Camera Rental Application
Phase-End Project Problem Statement

Project Agenda: Create a Camera Rental Application using command line interface, core java concepts, and data structures in Java.

Source Code:

```
import java.util.ArrayList;
                import java.util.List;
                import java.util.Scanner;



                class Camera {
                    private String brand;//instance of an variables.
                    private String model;
                    private double perDayPrice;
                    private boolean isRented;

                    public Camera(String brand, String model, double perDayPrice) {
                        this.brand = brand;//refrence of an variables
                        this.model = model;
                        this.perDayPrice = perDayPrice;
                        this.isRented = false;
                    }
```

```
    public String getBrand() {//getters
        return brand;
    }

    public String getModel() {
        return model;
    }

    public double getPerDayPrice() {
        return perDayPrice;
    }

    public boolean isRented() {
        return isRented;
    }

    public void setRented(boolean rented) {
        isRented = rented;
    }

    @Override
    public String toString() {
        return brand + " " + model + " - " + perDayPrice;
    }
}

class Wallet {
    private double balance;
    public Wallet() {
        this.balance = 0.0;
    }

    public double getBalance() {
        return balance;
    }

    public void deposit(double amount) {
        balance += amount;
    }

    public boolean withdraw(double amount) {
        if (amount <= balance) {
            balance =balance-amount;
            return true;
        }
        return false;
    }
}


public class CameraRentalApp {
    public static List<Camera> cameraList = new ArrayList<>();
    private static Wallet wallet = new Wallet();

    public static void main(String[] args) {
        boolean exit = false;
        Scanner scanner = new Scanner(System.in);
            displayWelcomeScreen();
            manageMainMenu();




        scanner.close();
    }

    private static void displayWelcomeScreen() {
          System.out.println("+---------------------------------+");
        System.out.println("|   WELCOME TO CAMERA RENTAL APP   |");
        System.out.println("+---------------------------------+");


        System.out.println("PLEASE LOGIN TO CONTINUE -");
        Scanner sc=new Scanner(System.in);

        System.out.println("USERNAME -");
        sc.next();
        System.out.println("PASSWORD -");
        sc.next();


        System.out.println("                ");
        System.out.println("Login Succesful");

        System.out.println("                ");
```

