*Technical University of Cluj-Napoca  
Faculty of Automation and Computers  
Department of Computer Science  
2nd Semester 2012-2013*

*Programming Techniques*

*Homework 2*

Pomian Aurelian – 30425

Contents

1. Problem specification …………………………………… 3
2. Design ……………………………………………………. 4
   1. Use Case Diagram …… . . . …………………. ….. 4
   2. Class Diagram ………………. . . .………………… 5
   3. Sequence Diagrams…..…. . ……………………… 6
   4. Activity Diagram..………………………………….. 7
   5. User Interface . . . ...……………………………….. 9
3. Packages . . . . . . . . . …………………………………… 13
4. Conclusions ……………………………………………… 14
5. References ……………. . . . . .…………………………. 14
6. ***Problem Specification***

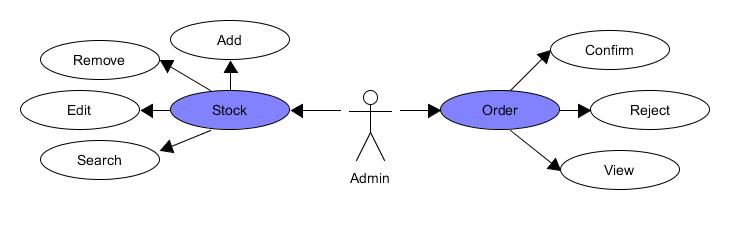
**Homework 2**

The objective of the first homework is the following: “Consider an application OrderManagement for processing customer orders. The application uses (minimally) the following classes: Order, OPDept (Order Processing Department), Customer, Product, and Warehouse. The classes OPDept and Warehouse use a BinarySearchTree for storing orders.” In other words we must create a program to simulate an online shop. The program must contain a customer interface with its specific actions and a admin interface, again with its specific actions.

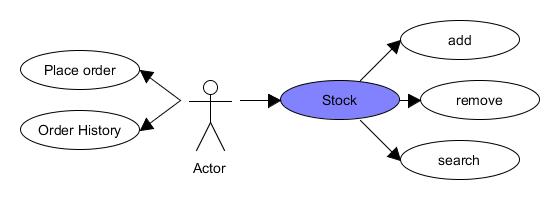
To solve this problem I first considered using a StateManager to manage and organize the UI windows ( MainMenu, Customer, Admin, CreateNew ). Then I started to implement the required classes ( OPDept, Warehouse, … ) . To store information about user accounts, orders, products in stock I used text files, that will append write when adding new products, orders or that will rewrite when updating the data. This way both customer and admin can access warehouse for their own purpose.

