**Project Report**

**Please run AppStarter instead of App due to a javafx bug**

**Abstract:**

**VideoCo is a small video store with 2 stores in Toronto. VideoCo wants to expand its business and start renting videos worldwide through the Internet. They plan to establish a few warehouses scattered around the globe and a new computer system. Your help is needed in specifying this complex project.**

**In order to make above activities happen, I designed and implemented this application – online video store management system. This application mainly employs MVC and singleton design patterns and provide multiple useful functions for Customers, admins, operators and warehouse admins.**

**With the help of my video store management system:**

* **Registered customers can order movies online.**
* **Registered customers can pay their orders online.**
* **Unregistered customers can dial in to ask an operator to place an order.**
* **System admin can manage database like adding/removing movies, change order status, remove/modify customer account.**
* **Warehouse admin can view orders waiting to be shipped and change the order status to shipped.**

**Outline:**

1. **Business rules assumptions**
2. **Database**
   1. **Database format**
   2. **Schema description**
3. **Test case**
4. **Business rule assumption**

* **For each order, only 1 copy of each movie can be added.**
* **Current database is empty, i.e. you’ll need to sign up as admin and use admin account to add movies into database first before you try may customer-related functions.**
* **Customers cannot place new order if this customer has an unpaid order**
* **Since calculating real distance between places requires another level of knowledge, the app will randomly choose a warehouse to simulate ‘Choosing the closest warehouse’ for now**
* **Customer must pay with a credit card with the name registered with the account**
* **The business process of order status is unpaid-paid-out for delivery-delivered-returned**
* **Root user can update database to check late returned orders to simulate the daily checks. The update is assumed to happen daily, which means if it’s pressed multiple time within a day the late fees will be wrong**
* **‘Hourly send orders to warehouse’ is simulated by presenting paid orders when warehouse page is opened for now.**
* **For company email, admins’ emails end with @videostoreadmin.com, operators’ end with @videostoreoperator.com, warehouse admins’ end with @storevideowarehouse.com**
* **Password for root is root.**

1. **Database**
   * **Database format**

**MySQL database.**

* + **Schema description**

**There are 6 tables in the database.**

**User table is used to store information of registered customers.**

**Admin table is used to store credentials of system admin.**

**Operator table is used to store credentials of operators.**

**Warehouseadmin table is used to store credentials of warehouse admin.**

**Movie table is used to store information of each movie.**

**Orders table is used to store information of orders of all statuses**

**A copy of schema can be found in the same folder.**

1. **Test Cases Coverage 90%**

**I’ve designed this general test case to test the application.**

* + **Step 1: Login with empty credentials, wrong credentials**
  + **Step 2: Register two customer accounts, one in ON and one in another province**
  + **Step 3: Register one admin account**
  + **Step 4: Register one operator account**
  + **Step 5: Register one warehouse admin account**
  + **Step 6: Login with admin account and wrong/empty password**
  + **Step 7: Login in with the admin account**
  + **Step 8: Insert 4 movies, one of them has quantity 1**
  + **Step 9: Remove one of the movies, validate database**
  + **Step 10: Update one of the movies, validate database. Update customer info and validate database**
  + **Step 11: Logout, login with first customer account with multiple credentials**
  + **Step 12: Update account info, change something**
  + **Step 13: Put some movies into cart, try to put a movie which is already in car into cart**
  + **Step14: Remove one movie from cart and put in again, validate quantity left**
  + **Step 15: Place the order**
  + **Step 16: Try to place another order**
  + **Step 17: Pay the order with wrong card number, wrong name and correct info**
  + **Step 18: Logoff, login as warehouse admin, ship the order**
  + **Step 19: Logoff, login as admin, change the order status and delivery method**
  + **Step 20: Logoff, login as the second customer, place an order**
  + **Step 21: Cancel the order just placed, and place another one, test if a movie can be added to cart with quantity 0.**
  + **Step 22: Pay the order, logoff, login as warehouse admin, ship order**
  + **Step 23: Login as the second customer, try to cancel the shipped order**
  + **Step 24: Put something into cart, logoff, login again and check cart**
  + **Step 25: Logoff, login as operator, place an order and retrieve order information**
  + **Step 26: Login as admin, change the order status to paid**
  + **Step 27: Update the reward points of the first customer to 10**
  + **Step 28: Login as the first customer, verify reward points in account page**
  + **Step 29: Place an order and pay with reward points**
  + **Step 30: Login as admin, change all order status to delivered, delete this admin account**
  + **Add a test to test deleting customer account**
  + **Step 31: Login as root, change system time to 20 days later, update system**
  + **Step 32: Login as customers, pay the late fee**