Experiment 1.2

Student Name: Anushka Kotiyal	**UID: 22BCS13559**
Branch: CSE	**Section: 22BCS_KRG_IOT_3B**
Semester: 6th	**Date of Performance: 14/01/25**
Subject: Project Based Learning In Java	**Subject Code: 22CSH-359**

Aim: Design and implement a simple inventory control system for a small video rental store

Objective: To design and implement a user-friendly inventory control system for a small video rental store, enabling efficient management of video inventory, including functionalities for adding, renting, and returning videos.

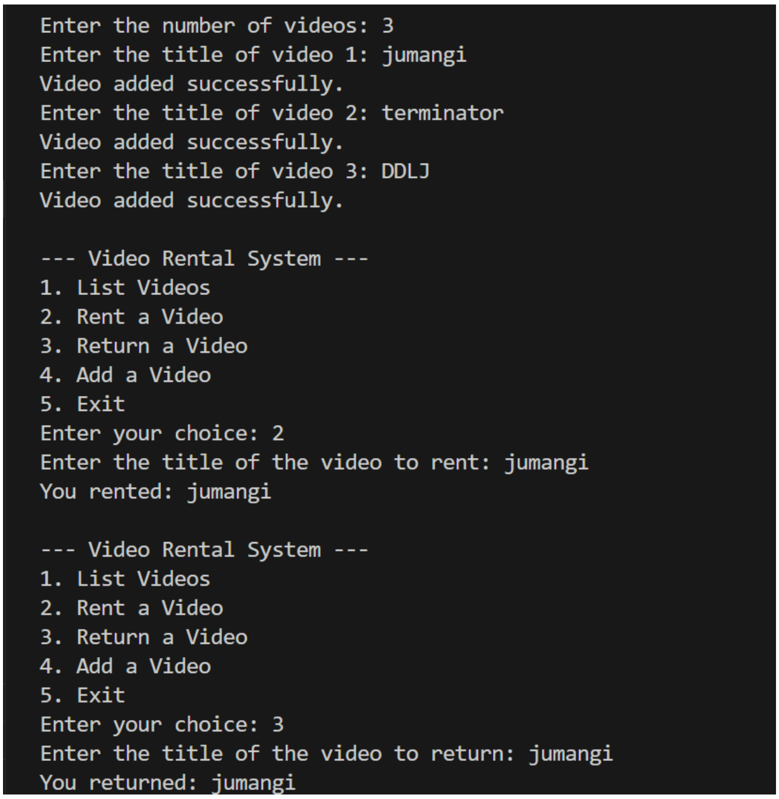
Algorithm:

- **Define Classes:**
  - **Video**: To represent each video, with attributes such as video ID, title, genre, and availability status.
  - **Inventory**: To manage the list of videos, including adding and removing videos from the inventory.
  - **Customer**: To represent customers, with attributes such as customer ID, name, and rented videos.
  - **RentalSystem**: To control the process of renting and returning videos.
- **Video Class:**
  - Define the video with attributes such as `videoID`, `title`, `genre`, and `isAvailable`.
  - Define methods to mark the video as rented and returned.
- **Inventory Class:**
  - Maintain a list of videos (`ArrayList<Video>`).
  - Implement methods to add new videos, display available videos, and check if a video is available.
- **Customer Class:**
  - Define a list to store rented videos.
  - Implement methods to rent a video (if available) and return it.
- **RentalSystem Class:**
  - Handle the main functionality: list available videos, allow customers to rent and return videos, and display the inventory status.

