GROUP ASSIGNMENT

TECNOLOGY PARK MALASIA

AAPP010-4-2-PWP

PROGRAMMING WITH PYTHON

UCDF2209ICT(SE)

OOI DUN TZI TP071308

LIEW JUN XUAN TP072101

TAN KAI BOON TP072047

CHIN BO ZE TP071139

LECTURER: SATHIAPRIYA A/P RAMIAH

Contents

1.0 Introduction and assumptions

Tron is a company that has been around since 2022. It is an enrolled laptop repair service provider and owns several branches in Kuala Lumpur. Therefore, they need a laptop repair service management system to stabilise their business. The laptop repair services management system contains four characteristics, and they are respectively “receptionist”, “technician”, “customer”, and “admin”. Each of these four characteristics has its own role, and it can’t have one without the other.

Role of receptionist:

Receptionists are the main support for the customer. The role of the receptionist is to help customers register their accounts, services and receive payments, but it can also be done by the customer itself by using the Tron application. Moreover, receptionists will base on “customer_information.txt” file to generate a receipt for customers who have paid.

Role of customer:

Customer is one of the main characteristics of the laptop repair services management system, which also provides the first instructions for the hole process when they register their account and services by themselves without passing through a receptionist by registering on the sign-in page. Because of that technician only can view the services that customers registered and fix the collection date based on the date registered by the customer. Moreover, customers can change their services, profile and view their services descriptions after technician has approved.

Role of technician:

Technician is the crucial role that service customer, first technician will view the requested service by the customer and satisfied their demands. After technician completed the requested service by the customer, technician will add the description and laptop collection date upon completion of service to inform the users. Moreover, technician will daily update the description in order for customers to get answers in the fastest way as possible. As a results, if customer change profile or services it will need to wait one working days for technician to update.

Role of admin:

Admin are the significant role of the systems, which managing information of the worker of the company. The worker needs to register themselves either receptionist or technician and the rest information to admin, and admin will store the information into worker_information.txt text file. Furthermore, admin can view the report of the company either the service report or the total income to make better decision. If the worker intends to change their password, they can only change through admin.

2.0 Design of the program

OOI DUN TZI (CUSTOMER)

```
#1
function menu():
  display"""
            1. Service type, Service fee
            2. Normal, Urgent
            3. Remove virus, malware or spyware, RM50.00, RM80.00
            4. Troubleshoot and fix computer running slow, RM60.00, RM90.00
            5. Laptop screen replacement, RM380.00, RM430.00
            6. Laptop battery replacement, RM180.00, RM210.00
            7. Operating System Format and Installation, RM100.00, RM150.00
            8. Data backup and recovery, RM80.00, RM130.00
          """
end function()
```

Figure1: Customer menu ()

```
#2
function description(username, password):
    openFile("description.txt", "read") as f2:
        loop each line in f2:
            split line into input_username, input_password, service_type, service_fee, service_fee_amount, date_collection using ","
            if input_username equals username and input_password equals password then:
                display "Username:",{username} \n"Service type:",{service_type} \n"Service fee:", {service_fee} \n"Service fee amount:", {service_fee_amount} \n"Date collection:", {date_collection}
                Exit loop
        else:
            display "Your description are still pending..."
end function()
```

Figure2: Customer function description ()

```
#3
function update_customer_data(username, password, new_customer, new_gender, new_city, new_postcode, new_telephone):
    openFile("customers_information", "read") as f:
        set lines as read all lines from file f
    set user_update as false
    Loop each line in lines, with index i:
        split line into current_username, current_password, service_type, service_fee, service_fee_amount, gender, city, postcode, telephone, day, month, year, paid_amount using ","
        if current_username equals username and current_password equals password then:
            set line at index i as the list called "liens" to the following strings:
            concatenate the values of "{new_username}, {new_password}, {service_fee}, {service_fee_amount}, {new_city}, {new_postcode}, {new_telephone}, {day}, {month}, {year}, {paid_amount}\n"
            set user_update as true
            break
            Exit loop
    openFile("customers_information", "write") as f:
        write the content from the list lines to the file f
    return to user_update
end function()
```

Figure3: Customer update_customer_data ()

```
#4
function update_customer_details():
    display "Enter your current username and password"
    input input_username as "Username:"
    input input_password as "Password:"
    input new_username as "Enter new username:"
    input new_password as "Enter new password:"
    while true:
        input new_gender as "male/female:"
        if new_gender equals "male", "female" then:
            break
        else:
         display "Invalid choice. Please enter 'male' or 'female'."
    input new_city as "Enter new city:"
    while true:
        input new_postcode as "Enter new postcode:"
        if new_postcode is digit then:
            break
        else:
            display "Invalid input. Please enter digits."
    while true:
        input new_telephone as "Enter new contact number:"
        if new_telephone is digit ans length of new_telephone equals 10 then:
            break
        else:
            display "Invalid input. Please enter a 10-digit number."
    set user_update as call update_customer_data(input_username, input_password, new_username, new_password, new_gender, new_city, new_postcode, new_telephone)
    if user_update then:
        display "Profile update successfully."
    else:
        display "User not found. Profile not updated."
end function()
```

Figure4: Customer update_customer_details ()

```
#5
function change_service(username):
    call menu()
    create dictionary services as '1': ("Remove virus malware or spyware", {"normal": 50.00, "urgent": 80.00}),
                                  '2': ("Troubleshoot and fix computer running slow", {"normal": 60.00, "urgent": 90.00}),
                                  '3': ("Laptop screen replacement", {"normal": 380.00, "urgent": 430.00}),
                                  '4': ("Laptop battery replacement", {"normal": 180.00, "urgent": 210.00}),
                                  '5': ("Operating System Format and Installation", {"normal": 100.00, "urgent": 150.00}),
                                  '6': ("Data backup and recovery", {"normal": 80.00, "urgent": 130.00}),
    while true:
        input service_choice as "Enter the service number (1-6):"
        input service_fee_type as "Enter service fee type (normal/urgent):"
        if service_choice equals service and service_fee_type equals "normal", "urgent" then:
            create list new_service_fee as services[service_choice][1][service_fee_type]
            create list new_service_fee_amount as float("{:.2f}".format(new_service_fee))
            while true:
                input day as "Day register change:"
                if day is digits and "1" <= day <= "31" then:
                    break
                else:
                    display "Invalid input. Please enter 1-30/31"
            while true:
                input month as "Month register change:"
                set valid_month as "january", "february", "march", "april", "may", "june", "july", "august", "september", "october", "november", "december"
                if month is equals valid_month then:
                    break
                else:
                    display "Invalid input. Please enter a valid month (e.g., january)."
            while true:
                input year as "Year register change:"
                if year is digit and length year equals 4 then:
                    break
                else:
                    display "Invalid input."
                set paid_amount as 0
                while paid_amount < new_service_fee_amount:
                    input paid_amount as float "Enter your payment amount: RM"
                    if paid_amount >= new_service_fee_amount then:
                        break
                    else:
                        display "Invalid choice. Please enter an amount equal to or greater than RM{:.2f}. format(new_service_fee_amount)"
                display "Payment accepted. Thank you!"
                openFile ("customers_information.txt","r") as f:
                    read lines from f and assign to lines
                openFile ("customer_information.txt", "write") as f:
                    loop each line in lines:
                        split line into user_data using ","
                        if user_data[0] equals username then:
                            create list line as "{username},{user_data[1]},{services[service_choice][0]},{service_fee_type},{new_service_fee_amount},{user_data[5]},{user_data[6]},{user_data[7]},{user_data[8]},{day},{month},{year},{paid_amount}\n"
                            Exit loop
                        write line to f
                display "Service for '{username}' changed successfully to '{services[service_choice][0]}' ({service_fee_type}): RM{new_service_fee_amount}"
                break
            else:
                display "Invalid service choice or service fee type. Please try again."
end function()
```

Figure5: Customer change_service ()

```
#7
function login_customer():
    set attempts as 0
    while attempts < 3:
        display "Log in"
        input input_username as "Username:"
        input input_password as "Password"
        set found as false
        openFile ("Customers_information.txt","read") as f:
        loop each line in f:
            split line into username, password, service_type, service_fee, service_fee_amount, gender, city, postcode,telephone_number, day, month, year, paid_amount using ","
            if input_username equals username and input_password equals password:
                set found as true
                display "You have successfully logged in"
                display "Your current requested services: {service_type}, {service_fee}, RM{service_fee_amount}"
                Exit loop
                while true:
                    display "1) Edit Service\n2) Edit Profile\n3) View Description"
                    input choose2 as "Choose your page (1/2/3):"
                    if choose2 equals "1":
                        call change_services(input_username)
                    elif if choose2 equals "2":
                        call update_customer_details()
                    elif if choose2 equals "3":
                        call description(input_username, input_password)
                    else:
                        display "Invalid answer"
                        continue
                    while true:
                        input continue_choice as "Do you want to continue? (yes/no):"
                        if continue_choice equals "yes" or continue_choice equals "no"
                            break
                        else:
                            display "Invalid choice. Please enter 'yes' or 'no'."
                    if continue_choice equals "no":
                        break
                break
        close file f
        if found:
            break
        else:
            display "Incorrect username or password, please try again"
            increment attempts by 1
    if attempts equals 3:
        display "Maximum login attempts reached. Exiting."
    else:
        display "Thanks for using our application"
end function()
```

