Name: Bhumika R

Employee ID: 2580388

Camera Rental Application

Problem Statement: Build a peer-to-peer camera rental application (Java).

Sprint Planning:

Sprint1 (week1)

- **Front End Development**

Day1- Project Setup and Planning

- Setting the development environment
- Project Scope & Objectives
- Requirements Analysis

Day 2-3 Design

- Design the Application
- Create the mock-ups
- Layout the structure

Day4-5 Implementation

- Front End Implementation
- Main Layouts
- Create Lists

Day 6-7 Authentication

- Create the user Registration
- Set the authentication.

Sprint 2(week2)

- **Back End**

Day 8 Database Setup

- Set database to store camera information
- Define schemas

Day9-10 Camera Management

- Functionalities of all cameras
- View Available cameras

Day 11-12 Wallet

- Create wallet
- Functionalities for the wallet
- Balance Check up

Day 13-14 Renting of camera

- Functionalities of Rent
- Update camera Details when rented

Algorithm

1. Initialize the application

2. User Authentication

3. Main Menu

4. Display options

    - Add a camera
    - Remove a camera
    - View all Cameras
    - Rent a camera
    - My wallet
    - Exit

5. User selects the option

6. Based on selection :

    Option a Add

        - Prompt the user to enter details for camera(model,Name,price)
        - Generate a ID for camera
        - Add Cameras to all my cameras list

    Option b Remove

        - Display all the cameras
        - Prompt the user to select which camera to be removed

Option c View all Cameras

- View all cameras in the list

Option d Rent a camera

- Display all the cameras in the list available with status
- Prompt the user to select the Camera Id
- Check the Amount (balance in the Wallet)
- If balance is sufficient then point the camera as rented and detect the amount from the wallet

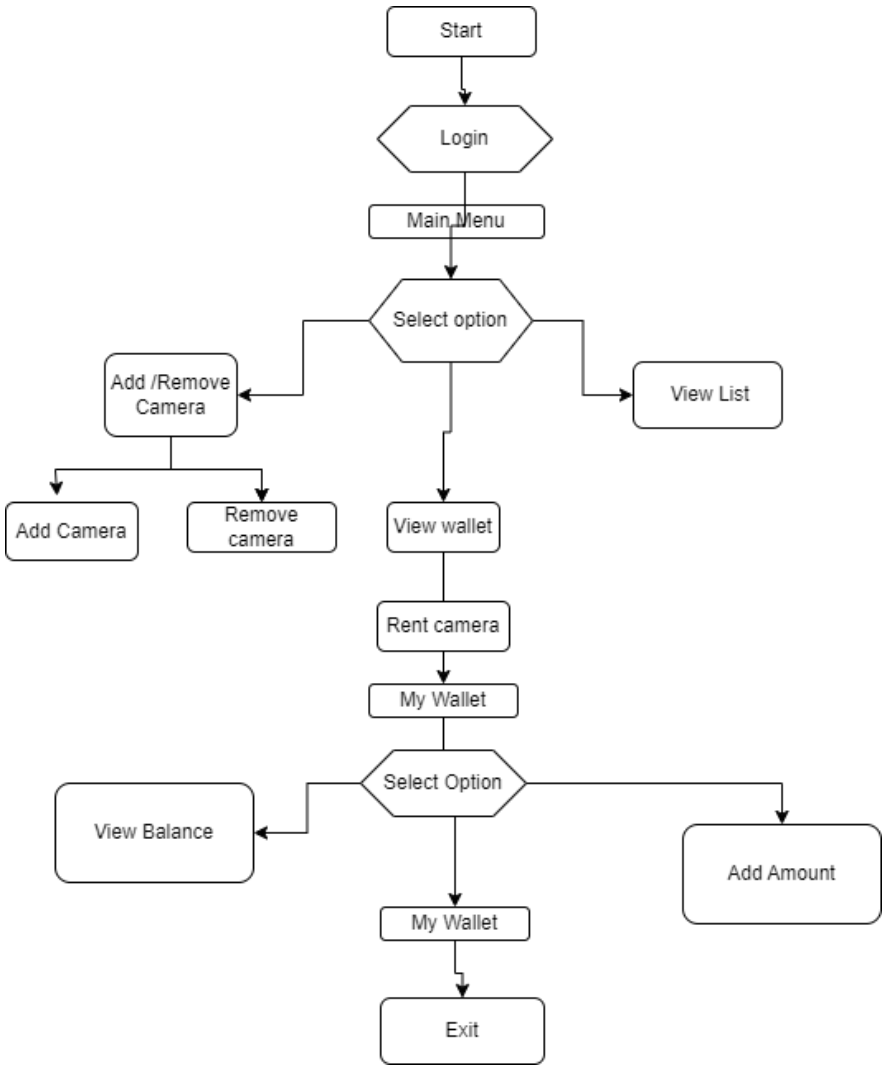
Option e My Wallet

- Display Balance
- If balance is not sufficient prompt the user to add amount to the wallet

Option f Exit

- Terminate from the application

Flow Chart



Java Concepts Used

- Classes and Objects
- Encapsulation
- Constructor
- Inheritance
- Data Collection
- Method Overloading
- Getter and Setter Methods

Project Flow

Start

Login--->Login Success..?

--No—(Login)

--Yes—(Main Menu)

Main Menu

User Choice … (1) Add Camera

(2) Remove Camera

(3) View Camera

(4) Rent a Camera

(5) My Wallet

(6) Exit

(Add/Remove Camera)

(1) Add Camera
Input Details
Camera added-----yes -----Main menu
-----NO-----Add

(2) Remove Camera
Select camera
If camera rented—yes—cannot remove
---No---Can Remove

(3) View Camera
Display all Cameras

(4) Rent camera

Available-----yes----select camera

------NO----No camera to rent

(Select camera)

Wallet sufficient---yes---Rent

---No—(Insufficient Balance)

(5) My Wallet

Display Balance

If Insufficient----Add amount

(6) Exit

Terminate the application