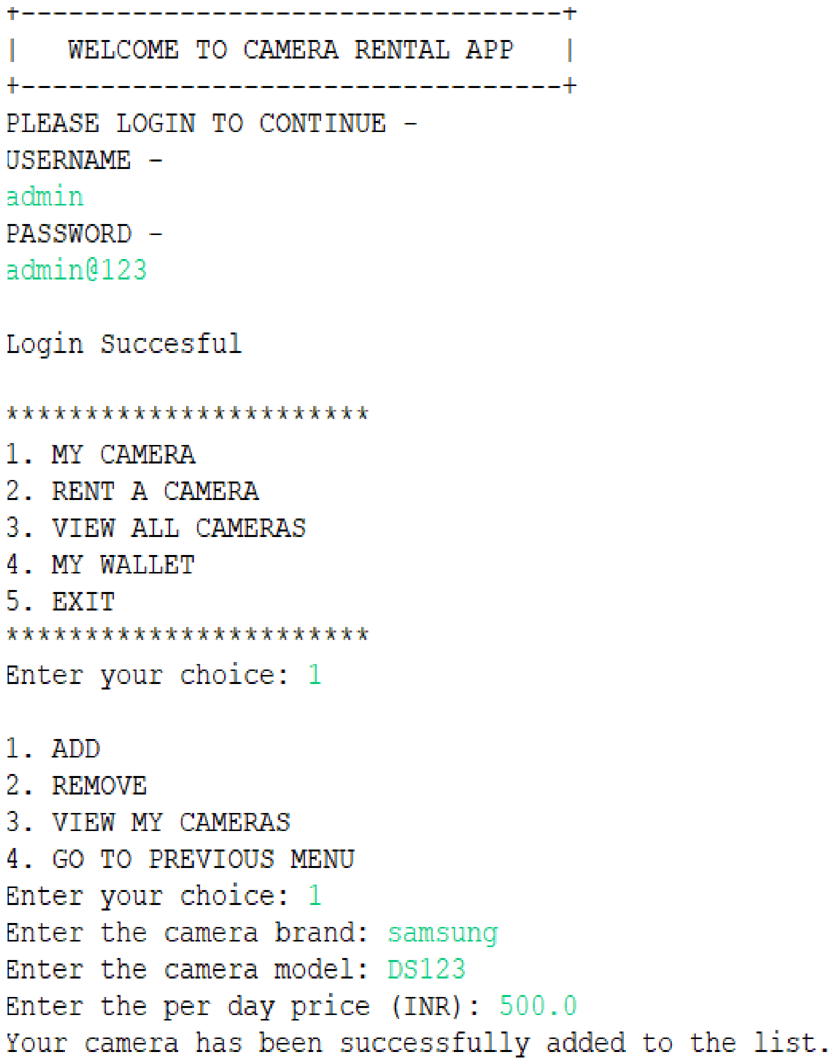
```
}

private static int getUserChoice(Scanner scanner) {
    System.out.print("Enter your choice: ");
    return scanner.nextInt();
}
private static void manageMainMenu( ) {
      boolean exit = false;
      System.out.println("***********************");
      System.out.println("1. MY CAMERA");
    System.out.println("2. RENT A CAMERA");
    System.out.println("3. VIEW ALL CAMERAS");
    System.out.println("4. MY WALLET");
    System.out.println("5. EXIT");
    System.out.println("***********************");

    Scanner scanner = new Scanner(System.in);
    int choice = getUserChoice(scanner);
    switch (choice) {
        case 1:
            manageMyCamera(scanner,false);
            manageMainMenu();
            break;
        case 2:
            rentCamera(scanner);
            manageMainMenu();
            break;
        case 3:
            viewAllCameras();
            manageMainMenu();
            break;
        case 4:
            manageWallet(scanner);
            manageMainMenu();
            break;
        case 5:
          /*  exit = true;
            System.out.println("Exiting the application...");*/
                 exitApplication();
            break;
        default:
            System.out.println("Invalid choice. Please try again.");
    }

}

private static void manageMyCamera(Scanner scanner, boolean displayMainMenu) {


        System.out.println("\n1. ADD");
        System.out.println("2. REMOVE");
        System.out.println("3. VIEW MY CAMERAS");
        System.out.println("4. GO TO PREVIOUS MENU");

        int choice = getUserChoice(scanner);
        switch (choice) {
            case 1:
                addCamera(scanner);
                manageMyCamera(scanner, false);// if we use true then the it will break and not excecute other
                break;
            case 2:
                removeCamera(scanner);
                manageMyCamera(scanner, false);
                break;
            case 3:
                viewMyCameras();
                manageMyCamera(scanner, false);
                break;
            case 4:
                  manageMainMenu();
                break;
            default:
                System.out.println("Invalid choice. Please try again.Please re-enter the  choice below");
                manageMainMenu();

        }
    }



private static void addCamera(Scanner scanner) {
    System.out.print("Enter the camera brand: ");
    String brand = scanner.next();
    System.out.print("Enter the camera model: ");
    String model = scanner.next();
    System.out.print("Enter the per day price (INR): ");
    double perDayPrice = scanner.nextDouble();
    Camera camera = new Camera(brand, model, perDayPrice);
    cameraList.add(camera);

    System.out.println("Your camera has been successfully added to the list.");

}

private static void removeCamera(Scanner scanner) {
    viewMyCameras();

    System.out.print("Enter the camera ID to remove: ");
    int cameraId = scanner.nextInt();

    if (cameraId >= 0 && cameraId < cameraList.size()) {
        cameraList.remove(cameraId);
        System.out.println("Camera successfully removed from the list.");
    } else {
        System.out.println("Invalid camera ID.");

    }

}


private static void viewMyCameras() {

    if (cameraList.isEmpty()) {
        System.out.println("No cameras present at this moment.");
```

```
            } else {
              System.out.println(        "--------------------------------------------------");
              System.out.printf("%-10s %-10s %-10s %-10s %-10s\n",
                      "CAMERA ID", "BRAND", "MODEL", "PRICE(Per day)", "STATUS");//%-10s it iused for
holding the data.
              System.out.println(       "---------------------------------------------");
               int id = 0;
               for (Camera camera : cameraList) {
                  System.out.printf("%-10s %-10s %-10s %-10.2f %-10s\n",
                          id++, camera.getBrand(), camera.getModel(),
                          camera.getPerDayPrice(), camera.isRented() ? "Rented" : "Available");



               }
            }

          }
          private static void rentCamera(Scanner scanner) {
            viewAllCameras();

            if (cameraList.isEmpty()) {
               System.out.println("No cameras available for rent at this moment.");
               return;
            }

            System.out.print("Enter the camera ID you want to rent: ");
            int cameraId = scanner.nextInt();

            if (cameraId >= 0 && cameraId < cameraList.size()) {
```