Figure6: Customer login_customer ()

```
#6
function signin_customer():
    input username as "Enter your name:"
    input password as "Enter your password:"
    call menu()
    while true:
        input service_choice as "Enter the service number (1-6):"
        if service_choice equals "1", "2", "3", "4", "5", "6" then:
            break
        else:
            display "Invalid service choice. Please try again."
    while true:
        input service_fee as "Enter service fee type (normal/urgent):"
        if service_fee equals "normal" and "urgent" then:
            break
        else:
            display "Invalid service fee type. Please try again."
    create dictionary services as '1': ("Remove virus malware or spyware", {"normal": 50.00, "urgent": 80.00}),
                                  '2': ("Troubleshoot and fix computer running slow", {"normal": 60.00, "urgent": 90.00}),
                                  '3': ("Laptop screen replacement", {"normal": 380.00, "urgent": 430.00}),
                                  '4': ("Laptop battery replacement", {"normal": 180.00, "urgent": 210.00}),
                                  '5': ("Operating System Format and Installation", {"normal": 100.00, "urgent": 150.00}),
                                  '6': ("Data backup and recovery", {"normal": 80.00, "urgent": 130.00}),
    create list service_type as services[service_choice][0]
    create list service_fee_amount as services[service_choice][1][service_fee]
    while true:
        input gender as "male/female:"
        if gender equals "male" and "female" then:
            break
        else:
            display "Invalid choice. Please try again."
    input city as "Current city:"
    while true:
        input postcode as "Current postcode:"
        if postcode is digit then:
            break
        else:
            display "Invalid input. Please enter digits."
    while true:
        input telephone as "Contact number:"
        if telephone is digit and length telephone equals 10 then:
            break
        else:
            display "Invalid input. Please enter a 10-digit number."
    while True:
        input day as"Day register:"
        if day is digit and "1" <= day <= "31" then:
            break
        else:
            display "Invalid input. Please enter 1-30/31"
    while true:
        input month as "Month register:"
        set valid_month as "january", "february", "march", "april", "may", "june", "july", "august", "september", "october", "november", "december"
        if month is equals valid_month then:
            break
        else:
            display "Invalid input. Please enter a valid month (e.g., january)."
    while true:
        input year as "Year register:"
        if year is digit and length year equals 4 then:
            break
        else:
            display "Invalid input."
        set paid_amount as 0
        while paid_amount < new_service_fee_amount:
            input paid_amount as float "Enter your payment amount: RM"
            if paid_amount >= new_service_fee_amount then:
                break
            else:
                display "Invalid choice. Please enter an amount equal to or greater than RM{:.2f}. format(service_fee_amount)"
        display "Payment accepted. Thank you!"
        set f to open file ("customers_information.txt", "append")
        write {username},{password},{service_type},{service_fee},{service_fee_amount},{gender},{city},{postcode},{telephone},{day},{month},{year},{paid_amount}\n to f
        close file f
        display "Record added. Thank you!"
end function()
```

Figure7: Customer signin_customer ()

```
#8
function mainpage():
    display "Mian Page"
    set choose as "1" and "2"
    display "1) Login \n2) Sign in"
    while choose not equals "1" and "2":
        input choose as "Login / Sign in (Enter Number):"
        if choose equals "1":
            call login_customer()
        elif choose equals "2":
            signin_customer()
            set log as "yes" and "no"
            while log not equals "yes" and "no":
                input log as "Do you want to login?(yes/no):"
                if log equals "yes":
                    call login_customer()
                elif log equals "no":
                    display "Thanks for the registration."
                    break
                else:
                    display "Invalid answer"
            break
        else:
            display "Invalid number!!"
end function()
```

Figure8: Customer mainpage ()

LIEW JUN XUAN (RECEPTIONIST)

```
Function receptionist(username,account,password)
    Dowhile True
        Display menu
        Prompt user for selection
        Read selectioR
        Dowhile selection is equal to 4
            End entire program
            Display "Logout Success"
        Dowhile selection is equal to 1
            Display instruction
            Prompt user for cusername,
            Read cusername
            If cusername is equal to q then
                Enddo
            EndIf
            Prompt user for cpassword
            Read cpassword
            Call menu()
            Dowhile True
                Prompt user for service_choice
                Read service_choice
                If service_choice is equal to number then
                    If service_choice is equal to 1,2,3,4,5,6 then
                        Enddo
                    Else
                        Display instruction
                        Continue
                    EndIf
                Else
                    Display warning
                    Continue
                EndIf
            Dowhile True
                Prompt user for service_fee
                Read service_fee
                If service_fee is not equal to Urgent or Normal then
                    Display reminder
                    Continue
                Else
                    Enddo
                Endif
            Prompt user for service_type,service_fee_amount
            Read service_type,service_fee_amount
            Dowhile True
                Prompt user for gender
                Read gender
                If gender is not equal to female or male then
                    Display instruction
                    Continue
                else
                    Enddo
                EndIf
            Dowhile True
                Prompt user for day,month,year
                Read day,month,year
                If month is not written in full name then
                    Display instruction
                    Continue
                Else
                    Enddo
                EndIf
            Dowhile True
                Prompt user for paid_amount
                Read paid_amount
                If paid_amount is lesser than service_fee_amount then
                    continue
                Else
                    Enddo
                Endif
            open "customer_information.txt" for append as file1
            Write customer information in file1
```

Figure9: Receptionist function part1

```
            Close file1
            continue
        Dowhile selection is equal to 3
            display menu
            Prompt user for choice
            Read choice
            If choice is equal to 3 then
                Enddo
            Elif choice is equal to 1 then
                Display instruction
                Dowhile True
                    Prompt user for new_username,new_password
                    Read new_username,new_password
                    If new_username is equal to q then
                        Enddo
                    Endif
                    Dowhile True
                        Prompt user for gender
                        Read gender
                        If gender is not equal to female or male then
                            Display instruction
                            Continue
                        Else
                            Enddo
                        Endif
                    Prompt user for city,postcode
                    Read city,postcode
                    Dowhile True
                        Prompt user for telephone
                        Read telephone
                        If telephone is equal to 10-digit number then
                            Enddo
                        Else
                            Display instruction
                            continue
                        Endif
                    Open "workers_information.txt" for read as file2
                    Loop file2 to read line by line
                        If account,username,password in file2 then
                            Open "workers_information.txt" for write
                            Replace the respective line with updated information
                            Close "workers_information.txt"
                            Display "Update Profile Successfully"
                            Enddo
                        EndIf
                    EndLoop
            Elif choice is equal to 2 then
                Open "workers_information.txt" for read
                Loop "workers_information.txt" and split with ","
                    If username is in line
                        If length of line is equal or more than 7 then
                            Display account,username,password,gender,city,postcode,telephone
                            Enddo
                        EndIf
                    EndIf
                EndLoop
            else:
                Display warning
                continue
            EndIf

        Dowhile selection is equal to 2
            Display customer list
            Prompt user for action
            Read action
            If action is equal to back to menu then
                Enddo
            EndIf
            If action is equal to customer number then
                Generate and Display receipt for the chosen customer
            Else
                Display "Invalid Number"
                continue
            Endif
    EndFunction
```

Figure10: Receptionist function part 2

```
Function login():
    Count is equal to 0
    Dowhile count is smaller than 3
        Prompt user for account,username,password
        Read account,username,password
        Open"workers_information.txt" for read
        Loop "workers_information.txt" to search line by line
            If account,username,password is equal to information in "workers_information.txt"
                If account is equal to receptionist then
                    Call receptionist(account,username,password)
                    Return True
                EndIf
                If account is equal to technician(account,username,password) then
                    Call technician(account,username,password)
                    Return True
                EndIf
                If account is equal to admin then
                    Call admin()
                    Return True
                EndIf
            EndIf
        EndLoop
        Count equal to count + 1
        If count is lesser than 3 then
            Display Attempt Remaining
        EndIf
        If count is equal to 0 then
            Display Warning
            Enddo
        EndIf
EndFunction
```

Figure11: Login ()

```
Start
    Dowhile True:
        Display Store name
        Prompt user for type
        Read type
        If type is equal to customer then
            Call mainpage()
            Enddo
        Elif type is not equal to staff or customer
            Continue
        Elif type is equal to staff then
            Call login()
            Enddo
        EndIf
End
```

Figure12: Execute Code

TAN KAI BOON (ADMIN)

```
START
FUNCTION StaffLogin()
    set count = 0
    prompt user for Username
    prompt user for Password
    set count = count + 1
    open file "workers_information.txt" in read mode as file
        set lines as read all line from file file
        loop each line in lines
            split line into data
            set stored_username as list data len 0
            set stored_password as list data len 1
            if stored_username equal username and stored_password equal password
                display "Login Success"
                Exit Loop
                return Username
                return
        display "Login Failed"
END Function
END
```

Figure13: StaffLogin()

```
START
FUNCTION ChangePassword(Usename)
    prompt user for NewPass
    open file "workers_information.txt" in read mode as file
        set lines as read all line from file file
    loop each lines according to the range and length, with index i
        set data as the lines that without spacing and split with ","
        set stored_username as the data at index 1
        IF stored_username equal to Username
            set index 2 in the data equal to NewPass
            Set lines [i] as join data into a string with commas and a newline character
            Open file "workers_information.txt" in write mode
                write lines to the file
            Display "Password updated successfully."
            return
        ENDIF
    Exit loop
    Display "Username not found. Password not updated."
END FUNCTION
END
```

Figure14: ChangePassword(Username)

```
START
FUNCTION TotalSalesByMonth():
    prompt user for month
    set total = 0
    Open file "customers_information.txt" in read mode as file
        loop each line in file
            set values as the lines that without space and split with ","
            IF the length of the values equal to 13
                set service_month as index 10 of values and remove spacing
                IF service_month convert to lowercase and month convert to lowercase are same
                    set income as convert index 4 of values to float
                    set total = total + income
                ENDIF
            ENDIF
        Exit loop
        return total
END FUNCTION
END
```

Figure15: TotalSalesByMonth()

```
START
    FUNCTION MonthlyReport()
        prompt user for month
        create service_data dictionary
        create summary dictionary
        Open file "customers_information.txt" in read mode as file
            loop each line in file
                set values as the lines that without space and split with ","
                IF the length of values greater or equal to 12
                    set service_type to the length 2 element in the values and remove spacing
                    set service_month to the length 10 element in the values and remove spacing
                    IF lowercase of service_month equal to lowercase of month
                        set name to length 0 of values
                        set day to length 9 of values
                        set year to length 11 of values
                        set entry as a formatted string with name, day, service_month and year
                        IF service_type is in service_data:
                            append entry to service_data[service_type]
                        ELSE
                            create a new list containing entry and assign it to
                            service_data[service_type]
                        IF service_type is in summary
                            increase summary[service_type] by 1
                        ELSE
                            set summary[service_type] to 1
                        ENDIF
                    ENDIF
                ENDIF
            Exit loop
        for each service, data in service_data
            IF data is not empty
                display "Service" + service
                for each entry in data
                    display entry
                Exit loop
                display spacing to add an empty line
            ENDIF
        Exit loop
        Display"----------Summary----------"
        for each service, count in summary
            display service + "=" + count
        Exit loop
    END FUNCTION
    END
```

