|  |
| --- |
|  |

Ticketing system

Ass1B00107874

[David Mulhall] | [Software development] | [24/10/2017]

# Assignment Brief

Create a text based ticketing system that must take as inputs the Surname, Name of the Show and the number of people in the Party, the system must also take into account the number of members in a party and apply a discount to the stated ticket price should a party contain over 6 patrons.

Next the system will output a "Ticket" to the user that contains all the relevant information.

Finally, the system must ask the user if they want more tickets and if so start the booking procedure again.

# Pseudocode

* Import libraries
* Start more tickets loop
* Initialize variables
* Print first message to the screen and ask user for input (i.e. Name)
* Display list of available movies on screen and ask for user input
* Using a switch statement to determine movie choice
* Print third message to the screen and ask for user input (i.e. Number of patrons)
* Check whether the number of patron's is more than 6, if so set the ticket price to €22.50 otherwise set ticket price to €25.00
* Work out total cost by multiplying the ticket price by the number of patrons
* Print ticket to the screen
* Ask user if they wish to book more tickets
* Increase loop counter by 1 and return to start of loop. If not exit program with farewell message.
* Source Code

// Import required libraries

import java.util.Scanner;

import java.io.IOException;

import java.text.DecimalFormat;

// Declare class

class Ass1B00107874

{

public static void main(String [] args)throws IOException, InterruptedException

{

// Initialise transaction loop counter

int tickets = 1;

// Begin main loop

while (true){

// Create and initialise variables

clearScreen(); // call clear screen method

Scanner myinput = new Scanner(System.in);

String name = "";

String show = "";

String choice = "";

String more = "";

int guests = 0;

int loop = 0;

double price = 0.00;

double total = 0.00;

// Load first screen to user

clearScreen(); // call clear screen method

banner(); // call banner method

System.out.println(" Welcome to Ticketmaster");

System.out.println("");

System.out.println("Please enter your surname: ");

System.out.println("");

name = myinput.nextLine(); // Set name variable to next full line input

// Load second screen to user and begin error catching in loop

while(loop == 0)

{

clearScreen(); // call clear screen method

banner(); // call banner method

System.out.println("");

System.out.println("What show would you like to see? ");

System.out.println("");

System.out.println("");

System.out.println(" 1. Star Wars 2. Guardians of the Galaxy");

System.out.println("");

System.out.println(" 3. Hackers 4. Legally Blonde");

System.out.println("");

choice = myinput.next(); // Set movie choice variable to next input

// Use switch statement to set show variable and set loop exit condition to move to next section

switch(choice)

{

case "1": show = "Star wars";loop = 1;

break;

case "2": show = "Guardians of the Galaxy";loop = 1;

break;

case "3": show = "Hackers";loop = 1;

break;

case "4": show = "Legally Blonde";loop = 1;

break;

default: System.out.println("Please enter number of your movie choice ");

Thread.sleep(1000); // starts timer to wait 1 second before reloading page

}

}

// Load third screen to user and begin error catching in loop

while (loop == 1)

{

clearScreen(); // call clear screen method

banner(); // call banner method

System.out.println("");

System.out.println("How many people in your party? ");

System.out.println("");

if(myinput.hasNextInt()) // checks to make sure input is of type Int

{

guests = myinput.nextInt(); // Sets guests variable to be the next input

if(guests != 0) // Check to make sure that amount of tickets is more than 0

}

loop = 0; // set loop exit condition

}

}

else

{

myinput.next(); // looking for user to reinput

System.out.println("Please enter a valid number of tickets!");

Thread.sleep(1000); // starts timer to wait 1 second before reloading page

}

}

// Once out of error catching loop set price of ticket based on the number of guests

// Apply discounted price if more than 6 guests

If (guests>6)

{

price=22.50;

}

else

{

price=25.00;

}

// Set total variable to be displayed as equal to price times number of guests

total = (guests\*price);

clearScreen(); // call clear screen method

// call print ticket method and forward variables as method parameters

printTicket(name,show,guests,total);

more = myinput.next(); // waiting for user input to for more tickets

// Error catch loop looks to make sure that the user enters ONLY a Y or N.

