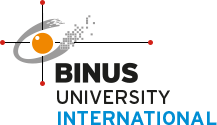
**JAVA PROJECT REPORT**



**ONLINE SHOPPING SYSTEM**

Submitted By:

Sunny Jovita

2301939046

FACULTY OF COMPUTER SCIENCE

BINUS INTERNATIONAL UNIVERSITY

Even Semester 2020



**BINUS UNIVERSITY**

**BINUS INTERNATIONAL**

**Assignment Cover Letter**

**(Individual Work****)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | | |  | |  | |
| **Student Information**: | **Given Names** |  | **Student ID Number** | |  | |
| 1. | Sunny Jovita |  |  | 2301939046 |  |
|  |  |
| **Course Code** | **: COMP6510** |  |  | **Course Name** | **: Programming Languages** |
| **Class** | **: L2BC** |  |  | **Name of Lecturer(s)** | **:** Jude Joseph Lamug Martinez, MCS |
|  |  |  |  |  |  |
| **Major** | **: Computer Science** |  |  |  |  |
| **Title of Assignment**  (if any) | **:** **Java Report** |  |  |  |  |
| **Type of Assignment**    **Submission Pattern** | **: Final Project** |  |  |  |  |
| **Due Date** | **: June 20 2020** |  |  | **Submission Date** | **: June 20 2020** |

The assignment should meet the below requirements.

1. Assignment (hard copy) is required to be submitted on clean paper, and (soft copy) as per lecturer’s instructions.
2. Soft copy assignment also requires the signed (hardcopy) submission of this form, which automatically validates the softcopy submission.
3. The above information is complete and legible.
4. Compiled pages are firmly stapled.
5. Assignment has been copied (soft copy and hard copy) for each student ahead of the submission.

# Plagiarism/Cheating

Binus International seriously regards all forms of plagiarism, cheating and collusion as academic offenses which may result in severe penalties, including loss/drop of marks, course/class discontinuity and other possible penalties executed by the university. Please refer to the related course syllabus for further information.

# Declaration of Originality

By signing this assignment, I understand, accept and consent to Binus International terms and policy on plagiarism. Herewith I declare that the work contained in this assignment is my own work and has not been submitted for the use of assessment in another course or class, except where this has been notified and accepted in advance.

Signature of Student: (Name of Student)

1.  (Sunny Jovita)

1. **PROJECT SPECIFICATIONS**
2. Abstract

Nowadays, everyone is having busy with their schedule or tasks whether it is urban area or rural. People out there, are so busy in their life until they don’t get enough of time to fulfill their needs properly. Therefore, a system was created to assist people’s necessary which is called online shopping. Online shopping is a system where consumers can directly order various things from internet without going anywhere, just by sitting at home or any place. It is a form of electronic commerce or e-commerce. More and more business industry are implementing web sites providing functionality for performing commercial transactions. As these days, it is reasonable to say that online shopping is becoming a commonplace. The objective of this system is to develop a general purpose from e-commerce store where any products, such as books, clothes, accessories, electronics, furniture, and others can be bought from home through internet.

1. Introduction

This project is an online shopping system and programmed in Java (IntelliJ IDEA) by using Swing and AWT packages to create a GUI (Graphical User Interface). The main aim of this project is to allow the customer to shop virtually using the internet and let customers to buy items of their desire from the store. The other goals are to improve the services of customers. It maintains the details of customer payments, product receipts, additional products, and also cancelling orders as well. The interface will give an easy and simple performance. The system will also provide some features for customers such as payment by cod (cash on delivery), cashless payment, secure registration, and others. For the payment selection, the customer will be asked to fill a payment option based on their preference (cashless or cash on delivery). In this system, customers are free to browse the catalogue and select products of interest. The selected items may be collected in a shopping cart, and it will count the total price of the products. At the checkout time, more information such as name, address, and telephone number of the customer will be needed to complete the transaction. A notification such as a pop up message box will be displayed as soon as the order has been confirmed.