Figure16: MonthlyReport()

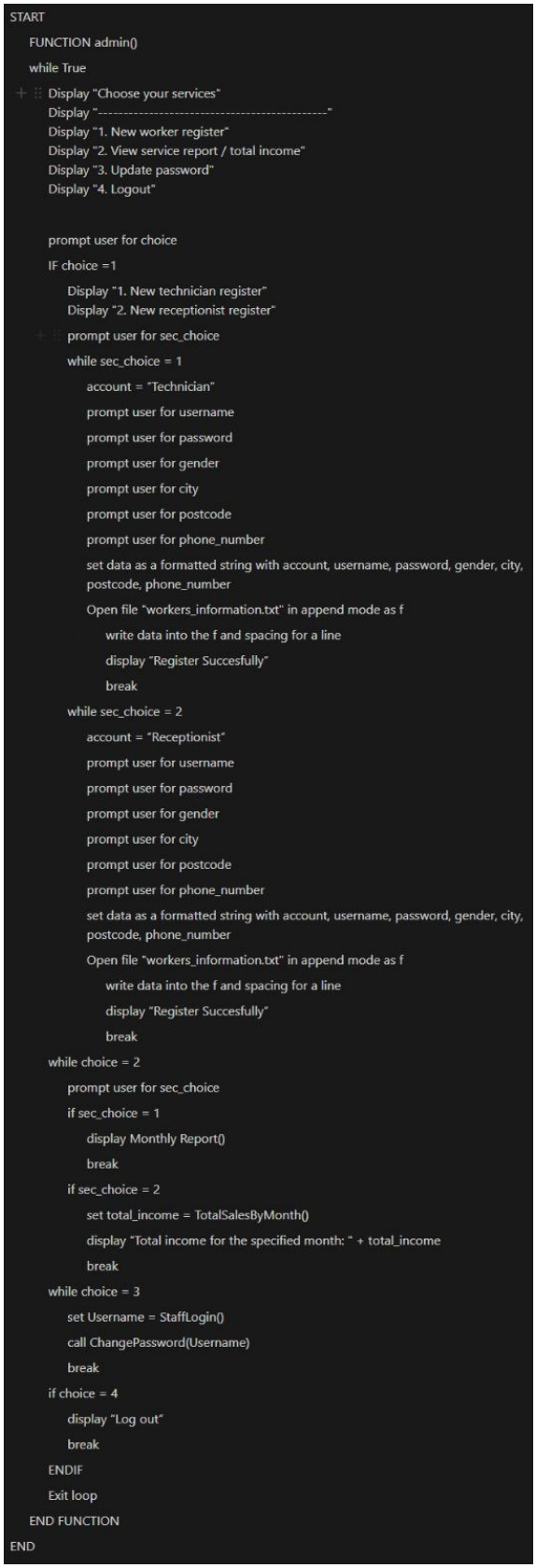


Figure17: admin()

CHIN BO ZE (TECHNICIAN)

```
Function update_workers_data(username, password, new_username, new_password, new_gender, new_city, new_postcode, new_telephone):
    Open "workers_information.txt" for reading and set as "f"
      Read lines from file  f and store them in the variable lines
      Close the file

    Set user_updated to False

    For each line in the list 'lines' along with its index 'i'
        Split the string 'line' by the comma (",") separator into account, current_username, current_password, gender, city, postcode, and telephone

        IF current_username == username AND current_password == password THEN
            Update the lines with the new information
            Set user_updated to True
            Break
        ENDIF

    Open "workers_information.txt" for writing and set as "f"
        Write the list of lines to the file "f"

    return user_updated
EndFunction
```

Figure18: Technician update_worker_data ()

```
Function update_workers_detail()
    Display "Enter your current username and password"
    Prompt user for input_username
    Prompt user for input_password
    Prompt user for new_username
    Prompt user for new_password
    Read input_username,input_password,new_username and new_password

    DOWHILE True
        Prompt user for new_gender
        Read new_gender
        IF new_gender "male" OR "female" THEN
            Break
        ELSE
            Display  "Invalid choice. Please enter 'male' or 'female'."
        ENDIF
    Prompt user for new_city
    ENDDO

    DOWHILE True
        Prompt user for new_postcode
        Read new_postcode
        IF new_postcode is digit THEN
            Break
        ELSE
           Display "Invalid input. Please enter digits."
        ENDIF
    ENDDO

    DOWHILE True
        Prompt user for new_telephone
        Read new_telephone
        IF new_telephone is digit AND 10 digit number THEN
            Break
        ELSE
            Display "Invalid input. Please enter a 10-digit number."
        ENDIF
    ENDDO

      IF user_updated THEN
          Display "Profile updated successfully."
      ELSE
          Display "User not found. Profile not updated."
      ENDIF
EndFunction
```

Figure19: Technician update_worker_detail()

```
Function read_customer_details()
    Open "customers_information.txt" for reading and set as "f1"
        For each line in the file 'f1' along with its line number and starting from 1
            Split the string 'line' by the comma (",") separator into  customer information
            Display "Line:",line_number
            Display customer information
EndFunction
```

Figure20: Tchnician read_customer_detail()

```
Function add_description(username, collection_date)
    Open "customers_information.txt" for reading and set as "f1"
        Set found to False
        for each line in the file "f1"
            Split the string 'line' by the comma (",") separator into customer information
            IF current_username == username THEN
                Set found to True
                Open "customers_information.txt" for appending and set as "f2"
                    write a new line with customer description
                Display "Record added. Thank you!"
                Break

        IF found is False THEN
            Display "Username not found in customer information. Record not added."
EndFunction
```

Figure21: Technician add_description()

```
Function technician(username,password)
    DOWHILE True
        Display menu
        Prompt user for choose2
        Read choose2
        IF choose2 == "1" THEN
            Display update_workers_details()
        ELIF choose2 == "2" THEN
            Display read_customer_details()
            Prompt user for input_username
            Prompt user for input_collection_date
            Read input_username and input_collection_date
            add_description(input_username, input_collection_date)
        ELSE
            Display "Invalid answer"
            Continue

        DOWHILE True
          Prompt user for continue_choice
          Read continue_choice
          IF continue_choice "yes" OR "no" THEN
            Break
          ELSE
            Display "Invalid choice. Please enter 'yes' or 'no'."
        ENDDO

        IF continue_choice == "no" THEN
            Break
    ENDDO
EndFunction
```

Figure22: Technician ()

3.0 Explanation of programming concepts

OOI DUN TZI (CUSTOMER)

1. Functions:

```
def mainpage():

    print("Main Page")
    choose = "1" and "2"
    print("1) Login\n2) Sign in")
    while choose != "1" and "2":
        choose = input("Login / Sign in (Enter Number):")
        if choose == "1":
            login_customer()
        elif choose == "2":
                signin_customer()
                log = "yes" and "no"
                while log != "yes" and "no":
                    log = input("Do you want to login?(yes/no):") .lower()
                    if log == "yes":
                        login_customer()
                    elif log == "no":
                        print("Thanks for the registration.")
                        break
                    else:
                        print("Invalid answer")
                break
        else:
            print("Invalid number!!")
```

Figure23: Customer function programming concept

The role of the function “mainpage()” is to sum up all the customer processes and other functions used in the customer process. For example, the functions that call in “mainpage()” are “signin_customer()” and “login_customer()”.

2. String Formatting:

```
print("{:^30}{:^100}".format( *args: "Service type", "Service fee"))
print("{:>67}{:>35}".format( *args: "Normal", "Urgent"))
print("{:<15}{:^62}{:^8}".format( *args: "1.Remove virus, malware or spyware", "RM50.00", "RM80.00"))
print("{:<15}{:^42}{:^28}".format( *args: "2.Troubleshoot and fix computer running slow", "RM60.00", "RM90.00"))
print("{:<15}{:>42}{:>35}".format( *args: "3.Laptop screen replacement", "RM380.00", "RM430.00"))
print("{:<15}{:>41}{:>35}".format( *args: "4.Laptop battery replacement", "RM180.00", "RM210.00"))
print("{:<15}{:>27}{:>35}".format( *args: "5.Operating System Format and Installation", "RM100.00", "RM150.00"))
print("{:<15}{:>42}{:>36}".format( *args: "6.Data backup and recovery", "RM80.00", "RM130.00"))
```

Figure24: Customer string formatting programming concept

In the function "menu(),” there are 8 strings using the format() method for string formatting. For example, the first string coded {:^30}{:^100} is to find the midpoint within a width of 30 and 100 characters and to display “Service type” and “Service fee”.

3. Loops and Conditional Statements:

```
while True:
    new_gender = input("male/female: ").lower()
    if new_gender in ("male", "female"):
        break
    else:
        print("Invalid choice. Please enter 'male' or 'female'.")
```

Figure25: Customer loops and conditional statement programming

In his loops condition, customers will be able to only choose “male” and “female” as their gender. If the customer chooses other than “male” or "female,” it will display “Invalid choice. Please enter 'male' or 'female'" and keep repeating the loop.

4. Lists and Dictionaries:

```
services = {
    '1': ("Remove virus malware or spyware", {"normal": 50.00, "urgent": 80.00}),
    '2': ("Troubleshoot and fix computer running slow", {"normal": 60.00, "urgent": 90.00}),
    '3': ("Laptop screen replacement", {"normal": 380.00, "urgent": 430.00}),
    '4': ("Laptop battery replacement", {"normal": 180.00, "urgent": 210.00}),
    '5': ("Operating System Format and Installation", {"normal": 100.00, "urgent": 150.00}),
    '6': ("Data backup and recovery", {"normal": 80.00, "urgent": 130.00}),
}
```

Figure26: Customer lists and dictionaries programming

“services” is a dictionary that uses the numerical keys ‘1’ to ‘6’ to represent the six types of services. Each numerical key contains a service type and an inner dictionary of ‘normal’ and ‘urgent’ to corresponding service fees. For example, if customers choose ‘1’ and "normal” it will store the information in “Remove virus malware or spyware” and 50.00.