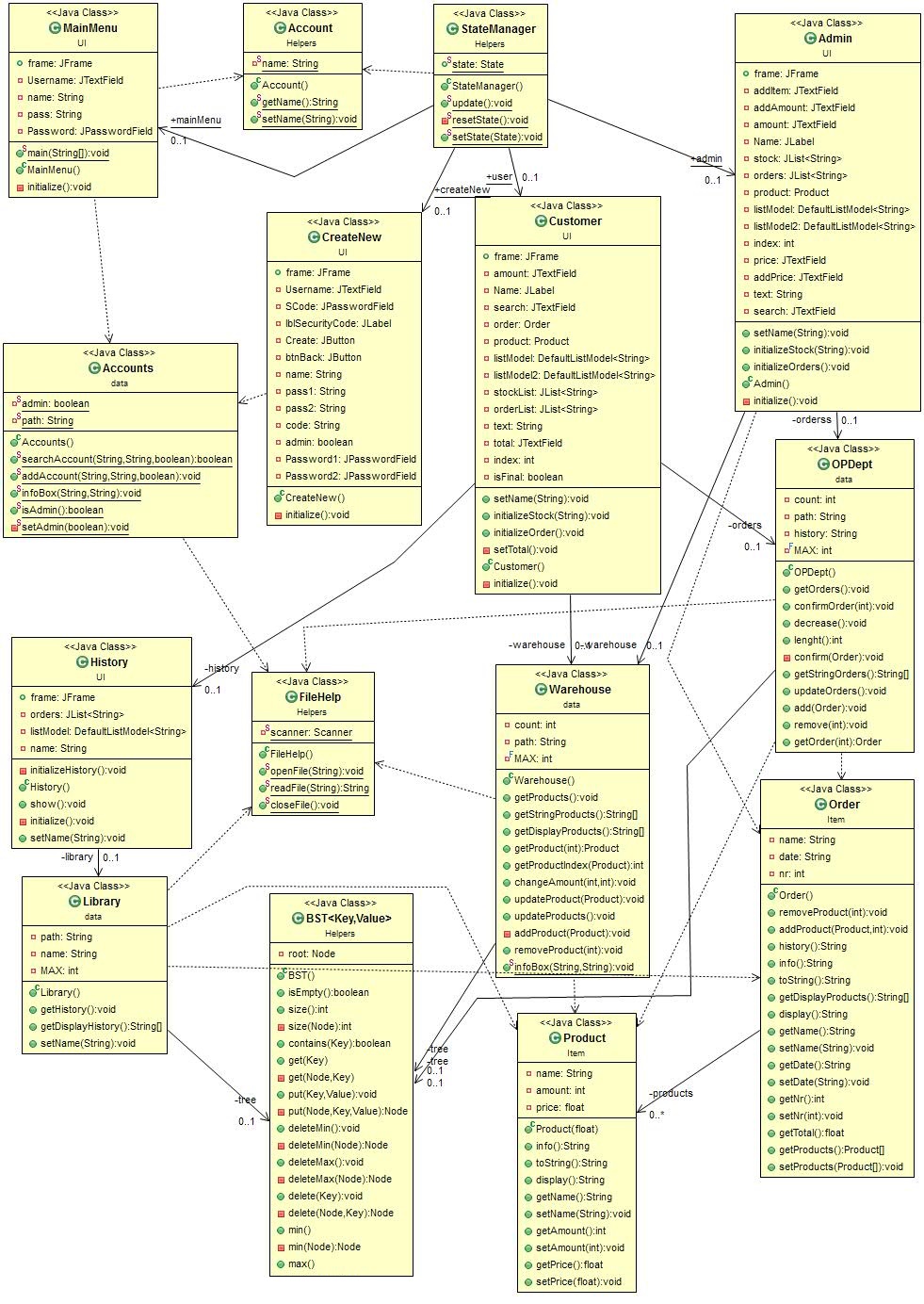
1. **Design**
   1. Use Case Diagrams

For Admin View:



For Customer View:

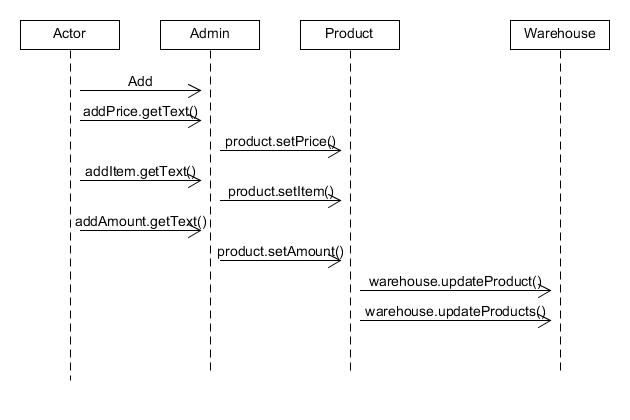


* 1.  Class Diagram

The class diagram show what classes the program contains, what methods and variables they have and how they are connected.

