Shaheryar Abid - 501110694
Tim Rozer - 501082203
Rohan Chedde - 501123581
Maninder Arora - 501041960

TA: Jorge Lopez
Course: CPS510
Professor: Abdolreza Abhari

# Online Movie Store

## Description:

The topic of our choice is an Online Movie Store. The store will be hosted on the internet and it will allow customers to purchase movies. Users will be presented with a catalog of movies they can sort through in a variety of manners and purchase. They will have the option of using the application as a guest or have a membership account which will grant them added functionalities and benefits. The membership holders will have points that will accumulate with each purchase and they will be able to redeem it for a discount on their purchases. They will also be able to add movies they want to purchase in the future to a wishlist.
The administrator has special access to the customer accounts and movie catalog which will grant them the authority to add or delete movies and customers as they wish.

## Functions:

Our online movie store implements a variety of features to make the user experience more streamlined and have customizability on the owner side. The results expected from the application include several customer features. Customers will have the option to use the application as a guest or as a member. Customers with membership accounts will be granted personalized features like a wishlist to add and favorite movies and view their transaction history. Having a function that allows users to create an account with a username and password to login and an abstract customer class will achieve this result. Moreover, a function that allows customers to add movies to a shopping cart that they can review and checkout will be necessary so customers can purchase movies with ease. Importantly, we will need a function to display general information about each movie, including title, reviews and ratings by having an abstract movie class as a starting point. The customer will be able to view their transaction history to keep track of their spendings. Additionally, customers will have the option to leave reviews on a movie. Lastly, there will be a function that allows the admin to add or delete movies or customers.

## Entities:

The first main entity in our movie store will be the **customer**. Customer will have an “adds to” relationship with the shopping cart, “has” with the account weak entity, and “accesses” with the catalog.
The **account** will have a key attribute of a unique username. Account has an “accesses” relationship with the **membership** entity.
Membership has a dynamic attribute of points respective to the customer. Branching from the membership with a “manages” relationship is the **wishlist** weak entity. The **shopping cart** entity will have a dynamic attribute of total cost and an “accesses” relationship with the payment entity.

**Payment** entity has a key attribute of payment authentication and multi-valued attribute of payment method. Furthermore, the **admin** is an entity also with a key attribute of a unique username. Admin has a one to many “manages” relationship with customer and movie catalog since admin can add or delete them. Moreover, another important entity is the **movie**. Movie has many attributes which are views(dynamic), title(simple), rating(dynamic) and price(simple) and a key attribute movie Id. **Catalog** is an entity that is the menu and has a one to many ‘has’ relationship with the **genre** entity because the catalog is a collection of all the genres. Genres being a weak entity have their own specific name as a simple attribute.

Customer

- Customer has an “adds to” relationship with the shopping cart, “has” with the account entity, and “accesses” with the catalog.
- It has 2 attributes : customerId and customerName.
  - customerId is an integer value and serves as the primary key
  - custome Name is a char value and cannot be NULL.
- Cardinality
  - It has a N to 1 relationship with Admin
  - It has a N to 1 relationship with Catalog
  - It has a 1 to 1 relationship with Shopping Cart
  - It has a 1 to N relationship with Payment
  - It has a 1 to 1 relation with Accoiunt
- Keys analysis of entities (primary/Foreign)
- It has one primary key of attribute customerId, it is used to keep track of all customer on the website.

Account

- Account has a “access” relationship with Membership and Shopping Cart entites.
- It has
- The **account** will have a key attribute of a unique username. Account has an “accesses” relationship with the **membership** entity.

- (relationship analysis)
- Attributes analysis
- Data type and length definition for attributes (integer, float, string, ...)
- Cardinality
- Keys analysis of entities (primary/Foreign)

| Entities | Attribues |
|---|---|
| Admin<br>(Superclass) | ● username<br>    ○ Type String<br>    ○ Primary Key<br>● pass<br>    ○ Type String |
| Customer | ● customerId<br>    ○ Type int<br>    ○ Primany Key |
| Account | ● username<br>    ○ Type String<br>    ○ Must be unique<br>● pass<br>    ○ Type String<br>● customerId references customer(customerId)<br>    ○ Type int<br>● points references membership(points)<br>    ○ Type String |
| Shopping Cart | ● orderID<br>    ○ Type int<br>    ○ Primary Key<br>● total<br>    ○ Type int<br>● Price<br>    ○ Type int<br>● movieId<br>    ○ Type int<br>    ○ Foreign Key references movie(movieId)<br>● customerId Type int<br>    ○ Foreign Key references customer(customerId) |
| Payment | ● Method<br>    ○ Type String<br>● Authen<br>    ○ Type int<br>    ○ Primary Key (unique) |
| Membeship | ● Points<br>    ○ Type int<br>    ○ Constraint points cannot be negative |
| Wishlist | ● movieId<br>    ○ Foreign Key references movie(movieId)<br>● customerId<br>    ○ Foreign Key references customer(customerId) |

| | |
|---|---|
| Catalog | <ul><li>Genre<ul><li>Type String</li><li>Primary Key (unique)</li></ul></li></ul> |
| Generes | <ul><li>movie<ul><li>Foreign Key references movie(movieId)</li></ul></li></ul> |
| Movie | <ul><li>movidId<ul><li>Type int</li><li>Primary Key</li></ul></li><li>movieName<ul><li>Type String</li><li>Cannot be NULL</li></ul></li><li>moviePrice<ul><li>Type Int</li><li>Defalt value 0</li></ul></li><li>Genre<ul><li>Type String</li><li>Foreign Key reference genres(genre)</li></ul></li></ul> |

Relationships

1. Admin has a 1 to many “manages” relationship with customer and and 1 to 1 with catalog
2. Customer has a 1 to 1 “adds to” relationship with Shopping Cart and 1 to many “makes payment” with Payment, and has a 1 to 1 “has” relationship with account.
3. Catalog has 1 to many “accesses” relationship with customer.
4. Shopping cart has a 1 to many “accesses” relationship with payment