5. Input and Output:

```
def signin_customer():
    username = input("Enter your name: ")
    password = input("Enter your password: ")
    ...
    print("Record added. Thank you!")
```

Figure27: Customer input and output programming concept

The role of “input” in the function “signin_customer()” is to request that customers enter their information. Moreover, when the customer enters all the information required in the function "signin_customer(),” it will display “Record added. Thank you!” as the output after the process has finished.

6. Error Handling:

```
def login_customer():
    attempts = 0
    while attempts < 3:
        ...
            print("Incorrect username or password, please try again")
            attempts += 1
    if attempts == 3:
        print("Maximum login attempts reached. Exiting.")
    else:
        print("Thanks for using our application")
```

Figure28: Customer error handling programming concept

In the function "login_customer(),” it only allows the user to have three chances to attempt the username and password. Therefore, if customers input the wrong username or password, it will be considered an invalid answer, and after three chances have passed, it will stop the loop and display “Maximum login attempts reached. Exiting.”

7. Data Types:

```
    paid_amount = float(input("Enter your payment amount: RM"))
```

Figure29: Customer data types programming concept

In this input, it’s using a float as the data type. Therefore, “paid_amount” only includes numerical information and those real numbers that have decimal points and those don't.

8. Control Flow:

```
    if choose == "1":
        login_customer()
    elif choose == "2":
            signin_customer()
```

Figure30: Customer control flow programming concept

The role of “if” and “elif” is to control the flow of the process. For example, customers will be able to choose “1” and “2” as the pages they want to go to, and if they choose “1” it will call “login_customer()” and if they choose “2” it will call “signin_customer()”.

9. Function Calls:

```
                elif choose2 == "2":
                    update_customer_details()
```

Figure31: Customer function calls programming concept

In this section, when customers input “2” in the index "choose2,” it will call “update_customer_details()” to continue the process in the function "update_customer_details(),” and this will be the role of function calls.

10. File Input/Output:
Read file:

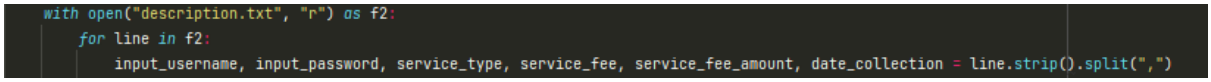


Figure32: Customer file input/output programming concept in read file

The role of the read file in this section is to open the “description.txt” file as f2 and read each line in f2.

Write file:

Figure33: Customer file input/output programming concept in write file

The role of the write file in this section is to open the “customers_information.txt” file as f and write the content from the list lines to the file f.

Append file:

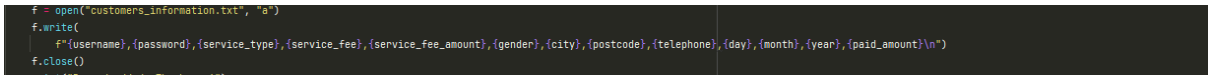


Figure34: Customer file input/output programming concept in append

The role of the append file in this section is to create a “customer_information.txt” file and allow customers to input their information in it.

11. Break and Continue Statements:

Continue Statements:

Figure35: Customer file break and continue statement programming concept in continue statement

In this section the role of continue statements is to skip the remaining code when customer input the wrong instruction and reprocess the loop.

Break Statements:

Figure36: Customer file break and continue statement programming concept in break statement

In this section the role of break statement is to terminate the loop when customers input the correct instruction.

LIEW JUN XUAN (RECEPTIONIST)

1. Define()

```
def receptionist(username, account, password):
```

Figure37: Receptionist define function

It is a function that I have defined in this assignment to handle the responsibilities of a receptionist, such as updating profiles, registering customers, and more.

2. While Loop

```
    while True:
        profile_update = False
        print("""            Menu
++++++++++++++++++++++++++++++++++++++++++++
1. Register Customer and Assign Service
2. Accept Payment and Generate Receipt
3. Update Profile
4. Exit
++++++++++++++++++++++++++++++++++++++++++++""")
        selection = input("Selection:")
```

Figure38: Receptionist while loop

A while loop is used to allow the user to navigate through various selections until specific conditions are detected, at which point the loop stops.

3. Print()

```
        print("""            Menu
++++++++++++++++++++++++++++++++++++++++++++
1. Register Customer and Assign Service
2. Accept Payment and Generate Receipt
3. Update Profile
4. Exit
++++++++++++++++++++++++++++++++++++++++++++""")
```

Figure39: Receptionist print

Print is frequently used to display essential information.

4. Input()

```
selection = input("Selection:")
if selection == "4":
    exit("Logout Success")
```

Figure40: Receptionist input

Input is used to store a string in variable.

5. For Loop

```
for stafflog in open("workers_information.txt", "r").readlines():
```

Figure41: Receptionist for loop

A for loop is used to iterate over sequences or collections

6. Split()

```
    line = line.split(",")
```

Figure42: Receptionist split

The split function is used to divide a string into a list of substrings.

7. If and Nested If statement

```
if service_choice.isdigit():
    if service_choice in ("1","2","3","4","5","6"):
        break
    else:
        print("Enter service number in 1-6")
        continue
else:
    print("Enter service number in digits")
    continue
```

Figure43: Receptionist if and nested if

An if statement is used to execute specific code based on a given condition. When one if statement is nested inside another, it allows for the creation of more complex conditional logic.

8. Enumerate

```
newfile = account + "," + new_username + "," + new_password + "," + gender + "," + city + "," + postcode + "," + telephone
for a,text in enumerate(file2):
    index = text.split(",")

    if account == index[0] and username == index[1] and password == index[2]:
        file2[a] = newfile + "\n"
        profile_update = True
        break
```

Figure44: Receptionist enumerate

Enumerate is used to iterate over elements in an iterable while keeping track of the element's index or position within the iterable.

9. Text file Processing

```
with open("workers_information.txt", "r") as file:
    file1 = open("customers_information.txt", 'a+')
```

Figure45: Receptionist Text file Processing

In general, there are three ways to open a file: 'r' for reading, 'a' for appending, and 'w' for writing. 'Read' is used to access the content in a text file without making changes, while 'append' is used to add data to the end of the file, and 'write' is used to add data to a file. The key difference is that 'write' will create a text file if it doesn't exist, but if it does exist, the content will be overwritten.

TAN KAI BOON (ADMIN)

1. Define ()

```
def TotalSalesByMonth():
    month = input("Enter the month (e.g., February) to calculate total income for: ")
    total = 0
    with open("customers_information.txt", "r") as file:
        for line in file:
            values = line.strip().split(',')
            if len(values) == 13:
                service_month = values[10].strip()
                if service_month.lower() == month.lower():
                    income = float(values[4])
                    total += income
    return total
```

Figure46: Define Function

The Define function is use for creating a reusable block of code that perform specific tasks, when need to calculate the total sales of the month can just using this function. This function is based on the month that user provide and grab the income of the specific month and sum it up to display to user.

2. While Loop

```
    while True:
        print('''          Choose your services
-----------------------------------------------
1. New worker register
2. View service report / total income
3. Update password
4. Logout''')
```

Figure47: While Loop

For this while loop function is for create an interactive menu for users, when the last program result is True, it will display the menu to the user. While loop is using for the program to keep running, presenting the menu repeatedly, until the user chooses to log out or exit the program.

3. Print ()

```
print("Register Succesfully")
```

Figure48: Print Function

Print function is basically just display out the element to the user, for instance, the program will print “Register Successfully”.

4. Input ()

```
username = input("Enter Username: ")
```

Figure49: Input Function

Input function is prompt the information that program need from the user to do the subsequent operation. For example, the input function of figure is to get the user username.

5. For Loop

```
for line in file:
```

Figure50: For Loop

The loop ensures that the program processes each line of data systematically until all lines have been examined.

6. Strip () & Split ()

```
values = line.strip().split(',')
```

Figure51: Strip & Split Function

When reading a file, there is plenty of information inside a single line, the Strip function and help us to remove the spacing between the string and the Split function can divide the string into smaller, more manageable.

7. File

Figure52: File

There is many of situation that the code needs to interact with the file, when open file, can either choosing to read, write or append command to perform the corresponding operation on the file.

8. Dictionary

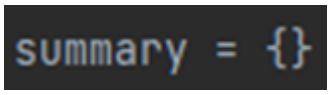


Figure53: Dictionary

Dictionary are powerful data structure for efficiently organizing and accessing data in Python, the ‘summary’ dictionary is used to keep track of the count of different service types.

CHIN BO ZE (TECHNICIAN)

1. Print ()

```
print("Profile updated successfully.")
```

Figure54: Receptionist print

The “print ()” function can print the specific message on the screen. For example, the “print (“Profile updated successfully.”)” statement can tell the workers the user profile is already updated successfully (Figure 54).

2. Input ()

```
input_collection_date = input("Enter the collection date: ")
```

Figure55: Receptionist input

The code uses the “input ()” function to enter various information. Such as “input (“Enter the collection date)” this statement can let the technician to enter collection time into the user’s information (Figure 55).

3. Repetition

```
while True:
    continue_choice = input("Do you want to continue? (yes/no): ").lower()
    if continue_choice in ["yes", "no"]:
        break
    else:
        print("Invalid choice. Please enter 'yes' or 'no'.")
```

Figure56: Receptionist while loop

The code uses “while” to specify a group of statements that are carried out repeatedly until a condition is satisfied. For example, in Figure 56, when the user enters either "yes" or "no," it will stop looping. If the user enters another word, it will result in the loop repeating the prompt until a valid choice is given.