// While not Y or N loop is maintained

while (!"Y".equalsIgnoreCase(more) && !"N".equalsIgnoreCase(more))

{

System.out.println("\nInvalid response. Try again.");

more = myinput.next();

}

// once loop exit condition is met user is prompted with the total number of transactions and system exits

if ("n".equalsIgnoreCase(more))

{

System.out.println(" Total number of transactions this session" +tickets);

System.out.println("");

System.out.println(" Have a great day!");

break; // system exit is made.

}

tickets++; // increase ticket count by 1

}

}

//Accesses system environment to clear the screen

public static void clearScreen()throws IOException, InterruptedException

{

new ProcessBuilder("cmd", "/c", "cls").inheritIO().start().waitFor();

}

public static void banner()

{

// Outputs ASCII art that reads "TICKETMASTER" System.out.println("##################################################################################");

System.out.println("### \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_ \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ###");

System.out.println("### /\_ \_\_/ \_/ \_\_\_\_/ //\_// \_\_\_\_/\_ \_\_/ / |/ / | / \_\_\_/\_ \_\_/ \_\_\_\_/ \_\_ ) ###");

System.out.println("### / / / // / / ,< / \_\_/ / / / /|\_/ / /| | )\_\_ ) / / / \_\_/ / /\_/ / ###");

System.out.println("### / / \_/ // /\_\_\_/ /| |/ /\_\_\_ / / / / / / \_\_\_ |\_\_\_/ // / / /\_\_\_/ \_, \_/ ###");

System.out.println("### /\_/ /\_\_\_/)\_\_\_\_/\_/ |\_/\_\_\_\_\_/ /\_/ /\_/ /\_/\_/ |\_/\_\_\_\_//\_/ /\_\_\_\_\_/\_/ |\_| ###");

System.out.println("### ###");

System.out.println("### (c) 1983 ###");

System.out.println("##################################################################################");

}

//Takes parameters sent from main method and uses them to populate "Ticket"

public static void printTicket(String info1, String info2, int info3, double info4)throws IOException, InterruptedException{

//Sets the format that will be used when displaying currency

DecimalFormat df = new DecimalFormat("¤¤ 0.00");

banner();

System.out.println(" Proudly presents");

System.out.println("");

System.out.println("");

System.out.println("Show : "+info2);

System.out.println("");

System.out.println("Surname : "+info1);

System.out.println("");

System.out.println("No. in party : "+info3);

System.out.println("");

System.out.println("Price : "+df.format(info4));

System.out.println("")

System.out.println("");

System.out.println("#############################################");

System.out.println(" TicketMaster retains the right to refuse admission");

System.out.println("");

System.out.println("");

System.out.println("");

System.out.println(" Would you like to purchase more tickets?");