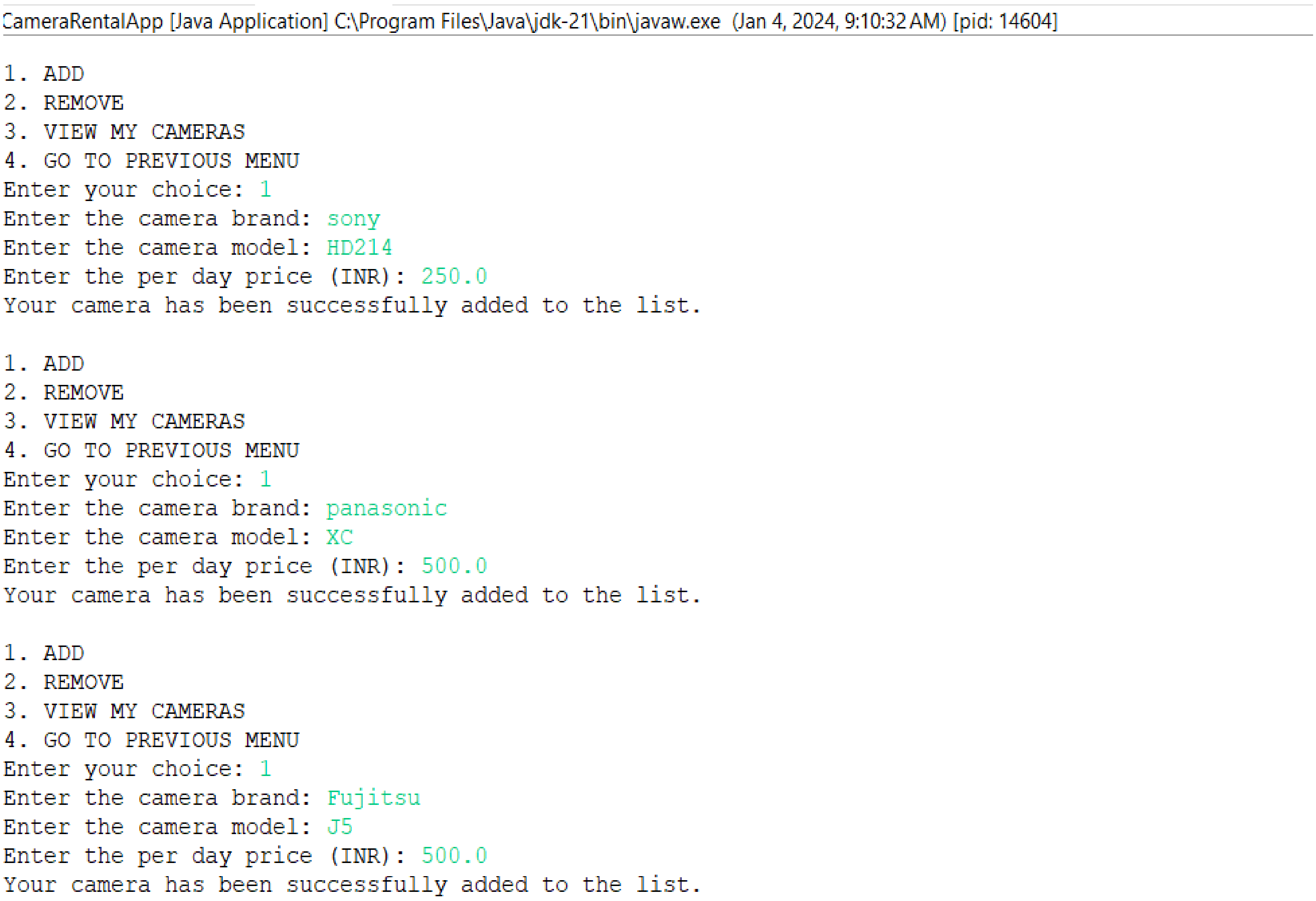
```
                                Camera camera = cameraList.get(cameraId);
                                if (camera.isRented()) {
                                    System.out.println("Camera is already rented.");
                                } else {
                                    if (wallet.getBalance() >= camera.getPerDayPrice()) {
                                        wallet.withdraw(camera.getPerDayPrice());
                                        camera.setRented(true);
                                        String output = "YOUR TRANSACTION FOR CAMERA " + camera.getBrand() + " " +
camera.getModel() +
                                                " WITH RENT INR." + camera.getPerDayPrice() + " HAS SUCCESSFULLY
COMPLETED";
                                        System.out.println(output);
                                    } else {
                                        System.out.println("Insufficient wallet balance. Please deposit the amount to your wallet.");
                                    }
                                }}
                             else {
                                System.out.println("Invalid camera ID.");
                            }
                        }


                        private static void viewAllCameras() {

                            System.out.println("\nFOLLOWING IS THE LIST OF AVAILABLE CAMERA(S)\n");
                            if (cameraList.isEmpty()) {
                                System.out.println("No cameras available at this moment.");
                            } else {
                                System.out.printf("%-10s %-10s %-10s %-10s %-10s\n",
        "CAMERA ID", "BRAND", "MODEL", "PRICE", "STATUS");
    int id = 1;
    for (Camera camera : cameraList) {
      System.out.printf("%-10s %-10s %-10s %-10.2f %-10s\n",
          id++, camera.getBrand(), camera.getModel(),
          camera.getPerDayPrice(), camera.isRented() ? "Rented" : "Available");

    }
  }

}

private static void manageWallet(Scanner scanner) {
   Scanner sc=new Scanner(System.in);
  System.out.println("\nMY WALLET\n");
  System.out.printf("Your current wallet balance is INR %.2f\n", wallet.getBalance());

  System.out.println("Do you want to deposit more amount to your wallet?");
  System.out.println("1. Yes");
  System.out.println("2. No");

  int choice = scanner.nextInt();

  switch (choice) {
    case 1:
      System.out.print("Enter the amount (INR): ");
      double amount = scanner.nextDouble();
      wallet.deposit(amount);
```

```
                    System.out.printf("Your wallet balance updated successfully. Current wallet balance: INR %.2f\n",
wallet.getBalance());
                    break;
                case 2:
                    break;
                default:
                    System.out.println("Invalid choice. Please try again.");


            }

        }

        private static void exitApplication() {
            System.out.println("Exiting the application... Goodbye!");
            System.exit(0);
        }
    }
```

