Coupon System project

Chen Alkabets / Yonathan Azgad

1. System overview
2. UML diagram
3. System Components
    a. DAO objects
    b. Connection pool
    c. Facade objects
    d. Daily expiration task
    e. Coupon System class
    f. Exceptions
4. Directory structure
    a. Packages
    b. Folders and additional files
    c. External libraries (*.jars)
5. System properties
6. Running the system
7. Logging example of running system for more than 24H

1. System overview

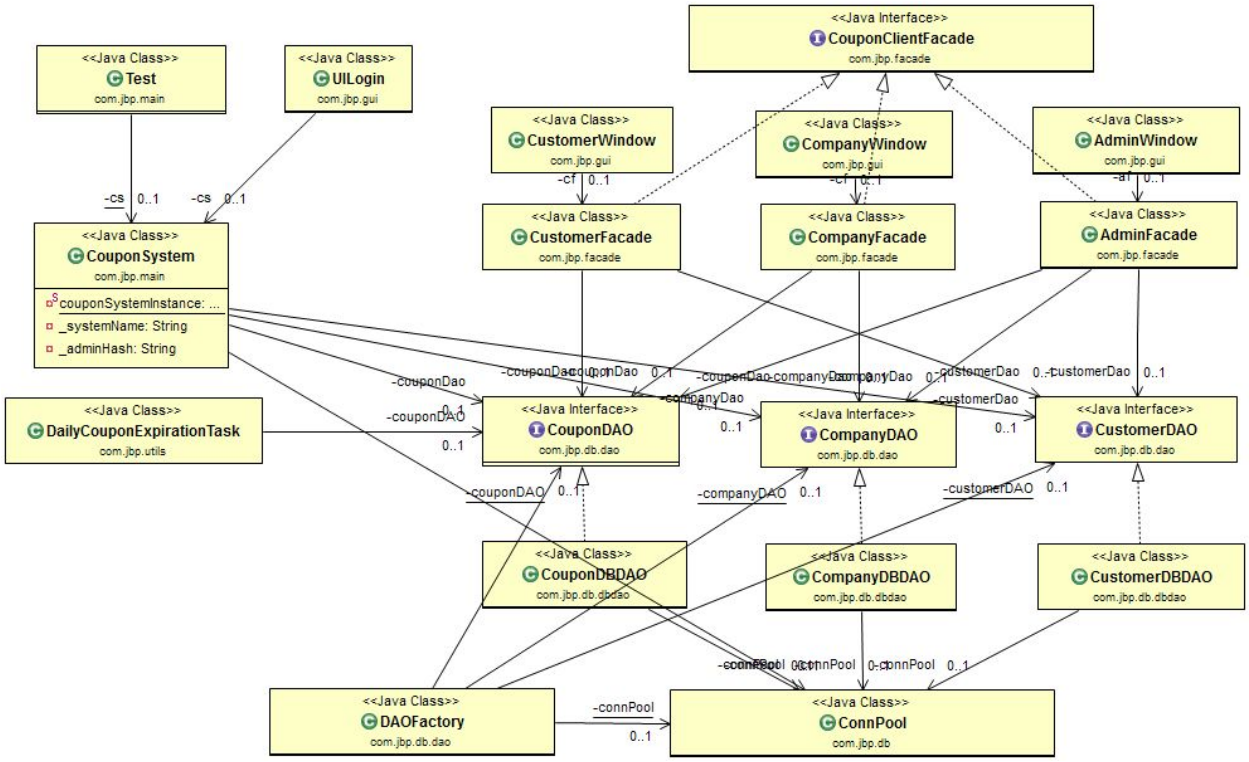
The CouponSystem enables companies to create coupons and customer purchase them.
The system can be logically divided into 3 modules :

1. Database module (bottom right side)
   using MySQL database hosted on the internet.
2. User module (upper right side)
   Facades and GUI.
3. CouponSystem main module (left side)
   Main system, management, configuration and logging.