4. File processing

```
with open("workers_information.txt", "w") as f:
```

Figure57: Receptionist Text file

The code uses file processing operations to read, write and create data for files. It uses the “open ()” function to open files and “w” means open a file for writing if a file exists. In Figure 57, it is shown that will open the “workers_information.txt” file and write the new information into the file and the file will use the “f” variable to write data to the "workers_information.txt" file.

5. Function

```
def update_workers_details():
```

Figure58: Function

A function in Python is some reusable code that accepts parameters as input, does computations, and then returns one or more results. So, we will use “def” to define a function. Invoking a function involves utilising its name, parentheses, and parameters in an expression. Figure 58 shows that a new function called update_workers_details is being defined. This function takes no parameters because there is no information in the brackets ().

6. Selection

```
if new_gender in ("male", "female"):
    break
else:
    print("Invalid choice. Please enter 'male' or 'female'.")
```

Figure59: Simple Selection

In the code “if” and “else” allow you to execute different blocks of code depending on whether a specific condition is true or false, and then choose between two alternative paths. For example, in Figure 59 if the user enter male or female then will break. When the user enters something other than male and female he will print “Invalid choice. Please enter ‘male’ or ‘female’.”

4.0 Screenshots of sample input/output

OOI DUN TZI (CUSTOMER)

Sign in Page:

```
Main Page
1) Login
2) Sign in
Login / Sign in (Enter Number):3
Invalid number!!
Login / Sign in (Enter Number):2
Enter your name: ooi
Enter your password: 123
        Service type                                    Service fee
                                                Normal                  Urgent
1.Remove virus, malware or spyware              RM50.00                 RM80.00
2.Troubleshoot and fix computer running slow    RM60.00                 RM90.00
3.Laptop screen replacement                     RM380.00                RM430.00
4.Laptop battery replacement                    RM180.00                RM210.00
5.Operating System Format and Installation      RM100.00                RM150.00
6.Data backup and recovery                      RM80.00                 RM130.00
Enter the service number (1-6): 7
Invalid service choice. Please try again.
Enter the service number (1-6): 1
Enter service fee type (normal/urgent): a
Invalid service fee type. Please try again.
Enter service fee type (normal/urgent): normal
male/female:a
Invalid choice. Please try again.
male/female:male
Current city:KL
Current postcode:57000
Contact number:1
Invalid input. Please enter a 10-digit number.
Contact number:0123456789
Day register:32
Invalid input. Please enter 1-30/31
Day register:14
Month register: 10
Invalid input. Please enter a valid month (e.g., january).
Month register: october
Year register:204
Invalid input.
Year register:2023
Enter your payment amount: RM40
Invalid choice. Please enter an amount equal to or greater than RM50.00
Enter your payment amount: RM50
Payment accepted. Thank you!
Record added. Thank you!
Do you want to login?(yes/no):yes
Log in
Username: ooi
Password: 123
You have successfully logged in
```

Figure60: Customer sign in page

Starting from the main page, customers will be required to choose “Login” or “Sign in” by entering “1” or “2” and in that section, customers will not be able to input anything other than “1” or “2”. If so, it will request that customers input again. When the customer inputs "2”, it will process the “sign in” page and require the customer to input their information. Some of the information is required to meet certain requirements. For example, “Enter the service number (1-6):” is only available input 1–6, “Enter service fee type (normal/urgent):” is only available input normal and urgent, “male/female” is only available input male and female, “Contact number:” is only available as a 10-digit number, “Day register:” is only available to input 1–31; “Month register:” is only available to input months written on letters, “Year register:” is only available to input 4-digit numerical; “Enter your payment amount:” is only available for input amounts that equal or are greater than the price. After the customer has input all the information required, it will continue to ask, “Do you want to login? (yes/no)” If yes, it will proceed to the login page.

Login page:

```
Main Page
1) Login
2) Sign in
Login / Sign in (Enter Number):1
Log in
Username: a
Password: a
Incorrect username or password, please try again
Log in
Username: a
Password: a
Incorrect username or password, please try again
Log in
Username: a
Password: a
Incorrect username or password, please try again
Maximum login attempts reached. Exiting.
```

Figure61: Customer fails login page

When customer attempts to login over than three time fails it will stop the process and display "Maximum login attempts reached. Exiting."

```
Main Page
1) Login
2) Sign in
Login / Sign in (Enter Number):1
Log in
Username: ooi
Password: 123
You have successfully logged in
Your current requested services: Remove virus malware or spyware, normal, RM50.0
```

Figure62: Customer successfully login page

If customer successfully login and it will display "You have successfully logged in" and display the service request that have been registered in sign in page or registered by receptionist.

Edit Service page:

```
1) Edit Service
2) Edit Profile
3) View Description
Choose your page (1/2/3):4
Invalid answer
1) Edit Service
2) Edit Profile
3) View Description
Choose your page (1/2/3):1
        Service type                                    Service fee
                                                Normal                  Urgent
1.Remove virus, malware or spyware              RM50.00                 RM80.00
2.Troubleshoot and fix computer running slow    RM60.00                 RM90.00
3.Laptop screen replacement                     RM380.00                RM430.00
4.Laptop battery replacement                    RM180.00                RM210.00
5.Operating System Format and Installation      RM100.00                RM150.00
6.Data backup and recovery                      RM80.00                 RM130.00
Enter the service number (1-6): 1
Enter service fee type (normal/urgent): normal
Day register change:14
Month register change: october
Year register change:2024
Enter your payment amount: RM50
Payment accepted. Thank you!
Service for 'ooi' changed successfully to 'Remove virus malware or spyware' (normal): RM50.0
```

Figure63: Customer edit service page

Customer will be able to change the service in this page and some of the information is required to meet certain requirements as in sign in page. Also, customers require to input the latest date that changed the service. After all the process it will display the latest service details that customer choose.

Edit Profile page:

Figure64: Customer successfully edit profile page

In edit profile user can only edit the “username”, “password”, “gender”, “city”, “postcode”, “contact number”. After everything it will display “Profile updated successfully”

Figure65: Customer fails view description page

If customers input the wrong username or password, it will not be able to update profile shown in the image above.

View Description page:

Figure66: Customer fails edit profile page

In this section when technician haven’t updated the “description.txt” file customer will not be able to view their description and will display “Your description are still pending…”.

Figure67: Customer successfully view description page

If customers successfully view their description it will display “Service type”, “Service fee”, “Service fee amount”, and “Service fee amount”.

Figure68: Customer confirmation to next progress

Finally, every time customers completed the page, they want to view it will continue ask “Do you want to continue? (yes/no)”. If yes it will continue to the process of choosing “Edit service”, “Edit profile”, and “View description”. If no it will finish the whole process.

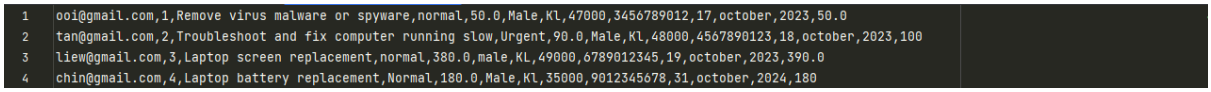


Figure69: Customer txt file

When every time customers register or receptionist register the new customer information it will auto save the data to “customers_information.txt” file. Moreover, when customer change their services or profile it will only change the portion of the information that they want to change.

LIEW JUN XUAN (RECEPTIONIST)

Login page

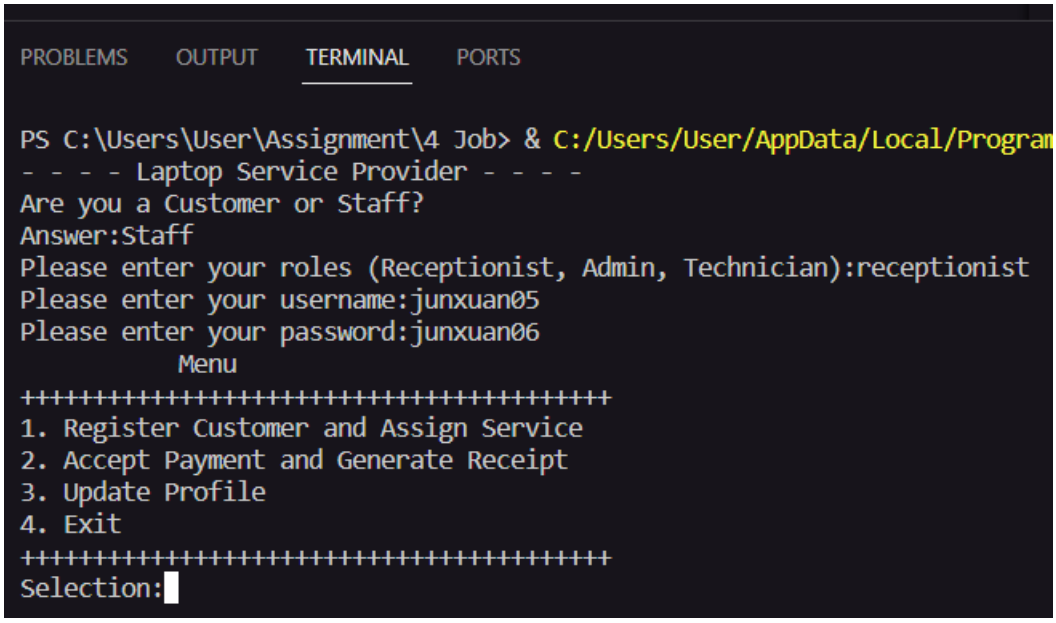


Figure70: Receptionist Login

When the program is executed, it will prompt the user to specify whether they want to log in as a customer or staff member. Upon choosing their account type, the user will be asked to input their username and password, which will be validated against the stored information in either the 'customer' or 'workers' text file. In the case of staff members, the user will also need to specify their role, such as 'receptionist' or 'admin. technician.

