

UCDF1905ICT(SE)

HAND OUT DATE: 6 2020

HAND IN DATE: 1 JUNE 2020

WEIGHTAGE:

**INSTRUCTIONS TO CANDIDATES:**

**1 Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing).**

**2 Late submission will be awarded zero (0) unless Extenuating Circumstances (EC) are upheld.**

**3 Cases of plagiarism will be penalised.**

**4 The assignment should be bound in an appropriate style (comb bound or stapled).**

**5 Where the assignment should be submitted in both hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hardcopy.**

**MUHAMMAD FUAD BIN ABDULLAH**

**TP055684**

# TABLE OF CONTENTS

[TABLE OF CONTENTS 2](#_Toc41911085)

[1.0 INTRODUCTION TO KHAYALAN CAFETERIA SYSTEM 4](#_Toc41911086)

[2.0 STORYBOARDS 5](#_Toc41911087)

[Part 0 – Form Style 5](#_Toc41911088)

[Part 1 – Login Interface 6](#_Toc41911089)

[Definition 7](#_Toc41911090)

[Part 2 – Staff Main Menu Interface 8](#_Toc41911091)

[Definition 9](#_Toc41911092)

[Part 3 – Customer Registration Interface 10](#_Toc41911093)

[Definition 11](#_Toc41911094)

[Part 4 – Customer Profile Interface 13](#_Toc41911095)

[Definition 14](#_Toc41911096)

[Part 5 – Customer Selection Interface 16](#_Toc41911097)

[Definition 17](#_Toc41911098)

[Part 5.1 – Customer Selection Interface Search Result 18](#_Toc41911099)

[Definition 19](#_Toc41911100)

[Part 6 – Menu Selection Interface 20](#_Toc41911101)

[Definition 21](#_Toc41911102)

[Part 7 – Order Summary Interface 22](#_Toc41911103)

[Definition 23](#_Toc41911104)

[Part 8 – Payment Interface 24](#_Toc41911105)

[Definition 25](#_Toc41911106)

[Part 9 – Receipt Interface 26](#_Toc41911107)

[Definition 27](#_Toc41911108)

[Part 12 – Splash screen Interface 28](#_Toc41911109)

[Definition 29](#_Toc41911110)

[3.0 USER MANUAL 30](#_Toc41911111)

[Usage Overview - Ordering 30](#_Toc41911112)

[Usage Overview – Registering 30](#_Toc41911113)

[Usage Overview – Modifying 31](#_Toc41911114)

[Usage – Logging in as staff 32](#_Toc41911115)

[Usage – Taking an order 39](#_Toc41911116)

[Usage – Cancelling an order 72](#_Toc41911117)

[Usage – Adding a remark to an order 78](#_Toc41911118)

[Usage – Registering a customer 83](#_Toc41911119)

[Usage – Accessing a customer’s profile 94](#_Toc41911120)

[Usage – Modifying a customer’s profile 101](#_Toc41911121)

[Usage – Deleting a customer’s profile 108](#_Toc41911122)

[Usage – Exiting the application 114](#_Toc41911123)

[4.0 TEST PLAN AND TEST CASES 121](#_Toc41911124)

[1.0 Splash Screen Interface 121](#_Toc41911125)

[2.0 Staff Login Interface 122](#_Toc41911126)

[3.0 Staff Main Interface 123](#_Toc41911127)

[4.0 Customer Registration Interface 125](#_Toc41911128)

[5.0 Customer Profile Interface 127](#_Toc41911129)

[6.0 Customer Selection Interface 130](#_Toc41911130)

[7.0 Food Selection Interface 131](#_Toc41911131)

[8.0 Drink Selection Interface 132](#_Toc41911132)

[9.0 Sides Selection Interface 133](#_Toc41911133)

[10.0 Order Summary Interface 134](#_Toc41911134)

[11.0 Payment Interface 135](#_Toc41911135)

[12.0 Receipt Interface 137](#_Toc41911136)

[5.0 CRITICAL ASSESSMENT 138](#_Toc41911137)

[6.0 REFERENCES 140](#_Toc41911138)

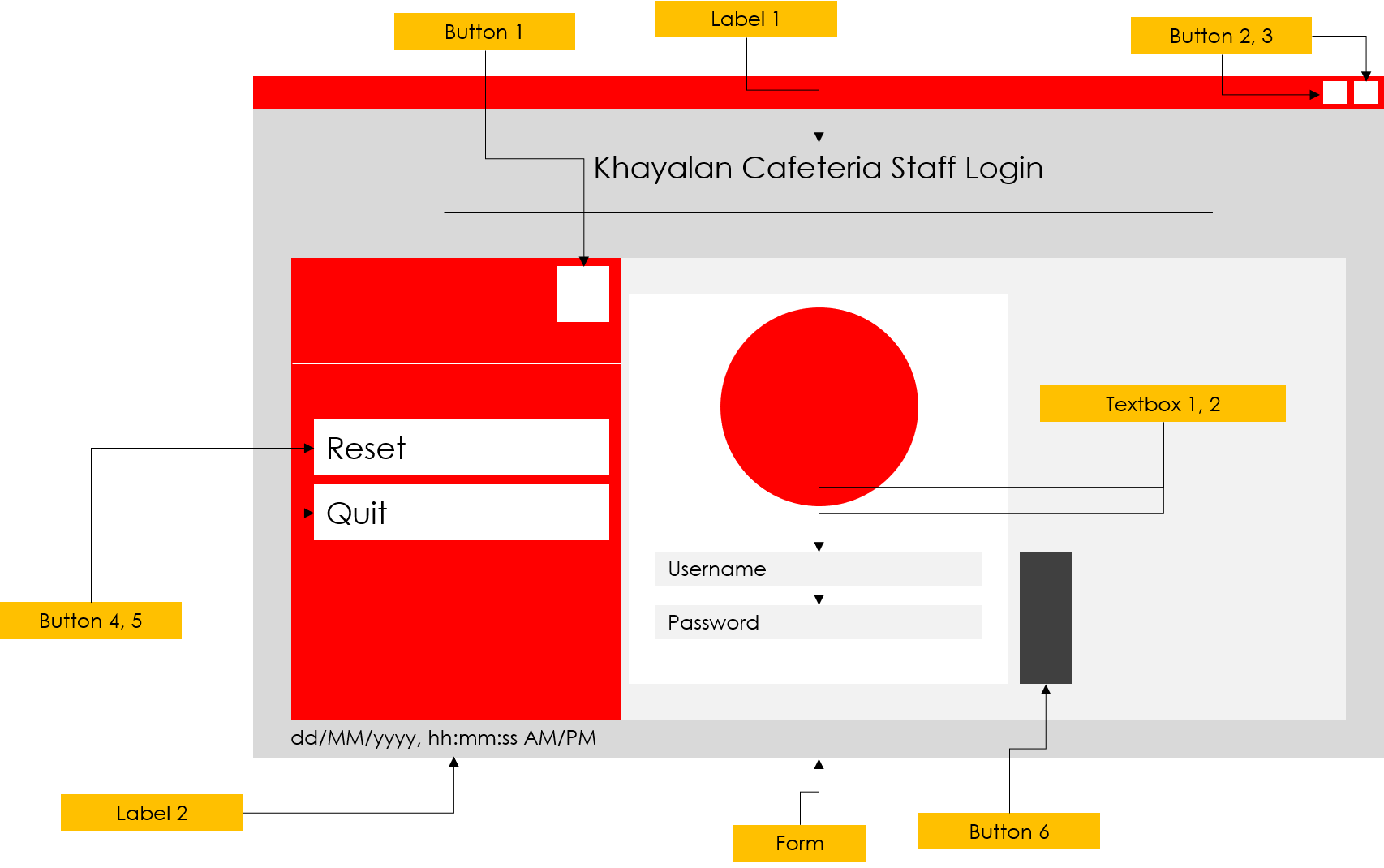
# 1.0 INTRODUCTION TO KHAYALAN CAFETERIA SYSTEM

Khayalan Cafeteria System is a system built to coordinate and manage a cafeteria from its method of handling oncoming customers to hard-working employees dealing with the customers from behind the counter of sales. The system is aspired to assist by reducing workload of repetitive and burdensome tasks which many small food and beverage (F&B) businesses must commit to every day as a normal work routine. This system is crafted specially to meet the requirements of an F&B business which includes a system that handles customer’s information and a thoroughly done, centralised transaction process system that comes with local save-to database feature whereby every order and transaction performed will be stored for future reference and security. This system will be programmed using Visual Basic.NET language and connected to an internal database handled by Microsoft Access to keep vital records that are required by the system in order to be able to function at expected performance.

