Camera Rental application

This application handles user authentication, camera management, rental processes, and wallet transactions. Here is a brief overview of the code:

1. **Initialization:**
   * Scanner and application-specific objects (**LoginManager** and **CameraRentalApp**) are initialized.
   * The user is welcomed to the Camera Rental App.
2. **User Authentication:**
   * Users are prompted to log in with a username and password.
   * An authentication loop ensures valid credentials before proceeding.
3. **Main Menu:**
   * After successful authentication, the user is presented with a main menu.
   * Options include managing personal cameras, renting a camera, viewing all cameras, checking the wallet, and exiting the application.
4. **Switch Statement:**
   * A switch statement handles the user's menu choice.
   * Depending on the choice, the application performs various operations.
5. **Camera Management (Case 1):**
   * Sub-menu for managing personal cameras (add, remove, view).
   * User input is collected to perform relevant operations on the camera list.
6. **Rent a Camera (Case 2):**
   * Displays the available cameras and prompts the user to select one to rent.
   * Validates the camera ID and rents the selected camera.
7. **View All Cameras (Case 3):**
   * Displays a list of all available cameras.
8. **Wallet Management (Case 4):**
   * Displays the current wallet balance.
   * Allows the user to deposit more money into the wallet.
9. **Exit Application (Case 5):**
   * Exits the application with a closing message.

* InputMismatchException is caught to handle invalid input and prompts the user for a valid integer choice.