1. **SOLUTION DESIGN**

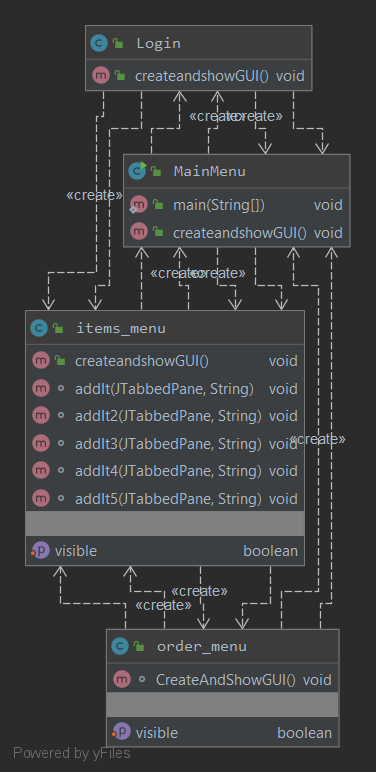
All the functionality about the system will be saved in classes named items\_menu, order\_menu, Login, and MainMenu. Inside each of the class, there are so many important functions that support the system well.

1. The needs



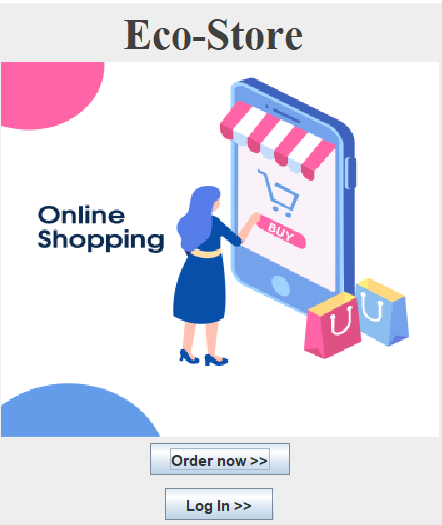
Before creating the code, there are some components that must be prepared to create the GUI. This program uses swing and AWT for the packages. This system also gives I.S.A (Interesting, Simple, and Attractive) display to satisfy the users by providing online details for all the products.

1. Class diagram

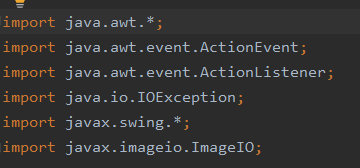


To represent a static view of an application, I created a UML class diagram or a structural diagram that can be used for visualizing, describing, and documenting from different aspects of this online shopping system. In this UML diagram, it shows a collection of classes (including their methods, constructors, properties, and inner classes) that are used for this online shopping system, and also their relationships or dependencies.

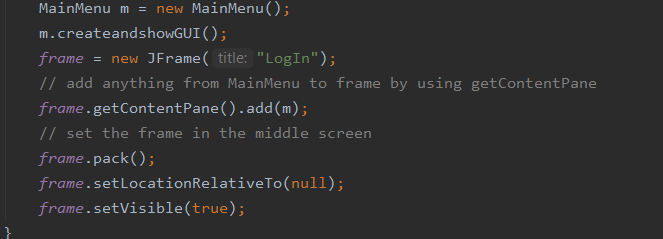
1. Cover display



This is the display of the main program. It shows the store’s name, a picture about online shopping, a login button for customers, and a button (“order now”) where the users could access and order the products directly without doing a login first. To create this frame, the system needs these kinds of packages:

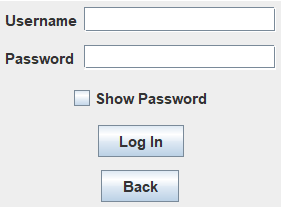


To build the main window where like labels, buttons, image are added within it, the program used JFrame since it is a container that provides a window on a screen. To create the login button and the order button, the system defines a class called JButton. This class is used to create a labeled button that has platform independent implementation. The result of this JButton will be displayed when the button is pushed.

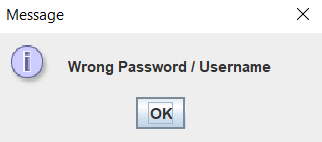


There are some methods that supports JFrame to create a good window. For example, pack() and setLocationRelativeto() methods are helping the frame to be set in the middle of our desktop screen.