This system is a mock application that is created solely for education purpose likewise, to fulfil an assignment under Visual Basic.NET module from the course. Any names or entities found in the application is purely coincidental and has no relation to the living nor dead. Khayalan Cafeteria System is fictional and purposely done to study the basics of programming and to explore available features provided by Visual Studio. The system is not eligible for redistribution and can only be used to be experimented, be studied, be tested and act as reference to create an actual system.

# 2.0 STORYBOARDS

## Part 0 – Form Style

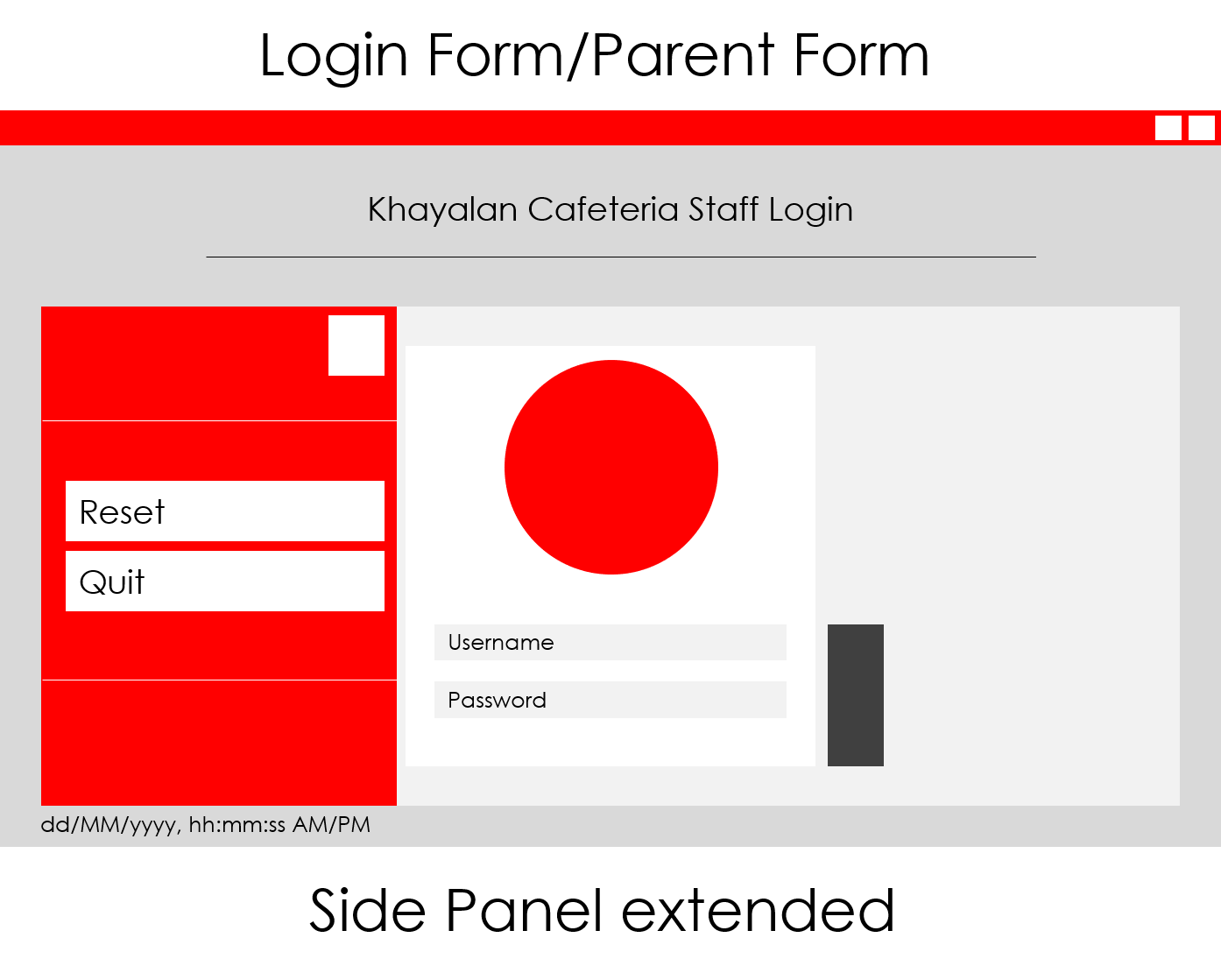


**Figure 0: Explaining the functions of every interactable elements on the interface**

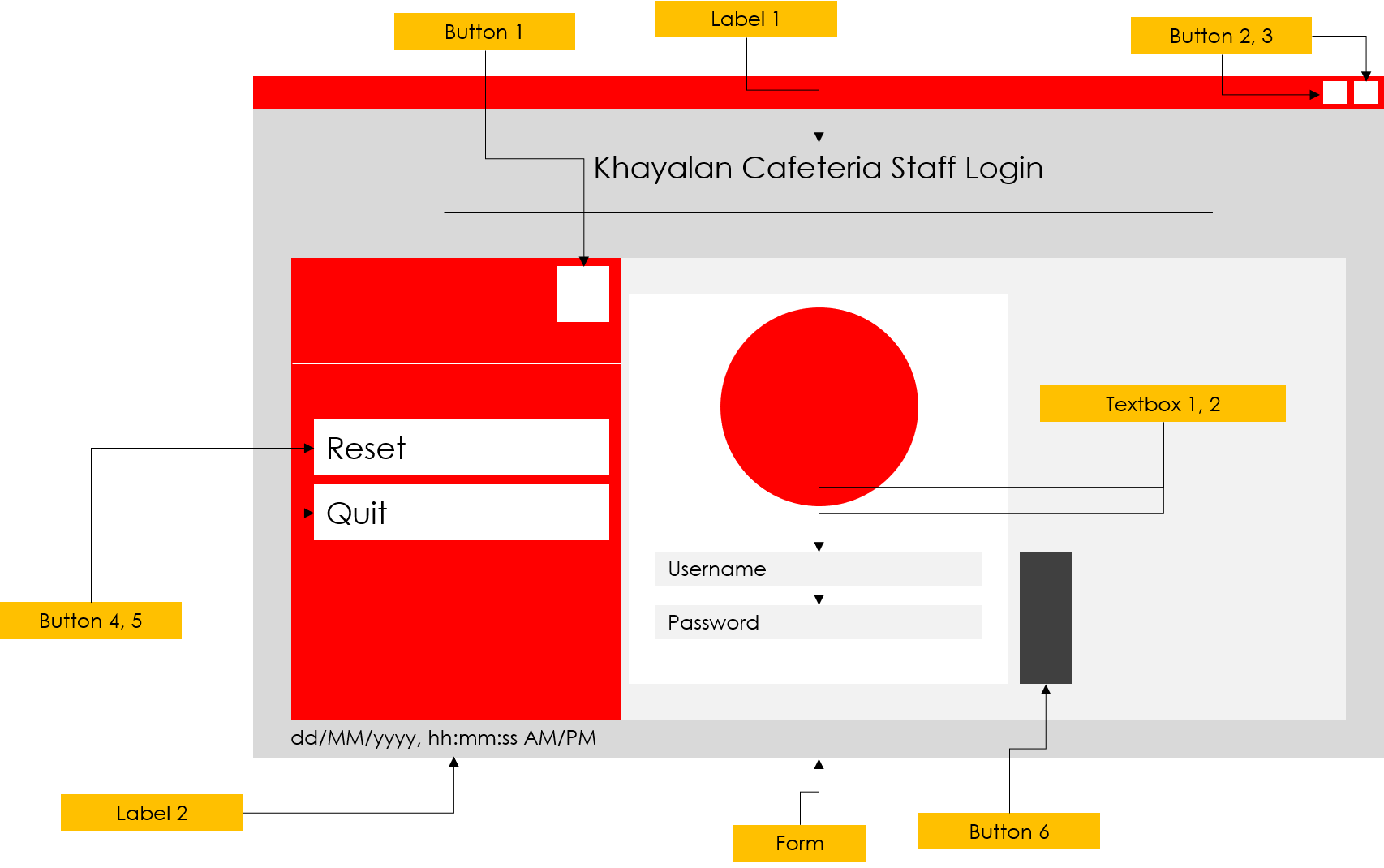
Figure 1 above describes the fundamental aspects of the form. The image shown above is a login interface drawn as a storyboard. The unique look of this form is due to its bright and contrasting colour. In the actual programme, it will use light gray, crimson and dark slate gray colour gradient for most of its controls.

