

# UML design for Trisyslogic Order Management System with Implementation by Object Oriented Programming in Java

**Ganeshwara Shanmugam**

*BSc (Hons) in Software Engineering, Asia Pacific University, Malaysia*

## ABSTRACT

*The objective of this paper is explaining about the implementation of Order Management System in UML also coded with Object Oriented Programming using Java. Various order management systems currently running with lack of functionalities for the business process. Some systems not supporting the business process linearly, simultaneously data protection and report generation also not able to perform. The implemented system will be following the system model in usecase diagram under Unified Modelling Language (UML). Java programming language fully supports with Object Oriented (OO) concepts have been followed in this proposed system. Hence this system will provide smooth operations of order management system through ease access, faster execution also supporting data protection.*

**Keywords:** Class, Data Protection, Implementation, Object Oriented, Usecase

## I.INTRODUCTION

TRISYSLOGICS SDN BHD (TSB) is one of the emerging wholesalers who sells everyday groceries, fresh produce and fresh foods. Situated in Kuala Lumpur, TSB provides its services to various retailers around Malaysia. Due to business expansions, TSB requires an order management system (POM) to automate the process of purchase orders to ensure that the purchases are conducted more effectively.

### 1. 1 OBJECTIVES

There are several objectives that have been listed by TSB that the developed system has achieve. The objectives are as listed below:

- ♦To ensure that the system have a login functionality to allow only authorized users to access the system. There will three users, Administrator, Sales Manager and Purchase Manager.
- ♦Based on the authorizations provided, the users must be able to add, update and delete information regarding users, items, suppliers, Daily Item Wise Sales, Purchase Requisitions and Purchase Orders.
- ♦To ensure there are no duplication in the data saved in the system.

## II. UML DIAGRAM

Unified Modelling Language (UML) diagrams are used to provide visualization to how a software is to be built with the help of a collection of diagrams [1]. To show Trisyslogics Order management system is to be build, the Use Case diagram, Class diagram and Activity diagram is drawn.

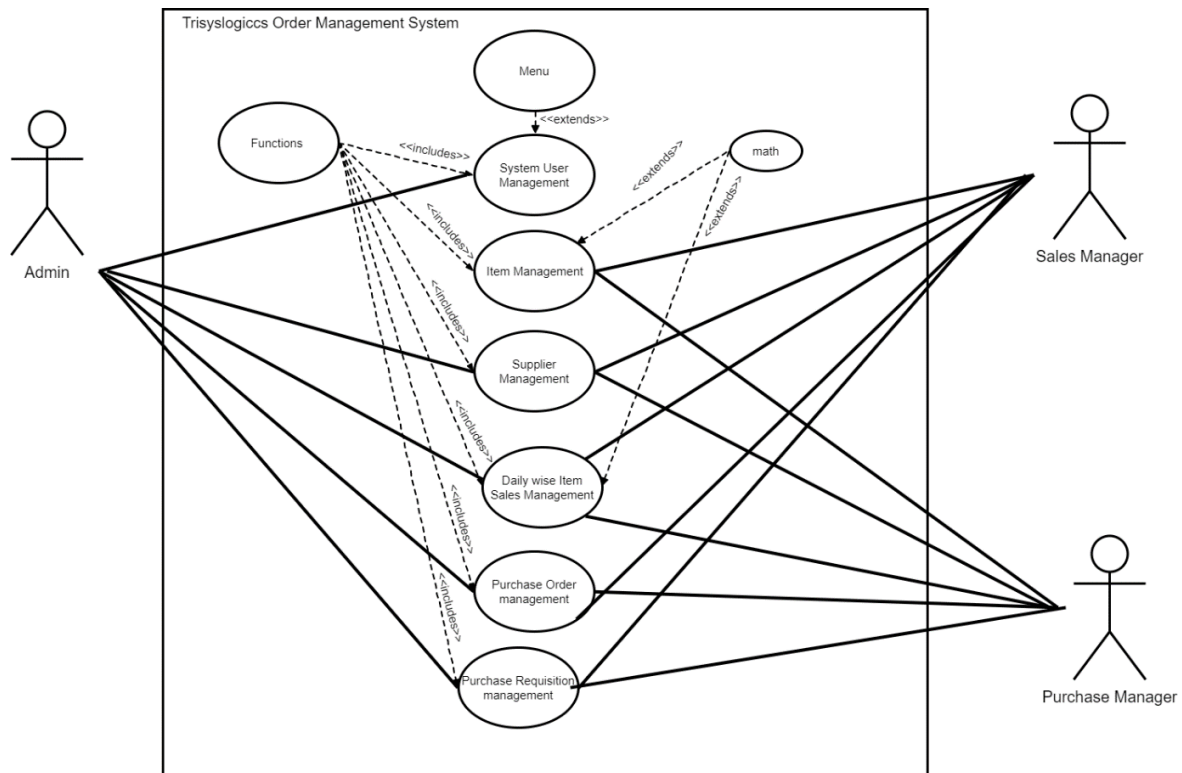


Figure 2. Use case diagram for Trisyslogics Order Management System

Figure 2. shows the list of functions within the order management system. There are three actors for the use case which the admin, sales manager and purchase manager also shown the class diagram in “Fig.21”.