* 1. Sequence Diagrams

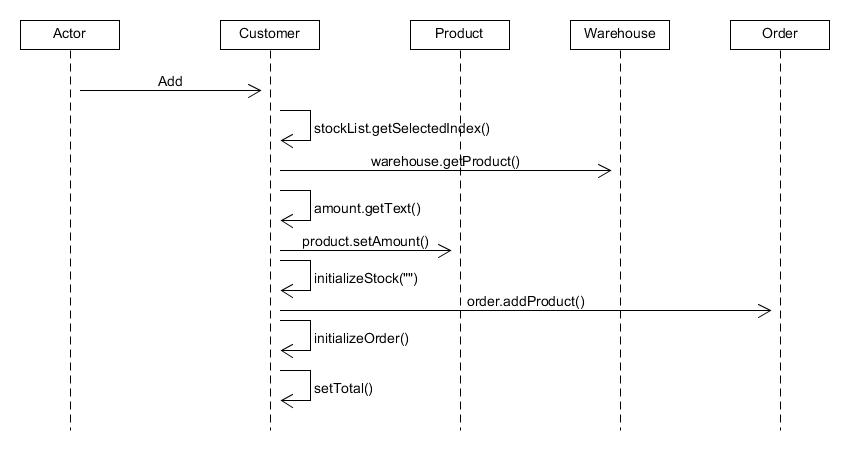
The sequence diagram shows the methods that are called and from what class when the user performs a action.

The sequence diagram below shows what happens when the admin wants to add a new product to the warehouse stock.

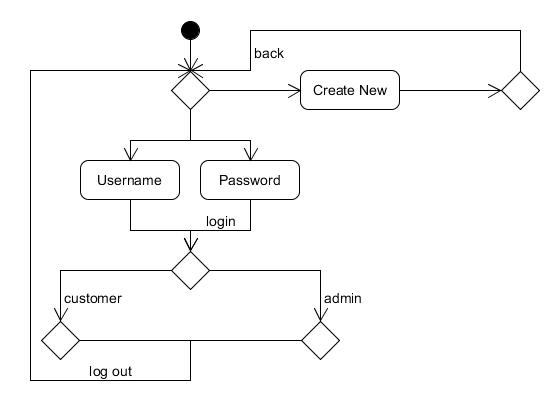
. . . . . . . . . . . . .. . . . . . .. . . . .

. . . . . . .. . . . . . . . .. . . . . . . . .. . . . .

.. . . . . . . . . .. . . . . . . . . . . . . .. . .. . . . . . . . . . . . . . .

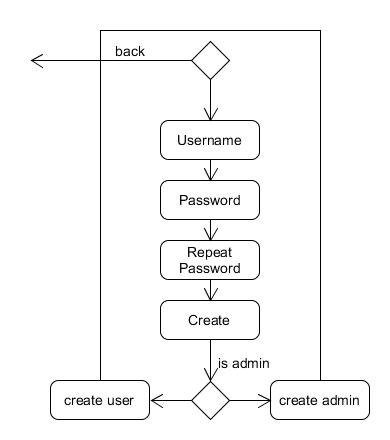
The next diagram shows what happens when the user wants to add a product from the warehouse to the order.

* 1. Activity Diagram

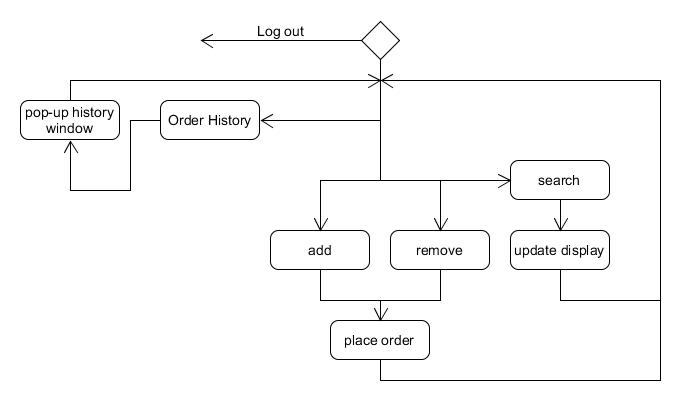
****The activity diagram captures the dynamic behavior of the system.