Besides that, the form has less of generic Windows influence and more of modern touch to it. This can be seen through its custom minimise and close buttons later on inside the document. This form purposely eliminates the ability to resize or maximise so that it will fit every screen, no matter of the resolution. Design of this form will be applied to the rest of the forms with slight variations but still inherits the original.

## Part 1 – Login Interface



**Figure 1.1: Login form with extended side panel.**

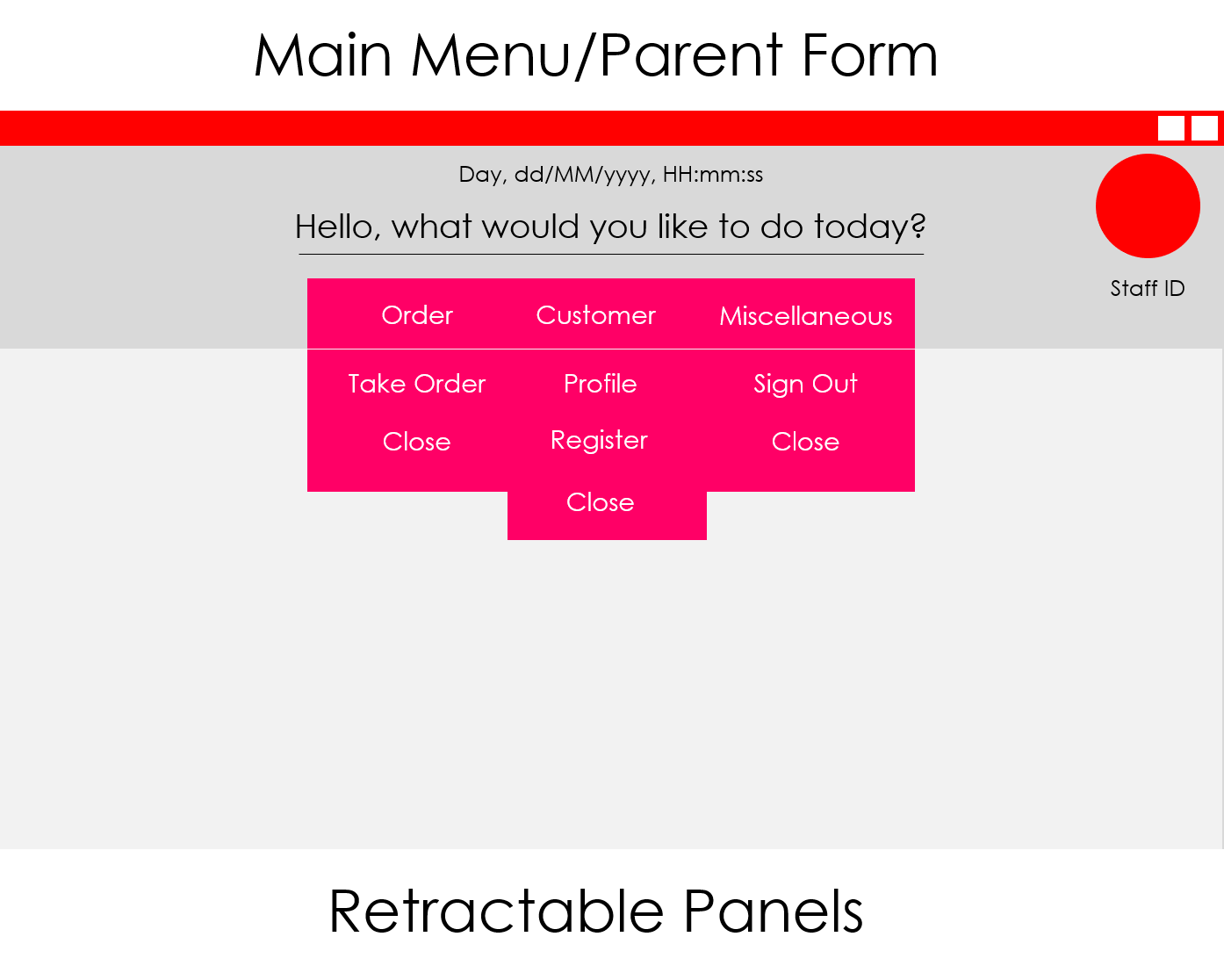


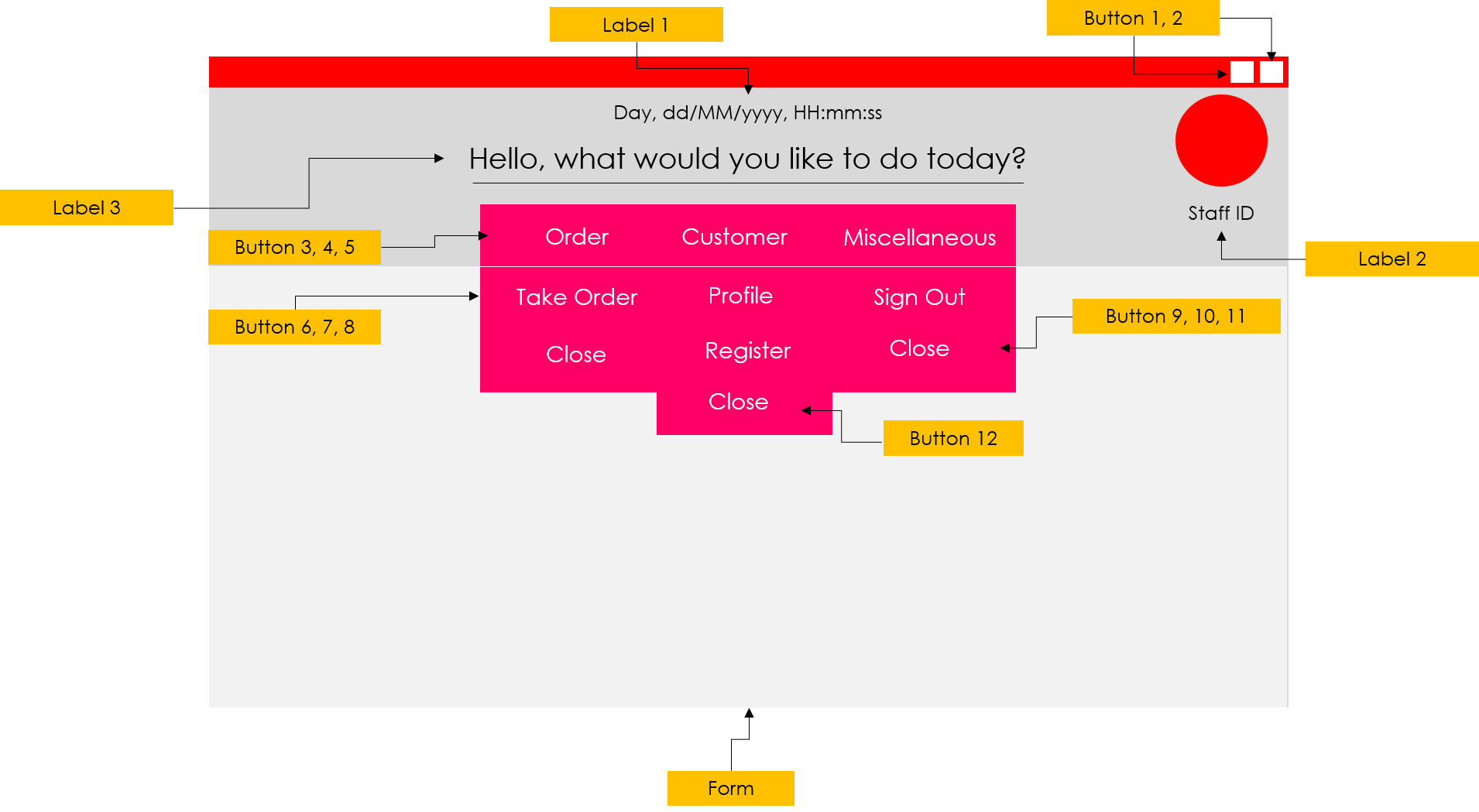
**Figure 1.2: Login form labelling.**

### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | staffLogin | The form |
| Button 1 | btnClosePane | Button to hide side panel |
| Button 2 | btnMinimise | Button to minimise form |
| Button 3 | btnClose | Button to close application |
| Button 4 | btnReset | Button to reset form |
| Button 5 | btnExit | Button to close application |
| Button 6 | btnLogin | Button to log into system |
| Text Box 1 | txtUsername | Textbox handling username |
| Text Box 2 | txtPassword | Textbox handling password |
| Label 1 | lblTitle | Label to show cafeteria name |
| Label 2 | lblClock | Label to display clock |

