Camera Rental Application

Developer- Shivam Jadhav

Sprint planning

Week 1

1. Using proper naming conventions and code documentation
2. Create app structure for start coding
3. Create login functionality for security.
4. Create My camera and add functions
5. Create menu for navigate through the app

Week 2

1. Create submenu in My camera
2. Create function like add , remove , view list and exit
3. Create my wallet for add money or view balance
4. Create next feature for renting the cameras
5. Create feature list all cameras

Ensure working of all feature and function of app

Details of Classes and Methods

1. **Camera:**

Class reprents a camera details like Brand , Model , Price per day

Use constructor to initialize camera brand,model, and price Per day

Use of setter and getter for access properties

2. **CameraRentalApplication:**

Class represents main method

It includes login feature

Methods used:

myCamera() - includes submenu add,remove,view

addCamera() - adds camera to the list

removeCamera() - remove camera from list

viewCamera() - shows list of camera after add or remove

listCamera() - show list of all cameras

rentCamera() - for rent the cameras

addOrViewWalletAmount() – add money to wallet and show balance remaining

OOP concepts used in the code

1. **Classes and Objects**
   In the code we define class like cameraRentalApplication and camera class. Object of these classes are used in the code.

2. **Constuctors**
   Constuctos like camera are created and initialize the object instances

3. **Encapsulations**
   Encapsulation is used in camera class where private access modifiers used for class members and call them using setter and getter methods to control access to these members.

4. **Inheritance**
   Inheritance allows you to create new classes by inheriting properties and methods from existing classes. In this code we are not using it.

5. **Polymorphism**
   Not used the polymorphism in this code

6. **Abstraction**
   abstraction is used in the code in class like camera and cameraRentalApplication class.

Algorithm

Initialize variables and data structures

Display a welcome message

Repeat until the user chooses to exit:

Display the main menu with options:

1. My Camera
2. Rent Camera
3. Add or view wallet amount
4. List all cameras
5. Exit

Ask the user to choose an option

If the user's choice is "My Camera" (1):

Repeat until the user chooses to return to the main menu:

Display the "My Camera" submenu with options:

a. Add Camera

b. Remove Camera

c. view Camera

d. Return to Main Menu

Ask the user to choose an option from the submenu

If the user's choice is "Add Camera" (a):

Ask for camera details (e.g.id, name, type, rent price)

Add the camera to the inventory

Display a message confirming the addition

If the user's choice is "Remove Camera" (b):

Ask for the camera ID or name to be removed

Remove the camera from the inventory

Display a message confirming the removal

If the user's choice is "List Camera" (c):

Display the list of available cameras

If the user's choice is "Return to Main Menu" (d):

Exit the submenu

If the user's choice is "Rent Camera" (2):

Display available cameras

Ask the user to select a camera to rent

Ask for rental details (e.g., rental period, customer information)

Reserve the camera for the user

Display a message confirming the rental

If the user's choice is "Camera List" (3):

Display the list of all available cameras

If the user's choice is "Wallet" (4):

Display the user's wallet balance

Provide options to add funds to the wallet

If the user's choice is "Exit" (5):

Exit the application

If the user's choice is invalid:

Display an error message and ask the user to choose again

Exit the application