Main

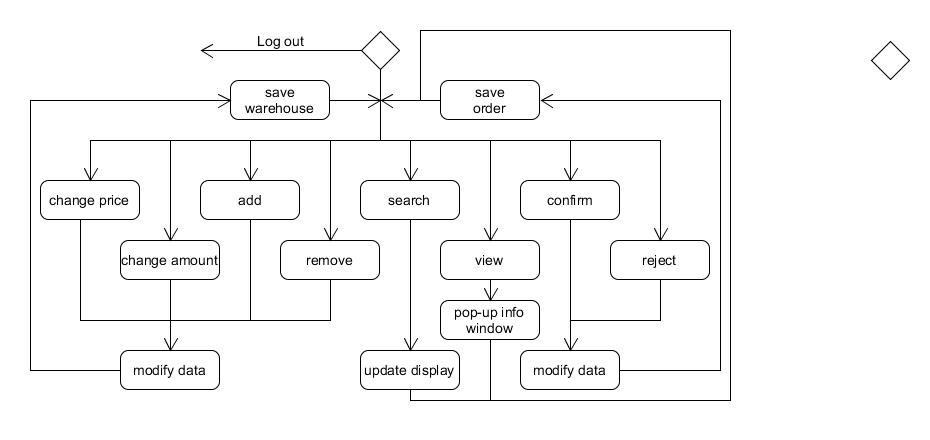
Menu

****

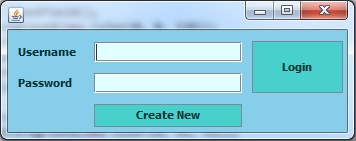
Create New

****User

Admin



* 1. User Interface
* Main Menu

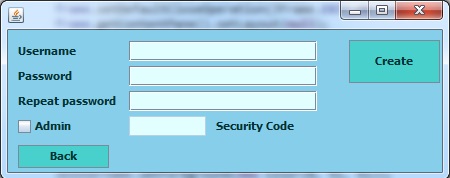


. . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

The Main Menu window is used only for Logging in or to go to the Create New window and create a new account.

It contains 2 labels, 2 text fields and 2 buttons.

* Create New



. . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . .

The Create New window is used to create a new account. Based on the checkbox “Admin” it will determine if the account is a customer or and admin.

To create an admin account you have to know the security code which hard-coded in the program (it is admin (for testing purpose)).

Error messages are implemented in order to ensure the user fills the required fields, such as “password field is empty” or “passwords do not match”.

There is a Back button which will return you the Main Menu window.

* Admin

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

The Admin interface is the most complex because it can do almost anything regarding the stock and the orders (It can’t modify orders).

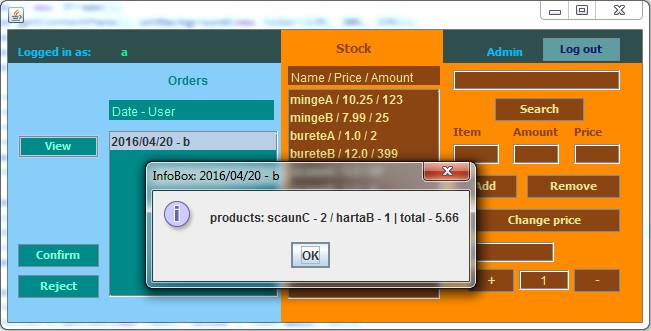
There 2 separate sections, one for orders and one for stock.

For stock the admin can search by name, add new products, remove products, modify the price or the amount.

For orders the admin can confirm , reject or view the order selected (view products and total price).