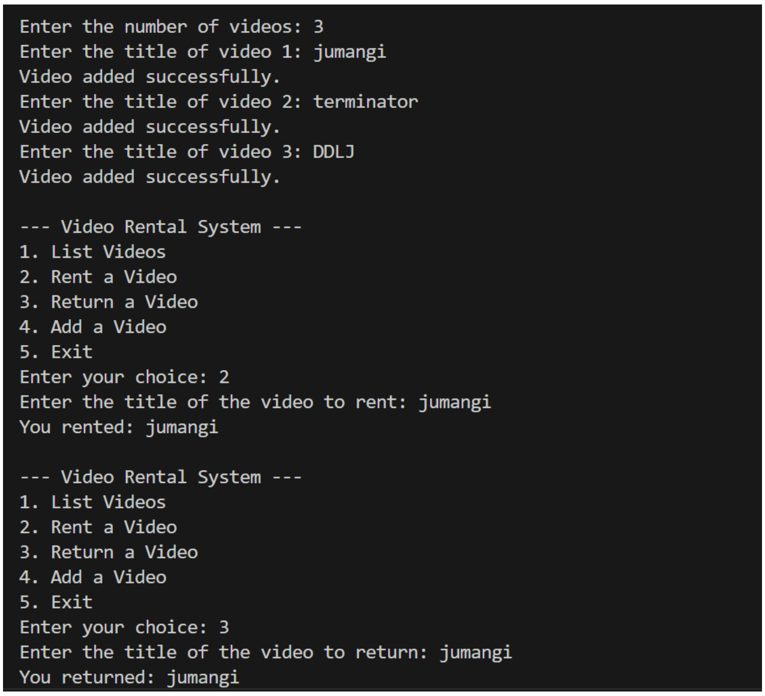
Code:

```
import java.util.*;
class Video {
    private String title;
    private boolean flag;
    private int rating;
    private int total = 0;
    public Video(String title) {
        this.title = title;
        this.flag = true;
        this.rating = 0;
    }
    public String getTitle() {
        return title;
    }
    public boolean isAvailable() {
        return flag;
    }
    public void rent() {
        if (flag) {
            flag = false;
            System.out.println("You rented: " + title);
        } else {
            System.out.println("Error: Video is already rented out.");
        }
    }
    public void returnVideo() {
        if (!flag) {
            flag = true;
            System.out.println("You returned: " + title);
        } else {
            System.out.println("Error: Video was not rented.");
        }
    }
    public void addRating(int rating) {
        if (rating < 1 || rating > 5) {
            System.out.println("Rating should be between 1-5");
        } else {
            total += rating;
            this.rating++;
            System.out.println("Thank you for rating!");
        }
    }
    public void display() {
        System.out.println("Title: " + title + " | Available: " + (flag ? "Yes" : "No") + " | Rating: " + (rating == 0 ? 0 : total / rating));
    }
}
class VideoStore {
    List<Video> videos = new ArrayList<>();
    public void addVideo(String title) {

        for (Video v : videos) {
            if (v.getTitle().equals(title)) {
                System.out.println("Already Available");
                return;
            }
        }

        videos.add(new Video(title));
        System.out.println("Video added successfully.");
    }

    public void avgRating(String title, int rating) {
        boolean rated = false;
        for (Video v : videos) {
            if (v.getTitle().equals(title)) {
                v.addRating(rating);
                rated = true;
                break;
            }
        }
        if (!rated) {
            System.out.println("No such video found.");
        }
    }

    public void listVideos() {
        if (videos.isEmpty()) {
            System.out.println("No videos available.");
        } else {
            for (Video v : videos) {
                v.display();
            }
        }
    }

    public void rentVideo(String title) {
        for (Video v : videos) {
            if (v.getTitle().equals(title)) {
                v.rent();
                return;
            }
        }
        System.out.println("Error: Video not found.");
    }
    public void returnVideo(String title) {
        for (Video v : videos) {
            if (v.getTitle().equals(title)) {
                v.returnVideo();
                return;
            }
        }
        System.out.println("Error: Video not found.");
    }
}

public class Project {
    public static void main(String[] args) {
        VideoStore store = new VideoStore();
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter the number of videos: ");
        int n = sc.nextInt();

        for (int i = 0; i < n; i++) {
            System.out.print("Enter the title of video " + (i + 1) + ": ");
            String title = sc.next();
            store.addVideo(title);
        }

        while (true) {
            System.out.println("\n--- Video Rental System ---");
            System.out.println("1. List Videos");
            System.out.println("2. Rent a Video");
            System.out.println("3. Return a Video");
            System.out.println("4. Add a Video");
            System.out.println("5. Exit");
            System.out.print("Enter your choice: ");

            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    store.listVideos();
                    break;
                case 2:
                    System.out.print("Enter the title of the video to rent: ");
                    String rentTitle = sc.next();
                    store.rentVideo(rentTitle);
                    break;
                case 3:
                    System.out.print("Enter the title of the video to return: ");
                    String returnTitle = sc.next();
                    store.returnVideo(returnTitle);
                    System.out.print("Enter the rating (1-5): ");
                    int rating = sc.nextInt();
                    store.avgRating(returnTitle, rating);
                    break;
                case 4:
                    System.out.print("Enter the title of the video to add: ");
                    String newTitle = sc.next();
                    store.addVideo(newTitle);
                    break;
                case 5:
                    System.out.println("Goodbye...");
                    sc.close();
                    return;
                default:
                    System.out.println("Invalid choice. Please try again.");
            }
        }
    }
}
```

