

|  |
| --- |
| KULLIYAH OF INFORMATION & COMMUNICATION TECHNOLOGY  DEPARTMENT OF COMPUTER SCIENCE  SEMESTER 2, 2015/2016 |

CSC 1103 OBJECT-ORIENTED PROGRAMMING

GROUP PROJECT

**( STOCK MANAGEMENT SYSTEM )**

**Team members:**

*Muhammad LaziemBin Shafie (1621781)*

*Nik Ahmad Faiz Bin Mohd Fauzi (1623399)*

*Muhammad Syafiq Bin Mohd Faudzy (1623765)*

*Muhammad Yusuf Bin Zainal Ariffin (1620815)*

**PREPARED FOR :**

*Madam Norzaliza Binti Mohd Nor*

**Submission Date:**

***20/12/2017***

**Abstract**

Our program Stock Management System is made to make stocking item for the warehouse. The main function for our system is that worker can add item that is not registered and update, delete and update the item registered into the system. The benefit that our application can provide for the user who use this system is that it will ease their life where there only need to input and output data from the system.

**Introduction**

First, when it come to Object Oriented Programming subject, it is something basic that we have to learn for our Final Year Project. It is the basic of all coding and programing, thus, when it come to this courses project, we have to come out with something simple yet brief enough let all four members of our team to get as much as possible from this courses. SMS which is software managing system was first,is the idea from our leader, Bro Laziem, which he got from the experience of working in a warehouse. We have actually have been thinking about so many possible thing to come out for this project but, if we do something complicated, the risk will be greater seems this is quite new for us.

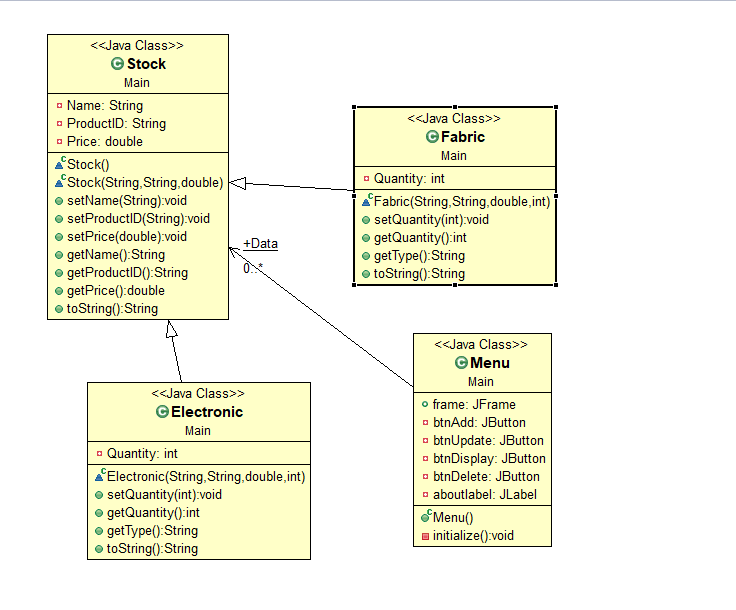
For the idea that we have all have been agreed to, it is simple, yet we still manage to include most of the stuff that we have learn in this project. The idea is just to understand how a storekeeper or a company store and update their products. So basically, we design this software to make it look exactly like a real one so to start with, we have design our security login page so the data can be only registered and deleted by the one who has been authorized later or in this case, only by the group members. It is just a simple login, using if else statement to match the security code that has been embedded to the coding, we manage to pull out that part easily. Then, one the user kick in the security code, they will automatically know what to do because we did our job to make this software understandable and easy to operate with. There’s no need a manual or a user guide in order to run this software, it is all simple and neat. Kicking in the right security code will bring the user right at the main menus where all the things that this software was meant to be built is in here just on the tip of user’s cursor.