## Part 2 – Staff Main Menu Interface



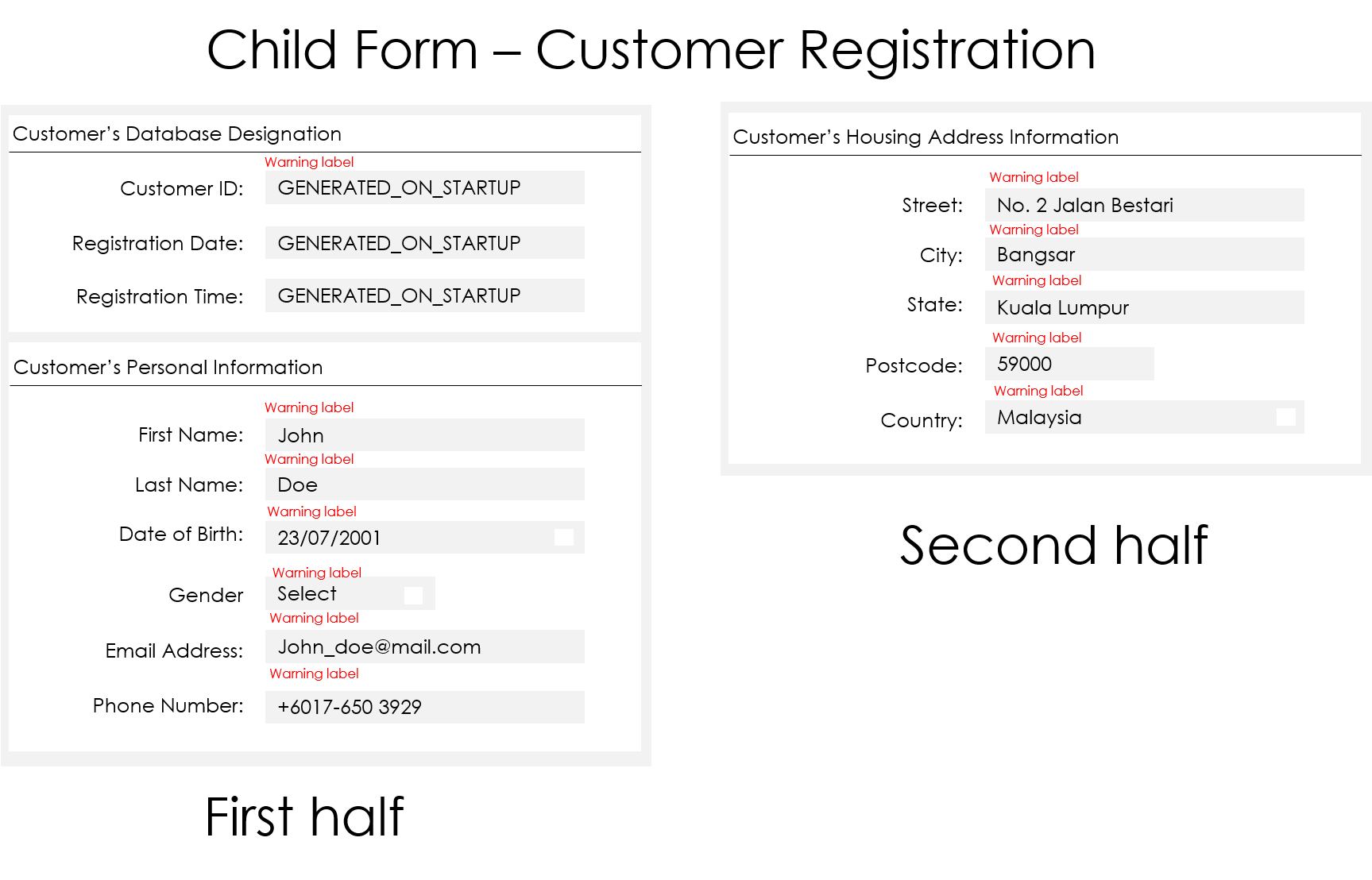
**Figure 2.1: Staff main menu with extendable panes containing interactable buttons.**

**Figure 2.2: Staff main menu labelling.**

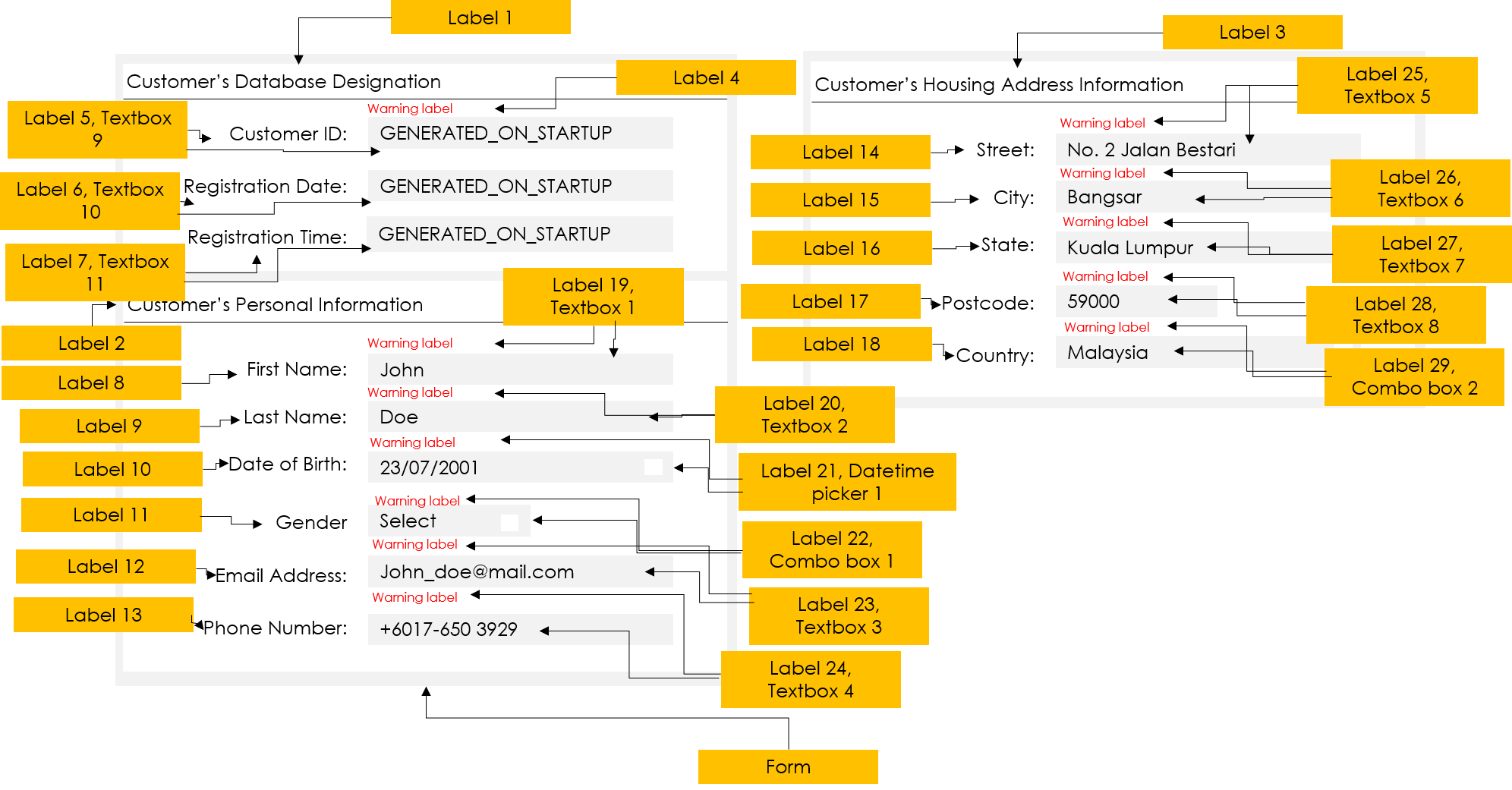
### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | staffMenu | The form |
| Button 1 | btnMinimise | Button to minimise form |
| Button 2 | btnClose | Button to close application |
| Button 3 | btnOrder | Button to show Order panel |
| Button 4 | btnCustomer | Button to show Customer panel |
| Button 5 | btnMiscellaneous | Button to show Miscellaneous panel |
| Button 6 | btnTakeOrder | Button to open Ordering form |
| Button 7 | btnProfile | Button to open Customer Profile form |
| Button 8 | btnLogOut | Button to return to Login menu |
| Button 9 | btnCloseOrder | Button to hide Order panel |
| Button 10 | btnRegister | Button to open Customer Registration form |
| Button 11 | btnCloseMiscellaneous | Button to hide Miscellaneous panel |
| Button 12 | btnCloseCustomer | Button to hide Customer panel |
| Label 1 | lblClock | Label to display clock |
| Label 2 | lblStaffID | Label to display Staff ID |
| Label 3 | lblTitle | Label to display system message |