The CouponSystem is configurable using the config/cs.properties file.
After changing the configuration, the application must be restarted to use the new values.

2. UML diagram was created by ObjectAid (Eclipse plugin).

Visual view of system logic.



3. System Components

a. DAO

DAO provides a common interface for C.R.U.D operations on a data-source. In this project the DBDAO is a concrete implementation of that interface which use the MySQL database as the underlying layer.

NOTE : if the data-source was XML files (instead of DB), an XMLDAO would be used to implement those operations on the XML files. no changes needed on the 'user code' since XMLDAO will implements the same DAO interface.

b. Connection pool

Singleton class to establish a DB connection pool to be used by all clients. Once system is on for the first time, a connection pool is created. Every new client will get an instace of the exist pool.

c. Facade

Facade provides a common simplified interface for user operations. in this project, 3 Facade objects were defined to provide those operations:

1. CustomerFacade : groups all operations of customers.

2. CompanyFacade : groups all operations of companies.
3. AdminFacade : groups all administration tasks.

d. Daily expiration task

A runnable object which is scheduled by the main CouponSystem program. the object will run once every 24H and will clean expired coupons (coupons that were expired already by the end-date field.

NOTE : 24H (1440 Minutes) is the default value, but can be configured in cs.properties file

e. Coupon System Singleton GUI class

This is the main system singleton class. it responsible for :

1. Load the cs.properties file.
2. Open a management JFrame (enables client login).
3. Initialize a connection pool.
4. Create 3 DAO's (using FactoryDAO for flexibility).
5. Start the Expiration task and schedule it for 24h.

f. Other GUI classes

Login window is a GUI which enable the user to login to the system.
3 login types available (Admin, Company and Customer), each will return a GUI JFrame upon successful login (with that particular Facade).
NOTE : This means that many users can be logged simultaneously.

1. AdminWindow : contains JTabbedPane with 3 panels :
    a. AdminCompaniesPanel
    b. AdminCustomersPanel
    c. AdminSystemPanel
2. CompanyWindow : a GUI for common companies operations.
3. CustomerWindow : a GUI for common customer operations UI.

g. Exceptions

Custom exception classes available for the system. The DAO receives SQL exceptions from the database and throws a new custom exception to the caller. The Facades also throws up the exceptions to be handled by the GUI and will show the appropriate message.

4. Directory Structure

```
├───config
│       cs.properties
│
├───docs
│       couponsystem.pdf
│       csuml.png
│       csuml.ucls
│
├───src
│   ├───com
│   │   └───jbp
│   │       ├───beans
│   │       │       Company.java
│   │       │       Coupon.java
│   │       │       CouponType.java
│   │       │       Customer.java
│   │       │
│   │       ├───db
│   │       │   │   ConnPool.java
│   │       │   │
│   │       │   ├───dao
│   │       │   │       CompanyDAO.java
│   │       │   │       CouponDAO.java
│   │       │   │       CustomerDAO.java
│   │       │   │       DAOFactory.java
│   │       │   │
│   │       │   ├───dbdao
│   │       │   │       CompanyDBDAO.java
│   │       │   │       CouponDBDAO.java
│   │       │   │       CustomerDBDAO.java
│   │       │   │
│   │       │   └───xmldao
│   │       │           CustomerXMLDAO.java
│   │       │           XMLTester.java
│   │       │
│   │       ├───exceptions
│   │       │       CompanyCreationException.java
│   │       │       CompanyNotFoundException.java
│   │       │       CompanyRemovalException.java
│   │       │       CompanyUpdateException.java
│   │       │       CouponCreationException.java
│   │       │       CouponNotAvailableException.java
│   │       │       CouponRemovalException.java
│   │       │       CouponUpdateException.java
│   │       │       CustomerCreationException.java
│   │       │       CustomerNotFoundException.java
│   │       │       CustomerRemovalException.java
│   │       │       CustomerUpdateException.java
│   │       │
│   │       ├───facade
│   │       │       AdminFacade.java
│   │       │       CompanyFacade.java
│   │       │       CouponClientFacade.java
│   │       │       CustomerFacade.java
│   │       │
│   │       ├───gui
│   │       │       AdminCompaniesPanel.java
│   │       │       AdminCustomersPanel.java
│   │       │       AdminSystemPanel.java
│   │       │       AdminWindow.java
│   │       │       CompanyWindow.java
│   │       │       CustomerWindow.java
│   │       │       UILogin.java
│   │       │
```

```
|   |       ├───main
|   |       |       ClientType.java
|   |       |       CouponSystem.java
|   |       |       Test.java
|   |       |
|   |       └───utils
|   |               DailyCouponExpirationTask.java
|   |               Severity.java
|   |               Utils.java
|   |
|   └───images
|           company.png
|           coupon.png
|           customer2.png
|           system.png
|
└───xml
        companies.xml
        company_coupon.xml
        coupons.xml
        customers.xml
        customer_coupon.xml
```