### 2.1 CLASS DIAGRAM

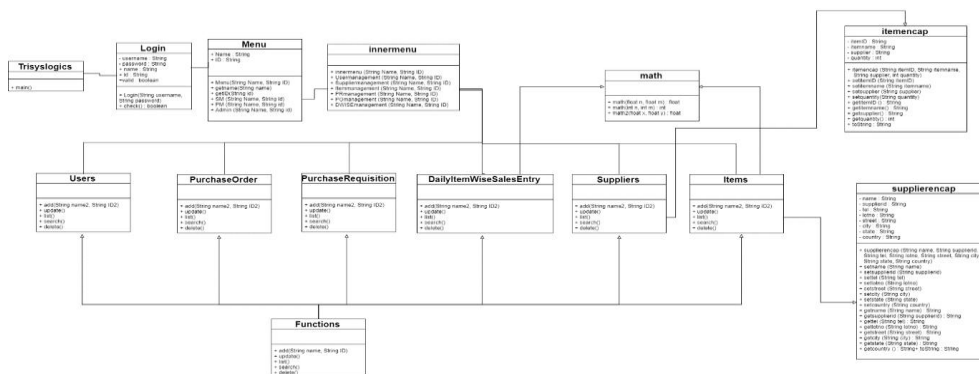


Figure 2.1 Class Diagram

## 2.2 ACTIVITY DIAGRAM

The activity diagram aids in understanding the flow of the process for each class. There are three important classes that are shown below “Fig.2.2” which are the login class, menu class and supplier class.