## Part 3 – Customer Registration Interface



**Figure 3.1: Customer registration interface separated into two parts.**

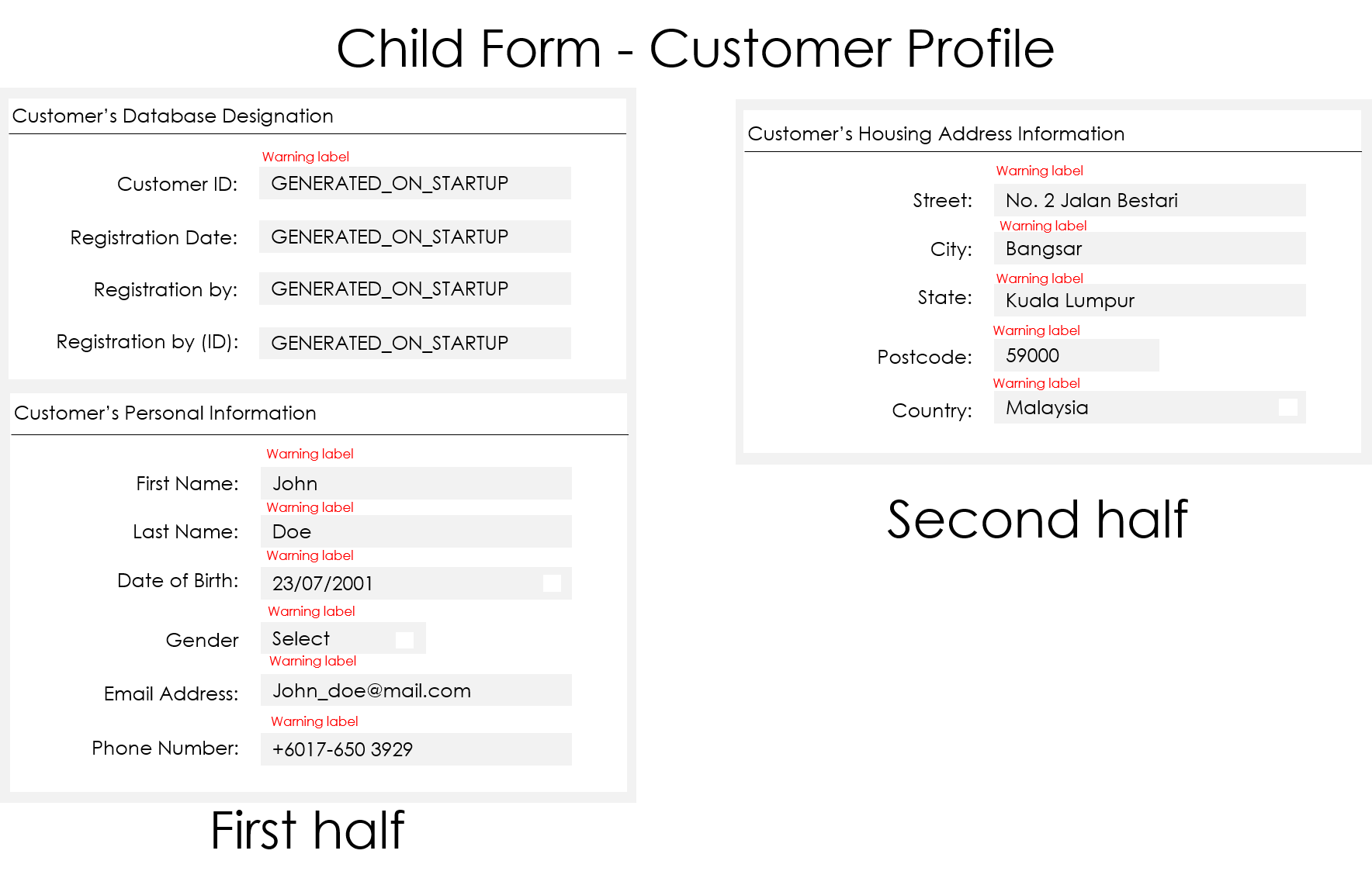


**Figure 3.2: Customer registration interface labelling.**

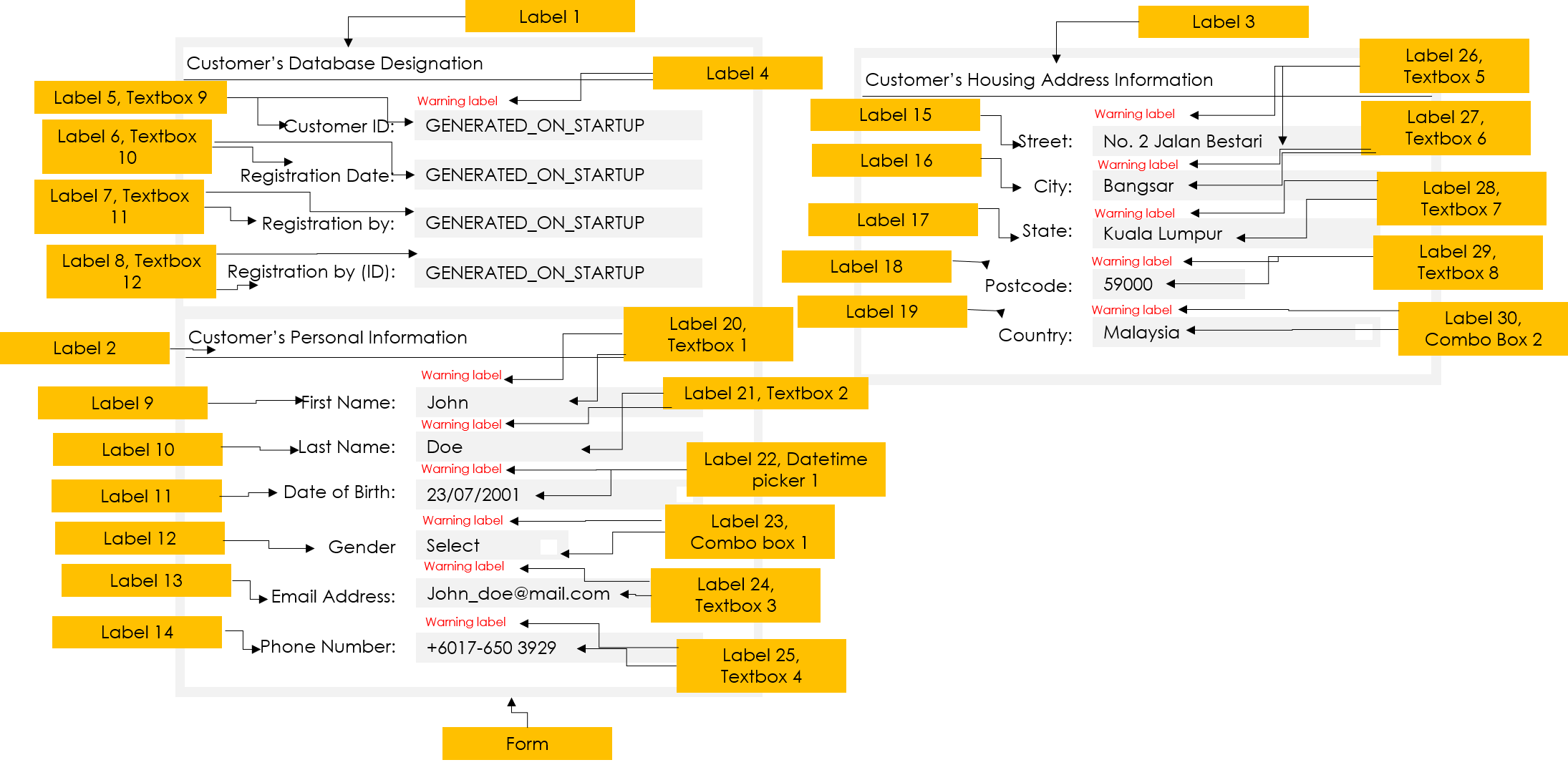
### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | customerregister | The form |
| Label 1 | lblCustDBHeader | Label displaying customer database information title |
| Label 2 | lblCustPerHeader | Label displaying customer personal information title |
| Label 3 | lblCustHouHeader | Label displaying customer housing address title |
| Label 4 | lblCustIDWarn | Label to display warning message for Customer ID |
| Label 5 | lblCustID | Label displaying Customer ID title |
| Label 6 | lblRegDate | Label displaying Registration Date title |
| Label 7 | lblRegTime | Label displaying Registration Time title |
| Label 8 | lblFirstName | Label displaying First Name title |
| Label 9 | lblLastName | Label displaying Last Name title |
| Label 10 | lblDOB | Label displaying Date of Birth title |
| Label 11 | lblGender | Label displaying Gender title |
| Label 12 | lblEmail | Label displaying Email Address title |
| Label 13 | lblPhone | Label displaying Phone Number title |
| Label 14 | lblStreet | Label displaying Street title |
| Label 15 | lblCity | Label displaying City title |
| Label 16 | lblState | Label displaying State title |
| Label 17 | lblPostcode | Label displaying Postcode title |
| Label 18 | lblCountry | Label displaying Country title |
| Label 19 | lblFirstNameWarn | Label to display warning message for First Name |
| Label 20 | lblLastNameWarn | Label to display warning message for Last Name |
| Label 21 | lblDOBWarn | Label to display warning message for Date of Birth |
| Label 22 | lblGenderWarn | Label to display warning message for Gender |
| Label 23 | lblEmailWarn | Label to display warning message for Email Address |
| Label 24 | lblPhoneWarn | Label to display warning message for Phone Number |
| Label 25 | lblStreetWarn | Label to display warning message for Street |
| Label 26 | lblCityWarn | Label to display warning message for City |
| Label 27 | lblStateWarn | Label to display warning message for State |
| Label 28 | lblPostcodeWarn | Label to display warning message for Postcode |
| Label 29 | lblCountryWarn | Label to display warning message for Country |
| Text Box 1 | txtFirstName | Textbox to accept First Name input |
| Text Box 2 | txtLastName | Textbox to accept Last Name input |
| Text Box 3 | txtEmail | Textbox to accept Email Address input |
| Text Box 4 | txtPhone | Textbox to accept Phone Number input |
| Text Box 5 | txtStreet | Textbox to accept Street input |
| Text Box 6 | txtCity | Textbox to accept City input |
| Text Box 7 | txtState | Textbox to accept State input |
| Text Box 8 | txtPostcode | Textbox to accept Postcode input |
| Text Box 9 | txtCustID | Textbox to display generated Customer ID |
| Text Box 10 | txtRegDate | Textbox to display generated Registration Date |
| Text Box 11 | txtRegTime | Textbox to display generated Registration Time |
| Combo Box 1 | cbxGender | Combo box for Gender selection |
| Combo Box 2 | cbxCountry | Combo box for Country selection |
| Datetime Picker 1 | dtpDOB | Datetime picker for Date of Birth |