a. Packages :

com.jbp.beans : *classes that describe the beans used in the system.*

com.jbp.db.* : *classes and interfaces for accessing the database.*

com.jbp.exceptions : *custom exceptions classes.*

com.jbp.facade : *Facade classes that provides user interaction.*

com.jbp.gui : *Swing classes of the system main app, and for client login and operations.*

com.jbp.main : *CouponSystem main class, and a Test.java utility.*

b. Additional folders and files :

/images : *Contains *.png icons for Swing components*

/config/cs.properties : *Main configuration file of the system. values loaded upon starting the application.*

/docs/csuml.png : *UML diagram image..*

/docs/csuml.ucls : *UML diagram file, created with ObjectAid Eclipse plugin..*

/docs/couponsystem.pdf : *This document.*

/xml : *5 files that represents the database structure as 5 XML files. although not fully implemented, it shows how the data source can be replaced without changing the client code. since client code relies on DAO interfaces, implementation can be changed without notify the user. changing includes changes in the DAOFactory class that is responsible to create the concrete object.*

c. External libraries (*.jar):

JTattoo-1.6.11.jar : *Provides LookAndFeel templates that can be used by modifying the config/cs.properties file.*

mysql-connector-java-5.1.35-bin.jar : *JDBC connector, provides access to a mysql database.*

5. System properties

System properties is configurable via config/cs.properties. below is the content of the file : *NOTE : after changing this file the application must be restarted to get the new values..*

```
# cs.properties file holds global application configuration.
# invalid parameters may lead to unexpected behavior
##############################################################################

# md5(1234) = 81dc9bdb52d04dc20036dbd8313ed055
# admin should login with password of '1234'
# md5 calculator : http://www.md5.cz/
ADMIN_HASH=81dc9bdb52d04dc20036dbd8313ed055

# available templates name appears below
LOOK_AND_FEEL_TEMPLATE=com.jtattoo.plaf.graphite.GraphiteLookAndFeel

# max TCP connections in pool
MAX_DB_CONNECTIONS=2

# global name for this singleton system
SYSTEM_NAME=KoopOnLine-2015

# server ip address/credentials and optional parameters
# allowMultiQueries=true need to be passed to enable few queries at once
# ie . 'SELECT .... ; SELECT ....'
DB_URL=jdbc:mysql://appsrv.kukinet.net/DBCoupons?allowMultiQueries=true
DB_USERNAME=admin
DB_PASSWORD=1234

# 1440M = 24H
THREAD_INTERVAL_MINUTES=1440

# enable / disable system logs
LOGGING_ENABLED=true

# lookAndFeel templates
##############################################################################
# com.jtattoo.plaf.aero.AeroLookAndFeel
# com.jtattoo.plaf.aluminium.AluminiumLookAndFeel
# com.jtattoo.plaf.bernstein.BernsteinLookAndFeel
# com.jtattoo.plaf.fast.FastLookAndFeel
```