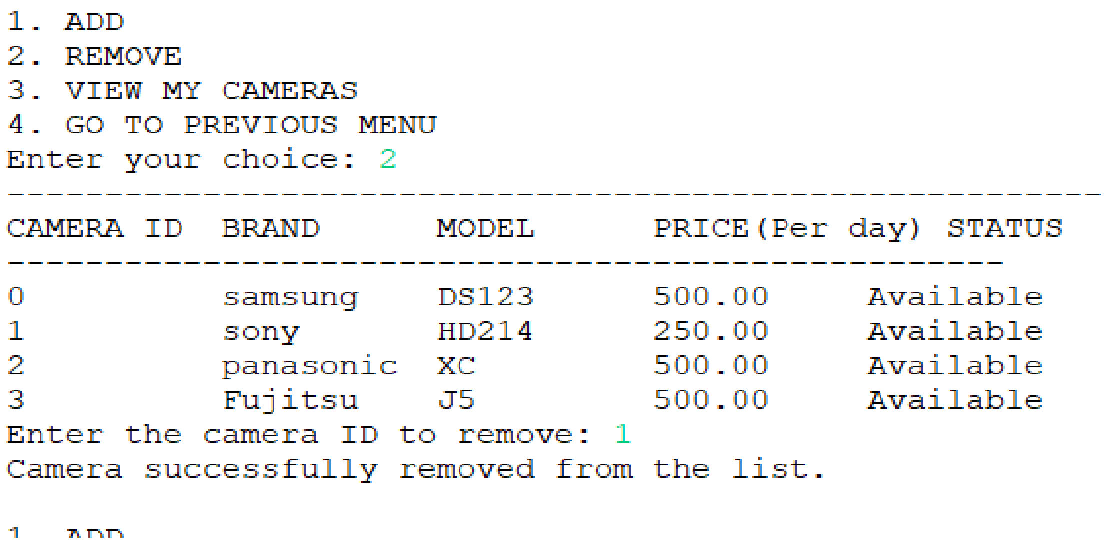
Output:



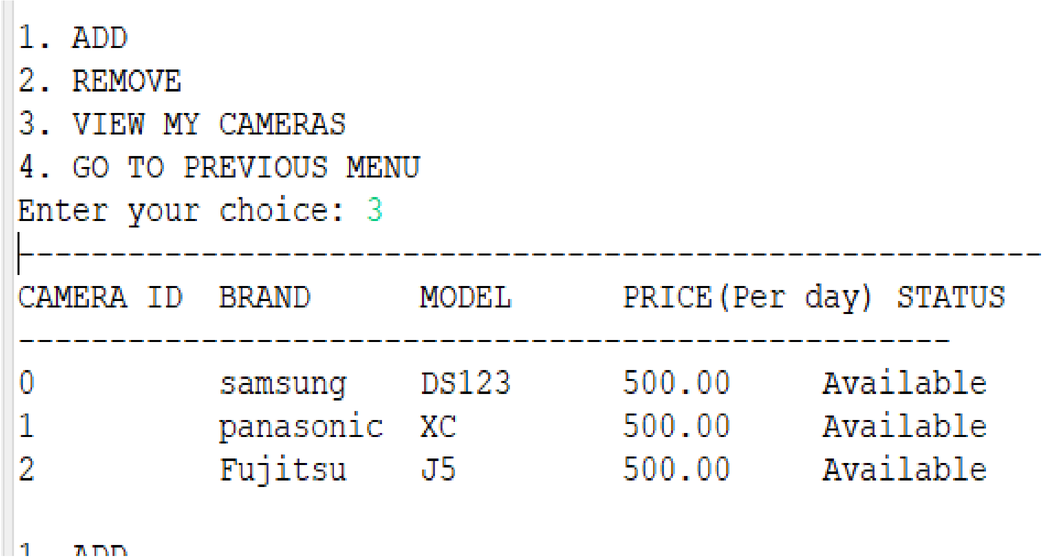
```
+----------------------------------+
|   WELCOME TO CAMERA RENTAL APP   |
+----------------------------------+
PLEASE LOGIN TO CONTINUE -
USERNAME -
admin
PASSWORD -
admin@123

Login Succesful

***********************
1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5. EXIT
***********************
Enter your choice: 1

1. ADD
2. REMOVE
3. VIEW MY CAMERAS
4. GO TO PREVIOUS MENU
Enter your choice: 1
Enter the camera brand: samsung
Enter the camera model: DS123
Enter the per day price (INR): 500.0
Your camera has been successfully added to the list.
```