## Part 4 – Customer Profile Interface



**Figure 4.1: Customer profile interface separated into two parts.**

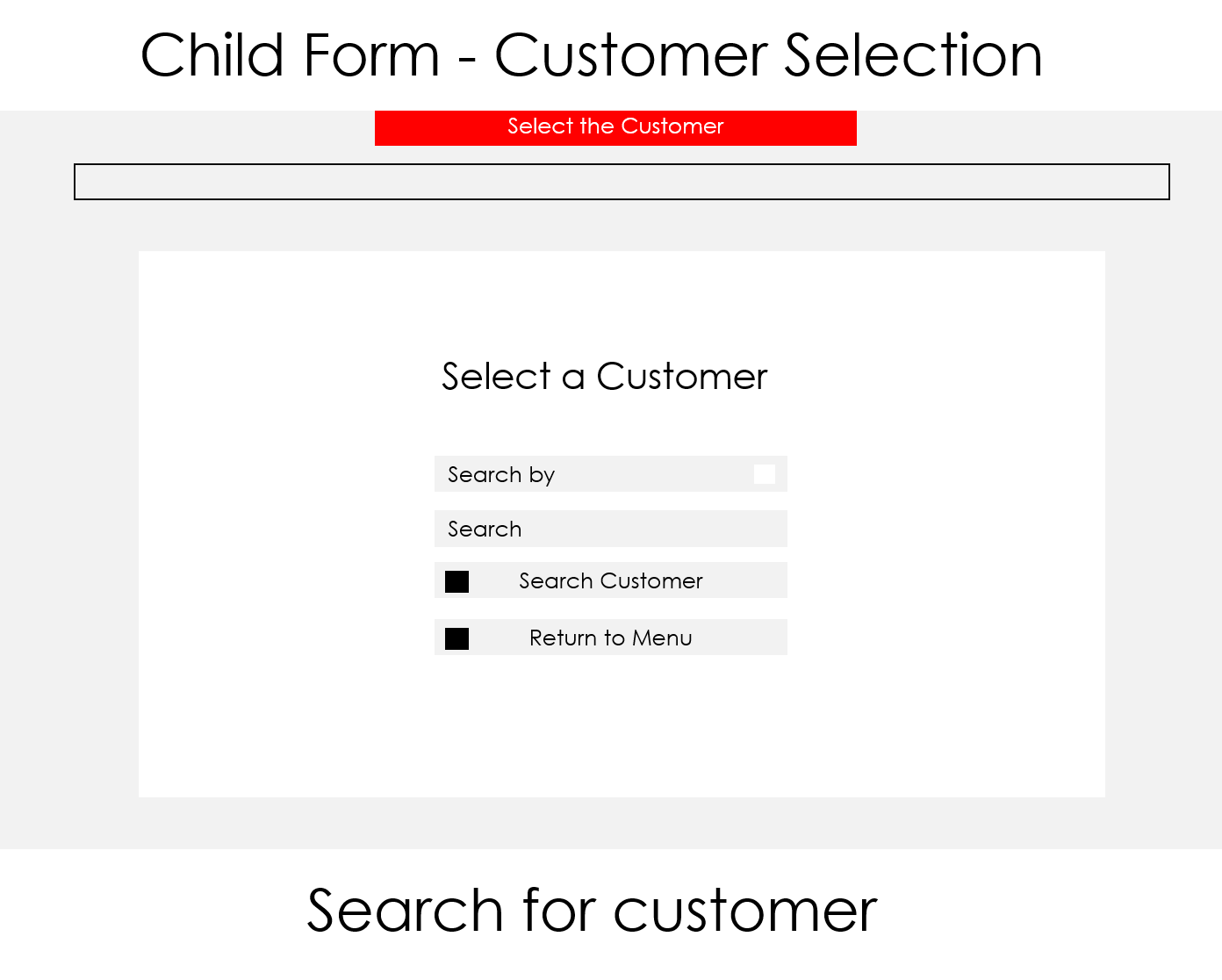


**Figure 4.2: Customer profile interface labelling.**

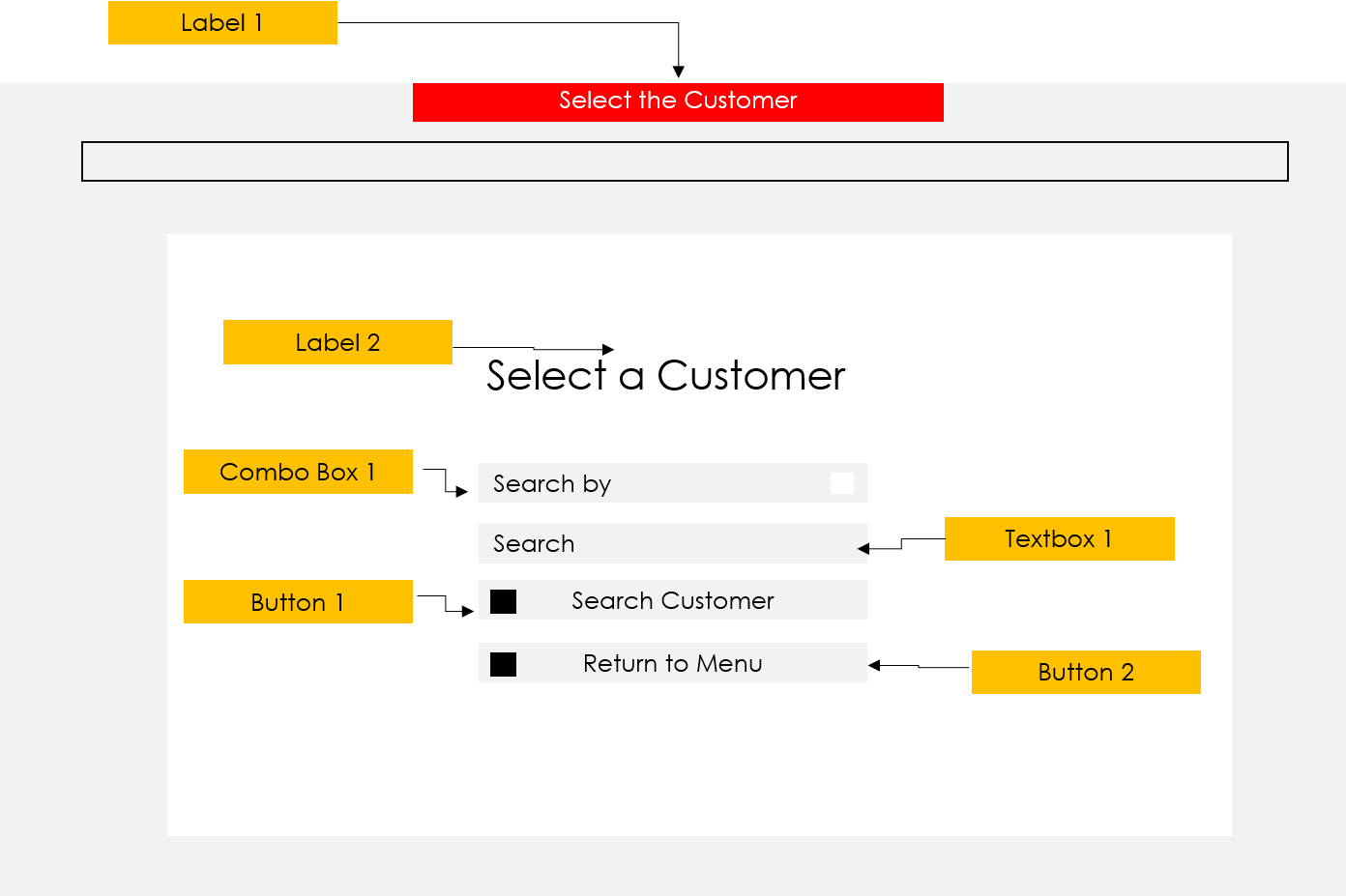
### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | customerprofile | The form |
| Label 1 | lblCustDBHeader | Label displaying customer database information title |
| Label 2 | lblCustPerHeader | Label displaying customer personal information title |
| Label 3 | lblCustHouHeader | Label displaying customer housing address title |
| Label 4 | lblCustIDWarn | Label to display warning message for Customer ID |
| Label 5 | lblCustID | Label displaying Customer ID title |
| Label 6 | lblRegDate | Label displaying Registration Date title |
| Label 7 | lblRegBy | Label displaying Staff Name title |
| Label 8 | lblRegByID | Label displaying Staff Name ID title |
| Label 9 | lblFirstName | Label displaying First Name title |
| Label 10 | lblLastName | Label displaying Last Name title |
| Label 11 | lblDOB | Label displaying Date of Birth title |
| Label 12 | lblGender | Label displaying Gender title |
| Label 13 | lblEmail | Label displaying Email Address title |
| Label 14 | lblPhone | Label displaying Phone Number title |
| Label 15 | lblStreet | Label displaying Street title |
| Label 16 | lblCity | Label displaying City title |
| Label 17 | lblState | Label displaying State title |
| Label 18 | lblPostcode | Label displaying Postcode title |
| Label 19 | lblCountry | Label displaying Country title |
| Label 20 | lblFirstNameWarn | Label to display warning message for First Name |