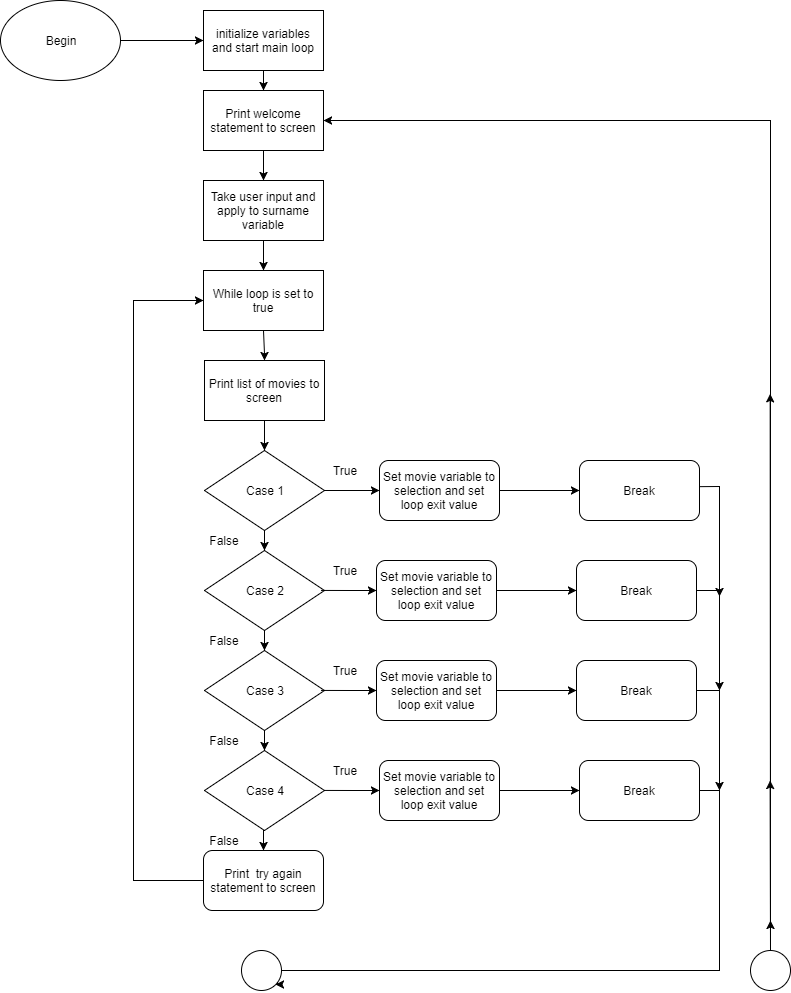
System.out.println(" Y/N");

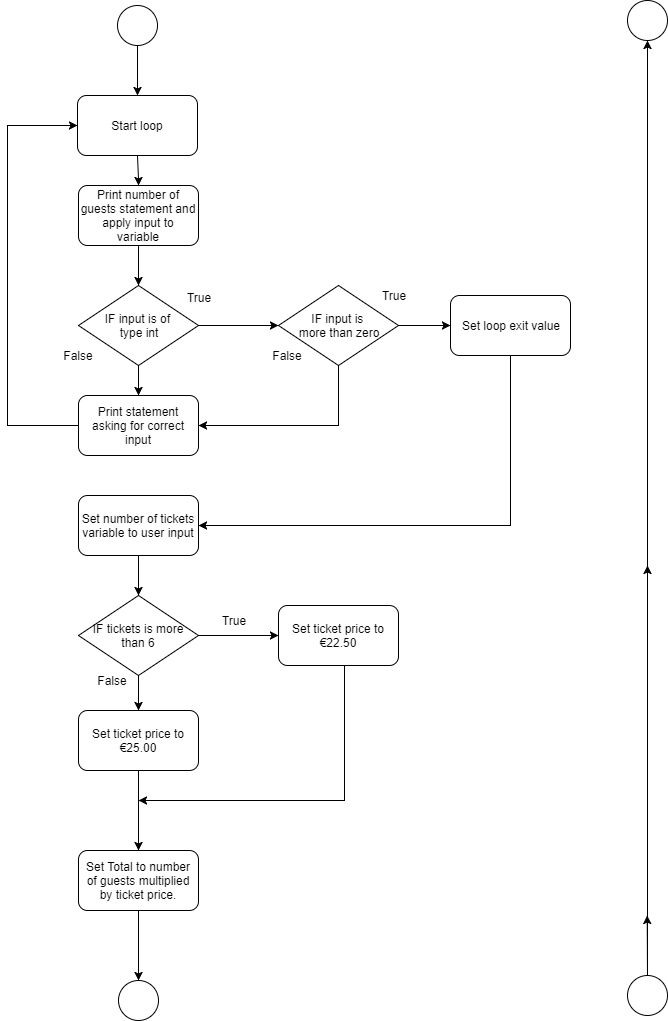
System.out.println("");