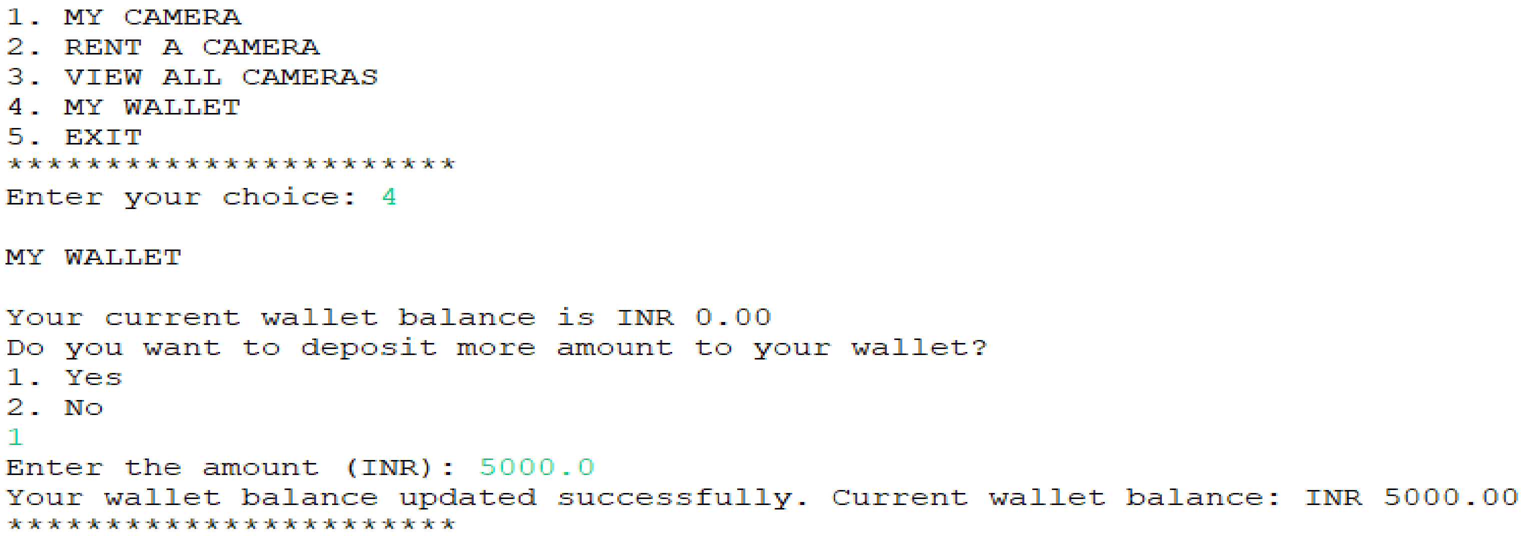
```
CameraRentalApp [Java Application] C:\Program Files\Java\jdk-21\bin\javaw.exe  (Jan 4, 2024, 9:10:32 AM) [pid: 14604]

1. ADD
2. REMOVE
3. VIEW MY CAMERAS
4. GO TO PREVIOUS MENU
Enter your choice: 1
Enter the camera brand: sony
Enter the camera model: HD214
Enter the per day price (INR): 250.0
Your camera has been successfully added to the list.

1. ADD
2. REMOVE
3. VIEW MY CAMERAS
4. GO TO PREVIOUS MENU
Enter your choice: 1
Enter the camera brand: panasonic
Enter the camera model: XC
Enter the per day price (INR): 500.0
Your camera has been successfully added to the list.

1. ADD
2. REMOVE
3. VIEW MY CAMERAS
4. GO TO PREVIOUS MENU
Enter your choice: 1
Enter the camera brand: Fujitsu
Enter the camera model: J5
Enter the per day price (INR): 500.0
Your camera has been successfully added to the list.
```



