Camera Rental Application Documentation

Table of Contents

1. Introduction

The Camera Rental Application is a Java-based command-line program designed to manage a camera rental service. It allows users to log in, view available cameras, rent cameras, manage their wallet balance, and perform various camera-related operations.

2. Features

2.1 User Authentication

- Users are required to log in using a username and password (default credentials: "admin" and "admin123").
- Three unsuccessful login attempts result in a temporary lockout.

2.2 Camera Management

- Users can view their rented cameras, add new cameras, and remove cameras from their inventory.
- Cameras are represented by the `Camera` class, which stores information such as ID, brand, model, rent per day, and status.

2.3 Wallet Management

- Users have a wallet balance that is deducted when renting a camera.
- The application prompts users to top up their wallet if the balance is insufficient.

2.4 Rental System

- Users can rent available cameras, and the application checks for availability and deducts the rent amount from the user's wallet.

2.5 Menu System

- The application provides a user-friendly menu system for easy navigation through different functionalities.

3. Usage

- Compile the Java source code using a Java compiler.
- Run the compiled program (`Camerarental.java`) to start the application.

- Follow the on-screen instructions to navigate through the application menus.

4. Code Structure

The code is organized into several key components:

- **Main Class (`Camerarental`):**
    - Manages user authentication, camera lists, wallet balance, and the main menu system.
- **Camera Class:**
    - Represents the properties of a camera, including ID, brand, model, rent per day, and status.
- **Display Functions:**
    - Includes functions for displaying the camera table, both for all cameras and only available cameras.
- **Wallet Management:**
    - Implements the functionality for topping up the user's wallet balance.

5. Security Considerations

- **User Authentication:**
    - The default credentials ("admin" and "admin123") are for demonstration purposes only. In a production environment, implement more secure authentication mechanisms.
- **Input Validation:**
    - While there is basic input validation, additional checks can be added to enhance security and prevent unexpected behavior.

6. Error Handling

- The application includes basic exception handling to catch input errors and provide informative messages to the user.

7. Future Improvements

- Enhance code readability by providing more meaningful variable names.
- Add comments to explain complex sections or logic.
- Implement a file-based system for persistently storing camera data.