**EXPERIMENT-1.2**

|  |  |
| --- | --- |
| **Student Name**: SANSKAR AGRAWAL | **UID:** 21BCS5914 |
| **Branch**: CSE | **Section/Group:** 806/B |
| **Semester**: 5th | **Date of Performance**: 24/08/2022 |
| **Subject Name**: PBLJ Lab | **Subject Code:** 20CSP-321 |

1. **Aim/Overview of the practical:**

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

1. **Task to be done/ Which logistics used:**

Eclipse IDE (Java)

**3. Programming:-**

**package** pblj\_lab;

**import** java.util.\*;

**class** Video

{

String title; **boolean**

Flag = **false**;

**int** avg = 0;

}

**class** VideoStore

{

**private** **static** **final** Scanner ***input*** = **new** Scanner(System.***in***);

String chek2;

Video beat[] = **new** Video[10];

**int** num\_video;

**void** addVideo() {

System.***out***.println("Enter " + num\_video + " Video Title:- ");

**for** (**int** i =0; i < num\_video; i++) {

beat[i] = **new** Video(); beat[i].title =***input***.nextLine();

}

System.***out***.println("Enter " + num\_video + " Video rating between 1 to 5:- ");

**for**(**int** i = 0; i <num\_video; i++) {

beat[i].avg= ***input***.nextInt();

}

}

**int** chekOut(**int** k) {

String chek1; System.***out***.println("chekout " +(k + 1)); chek1 = ***input***.next();

**for** (**int** i = 0; i < num\_video; i++) {

**if** (beat[i].title.equals(chek1) && (beat[i].Flag == **false**))

{ beat[i].Flag = **true**;

**return** -1;

}

**else** **if** (beat[i].title.equals(chek1) && (beat[i].Flag == **true**))

{

System.***out***.println("Failed to chekout: ");

**return** -1;

}

}

**return** 1;

}

**int** returnvideo(**int** k) {

System.***out***.println("Returning Video name: " + (k + 1));

chek2 =***input***.next();

**for** (**int** i = 0; i < num\_video; i++){

**if** (beat[i].title.equals(chek2) && beat[i].Flag == **true**) {

System.***out***.println("Video" + chek2 + " is returned");

**this**.reciveRating();

beat[i].Flag = **false**;

**return** -1;

}

**else** **if** (beat[i].title.equals(chek2) && beat[i].Flag == **false**) {

System.***out***.println("U cannot return this!");

**return** -1;

}

}

**return** 1;

}

**void** reciveRating() {

System.***out***.println("Enter the rating between 1 to 5: ");

**for** (**int** i =0; i < num\_video; i++) {

**if** (beat[i].title.equals(chek2) && beat[i].Flag == **true**) {

beat[i].avg = ***input***.nextInt();

}

}

}

**void** listInventory()

{

System.***out***.println("List of all Videos: ");

**int** total = 0;

**for** (**int** i = 0; i < num\_video; i++) {

**if**(beat[i].Flag == **false**) {

System.***out***.println(beat[i].title + " Not chekout");

}

**else** {

System.***out***.println(beat[i].title + " Chekout!"); total += 1;

}

**if** (beat[i].avg != 0) {

System.***out***.println("Rating:- " + beat[i].avg + " Star ");

}

}

System.***out***.println("Total number of chekout video: " + total);

}

}

**public** **class** VideoStoreLauncher {

**public** **static** **void** main(String[] args) {

VideoStore box = **new** VideoStore();

**int** chekout;

**int** ret;

Scanner in = **new** Scanner(System.***in***);

System.***out***.println("SANSKAR AGRAWAL UID-20BCS5914 ");

System.***out***.println("Number of video: ");

box.num\_video = in.nextInt(); box.addVideo();

System.***out***.println("How Many video u wants to checkout: 0 if u don't");

chekout = in.nextInt();

**int** chek = 1;

**int** chek1 = 1;

**if** (chekout != 0) {

**for** (**int** i = 0; i < chekout; i++) {

chek= box.chekOut(i);

**if** (chek == 1) {

System.***out***.println("Video Not Present");

}

}

}

System.***out***.println("How Many video u wants to Return: 0 if u don't");

ret =in.nextInt();

**if** (ret != 0) {

**for** (**int** i = 0; i < ret; i++) {

chek1 = box.returnvideo(i);

**if** (chek1 == 1) {

System.***out***.println("Worng input!");

}

}

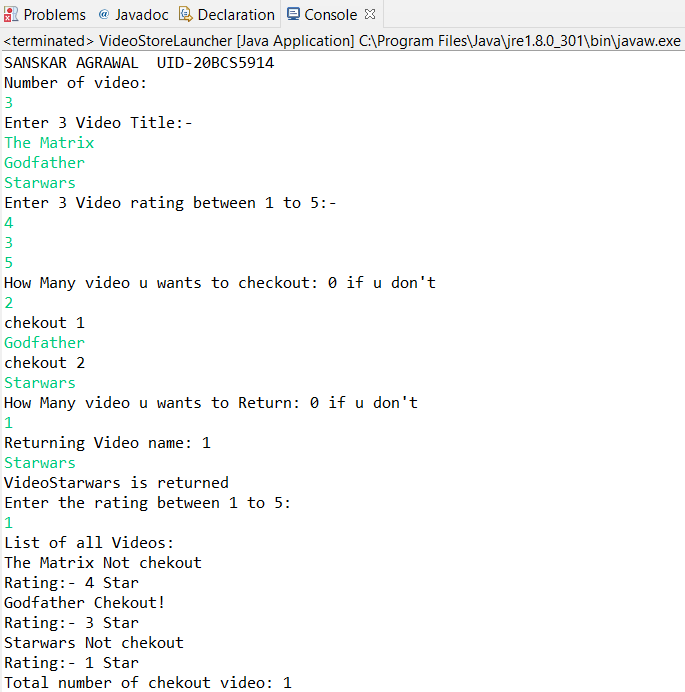
}

box.listInventory();

}

}

**5. Output:-**



**Learning Outcomes(What I have learnt):-**

1. To create a inventory control system in Java.
2. To generate analytical and conceptual ability related to fundaments of Java.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |