**Blue Box DVD/Blu Ray Rentals**

**Software System Design**

**Document Specification**

**Version 1.0**

**Date : 11/17/19**

**Prepared by EZ Software**

Table of Contents

Introduction 3

Application Architecture 3

System Functions 4

User Interface 10

Components 14

Database 16

Required Hardware and Software 19

1. **Introduction:**

Blue Box serves as a DVD/Blu-Ray rental service. This system will allow customers to search, browse and rent movies or documentaries etc. Customers can do it without any hassle and whole process is made to be as simple, convenient and seamless as possible.

These days thousands of movies come out in a year and some of them are blockbusters, some of them attract kids, some of them attract a specific type of audience. But for most people watching the movie once or twice in a movie theatre is not enough and they want to watch it again later sometime. So, instead of buying a DVD which would cost significantly more, we wanted to provide people with a service where they could rent out a copy and then when they are done watching it, they could return it.

Goals of the system are as follows:

* Provide an easy, affordable and convenient service to clients.
* Continually expand our search library and keep on adding to more material to accommodate the needs of a wider audience
* Instead of buying a DVD provide customers with a more economical option of renting one.

1. **Application Architecture:**

The application will have a UI on the front end where the customers could access the services provided to them. The UI will be installed on the kiosks that will be placed at the locations where we want to provide our services. On the backend we need database management where we could store all the information about the DVD/Blu-Ray and their copies and would also be able to store the customers information and which movies a particular customer has checked out and for how long. We also need a web client which would provide all the information about any movie that a customer is looking for, its rating, cast, runtime and so on. The architecture model we will be using would be Client server Model. We chose this model for the following reasons:

* It has powerful client processing capability
* User friendly graphic interface
* Its easy to upgrade or add new servers

All these factors are things that we need for our system. Below is the application architecture diagram.

Vending Machine

Vending Machine

Vending Machine

Vending Machine

Vending Machine

Backup Server

1. **Required System functions**

3.0.1 Search

1) Search for movies with various different inputs (genre, rating, search bar)

2) Search by relevance to title for search bar

3) Alphabetical list of the category otherwise

3.0.2 Rent

1) Add to cart

2) Checkout

3) Cart limit of three movies per rental

3.0.3 Return movies

1) Return disk to spot in machine inventory

2) Updates to inventory db and customer info to finish payment processing and final charges on the account

3.0.4 Login

1) Login checks against customer db

2) Makes sure they can rent still (no outstanding charge)

3.0.5 Add/remove movies

1) Update database manually

3.2.1 Search movie

|  |  |
| --- | --- |
| Name | Search movie |
| Trigger | The browser assesses the movie selection |
| Precondition | User is on the starting screen |
| Basic path | 1) User reads the various selections and makes one based on rating (drop down), genre (buttons along top), or by typing in the search bar and checking relevance to title  2) If its selected by genre, newest hottest or critically acclaimed movies, then the system creates and presents an alphabetical list of all movies in the specified genre in the database  3) The User selects a movie  4) System returns relevant movie info (description, director(s), lead actor(s), etc)  5) User proceeds to rental (adding bluray or movie to cart) or returns to search |
| Alternate path(s) | In step 2, if the User selects to search by rating, the system creates and presents an alphabetical list of all movies of the rating in the database.  3) Return to step 3  In step 2, if the User selects to search by keyword, the system presents a dialog box to enter the keyword or phrase.  3) The User enters a keyword or phrase.  4) The system searches the titles for all movies with that keyword or phrase and creates and presents a list of all such articles in the database. Return to step 3. |
| Post condition | Adds movie to cart for later checkout |
| Exception paths | User may abandon search at anytime |
| Other | The categories list is generated from the information provided when movies are posted and not predefined in the movie db. |

3.2.2 Rent movie

|  |  |
| --- | --- |
| Name | Rent movie |
| Trigger | User clicks checkout button by cart |
| Precondition | User has movies in cart |
| Basic path | 1. User selects the checkout button 2. Selects login or card 3. Displays price per day on screen 4. User selects rent and inputs payment option for card 5. Processes and stores payment info 6. Dispenses movie and updates inventory db (DVD or bluray) 7. Displays thank you message on screen restarts to title |
| Alternate path(s) | Selects login in step 2  3) Brings up two textboxes in a small login screen  4) Checks credentials against customer db  5) Returns bool canRent(no outstanding problems w payments)  Proceeds to step 6 |
| Post condition | Movie is dispensed by machine and up |
| Exception paths | Renter can cancel at anytime, timeouts will reset cart (inactive screen), pressing clear will reset cart |
| Other | None |

3.2.3 Return

|  |  |
| --- | --- |
| Name | Return |
| Trigger | Disk return at kiosk |
| Precondition | None |
| Basic path | 1) User inserts movie into dispense/return slot  2) Movie is returned to rightful place and appropriate inventory db for its ID is updated another copy  3) Updates customer db and stops rental charges, calculating a cost based on the daily amount and how many full 24 hour cycles its been out of the kiosk  4) Displays this total and says thank you for your return, calculatedtotal will be charged to your card  5) Processes payment |
| Alternate path(s) | If problem with payment method given at rental occurs in step 5, customer gets CanRent turned false until charge is cleared |
| Post condition | Movie is back in inventory and charges have stopped accruing on customers account |
| Exception paths | None |
| Other | None |