1. Login display

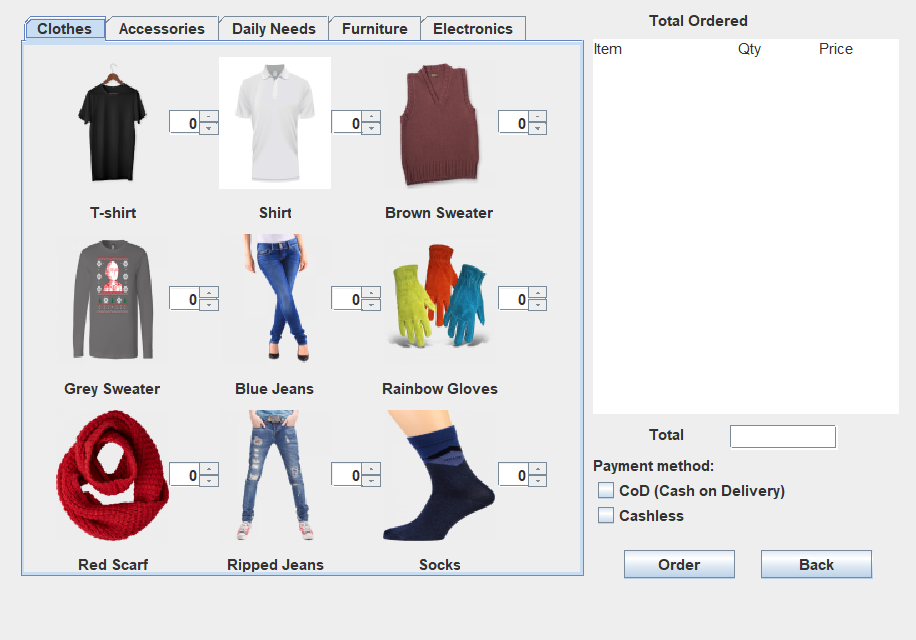


There are two options to help the users enter the menu display. The first one, they can click the “order now” button directly and the second is by input username and password. To make it more secure, the program provides a show password checkbox where the users are able to show or hide their password by clicking this show password checkbox. For the classes, the system uses JTextfield that creates a new empty text field, JCheckBox, JPasswordField, JButton, and also Jlabel.



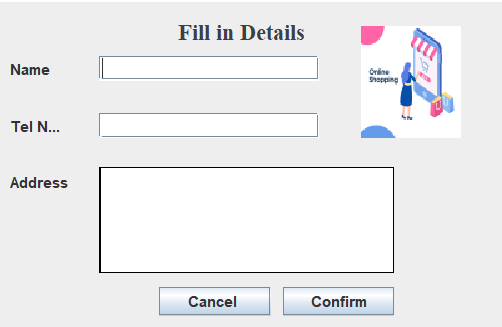
An alert box or a pop up message will be shown if the users enter the wrong password or username. This alert message will disappear if the users click the “OK” button.

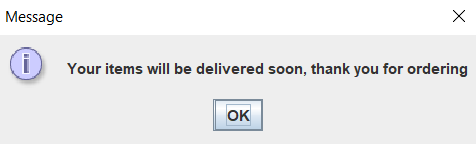
1. Main Menu display (include each panels)



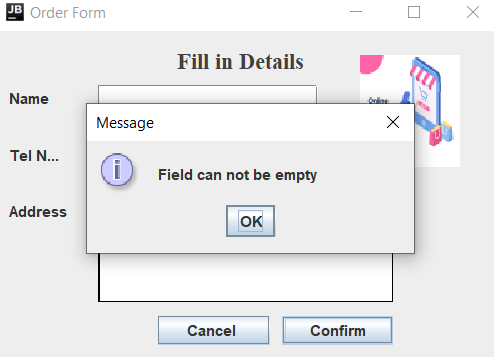
This is the final result of the main menu. As can be seen, there are some panels or tabs at the top of the screen that display products. Each of the tab will give a different objects view. The panel can be switched to another panel by pressing the button or by using keyboard arrows. On the right side of the main menu’s screen, there is a table that where the total items, price, quantities of the products are being placed.