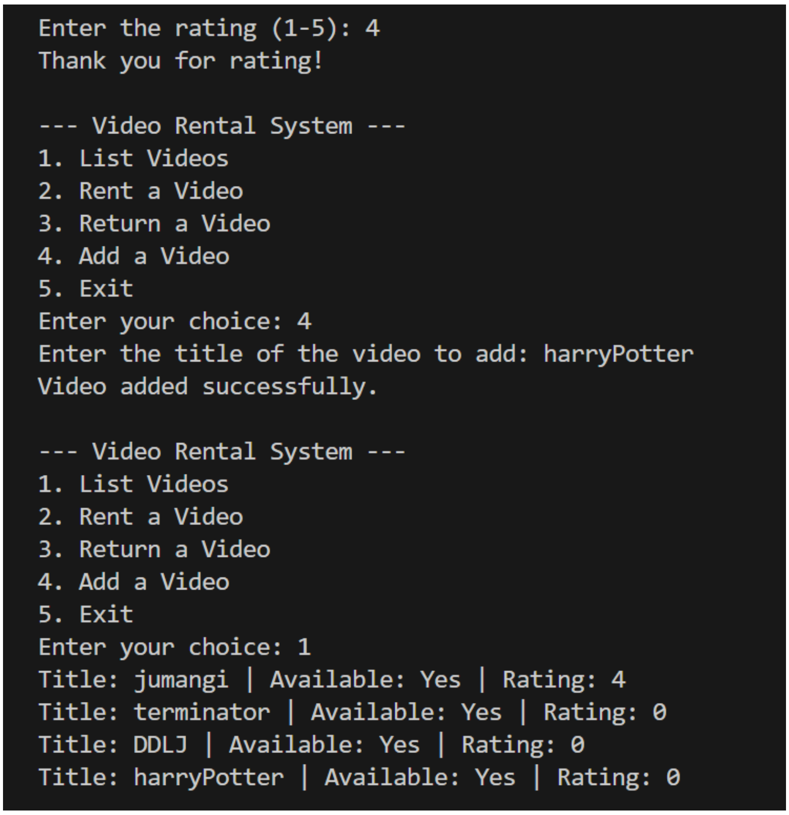
Output:



```
Enter the number of videos: 3
Enter the title of video 1: jumangi
Video added successfully.
Enter the title of video 2: terminator
Video added successfully.
Enter the title of video 3: DDLJ
Video added successfully.

--- Video Rental System ---
1. List Videos
2. Rent a Video
3. Return a Video
4. Add a Video
5. Exit
Enter your choice: 2
Enter the title of the video to rent: jumangi
You rented: jumangi

--- Video Rental System ---
1. List Videos
2. Rent a Video
3. Return a Video
4. Add a Video
5. Exit
Enter your choice: 3
Enter the title of the video to return: jumangi
You returned: jumangi
```



```
Enter the number of videos: 3
Enter the title of video 1: jumangi
Video added successfully.
Enter the title of video 2: terminator
Video added successfully.
Enter the title of video 3: DDLJ
Video added successfully.

--- Video Rental System ---
1. List Videos
2. Rent a Video
3. Return a Video
4. Add a Video
5. Exit
Enter your choice: 2
Enter the title of the video to rent: jumangi
You rented: jumangi

--- Video Rental System ---
1. List Videos
2. Rent a Video
3. Return a Video
4. Add a Video
5. Exit
Enter your choice: 3
Enter the title of the video to return: jumangi
You returned: jumangi
```



```
Enter the rating (1-5): 4
Thank you for rating!

--- Video Rental System ---
1. List Videos
2. Rent a Video
3. Return a Video
4. Add a Video
5. Exit
Enter your choice: 4
Enter the title of the video to add: harryPotter
Video added successfully.

--- Video Rental System ---
1. List Videos
2. Rent a Video
3. Return a Video
4. Add a Video
5. Exit
Enter your choice: 1
Title: jumangi | Available: Yes | Rating: 4
Title: terminator | Available: Yes | Rating: 0
Title: DDLJ | Available: Yes | Rating: 0
Title: harryPotter | Available: Yes | Rating: 0
```

Learning Outcomes:

- **Object-Oriented Design:** Learn to create and use classes for real-world entities.
- **Core Programming Skills:** Practice loops, conditionals, and methods for inventory operations.
- **Data Structure Usage:** Use `ArrayList` to manage dynamic data effectively.
- **User-Friendly Systems:** Design intuitive interfaces and handle errors smoothly.