| Label 21 | lblLastNameWarn | Label to display warning message for Last Name |
| Label 22 | lblDOBWarn | Label to display warning message for Date of Birth |
| Label 23 | lblGenderWarn | Label to display warning message for Gender |
| Label 24 | lblEmailWarn | Label to display warning message for Email Address |
| Label 25 | lblPhoneWarn | Label to display warning message for Phone Number |
| Label 26 | lblStreetWarn | Label to display warning message for Street |
| Label 27 | lblCityWarn | Label to display warning message for City |
| Label 28 | lblStateWarn | Label to display warning message for State |
| Label 29 | lblPostcodeWarn | Label to display warning message for Postcode |
| Label 30 | lblCountryWarn | Label to display warning message for Country |
| Text Box 1 | txtFirstName | Textbox to accept First Name input |
| Text Box 2 | txtLastName | Textbox to accept Last Name input |
| Text Box 3 | txtEmail | Textbox to accept Email Address input |
| Text Box 4 | txtPhone | Textbox to accept Phone Number input |
| Text Box 5 | txtStreet | Textbox to accept Street input |
| Text Box 6 | txtCity | Textbox to accept City input |
| Text Box 7 | txtState | Textbox to accept State input |
| Text Box 8 | txtPostcode | Textbox to accept Postcode input |
| Text Box 9 | txtCustID | Textbox to display generated Customer ID |
| Text Box 10 | txtRegDate | Textbox to display generated Registration Date |
| Text Box 11 | txtRegBy | Textbox to display Staff Name |
| Text Box 12 | txtRegByID | Textbox to display Staff Name ID |
| Combo Box 1 | cbxGender | Combo box for Gender selection |
| Combo Box 2 | cbxCountry | Combo box for Country selection |
| Datetime Picker 1 | dtpDOB | Datetime picker for Date of Birth |