1. Order menu display





The customer will be asked to fill the shipping information before they confirm their order. A thank you notification (a pop up message) will be displayed after they press the confirm button.



But, if the users have not fulfilled the form yet, an alert message will appear.

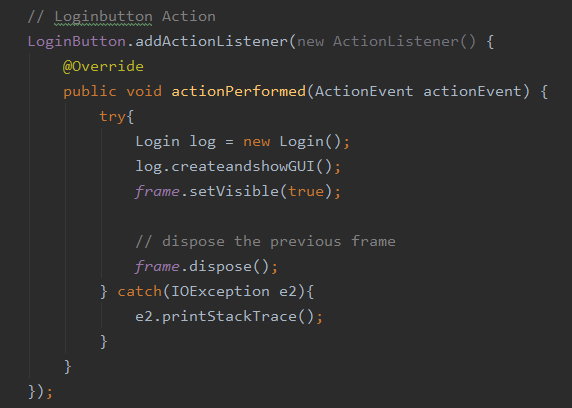
1. **DISCUSSION (how the system works)**

Since this online shopping system is divided into 4 classes, and they are related to each other in the forming of this program, there are some functions that the algorithms are kind a bit complicated to be solved. So in this part, I am going to explain those things separately (based on the class itself).

* Cover program

As mentioned before in the solution design (cover display’s part), this cover program is built by using JFrame, JPanel container classes, JButton and JLabel component classes. As it is known that a program will run well if there is two sections or more that are related to each other. It means that this program will work well when there is a displacement from this cover display to another section (whether it is login section or item’s menu section). JButton is used to connect those two sections. To make the button works, i used an interface called addActionListener.

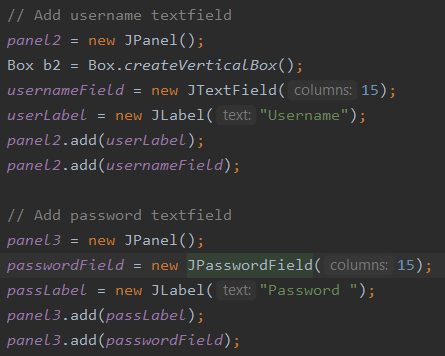




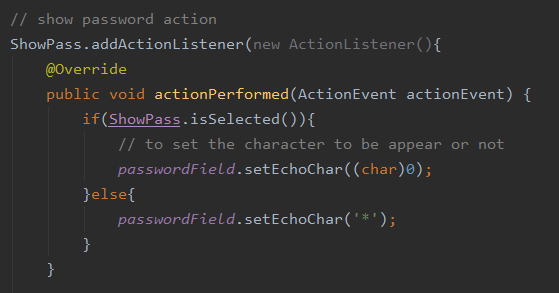
This addActionListener is an interface (from javax.awt.event.package) that allows the users to know when an action is performed in their JButton object, and react it to the intended class. The actionPerfomed function, in the picture is invoked automatically whenever the user click the button. For the conclusion, ActionEvent handles something that happens (in this case, when the button is being pressed, a signal from the button will be sent as an ActionEvent), and ActionListener will be the one who respond to that action.

* Login

In this login part, JFrame, JPanel, JButton, Jlabel are also needed to make this section. However, in order to make the textfield, to fill the password and the username, JTextfield and JPasswordfield are being used.