6. Running the system

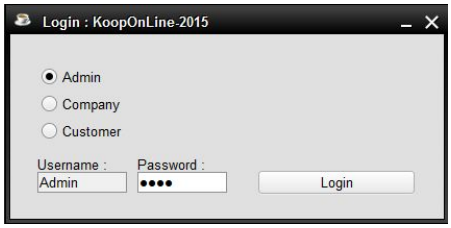
The project compressed asa jar file. following those steps to import to Eclipse :

1. Open a new Java project, give it a name and accept all defaults.
2. Right click on project → import → Archive File → [...path_to...]cs.jar .
3. Jar file will be extracted to the correct directory structure
4. 2 additional *.jar files provided, needed to build the project.
5. Run the com.jbp.main.Test.java

The main application window

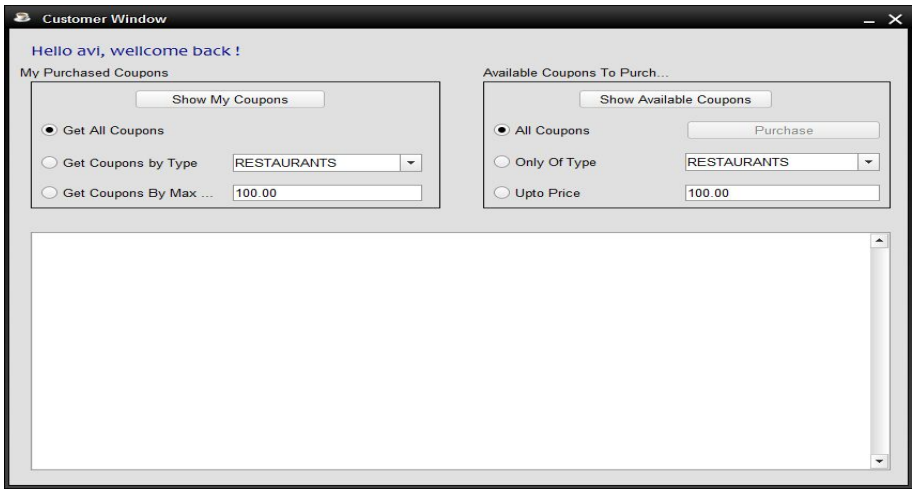
- *Showing the current configuration loaded from config/cs.properties file.*
- *Button to stop/restart the Daily Expiration Task.*
- *Button to shutdown (the 'X' button will not close the app !) - enforcing 'clean exit'.*
- *Login button to open a Login window.*

Pressing on Login will open the following Login window :



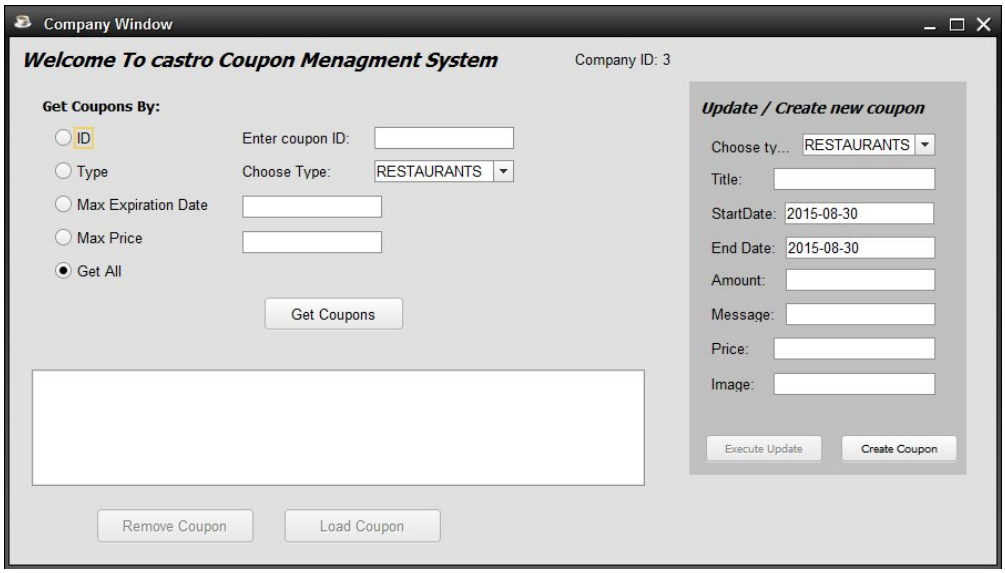
- *Customer login validated against db (case insensitive). upon successful login, a CustomerFacade will be returned and a CustomerWindow will open for that customer.*
- *Company login validated against db (case insensitive). upon successful login, a CompanyFacade will be returned and a CompanyWindow will open for that company.*
- *Admin login is authenticated by MD5 hash. The system calculates the MD5 hash of the provided password and compare against the HASH parameter in the cs.properties file. upon successful login, an AdminFacade will be returned and an AdminWindow will open. (the current admin password is 1234)*

