



اَبُو سَيِّدِي تَيْكُونُ لَوِي كِي مَارَا
UNIVERSITI
TEKNOLOGI
MARA

Cawangan Kedah
Kampus Sungai Petani

**SCHOOL OF INFORMATION SCIENCE,
COLLEGE OF COMPUTING, INFORMATICS AND MATHEMATICS
UNIVERSITI TEKNOLOGI MARA**

CDIM 144 - DIPLOMA IN INFORMATICS LIBRARY

**SUBJECT:
IML 208 PROGRAMMING FOR LIBRARIES**

**INDIVIDUAL ASSIGNMENT:
RENTAL GIRLFRIEND BOOKING SYSTEM**

**PREPARED BY
MOHAMAD AMIR AIMAN BIN ARIFFIN (2023685422)**

**CLASS:
KCDIM1443 E**

**PREPARED FOR:
SIR MOHD FIRDAUS BIN MOHD HELMI**

**SUBMISSION DATE:
18th DECEMBER 2024 (WEEK 10)**

Project Name: Rental Girlfriend Booking System

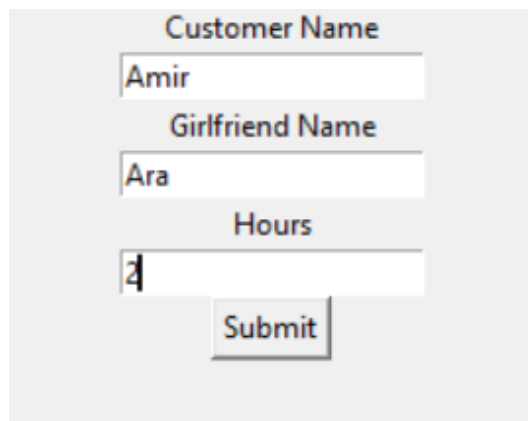
File name: Rentalgirlfriend.py

Prompt Data:

- i) Customer Name
- ii) Girlfriend Name
- iii) Hours

Function:

- i. Create the data



Customer Name

Amir

Girlfriend Name

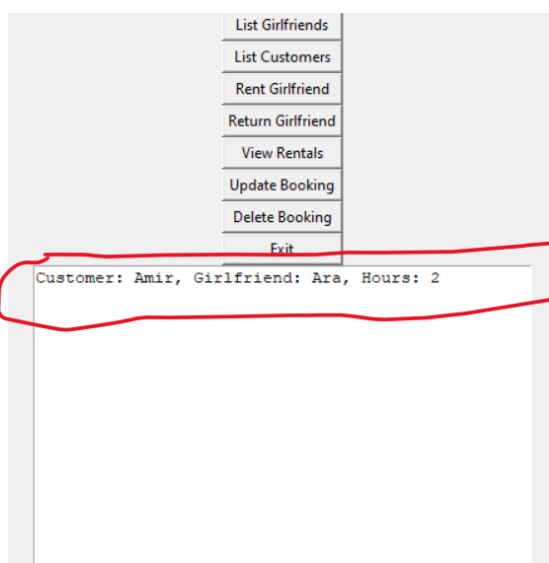
Ara

Hours

2

Submit

- ii. Read the data




List Girlfriends
List Customers
Rent Girlfriend
Return Girlfriend
View Rentals
Update Booking
Delete Booking
Exit

Customer: Amir, Girlfriend: Ara, Hours: 2

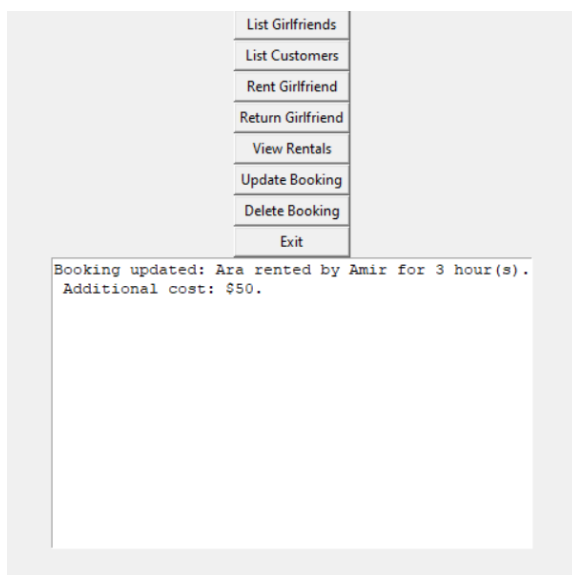
iii. Update the data

Before update the data of customer renting hours



A web form for updating customer data. It contains three text input fields and a submit button. The first field is labeled 'Customer Name' and contains the text 'Amir'. The second field is labeled 'Girlfriend Name' and contains the text 'Ara'. The third field is labeled 'New Hours' and contains the number '3'. Below the third field is a button labeled 'Submit'.

Booking update successfully



A menu of options and a confirmation message. The menu consists of a vertical list of buttons: 'List Girlfriends', 'List Customers', 'Rent Girlfriend', 'Return Girlfriend', 'View Rentals', 'Update Booking', 'Delete Booking', and 'Exit'. Below the menu is a text box containing the message: 'Booking updated: Ara rented by Amir for 3 hour(s). Additional cost: \$50.'

After update the data of customer renting hours

List Girlfriends
List Customers
Rent Girlfriend
Return Girlfriend
View Rentals
Update Booking
Delete Booking
Exit

Booking updated: Ara rented by Amir for 3 hour(s).
Additional cost: \$50.

iv. Delete the data

Before delete the data

Customer Name

Amir

Girlfriend Name

Ara

Submit

Order canceled successfully



No active rentals



Conditional Statement: Yes

```
choice = input("Enter your choice: ")
if choice == "1":
    system.list_girlfriends()

elif choice == "2":
    system.list_customers()

elif choice == "3":
    customer_name = input("Enter customer name: ")
    girlfriend_name = input("Enter girlfriend name: ")
    hours = int(input("Enter rental hours: "))
    system.rent_girlfriend(customer_name, girlfriend_name, hours)

elif choice == "4":
    girlfriend_name = input("Enter girlfriend name to return: ")
    system.return_girlfriend(girlfriend_name)

elif choice == "5":
    system.view_rentals()

elif choice == "6":
    customer_name = input("Enter customer name: ")
    girlfriend_name = input("Enter girlfriend name: ")
    system.delete_renting(customer_name, girlfriend_name)

elif choice == "7":
    customer_name = input("Enter customer name: ")
    girlfriend_name = input("Enter girlfriend name: ")
    new_hours = int(input("Enter new rental hours: "))
    system.update_booking(customer_name, girlfriend_name, new_hours)

elif choice == "8":
    print("Exiting the system. Goodbye!")
    break

else:
    print("Invalid choice. Please try again.")
```

GUI: Yes

List Girlfriends
List Customers
Rent Girlfriend
Return Girlfriend
View Rentals
Update Booking
Delete Booking
Exit

Result: Screenshot



Strengths:

1. User-Friendly Interface

- Buttons for each major function (e.g., listing, renting, returning, updating, and deleting bookings) make the system easy to navigate.
- Input windows are dynamically created with clear prompts for required details, reducing user confusion.

2. Comprehensive Functionality

- Supports all essential operations: listing girlfriends and customers, managing rentals, returning girlfriends, updating bookings, and deleting bookings.
- Ensures all use cases are covered within a single interface.

3. Real-Time Feedback

- The output text area provides immediate feedback for user actions, such as successful operations or error messages, enhancing user experience.

4. Modular and Scalable Design

- Each action has its own method, making it easier to maintain and extend the code.
- Additional functionality (like adding more fields or actions) can be integrated without significant restructuring.

5. Preloaded Data

- Preloading girlfriends and customers ensures the system is ready to use upon launch, providing a good starting point for testing or actual use.

