```
import pandas as pd
import matplotlib.pyplot as plt

# Load the CSV file once at the start
df = pd.read_csv("Task4a_data.csv")
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)  # Convert the Date column once for efficiency


def sales_menu():
    while True:
        print("\n##################################################")
        print("Welcome! Please choose an option from the list")
        print("1. Show sales for a specific car model")
        print("2. Show sales for all car models")
        print("3. Show sales for a specific salesperson")
        print("4. Exit")
        print("##################################################")

        choice = input("Please enter the number of your choice (1-4): ")
        if choice in ['1', '2', '3', '4']:
            return int(choice)
        else:
            print("Invalid choice. Please enter a number between 1 and 4.")


def get_car_model_choice():
    car_models = df['Car Model'].unique()
    for i, model in enumerate(car_models, 1):
        print(f"{i}. {model}")

    while True:
        choice = input(f"Please enter the number of your choice (1-{len(car_models)}): ")
        if choice.isdigit() and 1 <= int(choice) <= len(car_models):
            return car_models[int(choice) - 1]
        else:
            print("Invalid choice. Please select a valid car model number.")


def get_salesperson_choice():
    salespeople = df['Salesperson'].unique()
    for i, person in enumerate(salespeople, 1):
        print(f"{i}. {person}")
```

```
    while True:
        choice = input(f"Please enter the number of your choice (1-{len(salespeople)}): ")
        if choice.isdigit() and 1 <= int(choice) <= len(salespeople):
            return salespeople[int(choice) - 1]
        else:
            print("Invalid choice. Please select a valid salesperson number.")


def get_start_date():
    while True:
        try:
            start_date = input("Please enter the start date (DD/MM/YYYY): ")
            return pd.to_datetime(start_date, dayfirst=True)
        except ValueError:
            print("Invalid date format. Please use DD/MM/YYYY.")


def get_end_date():
    while True:
        try:
            end_date = input("Please enter the end date (DD/MM/YYYY): ")
            return pd.to_datetime(end_date, dayfirst=True)
        except ValueError:
            print("Invalid date format. Please use DD/MM/YYYY.")


def show_sales_all(start_date, end_date):
    filtered_df = df[(df['Date'] >= start_date) & (df['Date'] <= end_date)]
    print("\nSales Data for All Car Models:")
    print(filtered_df)

    total_sales = filtered_df['Value'].sum()
    print(f"\nTotal Sales for the selected period: £{total_sales:.2f}")

    # Plotting sales over time
    plt.figure(figsize=(10, 6))
    plt.plot(filtered_df['Date'], filtered_df['Value'], marker='o', linestyle='-', color='blue')
    plt.xlabel('Date')
    plt.ylabel('Sales Value (£)')
    plt.title('Total Sales Over Time')
    plt.grid(True)
    plt.show()
```

```
def show_sales_by_model(start_date, end_date):
    car_model = get_car_model_choice()
    filtered_df = df[(df['Date'] >= start_date) & (df['Date'] <= end_date) & (df['Car Model']
== car_model)]

    print(f"\nSales Data for {car_model}:")
    print(filtered_df)

    total_sales = filtered_df['Value'].sum()
    print(f"\nTotal Sales for {car_model} during the selected period: £{total_sales:.2f}")

    # Plotting sales over time for the selected car model
    plt.figure(figsize=(10, 6))
    plt.plot(filtered_df['Date'], filtered_df['Value'], marker='o', linestyle='-', color='green')
    plt.xlabel('Date')
    plt.ylabel('Sales Value (£)')
    plt.title(f'Sales of {car_model} Over Time')
    plt.grid(True)
    plt.show()


def show_sales_by_salesperson(start_date, end_date):
    salesperson = get_salesperson_choice()
    filtered_df = df[(df['Date'] >= start_date) & (df['Date'] <= end_date) & (df['Salesperson']
== salesperson)]

    print(f"\nSales Data for {salesperson}:")
    print(filtered_df)

    total_sales = filtered_df['Value'].sum()
    print(f"\nTotal Sales by {salesperson} during the selected period: £{total_sales:.2f}")

    # Plotting sales over time for the selected salesperson
    plt.figure(figsize=(10, 6))
    plt.plot(filtered_df['Date'], filtered_df['Value'], marker='o', linestyle='-', color='purple')
    plt.xlabel('Date')
    plt.ylabel('Sales Value (£)')
    plt.title(f'Sales by {salesperson} Over Time')
    plt.grid(True)
    plt.show()


def main():
    while True:
```

```
        choice = sales_menu()
        if choice == 4:
            print("Exiting the program. Goodbye!")
            break

        start_date = get_start_date()
        end_date = get_end_date()

        if choice == 1:
            show_sales_by_model(start_date, end_date)
        elif choice == 2:
            show_sales_all(start_date, end_date)
        elif choice == 3:
            show_sales_by_salesperson(start_date, end_date)


if __name__ == "__main__":
    main()
```