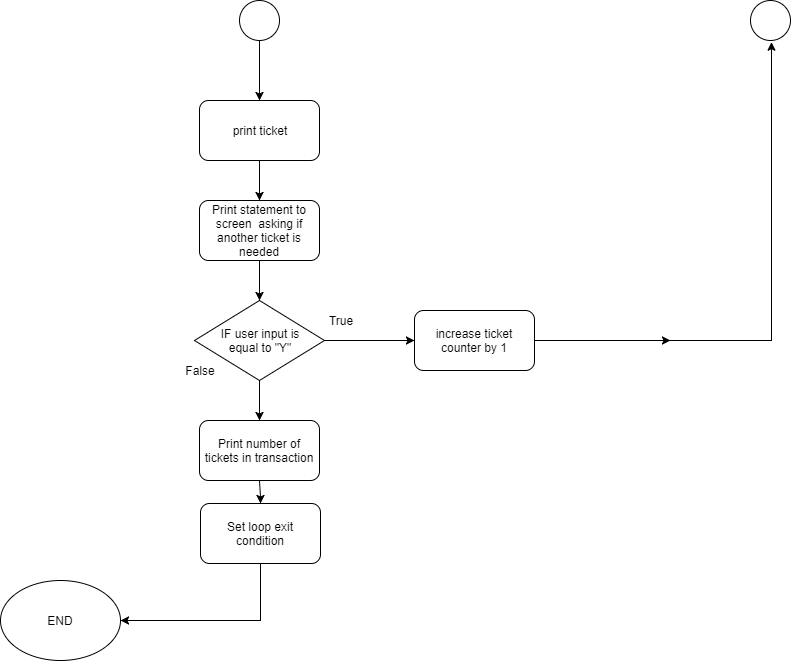
}

}.

Flow Chart







Test Cases

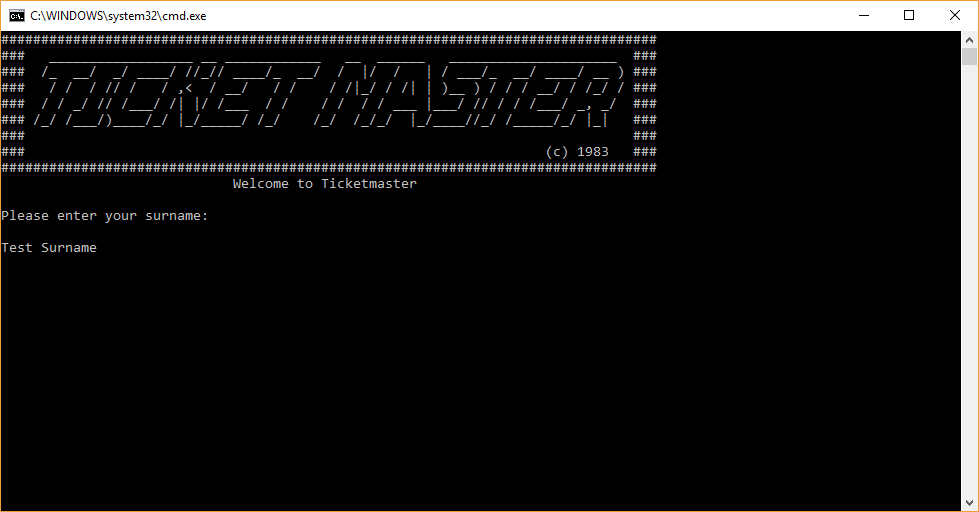
Listed below is a series of tests that will be ran in order to verify that all aspects of the program are working correctly and that any potential errors or failures are identified and rectified. Aspects tested will be :

1. Check that names with spaces or double barrel names can be used
2. Check that only a number and not letters can be used to choose movie
3. Check that the number of patrons is a number and not zero
4. Check that the price is set correctly if more than 6 patrons purchase tickets
5. Check to make sure that only "Y" or "N" are accepted for more tickets question
6. Check Ticket counter is working

