้ไปทำคำสั่งในฟังก์ชัน gotoMainWindow
264         self.btnBack.clicked.connect(self.gotoMainWindow)
265
266 @pyqtSlot() #ฟังก์ชันปุ่มในหน้าต่าง GUI ของฟอร์มเพิ่มข้อมูลลูกค้า
267 def gotoMainWindow(self): #ไปหน้า GUI ฟอร์มชื่อสินค้า
268     widget.addWidget(MainWindow())
269     widget.setFixedHeight(800)
270     widget.setFixedWidth(1280)
271     widget.setCurrentIndex(widget.currentIndex()+1)
272
273 def selectCapture(self): #เมื่อกดปุ่มถ่ายรูป
274     face_id = self.txtID.text() #แสดงค่า ID ถัดไป
275     face_thainame = self.txtThainame.text() #รับคำชื่อภาษาไทย
276     face_engname = self.txtEngname.text() #รับคำชื่อภาษาอังกฤษ
277
278     #ตรวจสอบการคลิกเลือกจำนวนเงินจาก Radio Button
279     if self.rBtn_20.isChecked():
280         face_money = 20
281     elif self.rBtn_50.isChecked():
282         face_money = 50
283     elif self.rBtn_100.isChecked():
284         face_money = 100
285     elif self.rBtn_200.isChecked():
286         face_money = 200
287
288     msgBox = QMessageBox() #เปิดหน้าต่างป๊อปอัพขึ้นมาให้มองกล่อง
289     msgBox.setIcon(QMessageBox.Information)
290     msgBox.setText("กรุณามองกล่องเพื่อสแกนใบหน้า:")
291     msgBox.setWindowTitle("บันทึกใบหน้า")
292     msgBox.setStandardButtons(QMessageBox.Ok)
293     returnValue = msgBox.exec()
294     if returnValue == QMessageBox.Ok:
295         pass
296
297     count = 0 #เปิดกล่องเพื่อสแกนใบหน้า
298     cam = cv2.VideoCapture(0)
299     cam.set(3, 640) # set video width
300     cam.set(4, 480) # set video height
301
302     while True:
303         ret, img = cam.read()
304         gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
305         faces = face_detector.detectMultiScale(img, 1.3, 5)
306         for (x, y, w, h) in faces:
307             cv2.rectangle(img, (x,y), (x+w, y+h), (255,0,0),2)
308             count += 1
309             #บันทึกรูปภาพไปโฟลเดอร์ face_image
310             cv2.imwrite("dataset/user." + str(face_id) + '.'
311                         + str(count) + ".jpg", gray[y:y+h,x:x+w])
312             cv2.imshow('image', img)
313
314         #ถ้ามีการกดปุ่ม Esc หรือวนถ่ายภาพจนครบ 30 ภาพ
315         k = cv2.waitKey(100) & 0xff
316         if k == 27 or count >= 30:
317             cam.release() #ปิดกล้อง
318             cv2.destroyAllWindows()
319             break #ออกจากการทำงาน
320

```