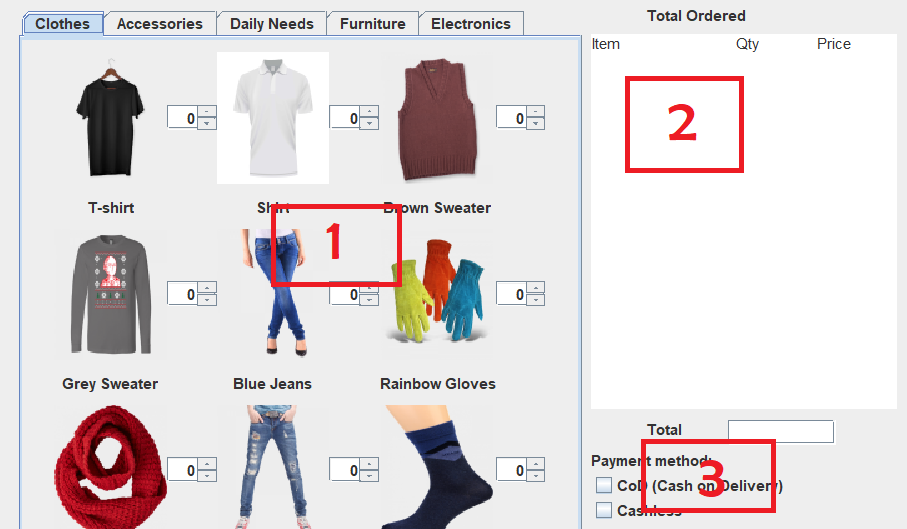


In order to make the password more secure or being hidden when the users typing it, I used JCheckbox to make the character appear or disappear.

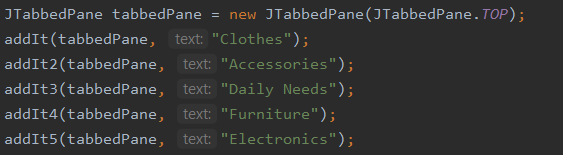


So, when the checkbox (named Showpass) is selected, the password will be set as a character, which is the password itself, it will be displayed inside the password text field. But when the checkbox is not selected, the password will be set to \* (a symbol, I used star in here) automatically. I also used the same algorithm (from the cover section) for the buttons as well.

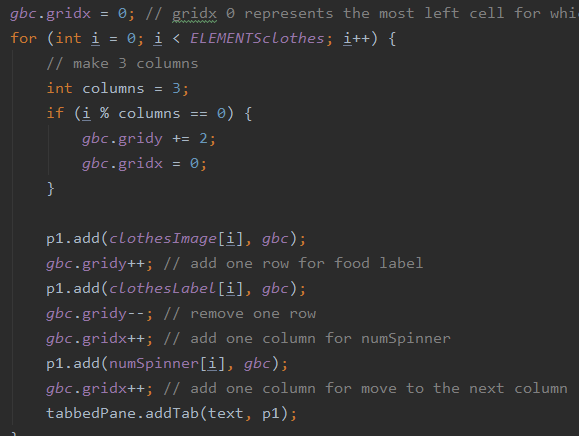
* Item’s menu



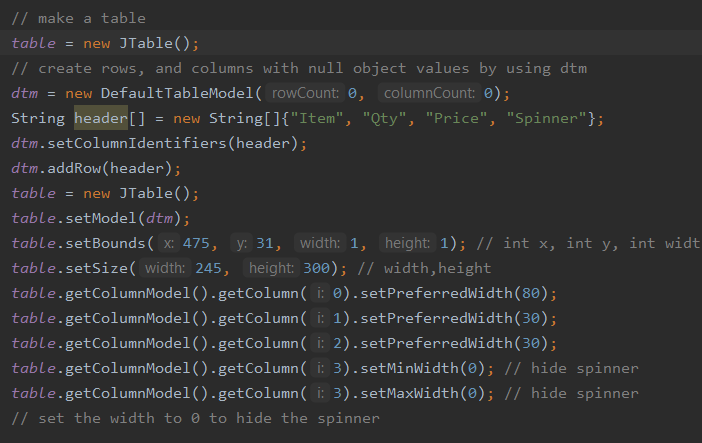
This system is divided into three important parts. The first one is for where the panels are created by using JPanel, and JTabbedPane. JTabbedPane is used to switch to another panel. By using JTabbedPane, it makes everything easier to be maintained by putting each section in the different panels.



To make all the pictures, labels, spinners, look organize, I set all of those by using gridx and gridy. Gridx will specify the column in which the component will be placed. Meanwhile for gridy, it specifies the row in which it will be placed.