Customer Window :



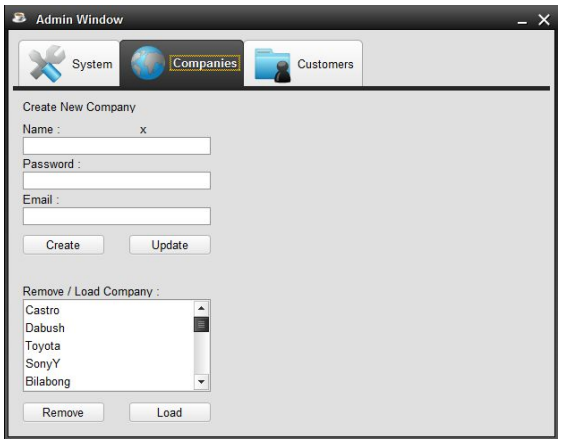
- *The left side provides access to coupons owns by this customer*
- *The right side provides access to coupons available for purchase*

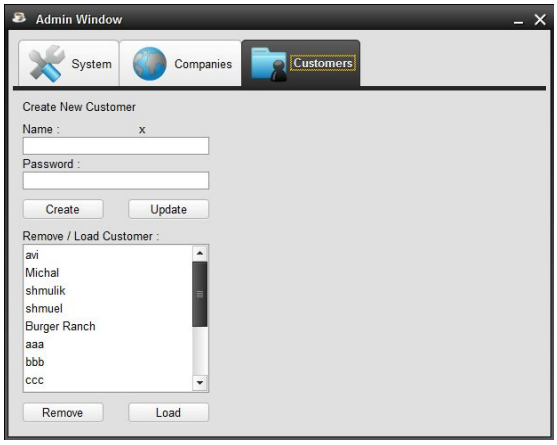
Company Window :



- *Provides access to coupons owns by this company.*
- *Options to create / remove / update a coupon.*

Admin Window contains 3 panels (System, Company, Customer)





- *System panel shows the cs.properties file and the current system uptime.*
- *Company panel provides administration tasks on companies (CRUD).*
- *Customer panel provides administration tasks on customers (CRUD).*

NOTES :

- Username / Passwords are case insensitive
- System main window cannot be closed by pressing on ‘X’.
- XML not implemented.
- Admin can see passwords (as a ToolTip on the password field).
- Closing the app can be done from the main window only.
- GUI windows can be closed without closing the main application.
- Tested on JRE 1.8.0_60 / windows 7.

