**Assignment 4 Design**

**Po Hong Lau, Lu Lu**

**Overview**

This project is to implement and design the automation inventory tracking system for a local movie rental store. Currently there are three types of movies/videos (in DVD media) to be tracked. Borrows and returns of items by customers are also to be tracked. There are four types of actions are desired in the system Borrow, Return, Inventory and History.

**Interaction between different classes**

**MovieCategory** class is the parent class of the other three types of movies type which is Comedy class, Drama class and Classic class. **MovieCategory** will represents a specific movie with a title, year and stores how many copies an available in the store. Movies must sort by the attributes therefore will be stored in a set and using vector to store the set of movies.

**MovieStore** class is represents an actual store which contains all the customer and movie category information. Therefore, **Customer** Class and **MovieCategory** will have interact with **MovieStore** class.

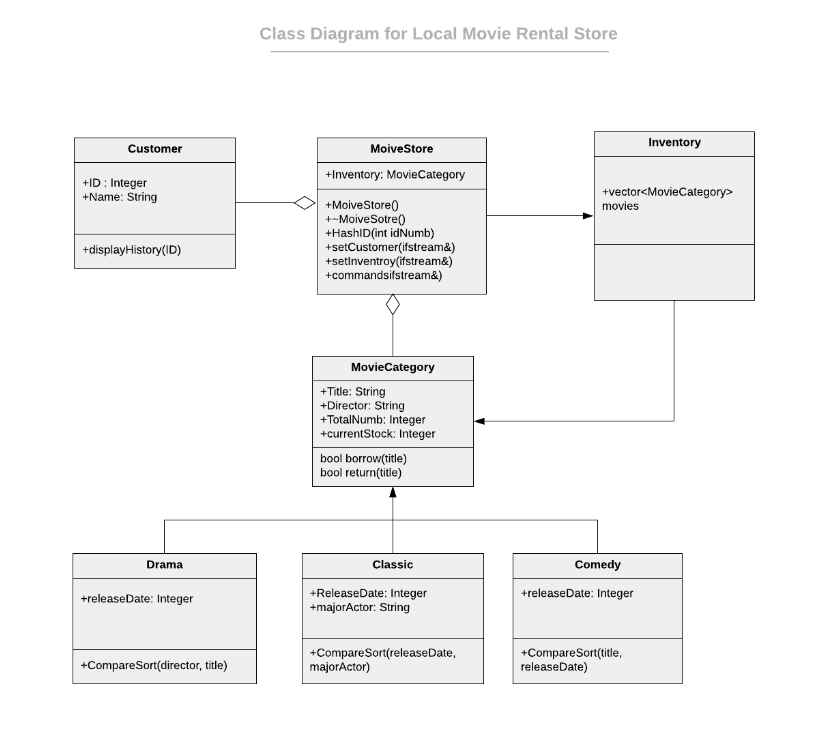
**Inventory** class stores Customer, Movies, and Commands data to conduct the customer’s action such as return or borrow. Therefore, will interact with **MovieCategory** class.

**Customer** class contains customer’s ID number, last name and first name information and contains a Borrow and Return action. Also, build up a customer’s history information. Customer can be identified by using their ID number therefore will be using a Hash Map.

**Objects in main**

The main will read the input text file and build up a movie inventory and customer list.

**Class diagram**

****

***Class descriptions***

**MoiveStore**

MoiveStore()

{

setCustomer(ifstream&)

setInventory(ifstream&)

command(ifstream&)

}

Void command(ifstream&)

{

Read each line in input command file

Check the first character on the line

If ( first character == vaild action)

{

Switch (first character in the file)

Case (“B”):

{

Borrow (title)

}

Case (“R”):

{

Return (title)

}

Case (“H”)

{

DisplayHistory(ID)

}

}

}

else

{

cout << “Invalid action “” << endl;

}

}

**MovieStore**

**Purpose**

The purpose of this class is to represents a rental store and read the input file customer and movie information and then build up a movie store inventory.

**Provided Interface**

Void movieStore();

**Description**

Default constructor to initial all data

**Parameters**

This method has no parameters

Void ~movieStore();

**Description**

A destructor to delete customer and movie information

**Parameters**

This method has no parameters

void HashID(int iDnumb);

**Description**

Using this Hash Map method to identify specific customer by using their ID number as parameters

**Parameters**

int iDnumb is the customer’s personal ID to identify specific customer

Void setCustomer(ifstream&);

**Description**

Implementation file for the customer information input (provided text file)

**Parameters**

Read the input file (provided text file) to get the customer information.

Void setInventory(ifstream&);

**Description**

Implementation file for the inventory information input (provided text file)

**Parameters**

Read the input movie file (provided text file) to build up a movie inventory

Void commands(ifstream&);

**Description**

Implementation file for the input commands (provided text file)

**Parameters**

Read the input commands file (provided text file) to create a command before being processed by the Inventory.

**MoiveCategory**

**Purpose**

The purpose of this class is to represents a specific movie with a title, year and stores how many copies an available in the local movie rental store.

**Provided Interface**

Bool borrow(title);

**Description**

A method that when the customer borrows the movie the stock will -1 for each item borrowed

**Parameters**

identify which movie is borrowed

Bool return(title);

**Description**

A method that when the customer returns the movie the stock will +1 for each item returned

**Parameters**

identify which movie is returned

**Customer**

**Purpose**

The purpose of this class is to implement file for the individual customers of the movie type.

**Provided Interface**

Void displayHistory(ID);

**Description**

Display the outputs of all the transactions of a customer

**Parameters**

ID is the customer’s ID to identify the specific customer

**Inventory**

**Purpose**

The purpose of this class is to implementation file for the inventory of the movie type by using vector movies must sort by the attributes therefore will be stored in a set and using vector to store the set of movies.

**Drama**

**Purpose**

The purpose of this class is to represents a movie with a distinct title and year belonging to the comedy and the local movie store owns at least one of the copies.

**Provided Interface**

CompareSort(director,title);

**Description**

Implementation of the movie sorting attributes. Sorted by Director then title

**Parameters**

Director represent the movie director and title represent the movie title.

**Classic**

**Purpose**

The purpose of this class is to represents a movie with a distinct title and year belonging to the comedy and the local movie store owns at least one of the copies.

**Provided Interface**

CompareSort(releaseData,majorActor);

**Description**

Implementation of the movie sorting attributes. Sorted by Release date then Major actor

**Parameters**

releaseData represent the movie’s release data and majorActor represents the major actor in the movie.

**Comedy**

**Purpose**

The purpose of this class is to represents a movie with a distinct title and year belonging to the comedy and the local movie store owns at least one of the copies.

**Provided Interface**

CompareSort(title,releaseData);

**Description**

Implementation of the movie sorting attributes. Sorted by title then Year it released

**Parameters**

Title represents the movie title and the release data represent the movie’s release data