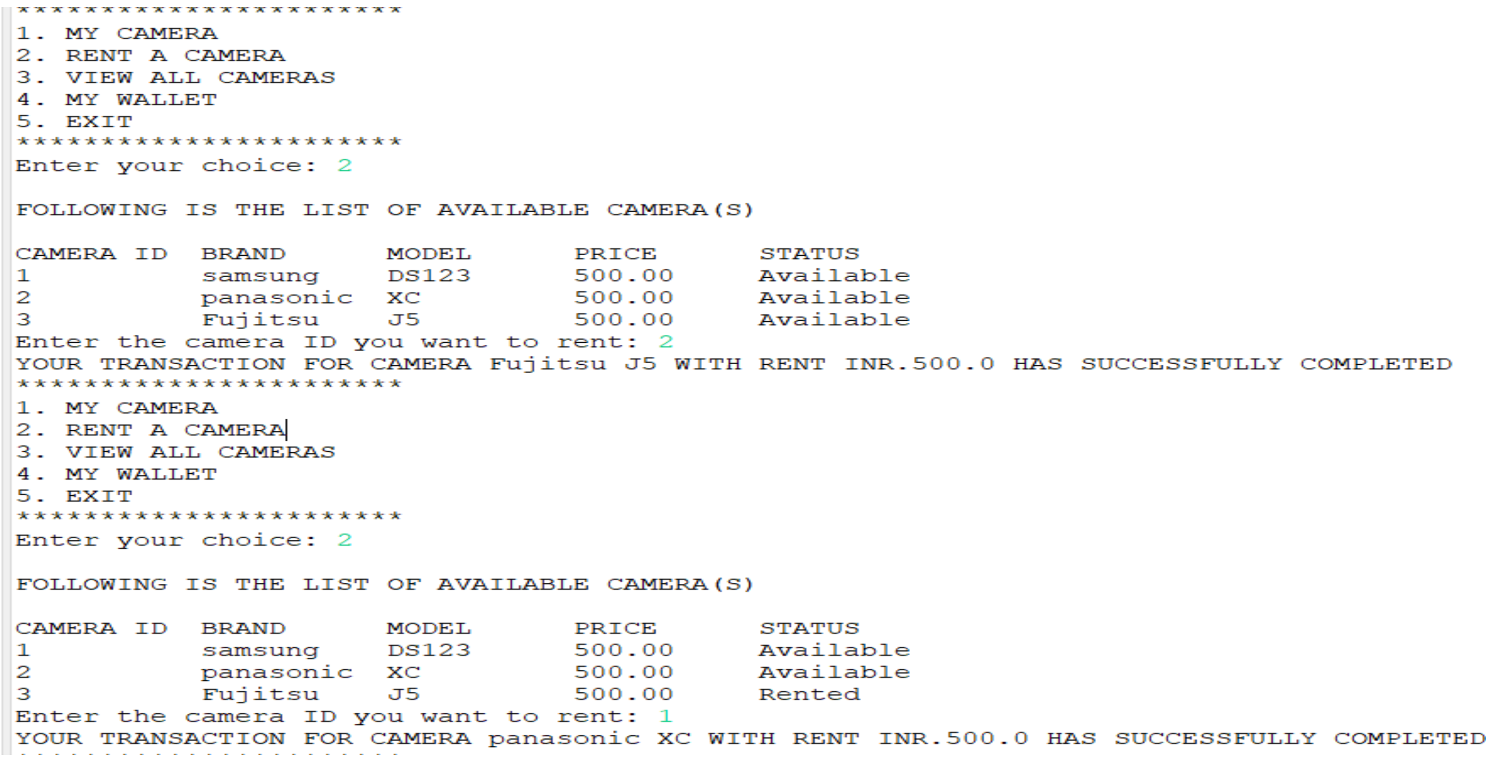
```
1. ADD
2. REMOVE
3. VIEW MY CAMERAS
4. GO TO PREVIOUS MENU
Enter your choice: 2
-------------------------------------------------------
CAMERA ID  BRAND      MODEL      PRICE(Per day)  STATUS
--------------------------------------------------
0          samsung    DS123      500.00     Available
1          sony       HD214      250.00     Available
2          panasonic  XC         500.00     Available
3          Fujitsu    J5         500.00     Available
Enter the camera ID to remove: 1
Camera successfully removed from the list.

1. ADD
```



