**A picture containing text, screenshot, font, line

Description automatically generated**

**TCP2201**

**Object-Oriented Analysis and Design**

**INDIVIDUAL ASSIGNMENT REPORT**

***Trimester 2, Session 2022 / 2023***

|  |  |  |
| --- | --- | --- |
| **No.** | **Student ID** | **Student Name** |
| **1** | **1221303085** | **Mannoj Sakthivel** |

Table of Contents

[Assumptions 2](#_Toc138010175)

[Use Case Diagram 3](#_Toc138010176)

[Class Diagram 4](#_Toc138010177)

[Sequence Diagram 5](#_Toc138010178)

[View Food Available 5](#_Toc138010179)

[Login 5](#_Toc138010180)

[MainFrame (The main system) 5](#_Toc138010181)

[System Design Pattern 6](#_Toc138010182)

[Extra Features from the system 6](#_Toc138010183)

# Assumptions

Only 1 person can use the system (Admin/Cashier) as there is only one username and password for the system (hard coded). There are 5 items being sold by the grocery store as the admin can only enter these 5 items or else the system will prompt an error message. The discount using royalty card number is not fixed (dynamic) since logically vouchers (royalty card number) will keep changing every month for almost any active store. The system is also future proof as different vouchers (royalty card number) can be added in the future for different types of discounts in the future. The admin is not able to key in the price because as in any real world system no Cashier/Admin can key in the price (usually done by scanning the QR of the product, thus no price is entered). Example, if an item is RM10 and the Cashier keys in RM5, he or she can steal RM5.

# Use Case Diagram

A picture containing diagram, text, line, drawing

Description automatically generated

# Class Diagram

A picture containing text, diagram, parallel, screenshot

Description automatically generated

# Sequence Diagram

## View Food Available

A picture containing diagram, line, text, plot

Description automatically generated

## Login

A diagram of a login and password

Description automatically generated with low confidence

## MainFrame (The main system)

A picture containing text, diagram, parallel, line

Description automatically generated

# System Design Pattern

For this project, I’ve implemented one design pattern in my code which is the observer design. The buttonAddItem button is equipped with an ActionListener that adheres to the observer pattern. This listener is responsible for detecting button clicks and informing the registered listener (the actionPerformed method) whenever the button is clicked. The advantages of using the observer design includes the decoupling. This decoupling facilitates GUI development by separating the user interface components from the data or model, resulting in a more modular design that is easier to work with. The Observer design pattern is frequently employed in graphical user interfaces (GUI) to achieve loose coupling between GUI components and the underlying data or model.

# Extra Features from the system

-Login system. The Admin/Cashier is able to log into the system to ensure no external parties (intruders) can access the system.

-Reset Button. The reset button is implemented in MainFrame.java. This button helps the Admin to reset the items already keyed in. This is incase the Admin accidentally entered a wrong item or quantity.

-Quantity. The quantity allows the Admin to add more than one item. This makes it easier for the Admin to get the price of multiple items quickly.

-Main Menu (JButton) is added into every page to allow the Admin to go back to a previous page if needed instead of starting the whole system again if he or she made a mistake.

-View Available Food page allows the Admin to look at the available food separately from the other pages.

-Event Handling (exception handling) is done throughout the system such as the quantity cannot be empty when keying in an item. This is one of many event handlings in the system.