**Full Stack Java Developer**

Phase-End Project Problem Statement



**Phase-End Project**

**Camera Rental Application**

**Project Agenda:** Build a peer-to-peer camera rental application

**Scenario:**

You have been hired by a company called rentmycam.io as a Full Stack Developer with the aim to create a prototype of a camera rental application.

The tasks that need to be performed by you are:

* Create a specification document detailing the product’s capabilities, its appearance, and user interactions
* Document the application flow
* Detail the number and duration of sprints required
* Plan more than two sprints to complete the application
* Set up Git and a GitHub account to store and track enhancements of the prototype
* Implement Java concepts in the creation of this application
  + This includes data structures for sorting and searching.
* Include the following operations in the application:
* List a camera along with its brand, model, and per-day rental amount
* Select a camera from a listing to rent
* Add or view wallet amount
* Navigate from the current execution context to the main context
* Close the application
* Create a welcome screen with the application name and developer details
  + Include options that the user can interact with to access various features of the application

**Application features:**

* The user can add a new camera, along with its details, to the existing camera list.
* The application can display the list of cameras available for rent.
* The list will include details like the camera brand, model, and per-day rent amount.
* The user can select a camera to rent for a day.
* The user will only be allowed to rent the camera if there is sufficient balance in the user’s wallet to fulfil the per-day rent amount. If not, a message is returned.

**Insufficient wallet amount**

* If there is sufficient balance, the amount will be deducted from the wallet.
* If the list of cameras is empty, an appropriate message is displayed: **No Data Present at This Moment**.
* The user can add or view the balance in the wallet through a sub-menu.
* The user can deposit an amount to increase the wallet balance.
* A message is displayed to indicate the status of the operation.
* The application implements concepts like exceptions, collections, and sorting techniques for source code optimization and increased performance.

**Tools required:**

* Eclipse or IntelliJ
* Java
* Git
* GitHub
* Scrum
* Search and Sort techniques
* Open-source documentation tool (For example: Google Docs)

**Screenshot:**

1. **Log in**

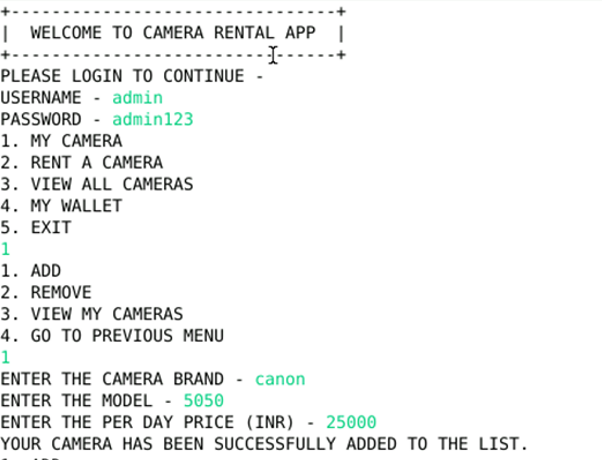
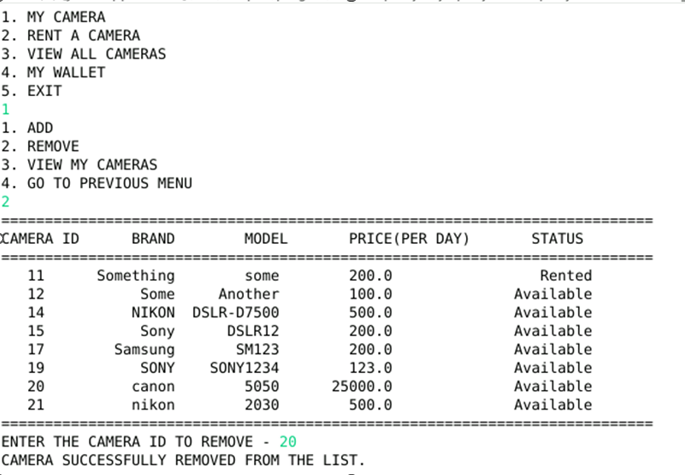
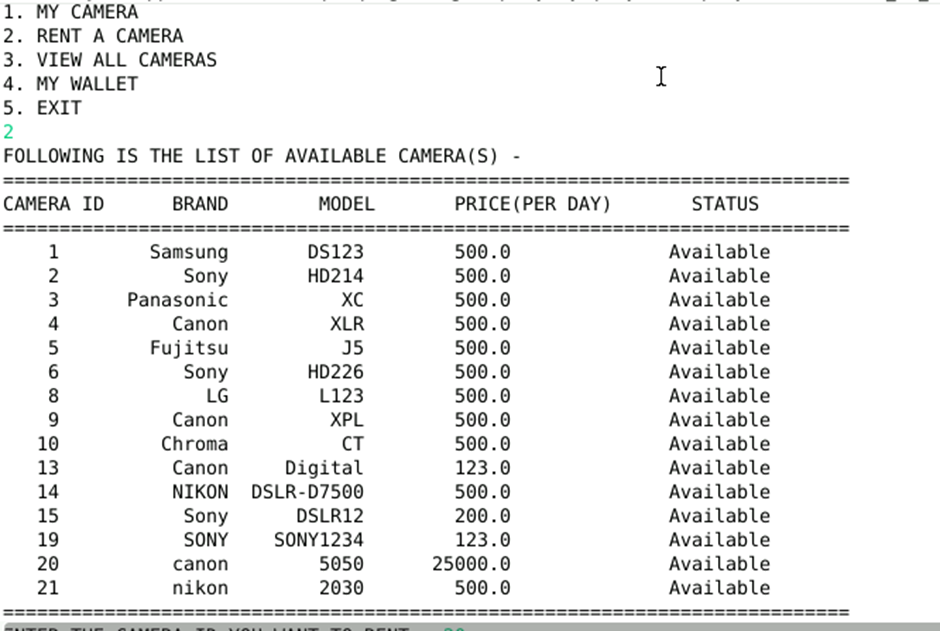
Graphical user interface, text, application