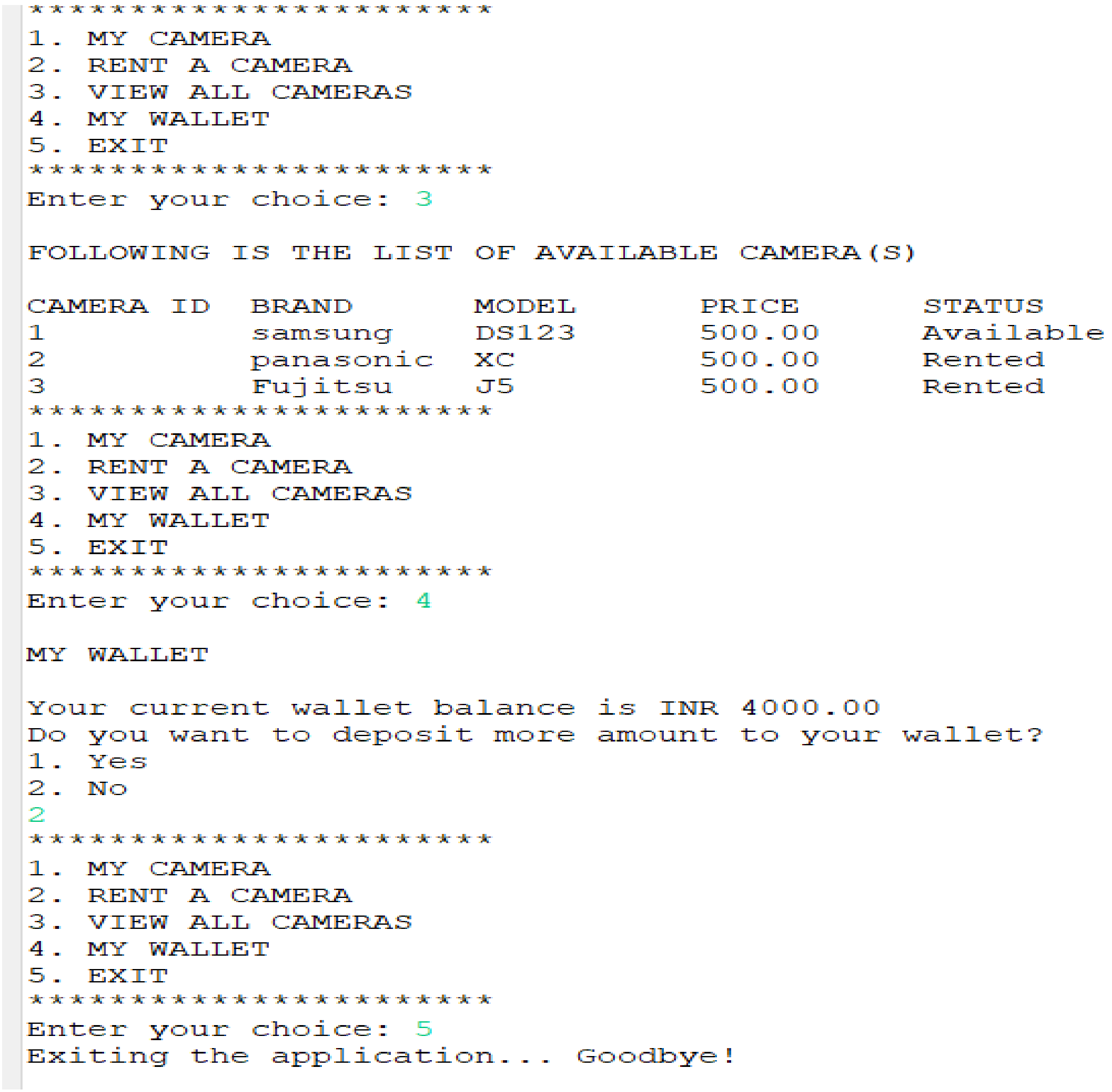
```
1. ADD
2. REMOVE
3. VIEW MY CAMERAS
4. GO TO PREVIOUS MENU
Enter your choice: 3
-------------------------------------------------------
CAMERA ID  BRAND      MODEL      PRICE(Per day)  STATUS
--------------------------------------------------
0          samsung    DS123      500.00     Available
1          panasonic  XC         500.00     Available
2          Fujitsu    J5         500.00     Available

1. ADD
```



```
1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5. EXIT
***********************
Enter your choice: 4

MY WALLET

Your current wallet balance is INR 0.00
Do you want to deposit more amount to your wallet?
1. Yes
2. No
1
Enter the amount (INR): 5000.0
Your wallet balance updated successfully. Current wallet balance: INR 5000.00
***********************
```



```
***********************
1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5. EXIT
***********************
Enter your choice: 2

FOLLOWING IS THE LIST OF AVAILABLE CAMERA(S)

CAMERA ID  BRAND      MODEL      PRICE      STATUS
1          samsung    DS123      500.00     Available
2          panasonic  XC         500.00     Available
3          Fujitsu    J5         500.00     Available
Enter the camera ID you want to rent: 2
YOUR TRANSACTION FOR CAMERA Fujitsu J5 WITH RENT INR.500.0 HAS SUCCESSFULLY COMPLETED
***********************
1. MY CAMERA
2. RENT A CAMERA
3. VIEW ALL CAMERAS
4. MY WALLET
5. EXIT
***********************
Enter your choice: 2

FOLLOWING IS THE LIST OF AVAILABLE CAMERA(S)

CAMERA ID  BRAND      MODEL      PRICE      STATUS
1          samsung    DS123      500.00     Available
2          panasonic  XC         500.00     Available
3          Fujitsu    J5         500.00     Rented
Enter the camera ID you want to rent: 1
YOUR TRANSACTION FOR CAMERA panasonic XC WITH RENT INR.500.0 HAS SUCCESSFULLY COMPLETED
***********************
```



```
***********************
1.  MY CAMERA
2.  RENT A CAMERA
3.  VIEW ALL CAMERAS
4.  MY WALLET
5.  EXIT
***********************
Enter your choice: 3

FOLLOWING IS THE LIST OF AVAILABLE CAMERA(S)

CAMERA ID  BRAND      MODEL      PRICE      STATUS
1          samsung    DS123      500.00     Available
2          panasonic  XC         500.00     Rented
3          Fujitsu    J5         500.00     Rented
***********************
1.  MY CAMERA
2.  RENT A CAMERA
3.  VIEW ALL CAMERAS
4.  MY WALLET
5.  EXIT
***********************
Enter your choice: 4

MY WALLET

Your current wallet balance is INR 4000.00
Do you want to deposit more amount to your wallet?
1.  Yes
2.  No
2
***********************
1.  MY CAMERA
2.  RENT A CAMERA
3.  VIEW ALL CAMERAS
4.  MY WALLET
5.  EXIT
***********************
Enter your choice: 5
Exiting the application... Goodbye!
```