### Activity Diagram for Login class

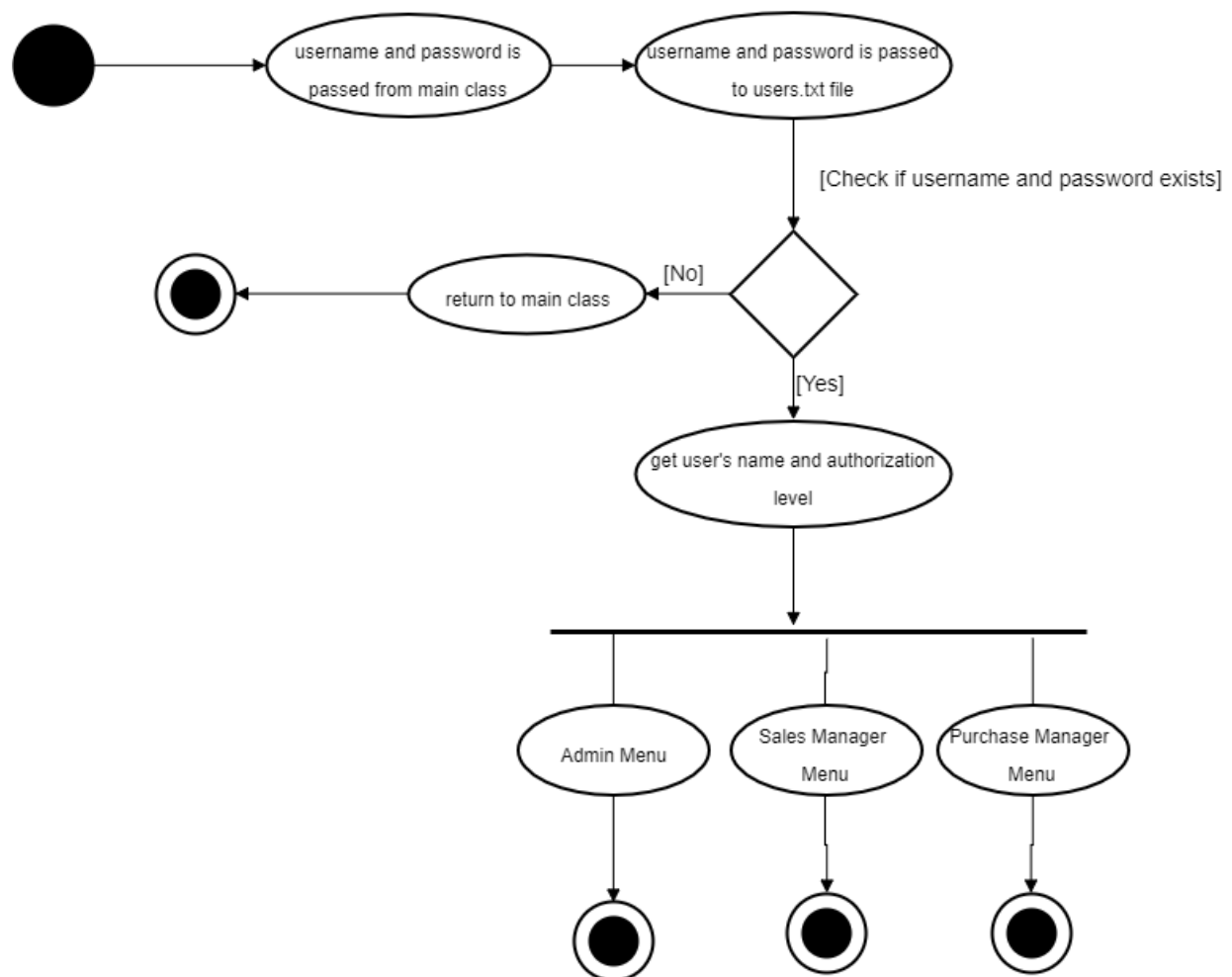


Figure 2.2 Activity Diagram for Login

## III.OBJECT ORIENTED PROGRAMMING

Object-oriented programming is a model that is derived from the concepts of “objects”. The objects referred here are contains data and methods. Implementation of object oriented programming into the programs allows them to be more flexible and maintainable. Object-oriented programming enables the easy understanding of how

the program works by bringing its behaviors and data into a single location [2]. Below are some of the object-oriented programming concepts that have been implemented into the program.

### 3.1 ENCAPSULATION

An encapsulation is an object-oriented concept that binds the data and code into a single unit [3]. The advantage of using encapsulation is that it allows the changes to be done to the class it has been implemented without affecting the encapsulated class [1].

```
public void setitemname(String itemname)
{
    this.itemname = itemname;
}

public void setsupplier(String supplier)
{
    this.supplier = supplier;
}

public void setquantity(int quantity)
{
    this.quantity = quantity;
}

//getttter

public String getitemID()
{
    return itemID;
}

public String getitemname()
{...39 lines }

public String getsupplier()
{
    return supplier;
}
```

Figure 3.1 Encapsulation Program

The “Fig.3.1” above shows a code snippet for the *itemencap* class. This concept has been implemented for the add method in the Items class. This is done to ensure that the data passed from the add method is gathered and passed correctly based on its requirements and then passed back to be saved into the file.

### 3.2 INHERITANCE

Inheritance is an object-oriented programming concept that allows the sharing of properties between classes. This method helps to make the data more manageable in the program. Inheritance can be invoked between classes with the use of several keywords. The keywords that are used in the program for implementing inheritance is “extends” and “implements”.

#### 3.2.1 EXTENDS

The “extends” keyword helps by enabling a class to inherit the properties of another class [4].

```
public class innermenu extends Menu {
```

The *innermenu* class contains all the menus for all the functions of the other classes such as the *users* class, *items* class, *supplier* class, *purchaserequisition* class, *purchaseorder* class and *dailyitemwisesalesentry* class. The *innermenu* class extends to the *Menu* class to enable it to access the Admin, SM and PM methods from the *Menu* class. With the help of the “extends” keyword, data such as the user name and user ID was passed down to the *innermenu* class. The type of inheritance used in this part of the program is single inheritance as it is only one class extending the other.

#### 3.2.2 IMPLEMENTS

The “implements” keyword is used to enable a class to inherit the properties from an interface [5].

```
public class Items implements Functions {
```

The code snippet above shows the keyword “implements” added to the *items* class. This keyword is used to inherit all the methods from the *Functions* interface. Once the interface is implemented into the class, the class must follow all the methods that interface contains.

### 3.3 INTERFACE

An interface is a type of reference that is similar to class in Java. An interface consists of abstract methods that will be inherited by the class that implements it [6].