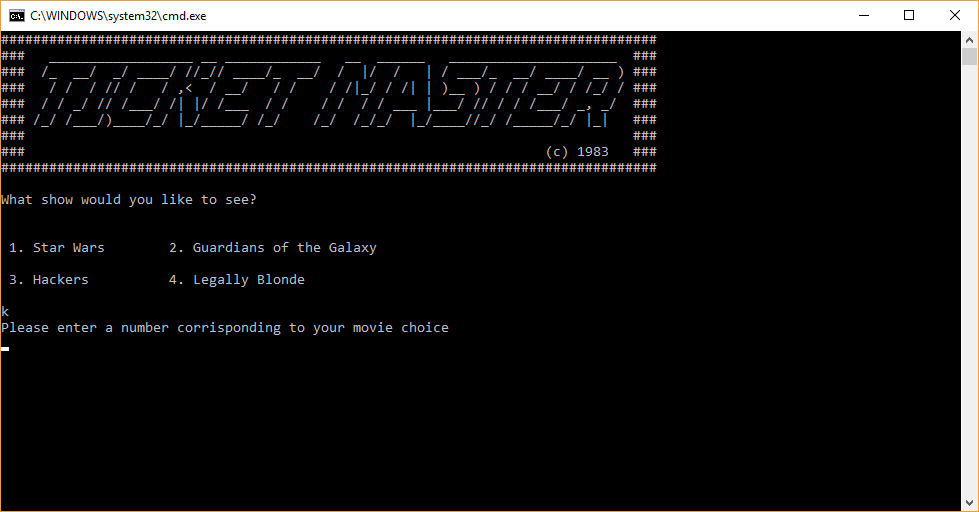
|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Expected  result | Actual  result |
| 1 | Check that names with spaces or double barrel names can be used | Message should display "Test Surname" | "Test Surname"  *\*Screen shot 1* |
|  |  |  |  |
| 2 | Check that only a number and not letters can be used to choose movie | Typing a character will result in a brief error text and then reload the screen | Error shown as expected  *\*Screenshot 2* |
|  |  |  |  |
| 3 | Check that the number of patrons, must be a number and not zero | Zero and letters will cause a brief error message to appear and then the screen will reload | Error shown as expected  *\*Screenshot 3* |
|  |  |  |  |
| 4 | Check that the price is set correctly if more than 6 patrons purchase tickets | 6 tickets will cost €150 and 7 tickets will cost €157.50 | Success!  *\*Screenshot 4*  *\*Screenshot 5* |
| 5 | Check to make sure that only "Y" or "N" are accepted for more tickets question | Numbers or other letters will prompt an error message to briefly appear | Only Y and N are accepted  *\*Screenshot 6* |
|  |  |  |  |
| 6 | Check Ticket counter is working | Ticket counter should show the number of completed ticket purchase cycles | Success!  *\*Screenshot 7* |
|  |  |  |  |