3.2.4 Login

|  |  |
| --- | --- |
| Name | Login |
| Trigger | Clicking login, either at checkout or in the cart display |
| Precondition | Being at start screen |
| Basic path | 1. Text boxes will pop up with login and password 2. User inputs username and password 3. System accesses customer account and determines if that account exists, and that the credentials are correct 4. Displays a welcome message to them, returns them to screen they clicked login on |
| Alternate path(s) | If in step 3 the password, username combination is incorrect  4) Display username and password are incorrect  5) Return to step 1 |
| Post condition | Customer account and payment details ready for kiosk to generate a rental |
| Exception paths | None |
| Other | None |

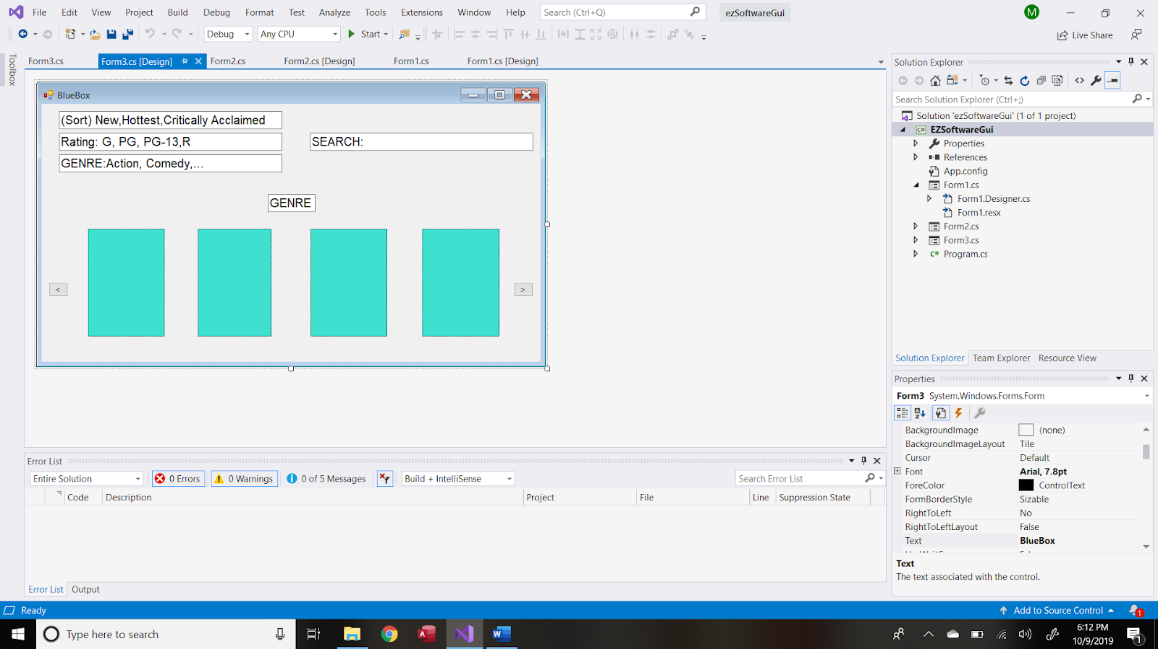
3.2.5 add/remove

|  |  |
| --- | --- |
| Name | Add/remove movie |
| Trigger | New movies to add to db |
| Precondition | None |
| Basic path | 1) Admin login to db from one of our computers  2) Manually add a movie row and its associated details |
| Alternate path(s) | Admin wants to remove movie from our db  2) Search for movie coming out of rotation  3) Delete it and its associated details |
| Post condition | Movie db updated so that movies and dvds of the new movie can be added/ and movies coming out of rotation are put on notice to be removed from kiosks |
| Exception paths | none |
| Other | none |

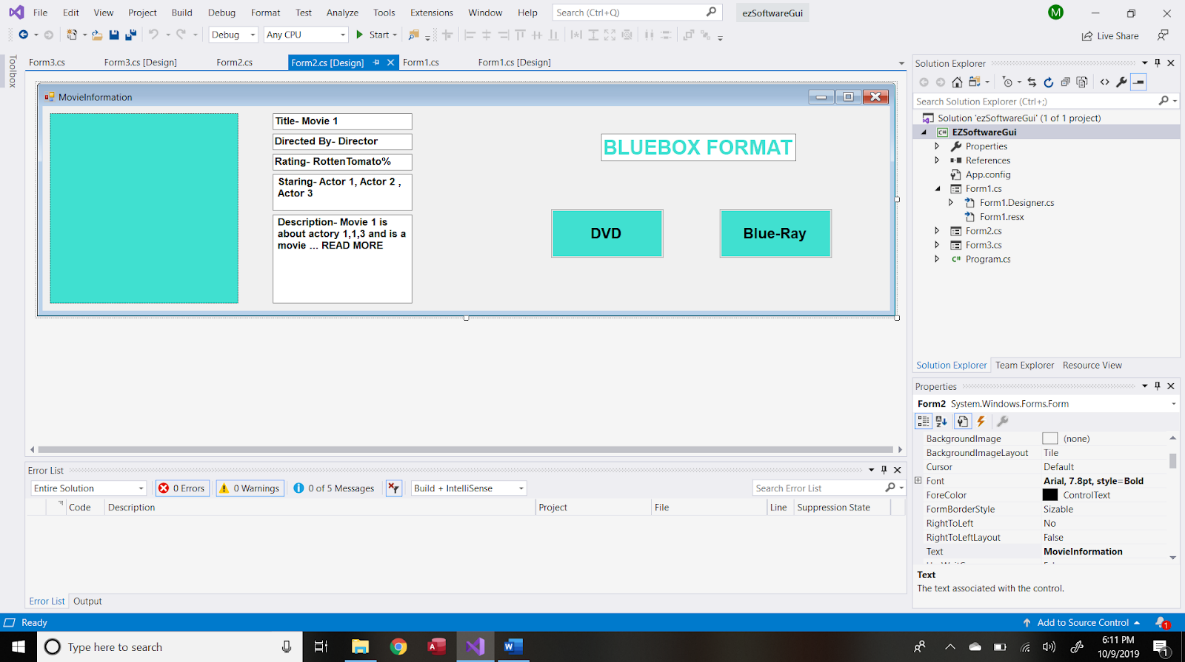
**4.0 Guest User interface prototypes**

4.1 Search movie

Start screen is sorted by hottest movies



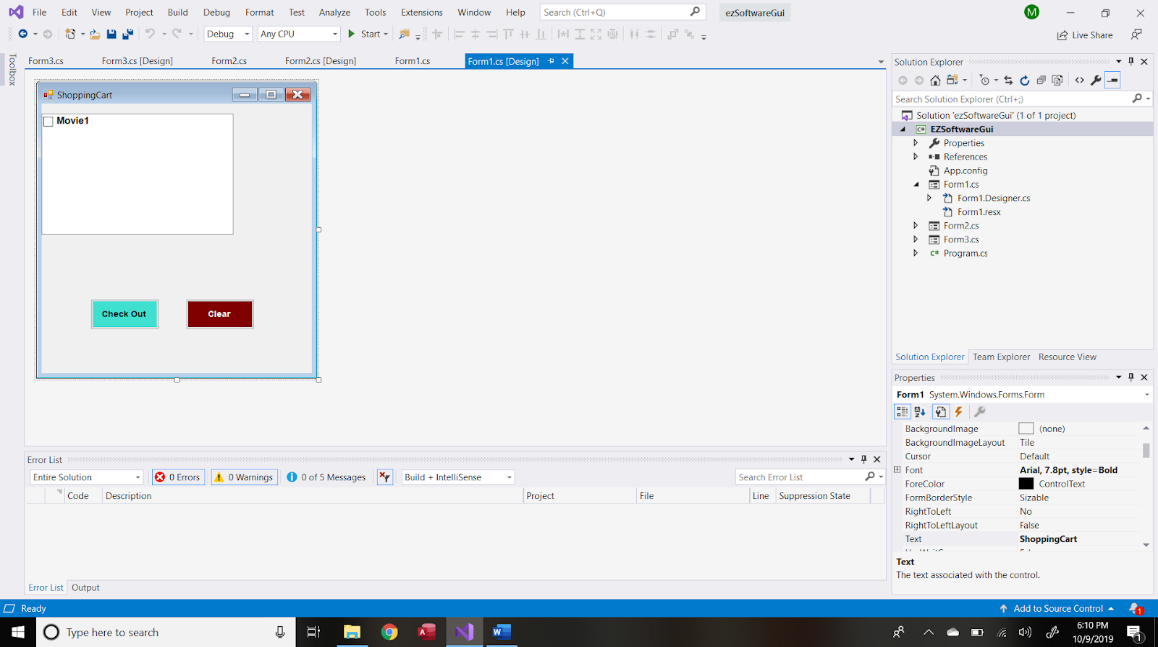
Selecting a rating or genre from the dropdown menu or to use the search bar will reorganize the movies shown to you by relevance. When users select a move they like they are brought to this screen



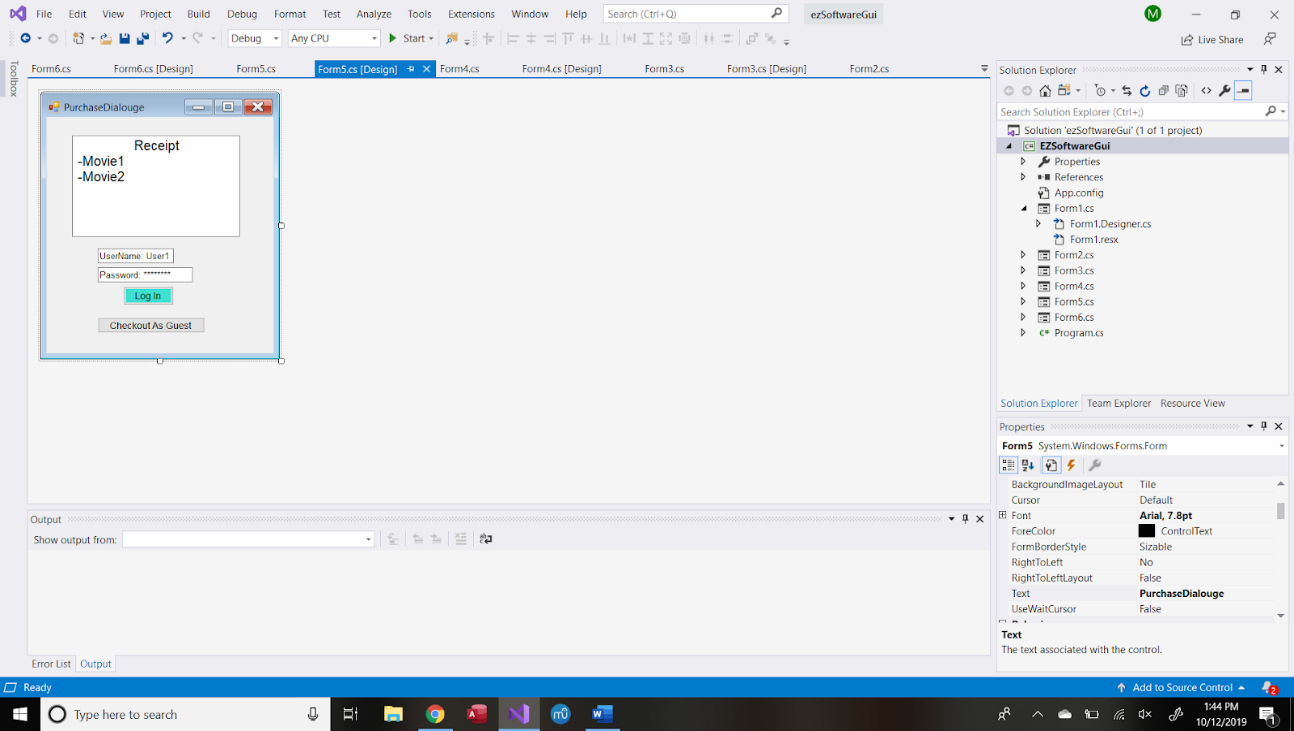
Which gives you all the relevant details of the rental details

4.2 Rent movie

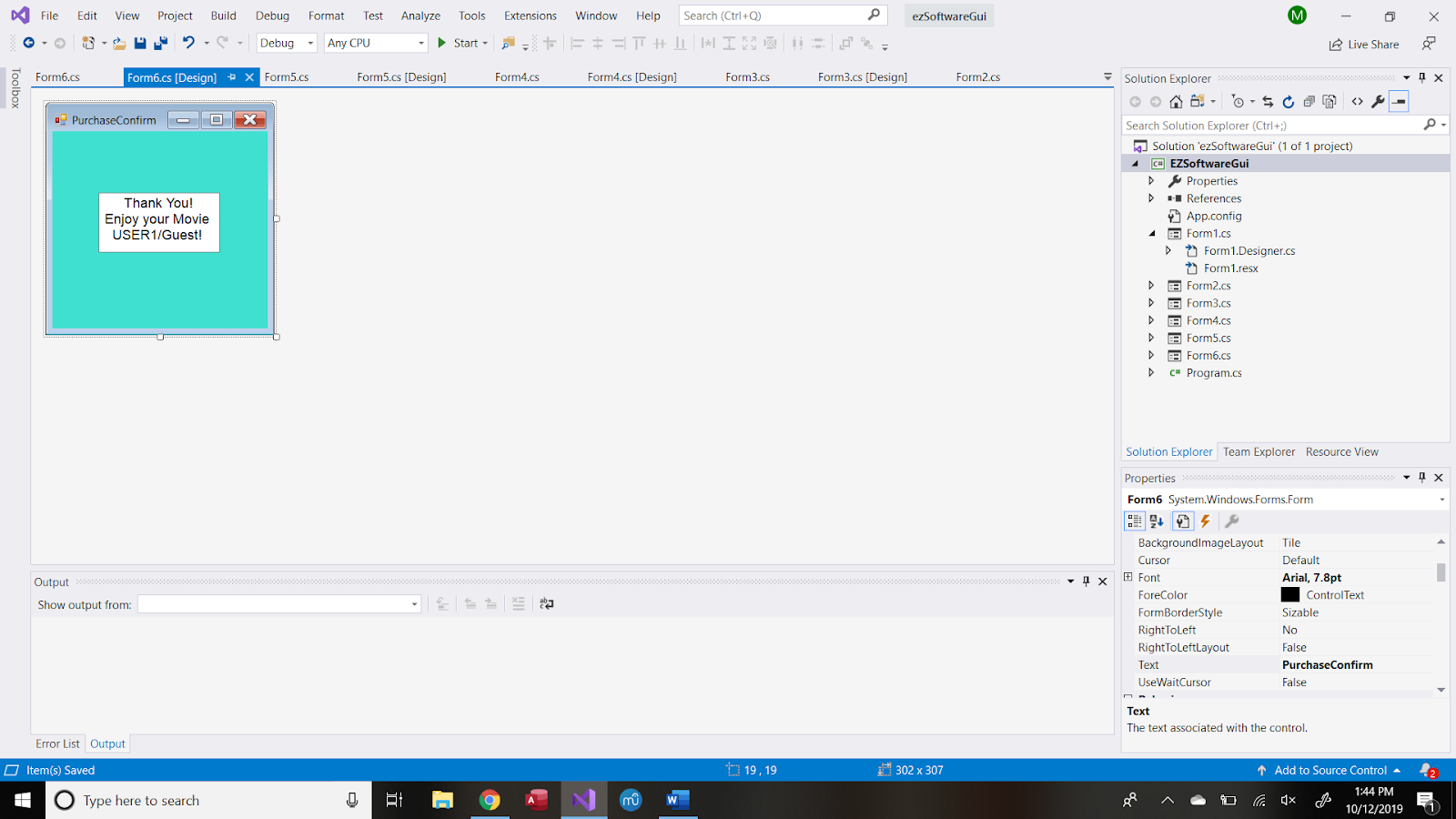
Upon clicking on blu ray or dvd from the selection screen above users will update their cart, shown on the right side of the screen on the kiosk and in the prototype below



When a user has selected all of the movies they would like to rent (up to 3 at a time) they can click the checkout at the bottom of the cart screen. Upon doing this, user is prompted to login or use their card.

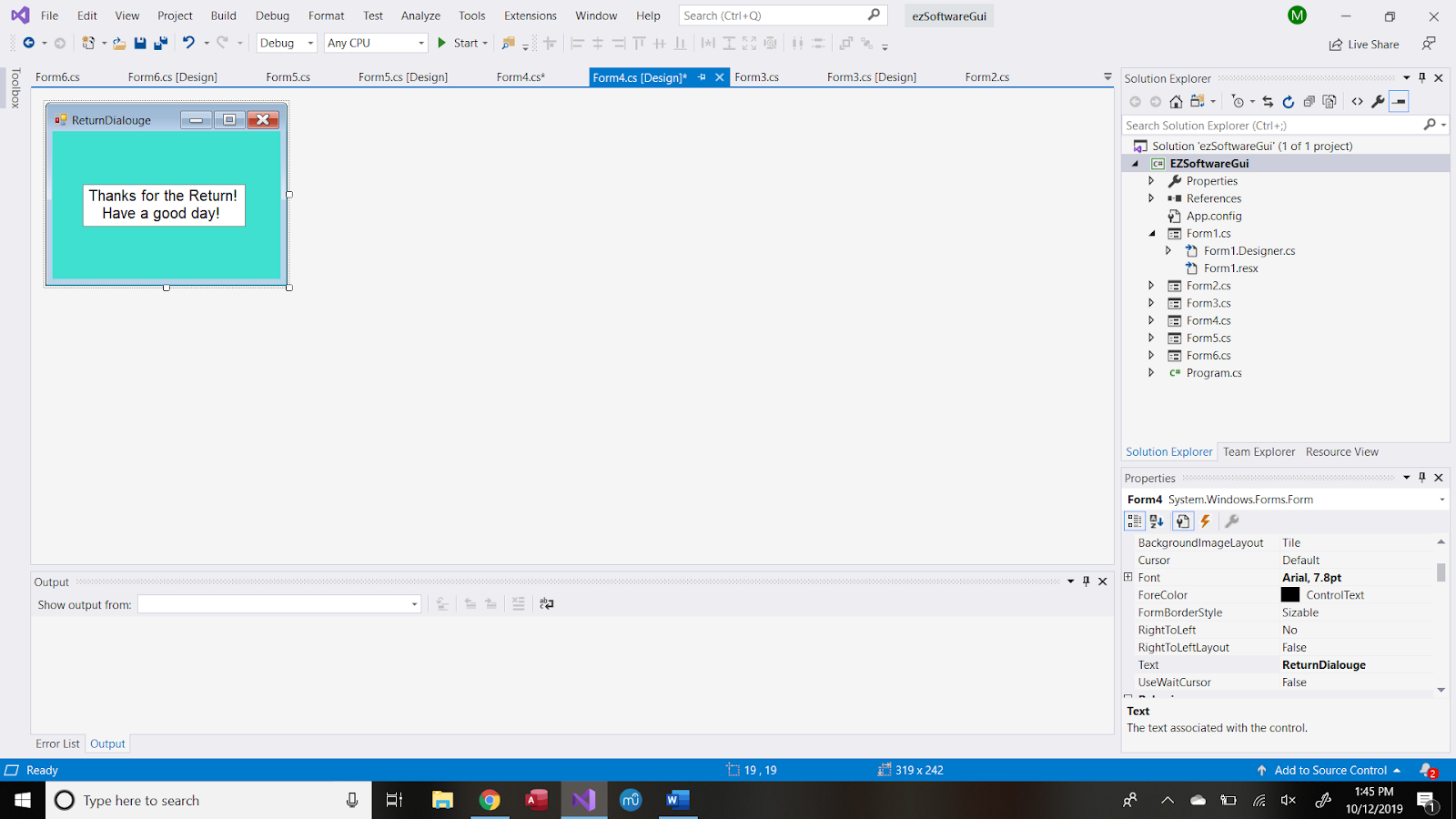


After checking out you are displayed a thank you message (shown below) for a minute before it resets to the starting page. Depending on the user info we have available we’ll say enjoy your movie guest or the name of the guest from the user login.



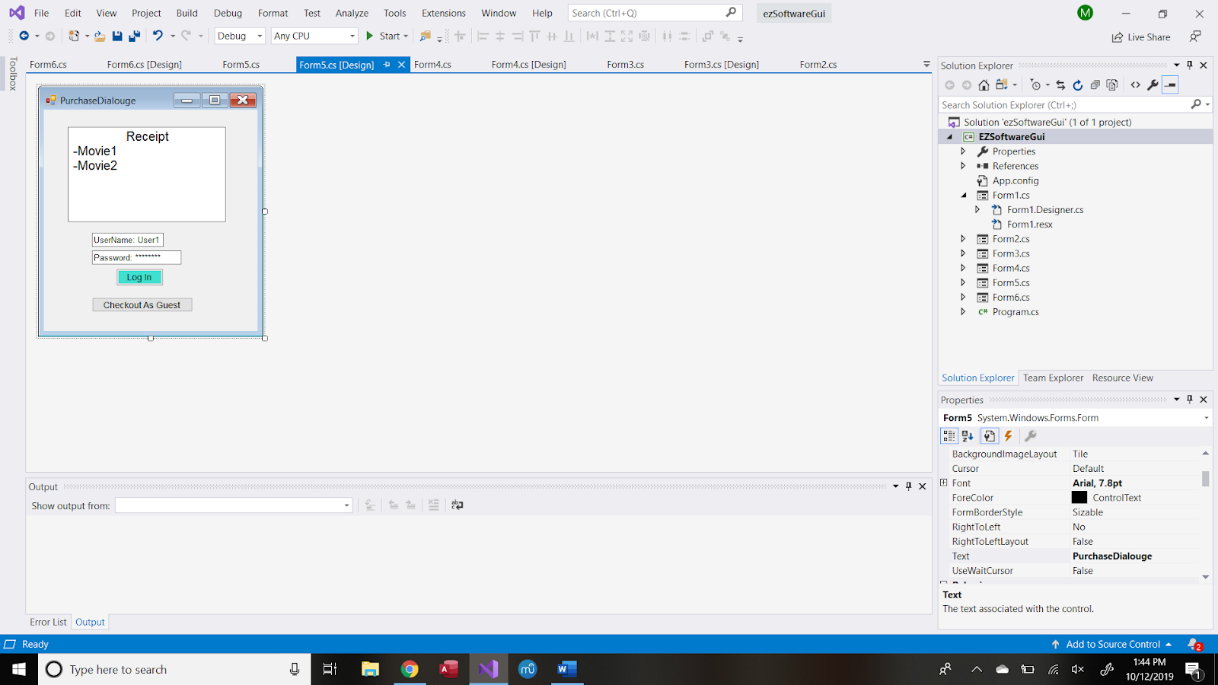
4.3 Return

When a disk is returned in the slot for movies the screen displays a thank you message shown below and the days since rented. Thanks for the return after (Date-RentStart) days! Have a good day!



4.4 Login

Logins will be done at the end when you would like to checkout, as described in 4.2. The displayed screen will have your cart and the login boxes or checkout as guest.



4.5 add/remove

Adding and removing movies to the db will be done from an outside terminal access, not on the kiosk, done manually by a script to interact with the db. No interaction with the GUI

**5.0 System Components**

It should be noted that our implementation will limit itselft to the

components deployed on the vending machines. Our components will follow

the MVC design pattern, thus diving our system into three interconnected

elements, the view, the model and the controller. Due to the size of the

system the view and the controller will be merged to form one component.

THE VIEW

Every of the app will be associated to a corresponding class. We

should then have at least three classes, one for the login, the browsing,

and the checkout. These classes will handle the input and output of data

associated with the uses case.

Name: View

Function Description:

Assure a smooth interaction between the user and the vending machine. Displays

data related to the uses cases and accept input from the user. Call the class of the

model component responsible to perform the actions associated to the use case.

Input Data

1. User's credential

2. User's movie choice

3. User's credit card information

Output Data

1. Confirmation message for every transaction

a. Payments

b. Renting

Dependencies

This view gets its data from the controller.