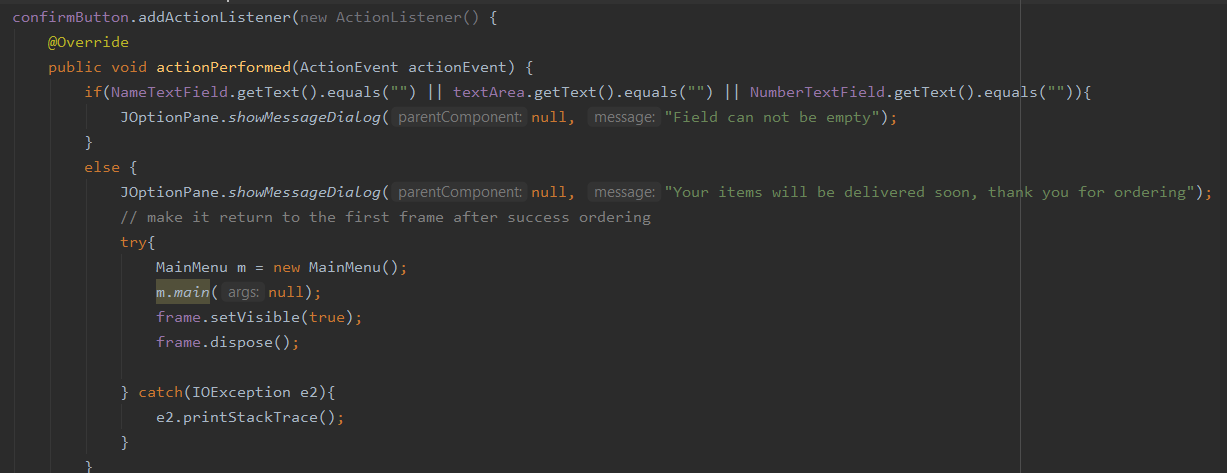
First, we set the gridx to be 0, since gridx 0 represents the left cell for where the component will be placed (means that the component will move from right to the left). After that, I created a loop condition to set the products organized inside the panel, by using 3x3 matrix form. The last step is just add the images, labels, spinners, inside that matrix and remove the cells that are excessed.



For the second part, I created a table by using JTable, and set the table to be zero columns and zero rows by using dtm (DefaultTableModel()). The next is set the title and the width for each of the row to be item, quantity, price, and spinner. At the last section, I hid the spinner’s row by setting the width to be zero (makes the table looks much better without the spinner, from what I know, commonly an online shopping system is just want to calculate the total price, not the total items that the user wants to buy).

* Order

For the last section, I made a pop up message for the users who have not filled all the details yet. They have to fill all the requirements before press the confirm button. Otherwise, a notification will appear.



To create the pop up message, I put a condition inside the confirm button ActionListener which is if the name, address, and the telephone text field are already got a text inside it, the order system is successful. But if not, it will show a message dialog which is “Field can’t be empty”.

1. **BENEFITS**

This system also offers a lot of facilities that could benefit the users, such as:

* Secure registration for customers (password and username will be asked).
* Browsing through the program to see the items that are in each category products like accessories, clothes, electronics, furniture, daily needs, etc.
* A shopping cart, including the total cost bar.
* Uploading the most purchased items in each category of products.
* 24x7 availability.
* The payment can be made through online mode or at the time of home delivery or cod depending upon the customer’s choice and convenience.
* A pop up notification will appear if the customers pressed the confirm button and order button already, however, they have not ordered yet.
* Users could cancel their order, and their shopping cart will go back to 0 (empty).
* Maximum amount of each product is 10.
* etc

1. **RESOURCES**
2. Change listener (for spinners) vs Action listener (for buttons)

<https://docs.oracle.com/javase/tutorial/uiswing/events/changelistener.html>

<https://stackoverflow.com/questions/53123562/itemlistener-vs-changelistener-vs-actionlistener>

1. Tabbed panes to create several components (like panels in the same frame)

<https://docs.oracle.com/javase/tutorial/uiswing/components/tabbedpane.html>

1. Create GUI

<https://www.guru99.com/java-swing-gui.html>

1. Pictures

<https://snipstock.com/image/png-images-pngs-broom-brush-29-png-48591?limit=reached>

1. GridBagConstraints for grid x, and grid y

<https://www.tutorialspoint.com/what-is-the-importance-of-the-gridbagconstraints-class-in-java>