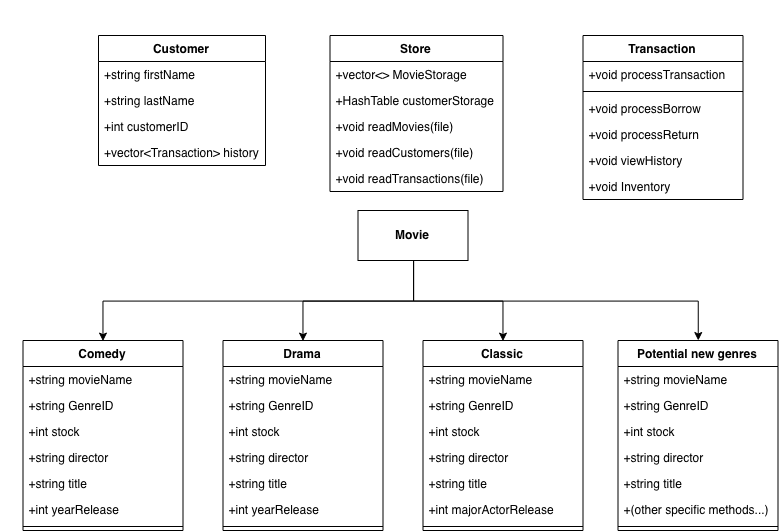
Names: Eric Feldman, Soo Kim, Will Hwang, Samreen Mohammed

## UML Class Diagram



## 

## 

## 

## Class Interactions

**Class:** Store

* Int stock: total stock
* Inventory vectors sorted
* Vector of drama movies
* Vector of comedy movies
* Vector of classic movies
* Creates customers
* Hash table of customers, with the ID as the key, and a customer object as the value
* Responsible for parsing Customer text file, transactions text file, and inventory text file
  + Customer text file: Customer will be created as a new Customer object with name and history. Customer is then added to hash table of customers
  + Transaction text file: Each transaction will be processed by creating a transaction object, and then calling the method processTransaction in the class Transaction.
  + Inventory text file: For example, for “F, 10, Nora Ephron, You've Got Mail, 1998”, the class will create Movie f = new Comedy(10, Nora Ephron, You've Got Mail, 1998). Then, the movie will be added to the correct inventory vector

**Class**: Customer

* Constructor: Customer ID, Name, History (vector of transaction objects)

**Class**: Movie

**Class:** Comedy

* Constructor: Stock, Director, Title, Year it released

**Class:** Drama

* Constructor: Stock, Director, Title, Year it released

**Class:** Classics

* Constructor: Stock, Director, Title, Major actor Release date

**Class:** Transaction

* Constructor: Transaction type, Customer, Movie
* Constructor: Transaction type, Customer
* processTransaction(int stock): takes care of the transaction and changes stock variable as needed. Creates a new entry in the customer’s history
* processBorrow(int stock, CustomerID, MediaType, MovieType): takes care of Borrow Transaction
* processReturn(int stock, CustomerID, MediaType, MovieType): takes care of Return Transaction
* viewHistory(CustomerID): Displays history of the passed in customer
* Inventory(): Displays the stocks in the store

## Main

The main file will be responsible for creating a new store object, and then passing in file names to the public Store methods readMovies, readCustomers, and readTransactions.

## .h Files

store.h: represents store. Keeps track of the stock and customers. Parses through files and creates objects and calls its methods accordingly

vector Dramas

vector Classics

vector Comedies

HashTable customers

Constructor: Store

readMovies(File)

readCustomers(File)

readTransactions(File)

customer.h: represents a customer. Keeps track of the customer’s ID and their history of transactions

Constructor: Customer(Customer ID, Name, vector History)

addToHistory(Transaction)

transaction.h: represents and processes transactions.

Constructor: Transaction(Transaction type, Customer, Movie)

Constructor: Transaction(Transaction type, Customer)

processTransaction()

movie.h: parent class that represents a general movie

comedy.h: subclass of movie that represents comedy movies

Constructor: Comedy(Stock, Director, Title, Year it released)

classic.h: subclass of movie that represents classical movies

Constructor: Classic(Stock, Director, Title, Major actor, Release date)

drama.h: subclass of movie that represents drama movies

Constructor: Comedy(Stock, Director, Title, Year it released)