The figure below shows the *Functions* interface that is implemented into the classes of the system. This interface consists of methods such as add, update, list, search, and delete. This method are common methods that are to be shared by most of the classes in the program. The methods are left empty as the function within the methods may differ from class to class.

```
package trisyslogics;

/**
 * @author Ganeshwara Shanmugam
 */
public interface Functions {

    public void add(String name, String ID);
    public void update();
    public void list();
    public void search();
    public void delete();
}
```

#### IV. SCREEN SHOTS OF TRISYSLOGICS ORDER MANAGEMENT SYSTEM

```

Trisyslogics
Order Management System

Username :
```

The figure above shows the login page for the Trisyslogics Order Management System. To enter the system, the users have to provide their credentials such as their username and password. There are three main users, Admin, Sales Manger and Purchase Manager.

```

Trisyslogics
Order Management System

Username :      Admin
Password :      admin

User :Tester
ID :Admin

ADMIN MENU
1.System User Mangement
2.Item Management
3.Supplier Management
4.Purchahse Requisition Management
5.Purchahse Order Management
6.Daily Item Wise Sales Management
7.Logout
8.Exit
Make a selection :
```

If the user has inserted the correct username and password, then the system will take the user to designated menu based on their authority level.

```
-----|
|               Trisyslogics               |
|               Order Management System     |
|-----|

Username :      adin

Password :      admn

Username or Password is incorrect. Try again [Y/N]
|
```

If the user fails to provide the correct username and password, then the system will reply “Username and Password is incorrect. Try again [Y/N]” if the user choses “Y”, then the user can retry to login. If “N” is chosen, then the system will exit.

```
-----|
|               ADMIN MENU                 |
|-----|

User :Tester
ID :Admin

1.System User Mangement
2.Item Management
3.Supplier Management
4.Purcahse Requisition Management
5.Purcahse Order Management
6.Daily Item Wise Sales Management
7.Logout
8.Exit

Make a selection : |
```

The figure above shows the admin main menu. The admin has the capability to access all the functions within the system. There are six systems. There is an option to exit the system and also log-out from the system.

The Figure below shows the Item Management menu. This menu is accessible to the Admin and the Sales manager. In this function, the users are able to add new item, update item, List the exiting items search for items and delete items. There is an additional option for the users to return to the main menu. The add new items function allows the admin and Sales manager to add new items to inventory. The users have to enter the product name, select the supplier and the quantity of the product. The item will be automatically generated a new ID at the beginning of the process. This ID is specific to one item.

```
|-----|  
|          ITEM MANAGEMENT          |  
|-----|  
Username : Tester  
1.Add New Item  
2.Update Item  
3.List Item  
4.Search Item  
5.Delete Item  
6.Back to main menu  
  
Enter your choice: |
```

## V.CONCLUSION

As a conclusion, Trisyslogic Order Management System was able to fulfil all the objective that was stated at the beginning of the project. Trisyslogic's Order Management System will be able to conduct its purchase and ordering operations more smoothly and efficiently to support Trisyslogics business expansion. The implementation of Object- Oriented Programming methods will make the program to easily modified and maintained by the developers as the methods implemented provides more flexibility. The drawbacks with the current Trisyslogic Order Management System is that the interface introduced is console based interface. The users will require some training before being able to use the system. Secondly, the information from the system is stored into a file. This provides minimal security to the data stored. Storing information into files can also affect the performance of the system. For future enhancements of the system, a graphical user interface will be implemented to enable easier transitions between menus and functions. The future implementation of a graphical user interface ensures a more user-friendly system. Another future enhancement to be considered is database connectivity. With the help of database connectivity in the future, the system will be able to store more information and also provide enhances the speed of writing and retrieval of information.

## VI.ACKNOWLEDGMENT

The author would like to share gratitude to Mr Umapathy Eaganathan, Lecturer in Computing, Asia Pacific University, Malaysia for the constant support and motivation also for Miss Angel Aron for her constant support which helped me to participate in this International Conference and also for journal publication.

## REFERENCES

- [1] Smartdraw, 2017. *What is a UML Diagram?*. [Online] Available at: <https://www.smartdraw.com/uml-diagram/> [Accessed 5 December 2017].
- [2] Singh, C., 2013. *OOPs concepts in Java*. [Online] Available at: <https://beginnersbook.com/2013/04/oops-concepts/> [Accessed 3 December 2017].
- [3] JavaTpoint, 2017. *Encapsulation in java*. [Online] Available at: <https://www.javatpoint.com/encapsulation> [Accessed 3 December 2017].

- [4] Tutorialspoint, 2017. *Java - Inheritance*. [Online] Available at:  
[https://www.tutorialspoint.com/java/java\\_inheritance.htm](https://www.tutorialspoint.com/java/java_inheritance.htm) [Accessed 3 December 2017].
- [5] Codejava, 2015. *Java Keyword: implements*. [Online] Available at: <http://www.codejava.net/java-core/the-java-language/implements-keyword> [Accessed 3 December 2017].
- [6] Tutorialspoint, 2017. *Java-Interfaces*. [Online] Available at:  
[https://www.tutorialspoint.com/java/java\\_interfaces.htm](https://www.tutorialspoint.com/java/java_interfaces.htm) [Accessed 3 December 2017].