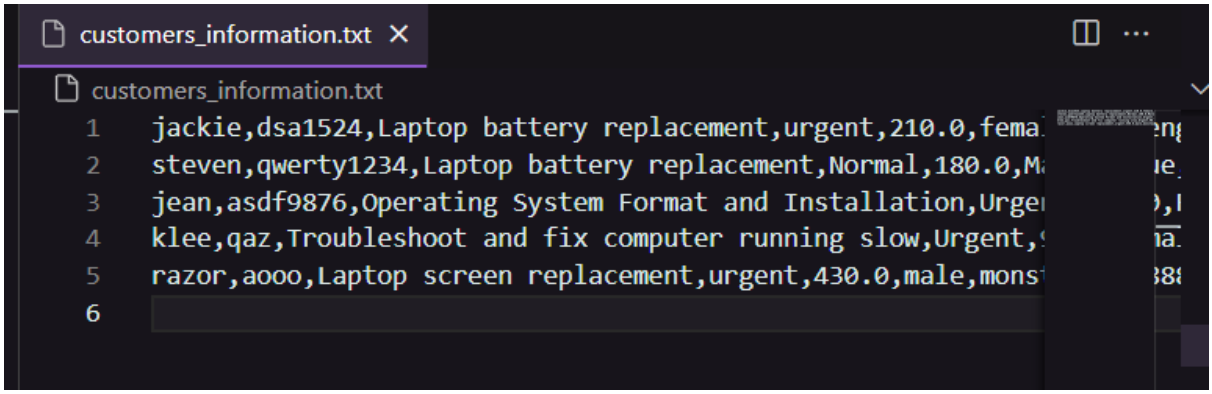
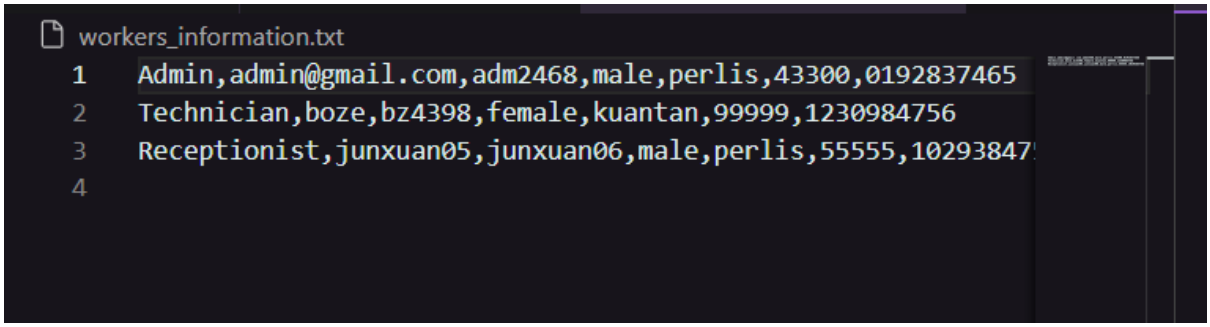


Figure 71: Receptionist default customer file



Figur72: Receptionist default worker file

In the case of the receptionist's responsibilities, after a successful login with the correct username and password, the program will display a menu with four options: registering customers and assigning services, accepting payments and generating receipts, updating profiles, or logging out. The receptionist can then select the specific task they want to perform.

Register Customer and Assign Service

```
Selection:1
Fill in Customer information, or type q to end
Register username:albert
Register password:albert1234
         Service type                                        Service fee
                                                    Normal                  Urgent
1.Remove virus, malware or spyware                  RM50.00                 RM80.00
2.Troubleshoot and fix computer running slow        RM60.00                 RM90.00
3.Laptop screen replacement                         RM380.00                RM430.00
4.Laptop battery replacement                        RM180.00                RM210.00
5.Operating System Format and Installation          RM100.00                RM150.00
6.Data backup and recovery                          RM80.00                 RM130.00
Register service (1-6)          :5
Urgent/Normal                   :urgent
Gender                          :male
City                            :perlis
Postcode                        :43300
Telephone                       :0123456789
Day                             :22
Month  (e.g.:January)           :january
Year                            :2023
Amount paided by customer       :900
Fill in Customer information, or type q to end
Register username:q
```

Figure 73: Receptionist Register Customer and Assign Service

In Figure 73, the receptionist's objective is to register customers and assign services. The 'register customer' function involves a loop that allows the receptionist to register multiple customers until they choose to stop the loop by typing 'q'.

customers_information.txt

```
1  jackie,dsa1524,Laptop battery replacement,urgent,210.0,fema
2  steven,qwerty1234,Laptop battery replacement,Normal,180.0,M
3  jean,asdf9876,Operating System Format and Installation,Urge
4  klee,qaz,Troubleshoot and fix computer running slow,Urgent,
5  razor,aooo,Laptop screen replacement,urgent,430.0,male,mons
6  albert,albert1234,Operating System Format and Installation,
7
```

Figure74: Receptionist Customer Text File after Register Customer

```
Register username:q
Return to the main page

           Menu
++++++++++++++++++++++++++++++++++++++++
1. Register Customer and Assign Service
2. Accept Payment and Generate Receipt
3. Update Profile
4. Exit
++++++++++++++++++++++++++++++++++++++++
Selection:2
```

Figure75: Receptionist Register Customer Return to menu

As a result, the receptionist will return to the menu page and must make a new selection.

Accept Payment and Generate Receipt

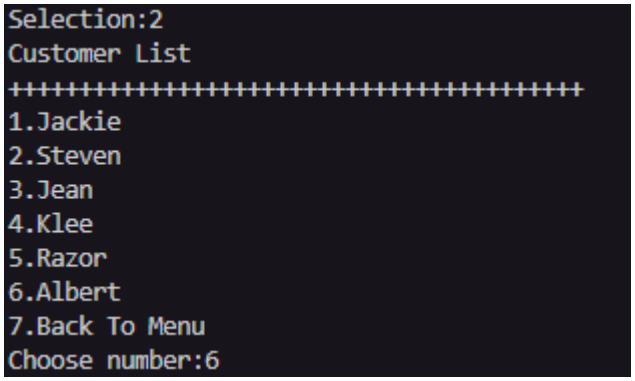


Figure76: Receptionist Accept Payment and generate

When the receptionist proceeds to the second function, 'accept payment and generate receipt,' the program will display a list of customer names. The receptionist can then select a customer from the list and generate a receipt for that customer.

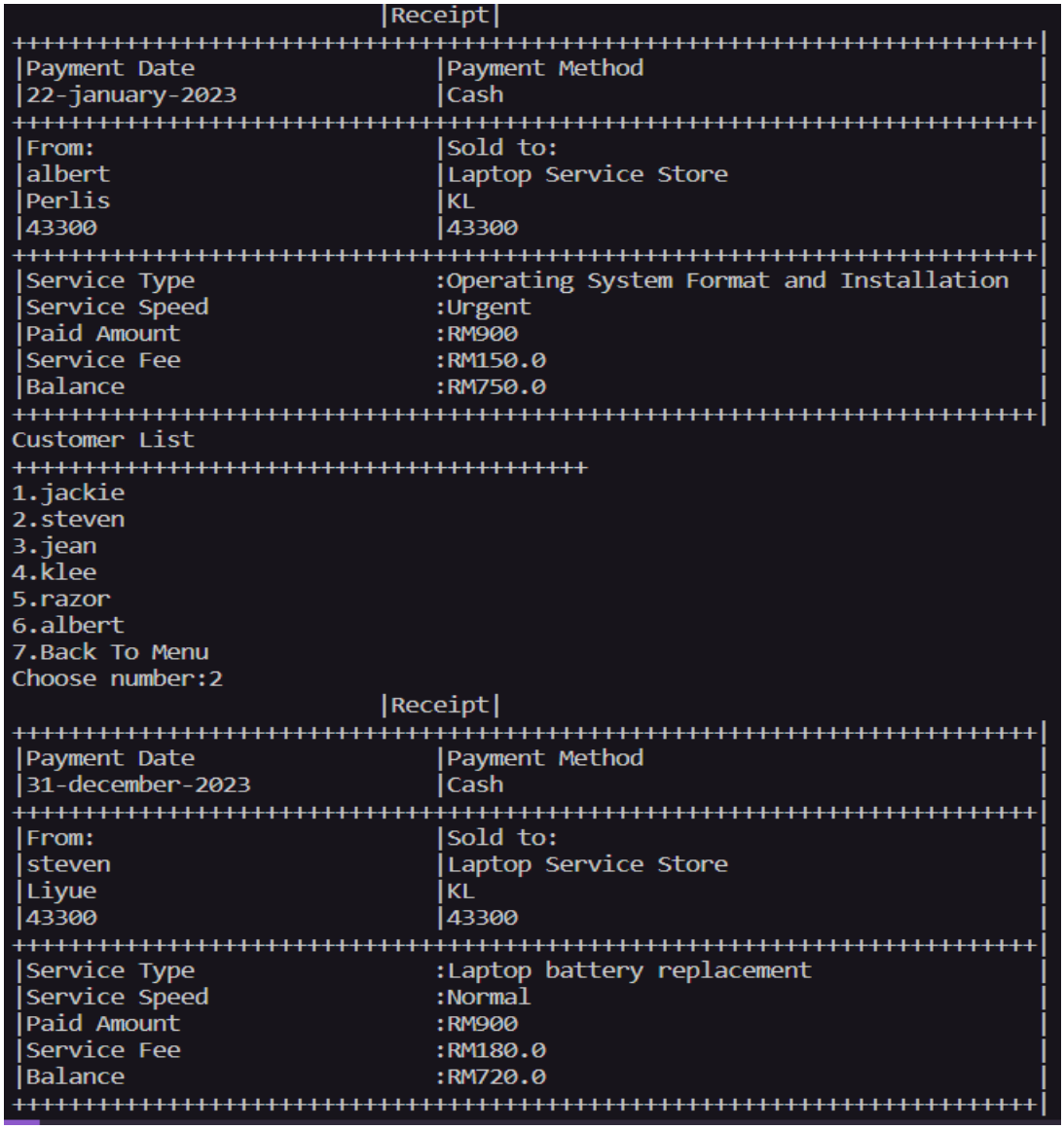


Figure77: Receptionist Sample Generate Receipt

For example, in Figure 78, the receptionist enters 'customer number 6,' and it is displayed in Figure 77. After that, the program returns to the loop and prompts the user to choose their next action. The user can generate receipts for multiple customers or return to the menu whenever they wish.

