**CAMERA RENTAL APPLICATION**

**Product Description:**

The focus of the project is to have a user view the cameras and rent the camera if he/she have the sufficient balance and can rent for a day. The user can also add amount to his wallet, to add or remove the cameras from the list and can end an application anytime.

**Capabilities:**

**Login:**

User can login with username and password. But for this project there is no user authentication as it is a prototype project. Any input will be taken as username and password.

**Camera Listing:**

Users can view the cameras from the list in the application and can use the camera id from the list for other functions of the application.

**Camera addition and removal:**

Users can add or remove the cameras from the list with the camera id given by the user.

**Rental Process:**

Users can initiate a rental request for a selected camera by camera id.

The application will handle payment processing.

**Wallet Management:**

Users can add amount to their wallet for rental transactions.

The wallet amount will be used to rent the cameras.

Users can view their wallet balance.

**Application Flow:**

The application will have a proper flow, guiding users through the rental process.

Each step of the process will be accompanied by appropriate instructions and error handling.

Appearance and User Interactions:

**Welcome Screen:**

The application will display app name with developer name.

Users can interact with various options on the welcome screen to access different features of the application.

**Main Menu:**

After authentication, users will be presented with a main menu.

The main menu will provide options to camera menu, rent a cameras, manage the wallet, and close the application.

**Listing Camera:**

Users can add a camera list with brand name, model, and rent amount. To remove a camera by camera id and view the camera list and have an option visit main menu

**Wallet Management:**

Users can add funds to their wallet by adding the amount to the wallet.

The application will display the current wallet balance.

Users can navigate back to the main menu from the wallet management screen.

**Application Exit:**

Users can choose to close the application, which will exit the program.

**Sprints and Duration:**

Based on the complexity of the application the project is done with sprints.

**Sprint 1:**

Set up Git and GitHub account for version control and collaboration.

Creating a class for Camera, CameraRentalApplication

Implement user registration and authentication.

Create the welcome screen and main menu.

**Sprint 2:**

Implementing the My camera and View camera functionality.

In My Camera menu adding methods for the options Add camera, remove camera, view camera list and navigate to previous menu.

**Sprint 3:**

Develop the rental process.

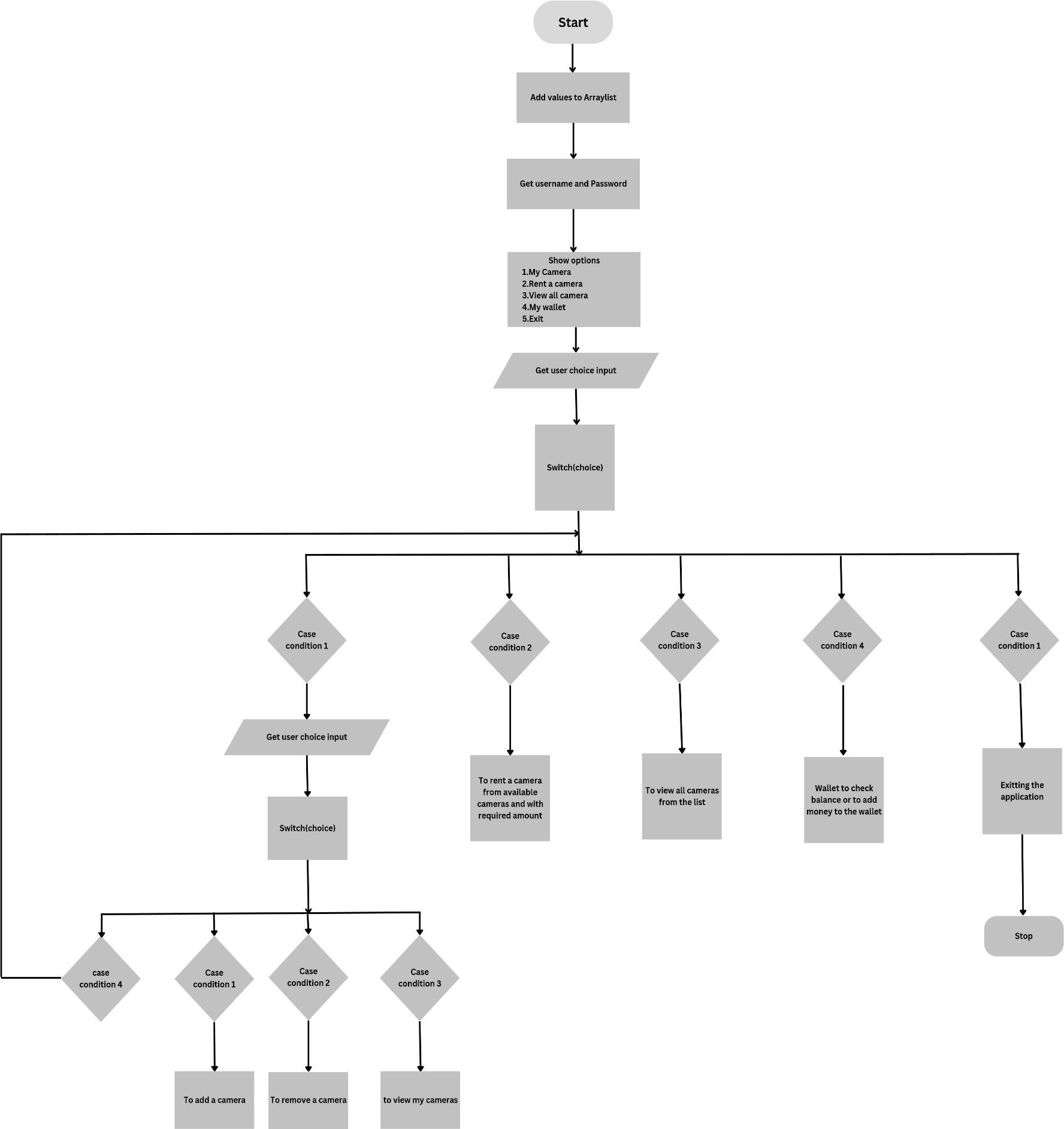
Implement wallet management features.

Add wallet amount management and transaction history.

**Sprint 4:**

Test and debug the application.

**Flow Chart:**



**Java Concepts and Data Structures:**

In the creation of this application, the following Java concepts and data structures can be utilized:

Object-oriented programming (OOP) principles for designing classes and their relationships.

Collections framework to store and manipulate camera data.

Searching algorithms to search for specific cameras.

**Tools required:**

* Eclipse
* Java
* Git
* GitHub
* Scrum
* Search techniques

**Github link:**

https://github.com/SridharA-5660/Java-FSD.git