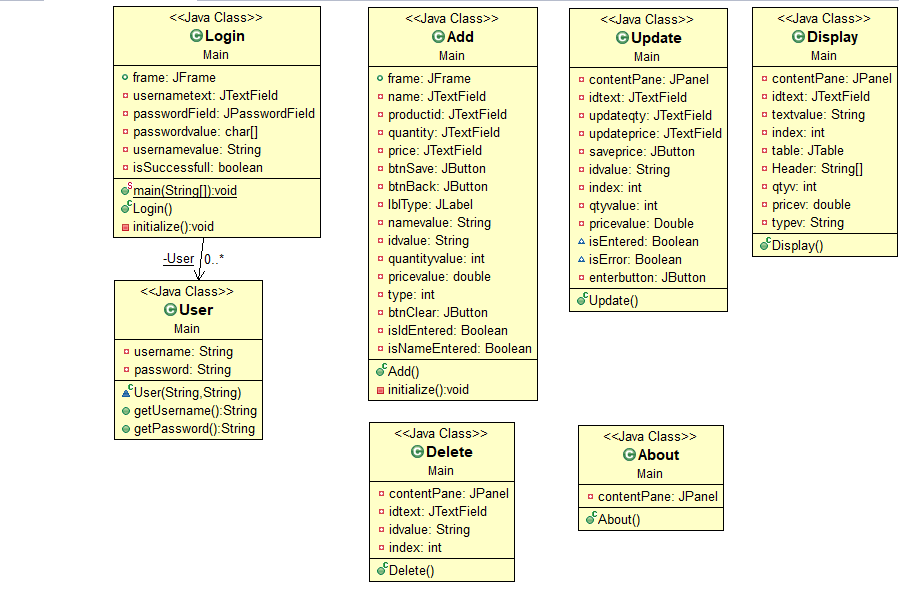
This software function stands with four different tasks which are add, update, display and delete. Add is used to add product and in this function, we had classified or arrange the data using arrays to sort and later to be called in another function. The product that being added will then be classed in their own type of product to ease and tidying the product list. Then, the update function is to edit the attributes of the product which for example the prices or the quantity left in the warehouse. Next, the display function is to call the data existed in the arrays and display the selected data or specific data the user prompt or enter in this function. Last but not least, the delete function is to delete all the attribute and the data itself in the arrays.

So left about the design and interface that we used in this project and it is all about the java. We manage to go through this phase only after we done a little bit of java coding and thanks to Web Technology subject, we can clearly understand how things work. We did a simple design to our project just because we find it look more adults and less kids for some reasons. The pixel counts and stuffs are quite tricky to put in order to make it look the same as what we imagine but we did our best and it looks satisfying as it is.

Last but not least, for what that we got from this project, actually we got so much from this little project, there are so many things that need to discover to make things do the way they should. If we only depend on what we can get in the class, this project will not be finished by now. There a lot of task and researches that we done by the order of our skilled leader, from there, each of us will understand something new and when we compiled it, we will explain each of our stuff to the other group member and at the end of the day, we will all understand what other group members are doing.

**UML Diagram**



****

**The Tasks Distribution**

|  |  |
| --- | --- |
| TASK NO | MEMBER |
| TASK 1 ( LOGIN )   * Coding * Interface * UML * Documentation | * Syafiq * Laziem * Faiz * Yusuf |
| TASK 2 ( MAIN MENU )   * Coding * Interface * UML * Documentation | * Yusuf * Laziem * Faiz * Syafiq |
| TASK 3 (ADD FUNCTION)   * Coding * Interface * UML * Documentation | * Syafiq * Faiz * Yusuf * Laziem |
| TASK 4 ( DISPLAY FUNCTION )   * Coding * Interface * UML * Documentation | * Laziem * Syafiq * Faiz * Yusuf |
| TASK 5 ( UPDATE FUNCTION)   * Coding * Interface * UML * Documentation | * Laziem * Faiz * Syafiq * Yusuf |
| TASK 6 ( DELETE FUNCTION)   * Coding * Interface * UML * Documentation | * Faiz * Laziem * Yusuf * Syafiq |
| TASK 7 ( FINALISING REPORT)   * Compilation * Checking * Printing / Binding | * Yusuf and Syafiq * Laziem * Faiz |