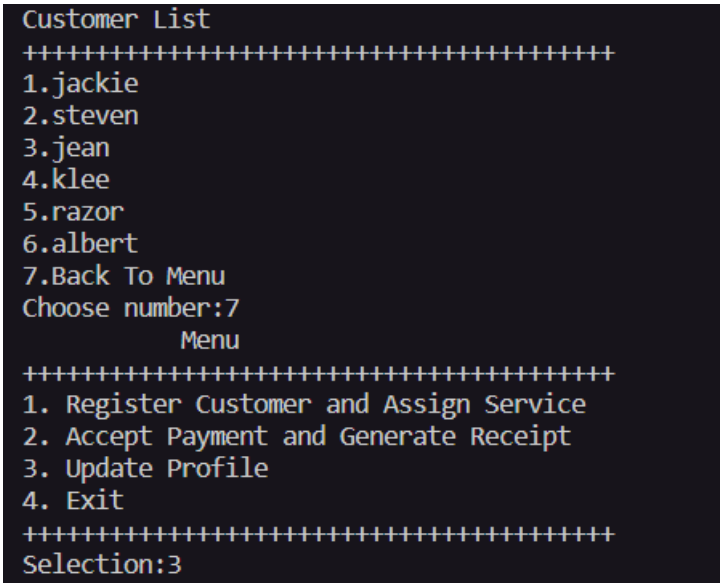


Figure78: Receptionist Generate Receipt Back To Menu

As shown in Figure 78, the user can successfully return to the main page by entering the number corresponding to back to menu. To proceed, the user can use the third function to update their profile. Receptionist accounts and profiles are initially created by the admin, and receptionists can update their own profiles.

Update Profile

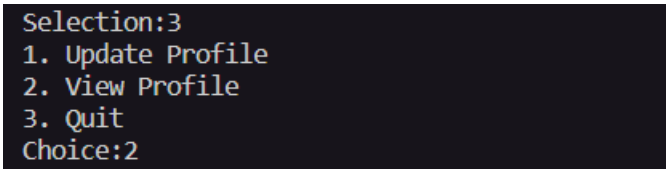


Figure79: Receptionist Update Profile Menu

In the update profile function, there are three options: updating the profile, viewing the current profile, and quitting, which return to menu.

```
Choice:2
Role: Receptionist
Username: junxuan05
Password: junxuan06
Gender: male
City: perlis
Postcode: 55555
Telephone: 1029384756
1. Update Profile
2. View Profile
3. Quit
Choice:1
```

Figure80: Receptionist View Profile

If the user selects 'view profile,' the current profile will be displayed, and the program will wait for the user to enter a number again.

```
Add Information, or type q to quit
New Username:junxuanZ2
New Password:asd1234
Gender:female
City:taiwan
Postcode:22222
Telephone:0123945678
Update Profile Successfully.
1. Update Profile
2. View Profile
3. Quit
Choice:2
```

Figure81: Receptionist Update

In figure 81, it is seeing that user want to update profile after viewing it. If user want to undo update profile by typing q, they will go back to previous page which is Figure 79, to do selection again.

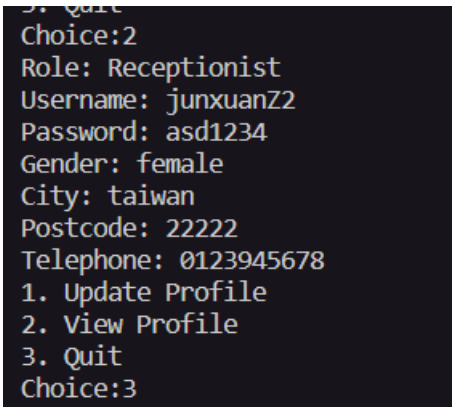
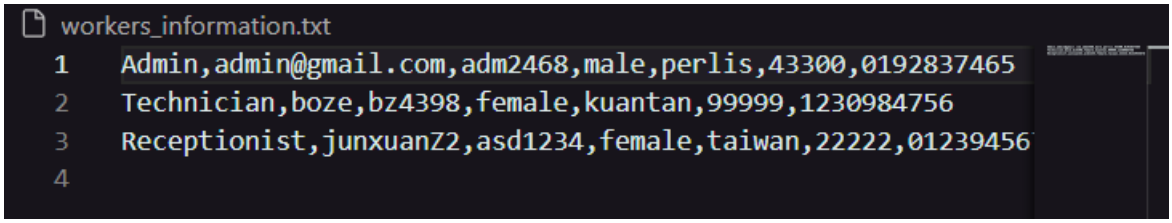


Figure82: Receptionist View Profile after Updating

Additionally, the user can update their username, password, gender, city, postcode, and telephone number if they choose to proceed. Furthermore, after updating, the user can immediately view their profile, which will now reflect the changes made in Figure 81.



Figur83: Receptionist Worker Text File after Updating Profile

Compared to the default worker text file, the line storing receptionist information has been modified.

Exit

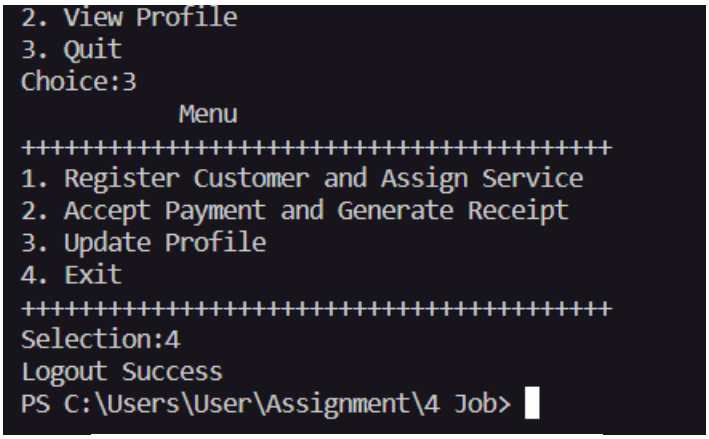


Figure84: Receptionist Exit

Finally, when the user chooses option 3, quit, they will return to the menu page. Within the menu page, the user can log out by selecting option 4.

TAN KAI BOON (ADMIN)

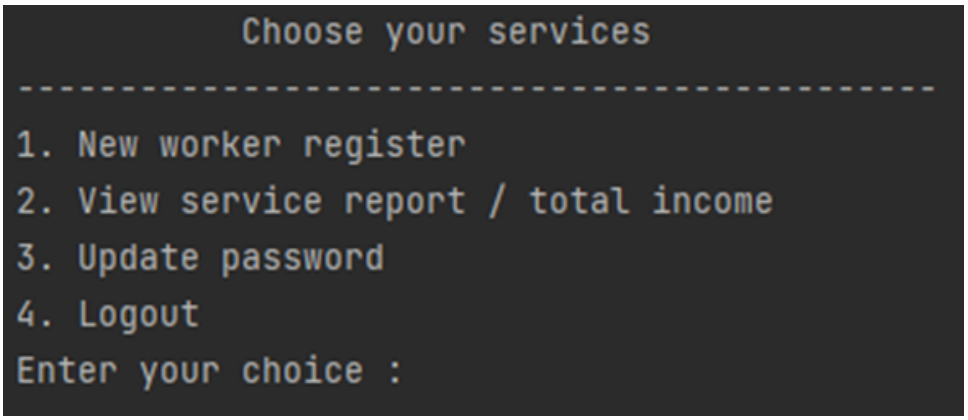


Figure85: Admin menu

When the program is running, it will display four option that can be choose which is new worker register, view service report/total income, update password and logout, user can enter the number to select the service.

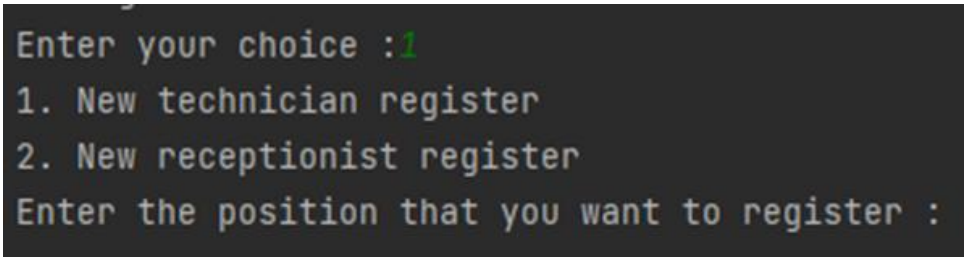


Figure86: Worker register

When the choice is 1, the program will display to ask user to register be a technician or receptionist.

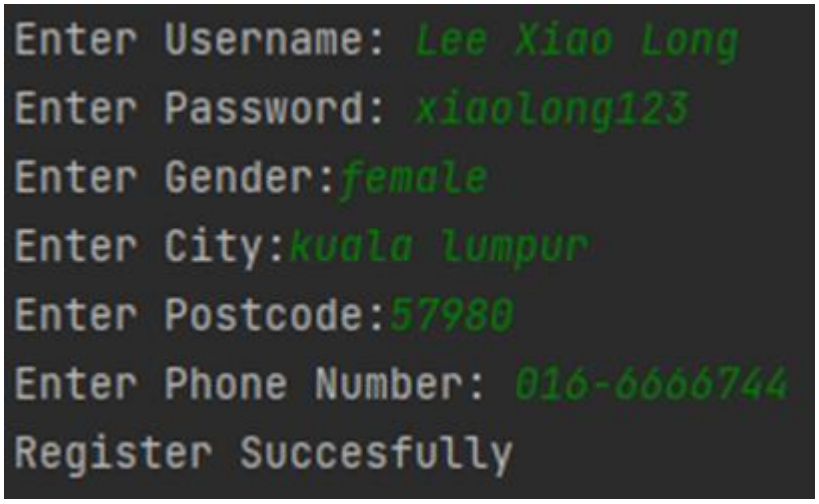


Figure87: Worker register successfully page and the information in file

After the selection, the program will prompt user for Username, Password, Gender, City, Postcode, and Phone Number. When user enter all the information it will store the information of the worker into the workers_information.txt text file and display “Register Successfully”.