Screen Shots

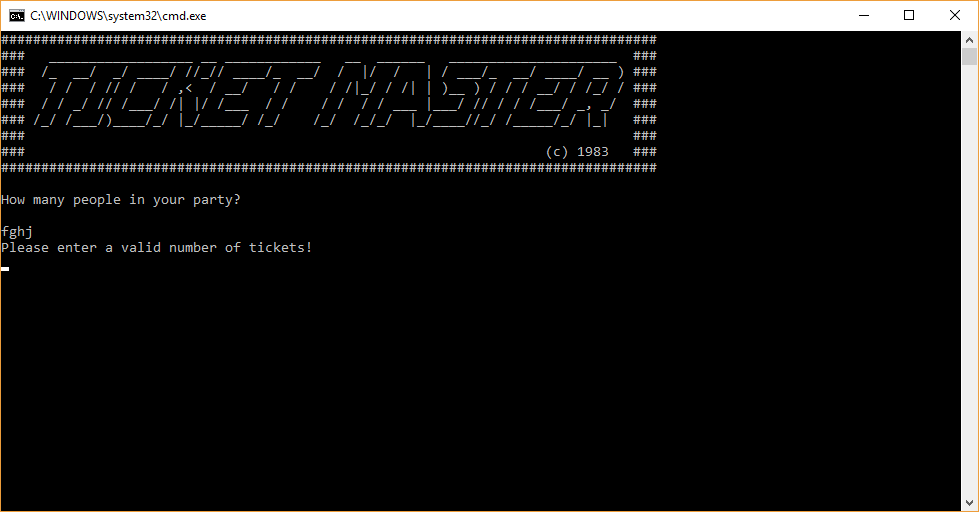
Screen #1



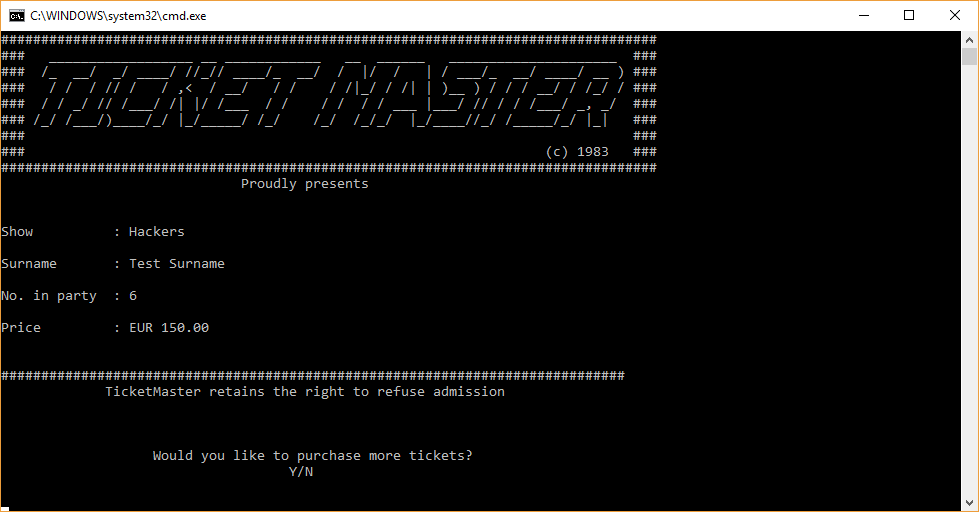
Screen#2



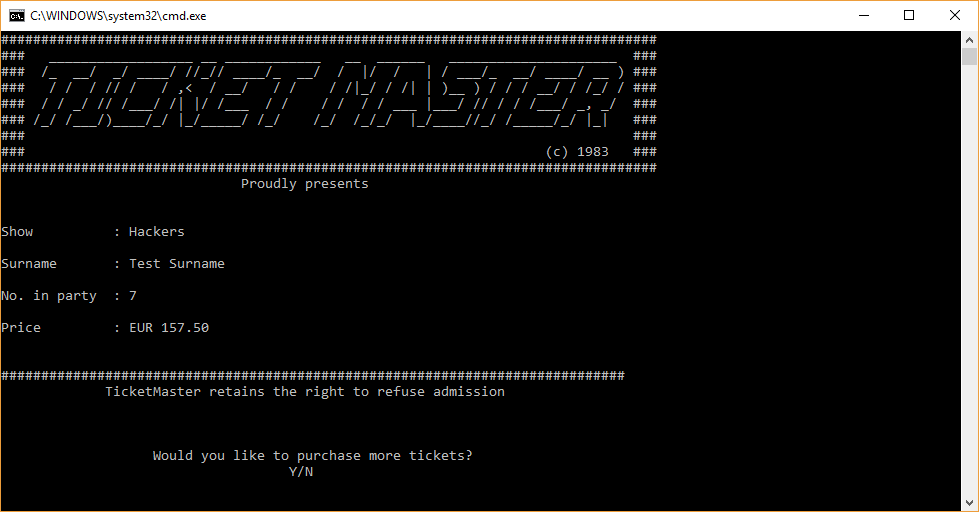
Screen #3



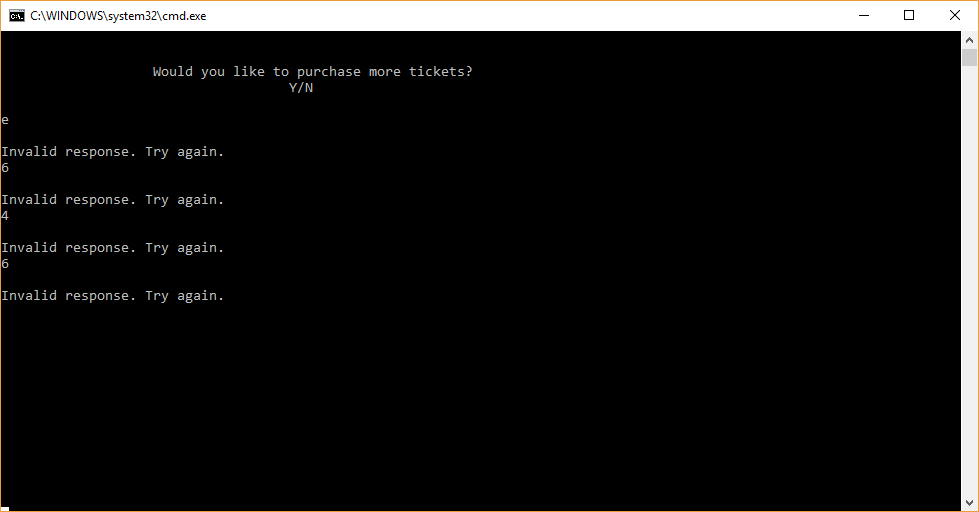
Screen #4

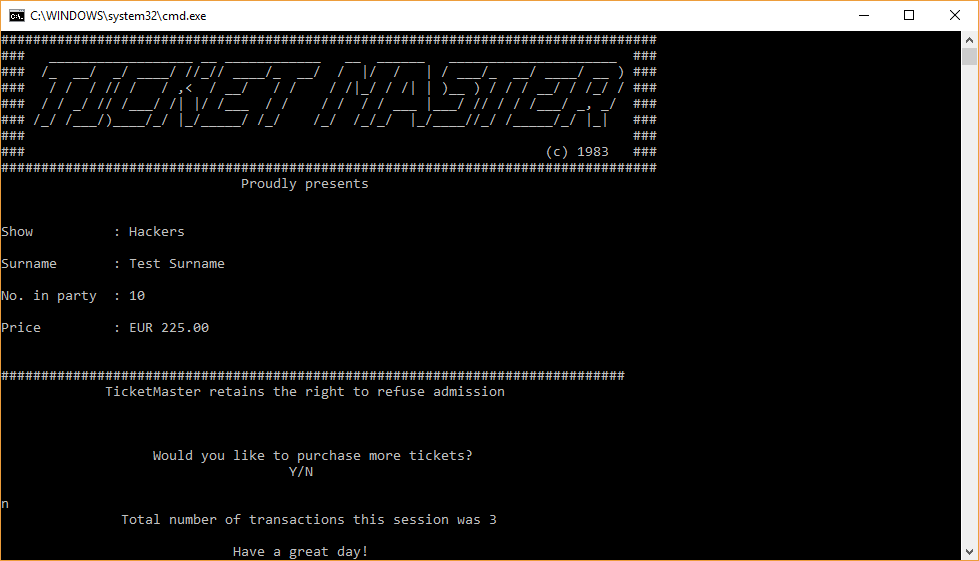


Screen #5



Screen #6



Screen #7

**Declaration Of Own Work**

**Student Name: David Mulhall**

**Student Number: B00107874**

**Course Title: BN026 Computer systems management**

**Lecturer: Ann Marie Cosgrave**

**Title of Work: Ticketing System**

**Set Submission Date: 13/11/2017**

I confirm that all this work is my own and that I have:

|  |  |
| --- | --- |
| * Clearly referenced/listed all sources as appropriate |  |
| * Given the sources of all pictures, data etc. that are not my own |  |
| * Any work of any other student(s) either past or present is clearly referenced |  |
| * Not submitted for assessment work previously submitted for any other course, degree or qualification |  |
| * Not incorporated any text acquired from external agencies other than extracts from attributed sources (including online facilities) |  |
| * Acknowledged in appropriate places any help that I have received from others (e.g. fellow students, technicians, statisticians, external sources) |  |
| * My work may be electronically checked for plagiarism, including, but not exclusively, by the use of plagiarism detection software and stored for future comparison |  |
| * Any false claim for this work will be penalised in accordance with the College’s regulations |  |

**Signature** ………………………………………..……….……………………..**Date** …………………….…..….