**The Design**

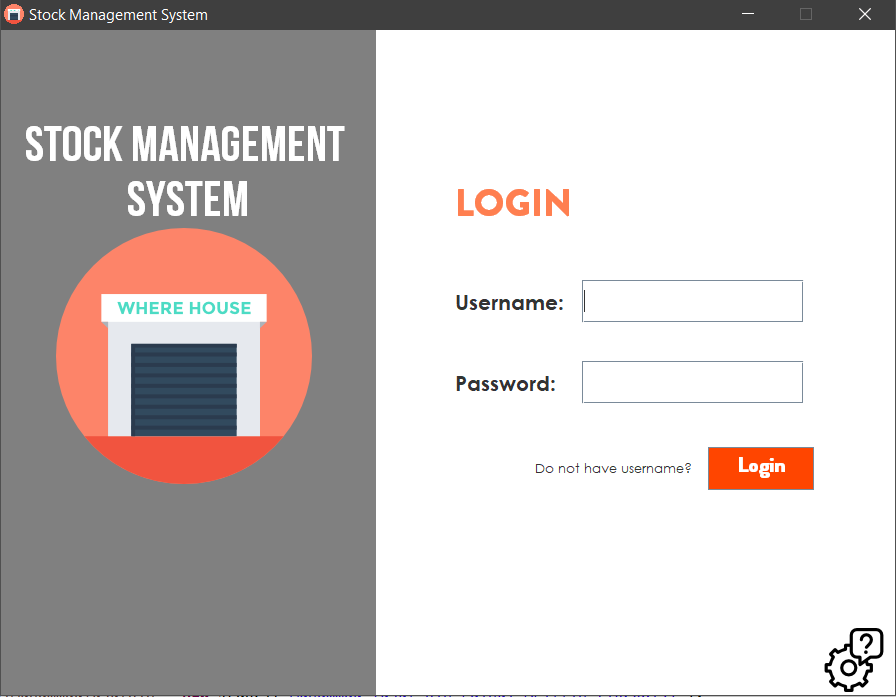
****

Figure 1 : Login

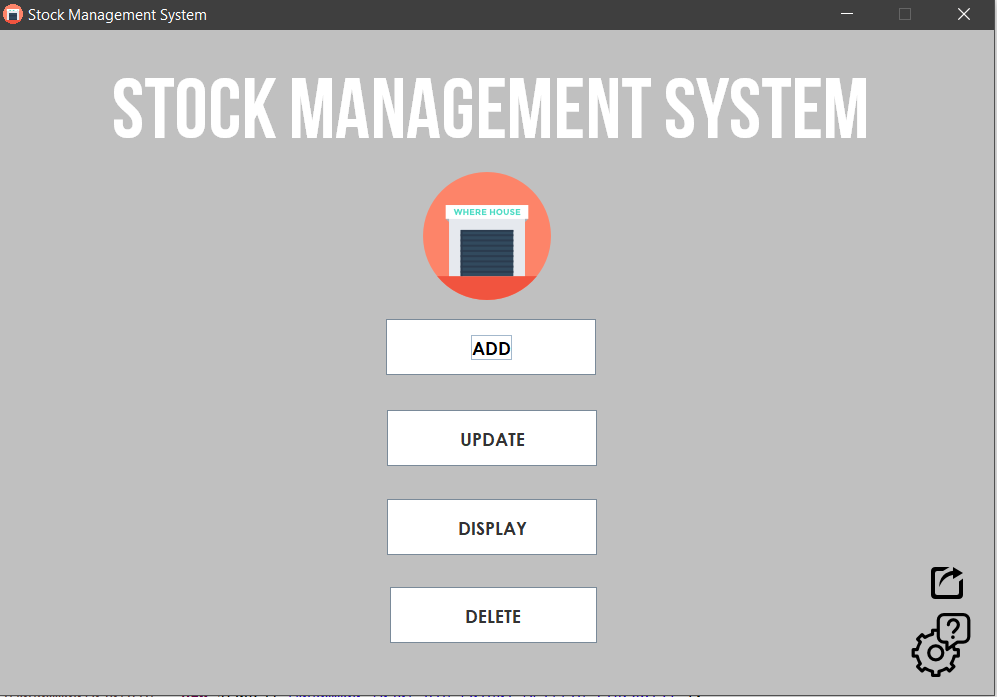


Figure 1 : Main Menu

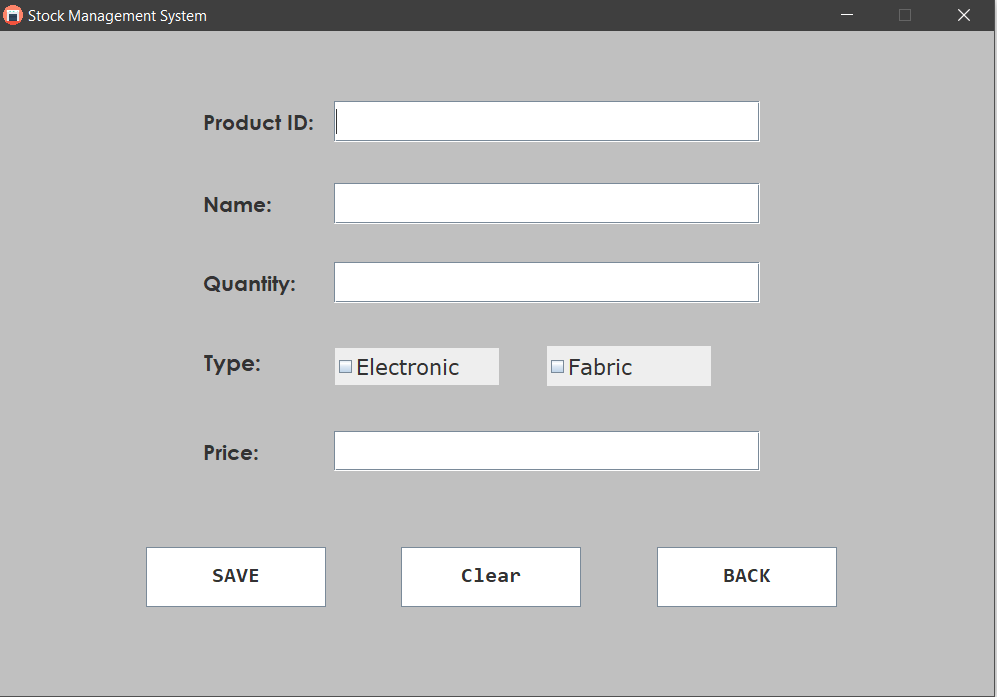


Figure 1 : Add Product interface

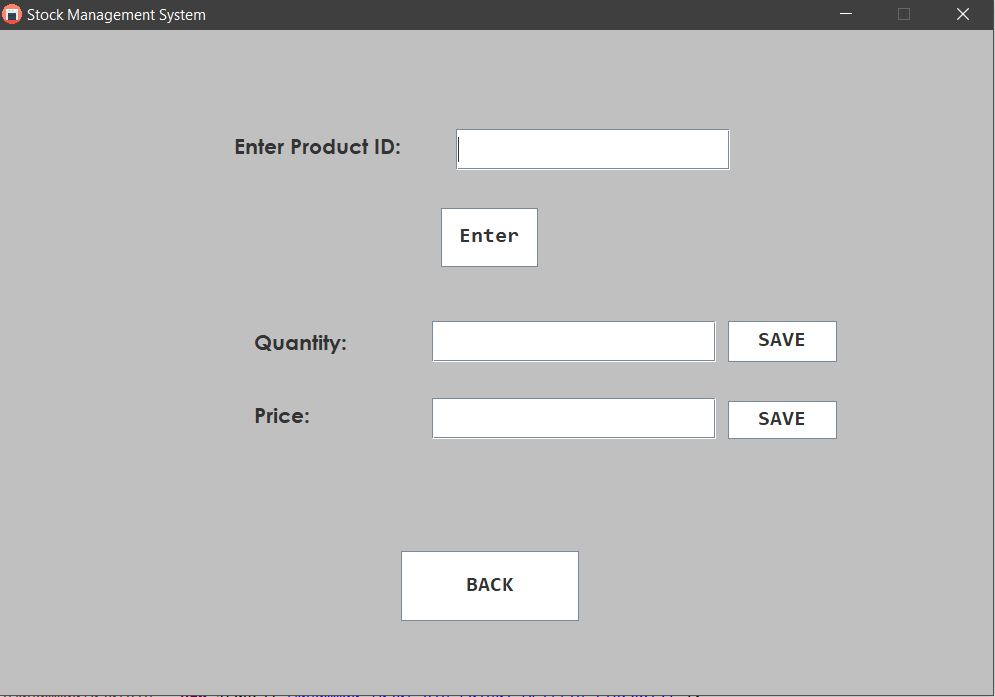


Figure 1 : Update Product Interface

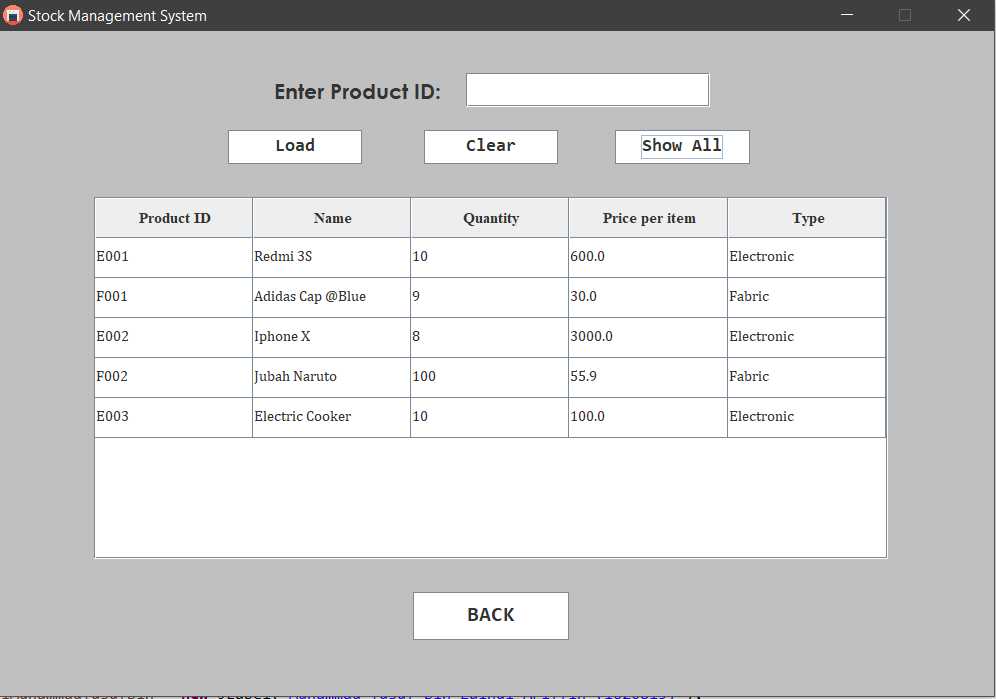


Figure 1 : Update Product Interface

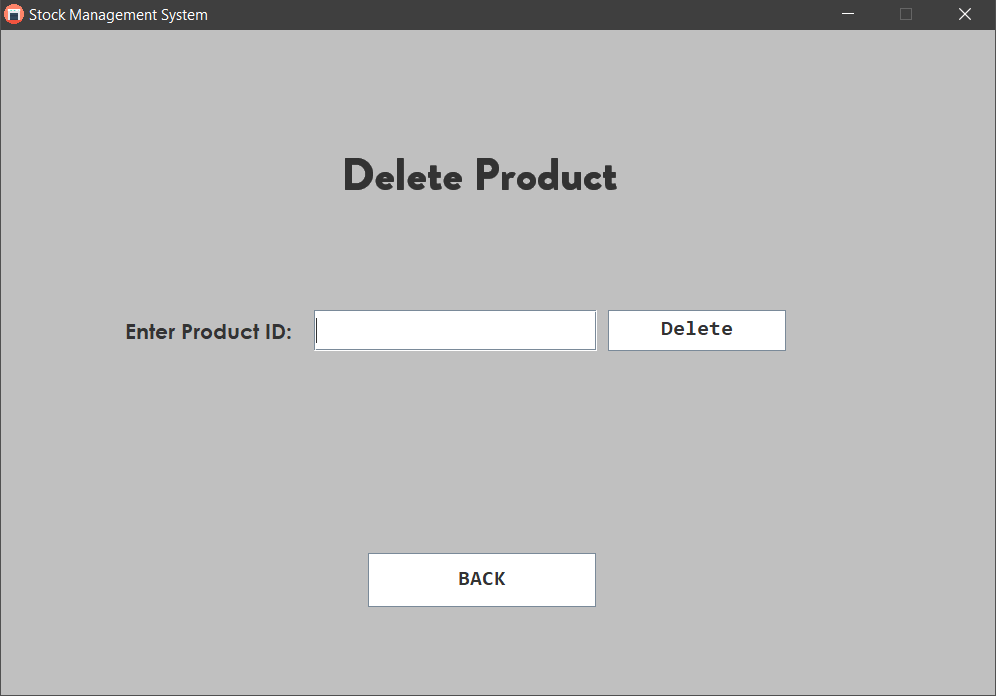


Figure 1 : Delete Product Interface

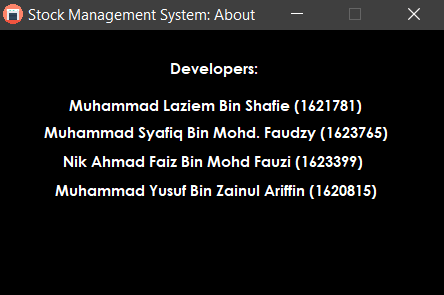


Figure 1 : About System Interface

**The List of Challenge**

1. To create a table in the update interface as we need the interface to ease for the user to see and use it.
2. The uses of permanent data (Either to use file / get output from the database). This is quite hard to learn as it consumes too much time to be learn, so we cancel the uses of permanent data
3. At the beginning stage of our project, we have the problem of our interface where the frame is stacked. When we try to move to another interface, the old interface is there and not removed. As a result, our interface become redundant and not nice to see.
4. We had the difficult top make our application robust.
5. We had difficult to learn the uses of array list, but we managed to learn it.