THE MODEL

Its the heart of the system as it performs all the operations from the

validation of the inputs, the calculation of the payments' amounts to

routing the information to the database in order to be stored.

Name: Model

Function Description:

Its Responsible of the logic of the system. Unlike the view classes, the classes

associated with the model component will be goal-oriented. As the system is

limited in scope and size, one class could handle all the logic of the

system. The model will

1. Process rents by checking the availability of the movies.

2. Check whether a user is allowed to rent a movie or not.

3. Update user's information when a dvd is returned.

3. Manage login and registration.

4. Send data to the database to be strored.

5. Calculation related to the payments such as fees.

Input Data

1. Data associated to the user's input.

Output Data

1. Any information related to the transactions that need to be displayed

to the customer.

Dependencies

It interacts with the controller and the database.

Name:The Controller

Function Description:

The Database

One class will handle all the database transactions. Besides, only one

instance of the given class will exist in the system.

Name:Database

Function Description:

Create, update, retrieve and store the records associated to the transactions.

Input Data:

1. Any formatted information coming from the model component.

Output Data:

1. The database records such as availalble movies and transaction history.

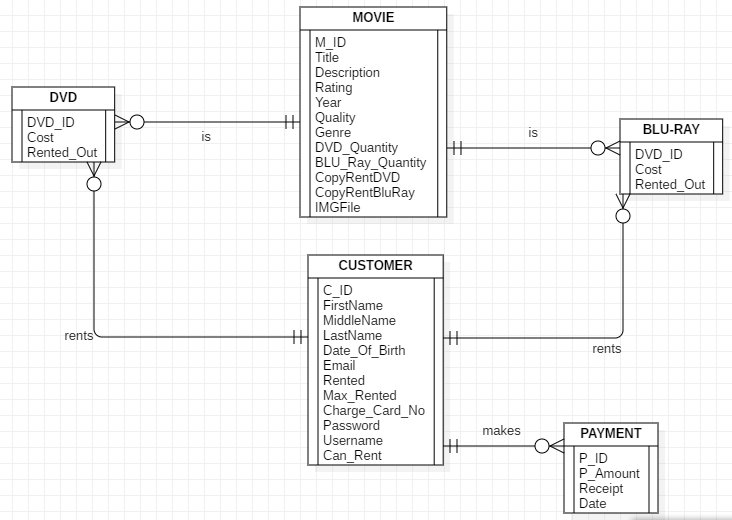
Dependencies:

It interacts with the model which sends data to be stored and read data

out of the database.

**6- Database Model**

E-R Data Model:



Logical Data Model:

MOVIE(M\_ID, Title, Description, Rating, Year, Quantity, Genre, DVD\_Quantity, BLU\_Ray\_Quantity, CopyRentDVD, CopyRentBluRay, IMGFile) Where M\_ID is the primary key.

CUSTOMER(C\_ID, FirstName, MiddleName, LastName, Date\_Of\_Birth, Email, Rented, Max\_Rented, Charge\_Card\_No, Password, Username, Can\_Rent) Where C\_ID is the primary key.

BLU-RAY(BLU\_ID, Movie\_ID, Cost, Rented\_Out) Where BLU\_ID is the primary key, and Movie\_ID is a foreign key that references MOVIE.M\_ID.

DVD(DVD\_ID, Movie\_ID, Cost, Rented\_Out) Where DVD\_ID is the primary key, and Movie\_ID is a foreign key that links to MOVIE.M\_ID.

PAYMENT(P\_ID, Customer\_ID, P\_Amount, Receipt, Date) Where P\_ID is the primary key, and Customer\_ID is a foreign key that refences CUSTOMER.C\_ID.

Physical Data Model:

SQL Code

MOVIE:

CREATE TABLE MOVIE(

M\_ID int NOT NULL PRIMARY KEY,

Title varchar(50) NOT NULL,

Description varchar(250) NOT NULL,

Rating int,

Year int NOT NULL,

Quantity int NOT NULL,

DVD\_Quantity int NOT NULL,

BLU\_Ray\_Quantity int NOT NULL,

CopyRentDVD int NOT NULL,

CopyRentBluRay int NOT NULL,

IMGFile varchar(50)

)

CUSTOMER:

CREATE TABLE CUSTOMER(

C\_ID int NOT NULL PRIMARY KEY,

FirstName varchar(50) NOT NULL,

MiddleName varchar(50),

LastName varchar(50) NOT NULL,

Date\_Of\_Birth date NOT NULL,

Email varchar(50) NOT NULL,

Rented int NOT NULL,

Max\_Rented int NOT NULL,

Charge\_Card\_No int NOT NULL,

Password varchar(50),

Username varchar(50),

Can\_Rent bit NOT NULL

);

BLU-RAY:

CREATE TABLE BLURAY(

BLU\_ID int NOT NULL PRIMARY KEY,

Movie\_ID int NOT NULL references MOVIE(M\_ID),

Cost money NOT NULL,

Rented\_Out bit NOT NULL

);