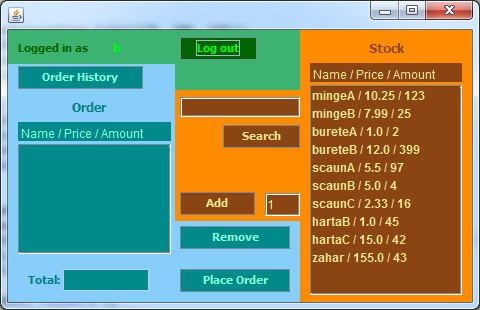
Confirming an order will send to order to order history.

Rejecting an order will remove that order and resupply the stock with that orders products.

Viewing an order:

Clicking view on a selected order will pop-up and info-box about the order, displaying the products and the total price of the order.

* Customer



. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

. . . . . . . . . . . . . . . . . . . . . . . . .

The customer view also contains 2 separate sections, one for order placement and one for the available stock.

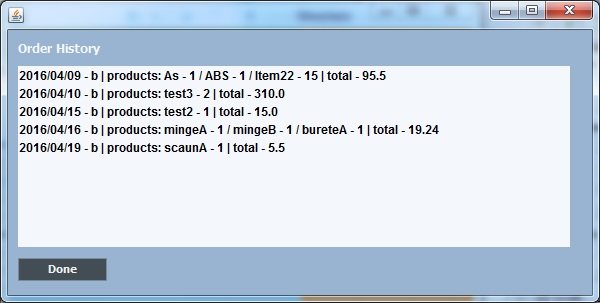
The customer can add/remove products to/from the order. He can also search products by name.

The customer has one uneditable text field used to display the total price of the current order.

The customer can only place an order while the “Total” field is not empty.

The Order History button will pop-up the History window and allows the customer to see all his past orders (only if the orders we’re confirmed)

Order History view:



The Order History view displays the date, username, products and the total price for each past confirmed order.

Done button will close the window.

Note: both admin and customer can logout and can see their account name on the top left of the window while logged in.

1. ***Packages***

A Java package is a mechanism for organizing Java [classes](http://en.wikipedia.org/wiki/Class_%28computer_science%29) into [namespaces](http://en.wikipedia.org/wiki/Namespace_%28computer_science%29). Java packages can be stored in compressed files called [JAR files](http://en.wikipedia.org/wiki/JAR_file), allowing classes to download faster as a group rather than one at a time. Programmers also typically use packages to organize classes belonging to the same category or providing similar functionality. A package provides a unique namespace for the types it contains. Classes in the same package can access each other's package-access members.

A package allows a developer to group classes (and interfaces) together. These classes will all be related in some way – they might all have to do with a specific application or perform a specific set of tasks.

For this application the following packages are imported, each of them having a certain role for the proper working of the application. We import them in the Gui Class (most of them relate to the user interface properties):

* **import java.awt:** Contains all of the classes for creating user interfaces and for painting graphics and images. A user interface object such as a button or a scrollbar is called, in AWT terminology, a component. The Component class is the root of all AWT components.
* **import javax.swing:** Typical Swing applications do processing in response to an event generated from a user gesture. For example, clicking on a JButton notifies all ActionListeners added to the JButton. That’s why we use this package for creating the user interface Gui.

1. ***Results***

The result of the homework is a user-friendly application that acts like an online shop, with a logging in window, create new account window, user-view and admin-view. For both admin and user the application provides the required activities that each can perform.

The interface is easy to use and understand.

1. ***Conclusions***

Achieving such a program may be hard both in terms of algorithms, graphical structure.

Some difficulties appeared while working with JLists that used to display orders, products and history, that took more time than expected.

Ways to upgrade the program:

* Add more functions for both admin and user such as:
  + Modifying orders ( admin )
  + Modifying accounts ( admin )
  + Add a message when rejecting an order which is sent to the customer ( admin )
  + Notifications ( from admin to customer (messages) )

1. ***References***

[*http://stackoverflow.com/*](http://stackoverflow.com/)

<http://www.google.ro/>

http://www.youtube.com/