```

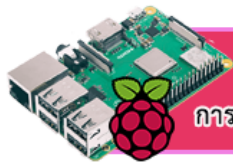
321 cur.execute("INSERT INTO tbCustomer Values(?,?,?,?)",
322             (face_id, face_thainame, face_engname, face_money))
323 cn.commit()
324 import face_training #เรียนรู้ใบหน้า โดยเรียกใช้ไฟล์ face_training.py
325 self.btnCapture.hide() #ซ่อนปุ่มถ่ายรูป
326
327 #หน้าต่าง GUI ของฟอร์มเติมเงิน
328 class AddMoney(QDialog):
329     def __init__(self):
330         super(AddMoney, self).__init__()
331         loadUi("AddMoney.ui", self) #โหลดหน้าต่างออกแบบฟอร์มฟอร์มเติมเงินขึ้นมา
332         self.btnAddMoney.clicked.connect(self.selectAddMoney)
333         self.btnBack.clicked.connect(self.gotoMainWindow)
334
335         #เปิดหน้าต่างป๊อปอัพขึ้นมาเพื่อบอกให้มองกล้อง
336         msgBox = QMessageBox()
337         msgBox.setIcon(QMessageBox.Information)
338         msgBox.setText("กรุณามองกล้องเพื่อสแกนใบหน้าในการเติมเงินค่ะ")
339         msgBox.setWindowTitle("ระบบเติมเงิน")
340         msgBox.setStandardButtons(QMessageBox.Ok)
341         returnValue = msgBox.exec()
342         if returnValue == QMessageBox.Ok:
343             pass
344
345         #วิเคราะห์ใบหน้า เพื่อตรวจสอบว่าตรงกับใบฐานข้อมูลหรือไม่
346         import face_training #เรียนรู้ใบหน้า โดยทำคำสั่งในไฟล์ training.py
347         path = 'dataset' #โฟลเดอร์ที่เก็บรูปใบหน้า
348         font = cv2.FONT_HERSHEY_SIMPLEX
349         capture = cv2.VideoCapture(0)
350         x = 0 #ตัวแปรในการวนลูป
351         id = 0
352         while True:
353             x += 1
354             ret, imgface = capture.read()
355             gray = cv2.cvtColor(imgface, cv2.COLOR_BGR2GRAY)
356             faces = face_detector.detectMultiScale(gray, 1.3, 5)
357             for (x,y,w,h) in faces:
358                 id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
359                 cv2.rectangle(imgface, (x,y), (x+w,y+h), (0,255,0), 2)
360                 #ค้นหาชื่อใบฐานข้อมูล
361                 cmd = cur.execute("SELECT * FROM tbCustomer WHERE ID = "
362                                 + str(id))
363                 profile = None
364                 for row in cmd:
365                     profile = row
366
367                 #ถ้าใบหน้าตรงกับรูปภาพใบหน้าที่มีในระบบ ให้แสดงชื่อและค่าความแม่นยำ
368                 if profile != None:
369                     confidence = " {:.1f}%".format(round(100 - confidence))
370                     cv2.putText(imgface, str(profile[1]) + str(confidence),
371                                (x+5,y+h-5), font, 1, (255,255,0), 1)
372                 #ถ้าใบหน้าไม่ตรงให้ขึ้นว่า Unknown
373                 else:
374                     cv2.putText(imgface, 'Unknown', (x+5,y-5), font, 1,
375                                (255,255,255), 2)
376                 moneyupdate = 0 #กำหนดตัวแปรเก็บเงินที่จะเติมเป็น 0
377
378             cv2.imshow('camera',imgface) #แสดงหน้าจอกล้อง
379
380             k = cv2.waitKey(10) & 0xff #ถ้ากดปุ่ม Esc หรือวนครบ 30 รอบก็จะออกจากลูป
381             if k == 27 or x >= 30:
382                 break
383
384             capture.release() #ปิดกล้อง
385             cv2.destroyAllWindows() #ปิดหน้าต่าง
386
387             #แสดงข้อมูลลูกค้าใน Textbox
388             self.txtUpdateID.setText(str(profile[0]))
389             self.txtUpdateName.setText(str(profile[1]))
390             self.txtUpdateMoney.setText(str(profile[3]))
391
392 @pyqtSlot()
393 def gotoMainWindow(self): #ฟังก์ชันปุ่มย้อนกลับไปหน้าหลักของโปรแกรม
394     widget.addWidget(MainWindow())
395     widget.setFixedHeight(720)
396     widget.setFixedWidth(1280)
397     widget.setCurrentIndex(widget.currentIndex()+1)
398
399 def selectAddMoney(self): #ฟังก์ชันปุ่ม Radio Button สำหรับเลือกจำนวนเงิน
400     if self.rBtn_20.isChecked():

```

```

401     n = 20
402     elif self.rBtn_50.isChecked():
403         n = 50
404     elif self.rBtn_100.isChecked():
405         n = 100
406     elif self.rBtn_200.isChecked():
407         n = 200
408
409     #อัพเดทจำนวนเงินที่เติมเข้าไปในฐานข้อมูล
410     cur.execute("UPDATE tbCustomer SET money = "
411                 + str(int(self.txtUpdateMoney.text()) + n)
412                 + " WHERE ID = "
413                 + str(self.txtUpdateID.text()))
414     cn.commit()
415     self.txtUpdateMoney.setText(str(int(self.txtUpdateMoney.text()) + n))
416     self.btnAddMoney.hide() #ซ่อนปุ่มเติมเงิน
417
418 app=QApplication(sys.argv) #เริ่มต้นสร้าง GUI
419 widget = QtWidgets.QStackedWidget() #สร้างหน้าต่าง GUI
420 widget.addWidget(MainWindow()) #เรียกหน้าต่างหลักขึ้นมาก่อนลำดับแรก
421 widget.setFixedHeight(800) #กำหนดความสูงหน้าจอ
422 widget.setFixedWidth(1280) #กำหนดความกว้างหน้าจอ
423 widget.show() #แสดงหน้าจอ
424 sys.exit(app.exec_())

```



วิดีโอการสอนเรื่อง

การเขียนโปรแกรมตู้จำหน่ายขนมอัตโนมัติจากบอร์ด Raspberry Pi



วิดีโอการสอนเรื่อง การเขียนโปรแกรมตู้จำหน่าย
ขนมอัตโนมัติจากบอร์ด Raspberry Pi

