Team Rito
Xiaojie (Aaron) Li, Joan Chirinos, Johnny Wong

Final Project Proposal

Command Line-Based Movie Rental Kiosk

Our team will create a movie rental kiosk based in the command line. This kiosk will have access to a large collection of movie titles of varying genres from various dates. This collection will be in a CSV file format. To view this collection of movies, we will implement a search engine for specific titles as well as categorization for general searching through our assortment of films like genre, year, and the length of the movie. The user will be able to rent as many movies as they want through our checkout system and finalize their order through our implemented credit card system. Our credit card system will have a pin and a card number, as well as expiration dates similar to real credit cards. Once the order is finalized, a receipt will be printed out showing the time and date the rental was purchased and what the title of the movie is.

We plan to have our credit card system be a subclass of BankAccount.java (possibly with modifications to the original) to utilize its methods. We will create a collection of valid credit cards in order to simulate a read world database of credit card numbers and pin codes. For further protection, there will be an encryption method for the pin number. In order to implement expiration dates for credit cards, we will have to learn more about how to use the Calendar class from java.utils. Another planned use we have for the Calendar class is to track the number of days a movie has been rented to give our kiosk the ability to procure the designated cost for rentals. To test this, we will write a method to fast forward to certain dates to be able to actually return the rental and pay fees that are due.

In this project, there will be error handling in all aspects to give our kiosk more realism. We will deal with issues such as the user having insufficient money, invalid credit card numbers and pins, and expired credit cards. The only real limitation of this project is that Calendar class attempts to use the user's local time, so someone trying to test this class would normally have to wait for weeks to finalize their testing. The fast-forward method discussed in the above paragraph is our way around this limitation.

Kiosk is an object, and it prints user dialogue, and other than that, its main purpose is to call the other classes which implement its features. We can separately build Kiosk apart from the other objects: an instance of CreditCard implements credit card methods, an instance of class called Checkout that solely introduces features to do with processing a rental, as well as an instance of Movie class in which it implements features that will return the attributes of movies, such as the year the movie was made, or the title. There is no inheritance involved in our project, because all the objects are independent, and Kiosk is the wrapper class in the sense that its sole purpose is to call the other object classes, as well as to output user information based on that. Various instance variables will be implemented first in the various objects, such as title, date, and genre variables for the Movie class, and date, number of rentals variables for the Checkout class. Due to this, our project will be developed bottom-up, since the most primitive variables and methods will be designed before rising in complexity.

Team Rito
Xiaojie (Aaron) Li, Joan Chirinos, Johnny Wong