7. Logging example of system running for more than 24h

```
# server ip address/credentials and optional parameters
example of log output during 24 hours :

2015-08-26 23:03:32.731      INFO   CouponSystem   Coupon system created.
2015-08-26 23:03:32.735      DEBUG  Utils          loadSystemParameters() invoked
2015-08-26 23:03:32.735      DEBUG  Utils          properties from file loaded to a hashmap
2015-08-26 23:03:32.736      DEBUG  CouponSystem   hashmap assigned to local vars, config applied.
2015-08-26 23:03:32.852      DEBUG  ConnPool       private c'tor invoked for this singleton.
2015-08-26 23:03:33.913      DEBUG  ConnPool       Connection object created.
2015-08-26 23:03:34.63               DEBUG  ConnPool       Connection object created.
2015-08-26 23:03:34.63               INFO   ConnPool       singleton c'tor finished. Total of 2 DB
Connections.
2015-08-26 23:03:34.631      INFO   DAOFactory     factory initialize connection pool
2015-08-26 23:03:34.632      INFO   DAOFactory     CustomerDBDAO instantiated.
2015-08-26 23:03:34.634      INFO   DAOFactory     CompanyDBDAO instantiated.
2015-08-26 23:03:34.635      INFO   DAOFactory     CouponDBDAO instantiated.
2015-08-26 23:03:34.635      INFO   CouponSystem   3 DAO objects created.
2015-08-26 23:03:34.635      INFO   CouponSystem   Thread scheduler started. interval = 1440 minutes.
2015-08-26 23:03:34.666      INFO   ExpirationTask Daily thread started.
2015-08-26 23:03:34.666      INFO   ConnPool       connection provided by pool. free connections 1/2
2015-08-26 23:03:34.75               INFO   CouponDBDAO    old coupons returned.
2015-08-26 23:03:34.75               INFO   ConnPool       connection returned to pool. free
connections 2/2
2015-08-26 23:03:34.75       INFO   ExpirationTask no old coupons to remove. cleaner thread has nothing
to do
2015-08-26 23:03:34.75               INFO   ExpirationTask Daily thread finished.
2015-08-26 23:03:35.94               DEBUG  UILogin        UI Login window created.
2015-08-26 23:03:39.175      DEBUG  CouponSystem   admin login invoked with pass = 1234
2015-08-26 23:03:39.175      DEBUG  CouponSystem   db hash = 81dc9bdb52d04dc20036dbd8313ed055
2015-08-26 23:03:39.175      DEBUG  CouponSystem   calculated hash = 81dc9bdb52d04dc20036dbd8313ed055
2015-08-26 23:03:39.175      DEBUG  CouponSystem   admin authenticated by md5.
2015-08-26 23:03:39.198      DEBUG  CompanyDBDAO   getAllCompanies() invoked.
2015-08-26 23:03:39.198      INFO   ConnPool       connection provided by pool. free connections 1/2
2015-08-26 23:03:39.271      INFO   CompanyDBDAO   companies list returned successfully.
2015-08-26 23:03:39.271      INFO   ConnPool       connection returned to pool. free connections 2/2
2015-08-26 23:03:39.306      DEBUG  CompanyDBDAO   getAllCompanies() invoked.
2015-08-26 23:03:39.306      INFO   ConnPool       connection provided by pool. free connections 1/2
2015-08-26 23:03:39.378      INFO   CompanyDBDAO   companies list returned successfully.
2015-08-26 23:03:39.378      INFO   ConnPool       connection returned to pool. free connections 2/2
2015-08-26 23:03:39.391      INFO   ConnPool       connection provided by pool. free connections 1/2
2015-08-26 23:03:39.463      INFO   CustomerDBDAO  customers list returned successfully.
2015-08-26 23:03:39.463      INFO   ConnPool       connection returned to pool. free connections 2/2
2015-08-26 23:03:39.463      INFO   AdminFacade    getAllCustomers() returns.
2015-08-27 23:03:34.666      INFO   ExpirationTask Daily thread started.
2015-08-27 23:03:34.67               INFO   ConnPool       connection provided by pool. free
connections 1/2
2015-08-27 23:03:34.7        INFO   ConnPool       connection returned to pool. free connections 2/2
2015-08-27 23:03:34.7        INFO   ExpirationTask no old coupons to remove. cleaner thread has nothing
to do
2015-08-27 23:03:34.7        INFO   ExpirationTask Daily thread finished.
```