Algorithm:

1. Start the application.
2. Initialize the camera list and wallet.
3. Prompt the user to log in with a username and password.
4. Validate the username and password. If invalid, go back to step 3.
5. If the login is successful, display the main menu with options.
6. Based on the user's input, perform the corresponding action:
    - If the user selects "MY CAMERA", go to step 7.
    - If the user selects "RENT A CAMERA", go to step 8.
    - If the user selects "VIEW ALL CAMERAS", go to step 9.
    - If the user selects "MY WALLET", go to step 10.
    - If the user selects "EXIT", end the application.
7. Handle the "MY CAMERA" functionality:
    - Display options: Add, Remove, View My Cameras, Go to Previous Menu.
    - Based on the user's input, perform the corresponding action:
    - If the user selects "ADD", go to step 11.
    - If the user selects "REMOVE", go to step 12.
    - If the user selects "VIEW MY CAMERAS", go to step 13.
    - If the user selects "GO TO PREVIOUS MENU", go back to step 6.
8. Handle the "RENT A CAMERA" functionality:
    - Display the list of available cameras.
    - Prompt the user to enter the camera ID to rent.
    - If the camera ID is valid, check if the camera is available and the user has sufficient balance in the wallet.
    - If both conditions are met, deduct the rental amount from the wallet, mark the camera as rented, and display a success message.
    - If any condition fails, display an error message.
    - Go back to step 6.

9. Handle the "VIEW ALL CAMERAS" functionality:

- Display the list of all cameras (rented and available).
- Go back to step 6.

10. Handle the "MY WALLET" functionality:

- Display the current wallet balance.
- Prompt the user if they want to deposit more money.
- If yes, prompt for the amount and deposit it into the wallet.
- Go back to step 6.

11. Handle the "ADD" functionality under "MY CAMERA":

- Prompt the user to enter the camera brand, model, and price per day.
- Create a new Camera object with an auto-incremented ID and the provided details.
- Add the camera to the camera list.
- Go back to step 7.

12. Handle the "REMOVE" functionality under "MY CAMERA":

- Display the list of cameras.
- Prompt the user to enter the camera ID to remove.
- Remove the camera with the corresponding ID from the camera list.
- Handle cases where an undesirable value for ID, i.e., other the integer is entered.
- Go back to step 7.

13. Handle the "VIEW MY CAMERAS" functionality under "MY CAMERA":

- Display the list of cameras that are added by the user.
- Go back to step 7.

Description for Source Code:

This Java source code represents a camera rental application. It consists of three classes: Camera, Wallet, and CameraRentalApp. The Camera class defines a camera object with properties such as brand, model, per day price, and rental status. The Wallet class manages the user's balance, allowing for deposits and withdrawals.

The main class, CameraRentalApp, provides a console-based interface for users to interact with the camera rental system. Users can log in, view a welcome screen, manage their cameras (add, remove, view), rent cameras, view all available cameras, manage their wallet (deposit), and exit the application. The application emphasizes a modular structure, encapsulation, and user-friendly menus.

Sprint Planning :

It took almost 9hrs to complete the task
for first 4 hrs the first option with 3 sub optuons i.e;ADD REMOVE VIEW CAmera
Next 5hrs for setting remaining options