DVD:

CREATE TABLE DVD(

DVD\_ID int NOT NULL PRIMARY KEY,

Movie\_ID int NOT NULL references MOVIE(M\_ID),

Cost money NOT NULL,

Rented\_Out bit NOT NULL

);

PAYMENT:

CREATE TABLE PAYMENT(

P\_ID int NOT NULL PRIMARY KEY,

Customer\_ID int REFERENCES CUSTOMER(C\_ID),

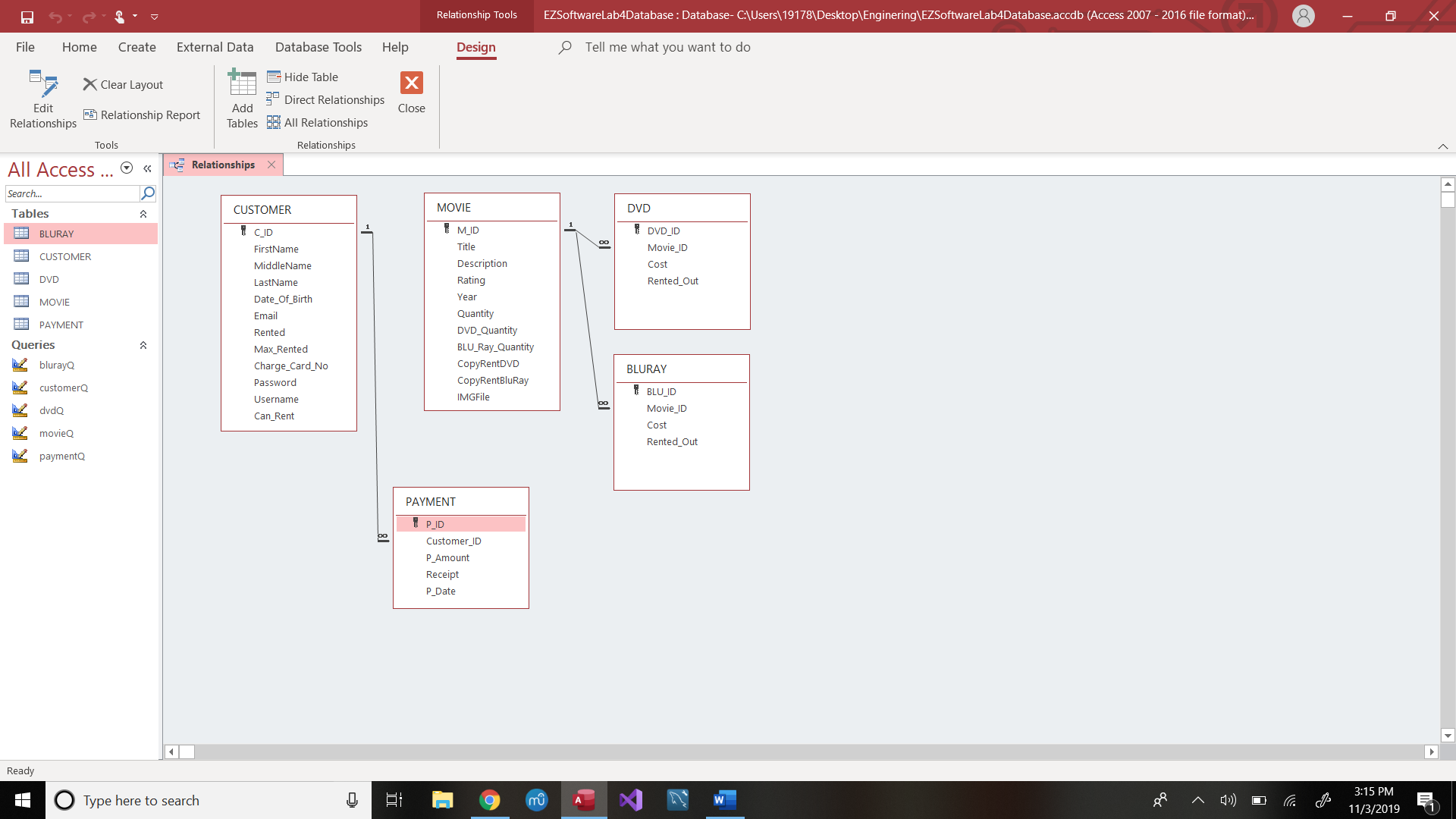
P\_Amount money NOT NULL,

Receipt int NOT NULL,

P\_Date date NOT NULL

);

USING MICROSOFT ACCESS:



**7- Required Software & Hardware**

Hardware design of CD vending machine- The machine for renting the CDs is built like a vending machine but with CDs and Blu-Rays. This is a very simple design and should be easy to incorporate. Also, since we are using CDs, we can fit many CDs into the machine since they do not take up a lot of space. The vending machine then has a touch screen display that runs only the companies renting software.

Software design of CD vending machine- Uses a extremely user interactive and responsive program. That assists with renting, and returning movies to the vending machine.