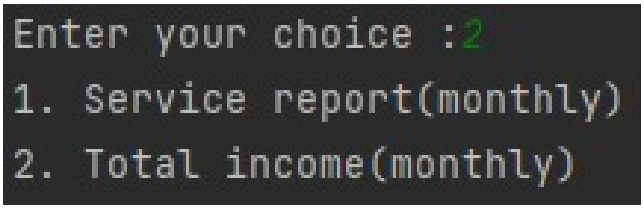


Figure88: Service report & summary

When the choice is 2, it will display to let user select either to view the monthly service report or the monthly total income.

```
Enter your choice: 1
Month (e.g., February): February
Service: Troubleshoot and fix computer running slow
Name: Thor   Date:  4 February  2080

Service: Laptop screen replacement
Name: Jonathon   Date:  2 February  2080

Service: Laptop battery replacement
Name: Lee   Date:  11 February  2080

Service: Operating system format and installation
Name: Ogg   Date:  12 February  2080

Service: Data backup and recovery
Name: Ooi   Date:  2 February  2080

Service: Remove virus malware or spyware
Name: Yasuo   Date:  3 February  2080

------------------Summary--------------------
Troubleshoot and fix computer running slow = 1
Laptop screen replacement = 1
Laptop battery replacement = 1
Operating system format and installation = 1
Data backup and recovery = 1
Remove virus malware or spyware = 1
```

Figure89: Monthly service report

When viewing the monthly service report, the program will prompt user for the month that user want to view about, for instance February, the program will display all the services in February which include the customer's name and the date and generate a summary of February.

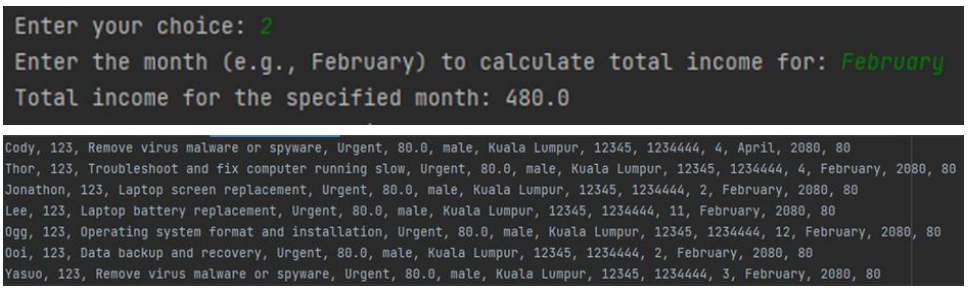


Figure90: Monthly income and original information of text file

When viewing the monthly total income, the program will prompt user for the specific month and calculate the total income according to the customers_informatin.txt text file then displays the final income to user.

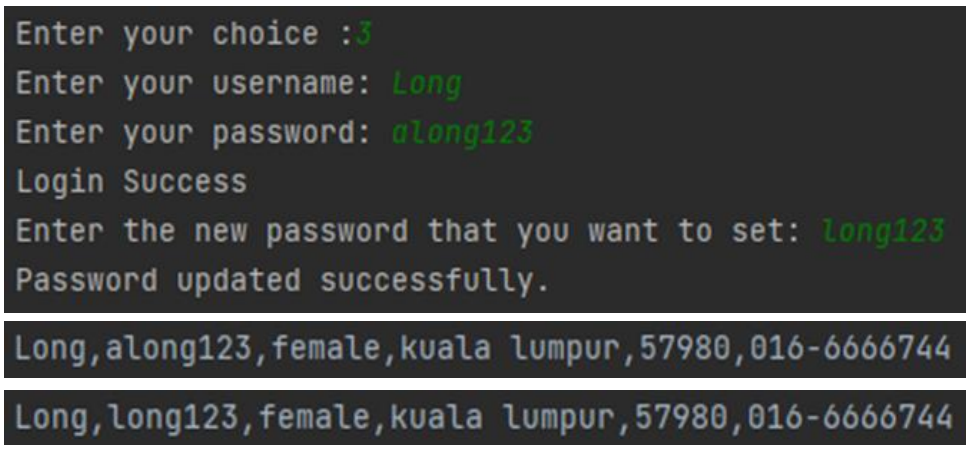


Figure91: User login and the modification of text file

When the choice is 3, the program will first prompt the username and password and read the workers_information.txt text file, if the username and password correct, it will display to let user enter the new password, when the user enter the new password, the old password will be replaced by new password.

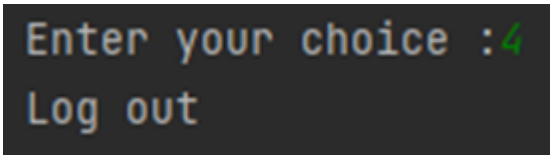


Figure92: Log out

When is choice is 4, the program will display Log out and directly end.

CHIN BO ZE (TECHNICIAN)

Login page:

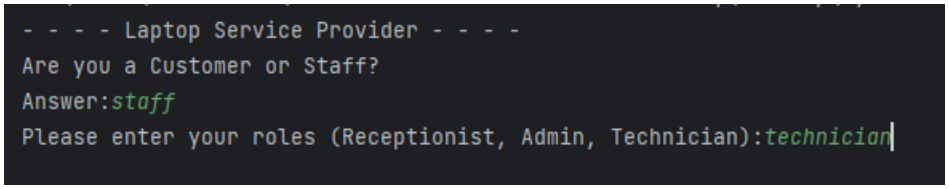


Figure93: Technician login page

When you want to login, the system will ask if you are a customer or staff member. If you choose a staff system, the system will allow you to choose your role, for example, Receptionist, Admin, or Technician.

Figure94: Technician Login

When you have completed the role selection part, you will need to enter the correct username and password to log in to the homepage. When you successfully log in, there will be two options：edit profile and add description, represented by 1 and 2 respectively. You can then choose your page.

Choosing service page:

```
Login Success
1) Edit Profile
2) Add description

Choose your page (1/2):1
Enter your current username and password
Username: technician@gmail.com
Password: 101
```

Figure95: Technician choosing

When choosing 1, you will first need to enter your current username and password.

Update Profile:

```
Choose your page (1/2):1
Enter your current username and password
Username: technician@gmail.com
Password: 101
Enter new username: boze0115@gmail.com
Enter new password: 123456
male/female: male
Enter new city:Pahang
Enter new postcode:27600
Enter new contact number:0126589764
Profile updated successfully.
Do you want to continue? (yes/no): no

Process finished with exit code 0
```

Figure96: Technician Update Profile Page

Next, you will be able to change your information, such as your username, password, gender, city, postcode, and contact number. When the filling is completed, the latest record will be updated in the system. At the end, the system will ask you if you want to continue.

View Customer Information:

```
Login Success
1) Edit Profile
2) Add description

Choose your page (1/2):2
Line 1:
Username: abc@gmail.com
Service type: Laptop battery replacement
Service fee: Urgent
Service fee amount:RM210

Line 2:
Username: dodo@gmail.com
Service type: Troubleshoot and fix computer running slow
Service fee: Urgent
Service fee amount:RM90
```

Figure97: Technician choosing Add Decription

But when you choose 2 on the homepage, you will first see all customer username, service type, service fee and service fee amount.

Add Description

```
Enter the username: abc@gmail.com
Enter the collection date: 18/11/2023
Record added. Thank you!
Do you want to continue? (yes/no): no

Process finished with exit code 0
```

Figure98: Technician Add

When you want to add a customer collection date, you will first enter the username of the designated customer and then enter the collection date. When the input is completed, the collection date will be updated in the designated customer's information.

5.0 Conclusion

In conclusion, Tron is a company with physical stores and mobile apps that provide services related to computer repairs. When users choose online services, they can choose the services they need more quickly and conveniently. As we went through the entire development process, we also faced some challenges, including in our communication with the team. When we complete tasks with our teammates, sometimes everyone has opinions and viewpoints that we do not understand, resulting in some misunderstandings or inconsistencies. Another challenge is that we often need to modify our code, which takes up a lot of time. Sometimes, the company may ask for new requirements to be added or old requirements to be changed, which may result in constant changes in the scope of work and the need to take more time to complete the work. When faced with the above challenges, we adopted some solutions to overcome these problems. First, we hold regular meetings to bring the team together to discuss issues in order to handle them more effectively. We also share our ideas and solutions to ensure everyone understands each other's perspectives. Next, we can create clear requirements documents and solutions, regularly confirming the company's needs with the team to reduce the frequency of changes and reduce the time required for work. We learned a few lessons when we were programming. For example, we gained a lot of new knowledge about programming. Because the field of programming is constantly evolving, we must continue to learn and pursue new knowledge to keep up with technological advances. Additionally, we learned how to do modular programming, breaking the code into parts. This makes the code easier to understand and maintain, while also promoting better team collaboration. There are still some improvements that can be made in this assignment. For instance, I hope that we can add a graphical user interface to this Python project. Currently, we only write code and execute it in the terminal. Introducing a graphical user interface can significantly enhance the program's efficiency and productivity. This is exemplified by its ability to provide graphical elements like buttons and menus, enabling users to perform actions more efficiently, ultimately leading to increased productivity. Beyond the assignment itself, the hard work and collaboration of each member within our group are the primary reasons we successfully completed this task. We are enthusiastic about sharing our thoughts and knowledge with one another, ensuring that everyone excels in their roles. There are no specific requirements for running the code; merely copying it or downloading the file and placing it into any compiler is adequate to run the program. Throughout the assignment, the code doesn't require the installation of additional extensions. All the functions used are built-in and readily available in Python. In conclusion,

this Python assignment has the potential to influence our future significantly, as it has been a valuable learning experience. Concepts such as cohesion, communication, and diligence have better equipped us to tackle various challenges.

6.0 References

There are no sources in the current document.