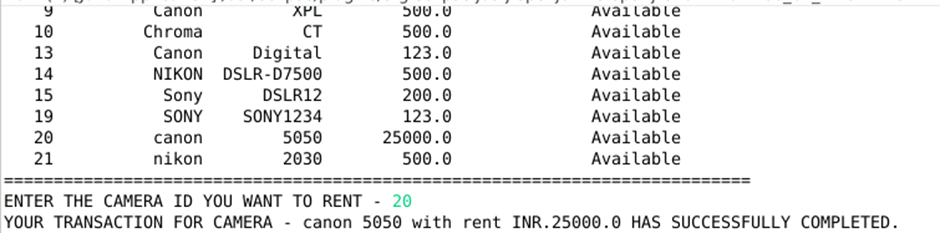
Description automatically generated

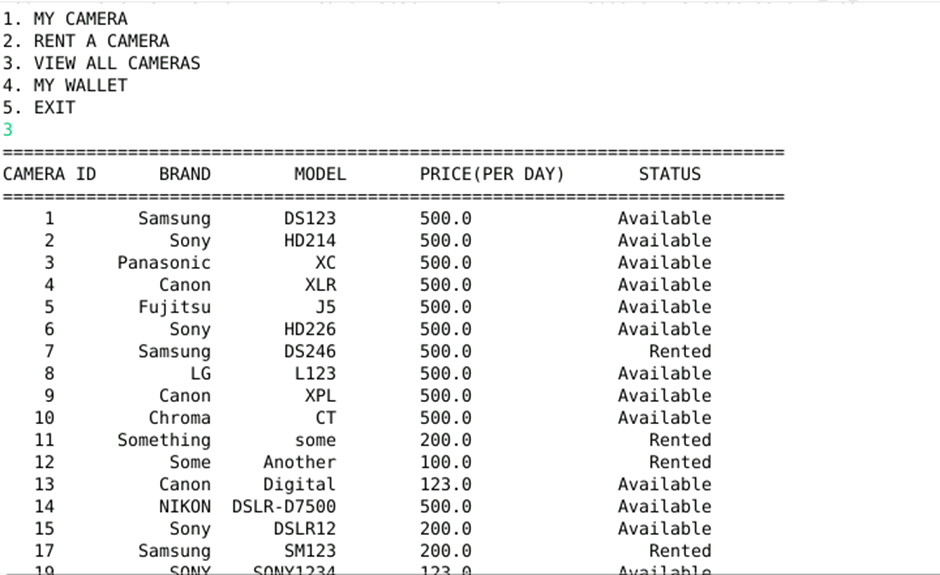
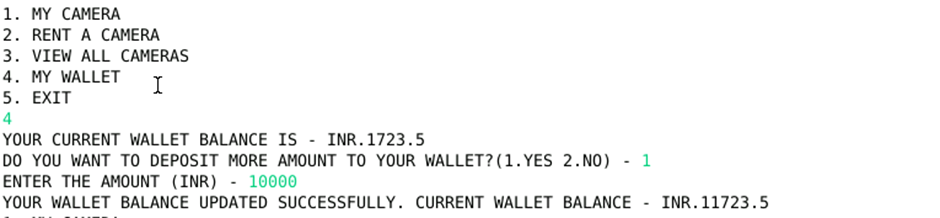
1. **Options to be displayed:**

Text, letter

Description automatically generated

1. **Add camera:**
2. **Remove camera:** 
3. **Rent a camera:**



1. **View all cameras:**
2. **My wallet or Add money:**
3. **Failed transaction:**