## Part 5 – Customer Selection Interface



**Figure 5.1: Customer selection interface main menu.**

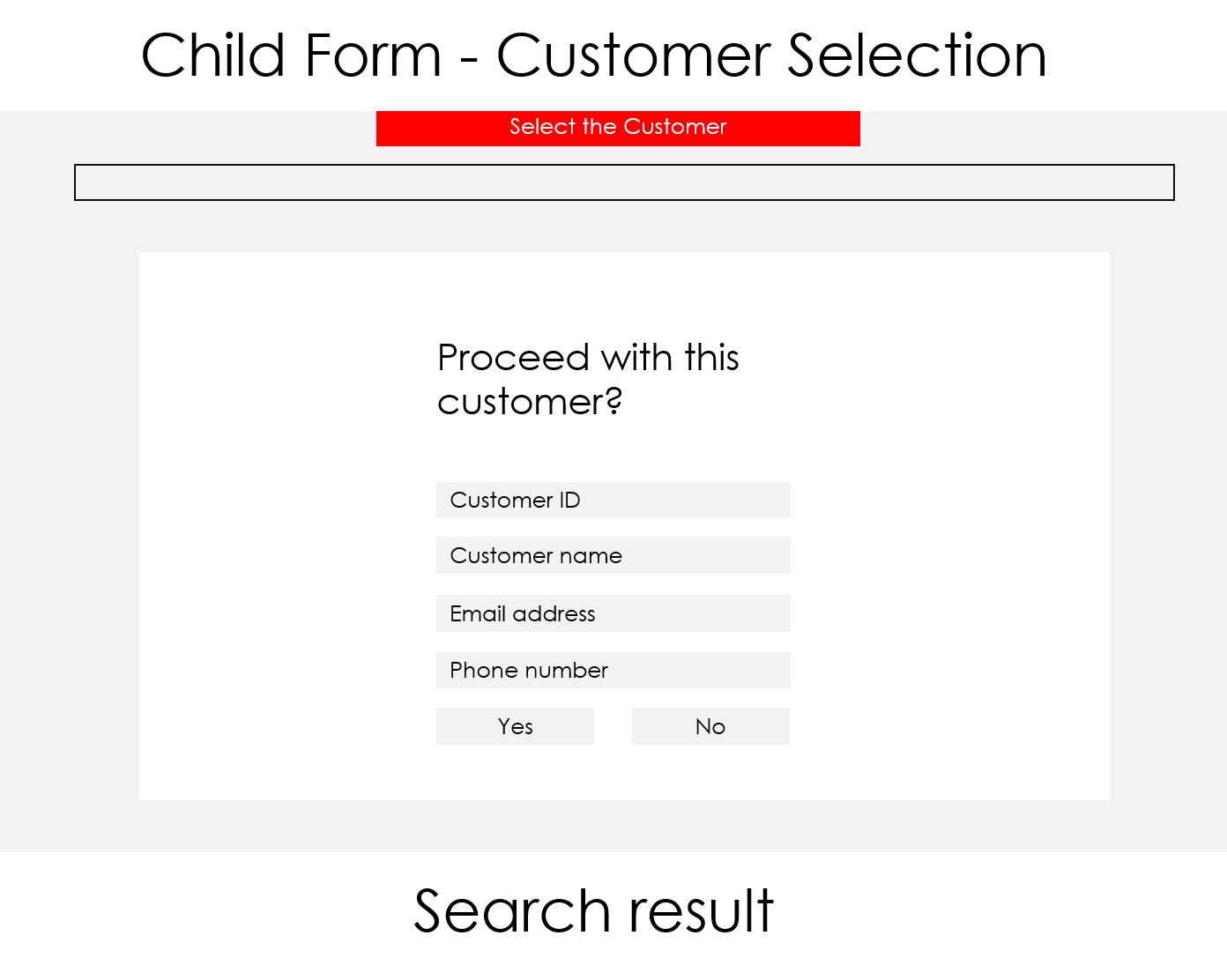


**Figure 5.2: Customer selection interface labelling.**

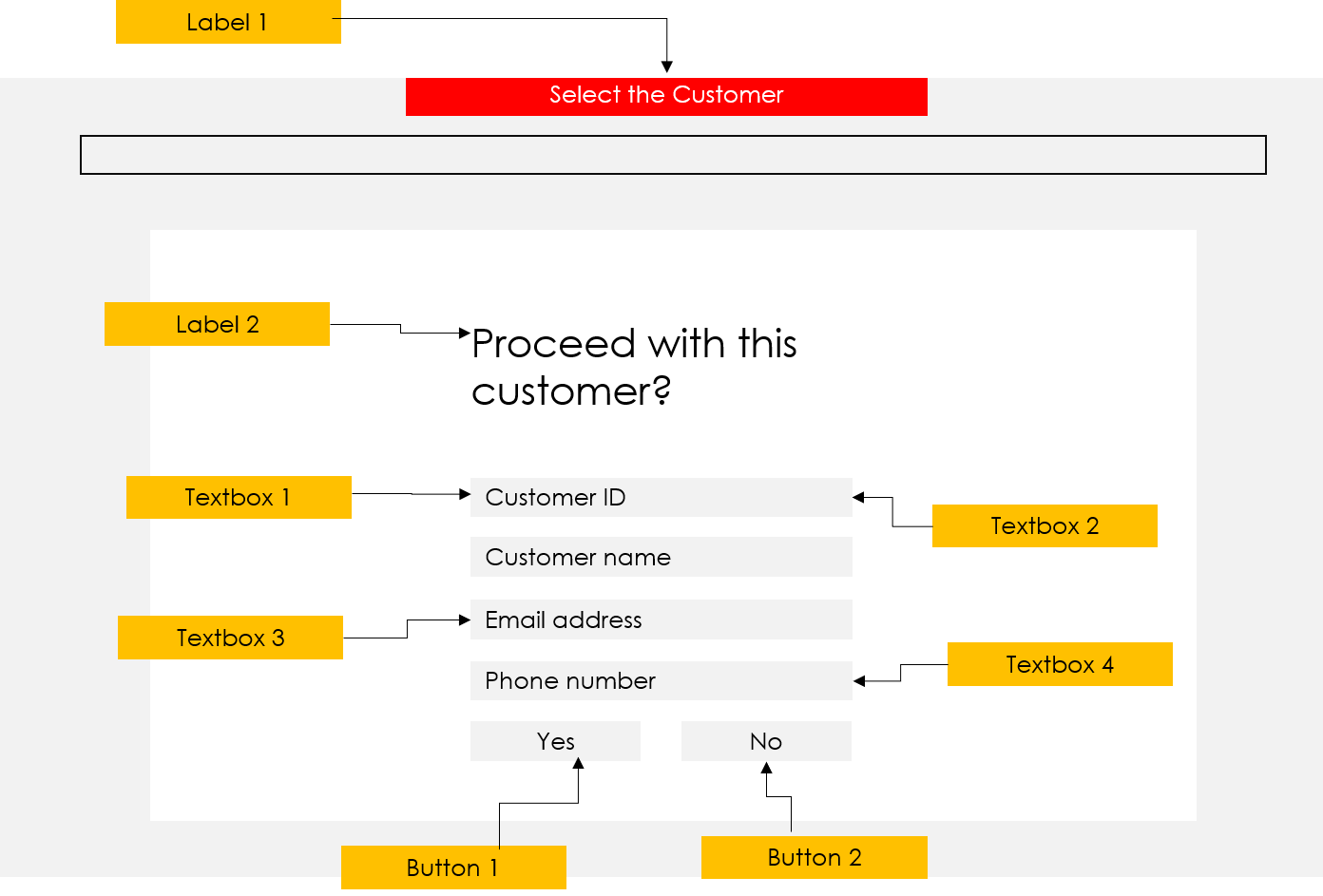
### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | customerselection | The form |
| Label 1 | lblHeader | Label displaying the header |
| Label 2 | lblTitle | Label displaying title |
| Combo Box 1 | cbxCriteria | Combo box for criteria selection |
| Text Box 1 | txtSearch | Textbox to accept user search keyword |
| Button 1 | btnSearch | Button to search customer |
| Button 2 | btnReturnMenu | Button to return to menu |

## Part 5.1 – Customer Selection Interface Search Result



**Figure 5.1: Customer selection interface fetched result.**

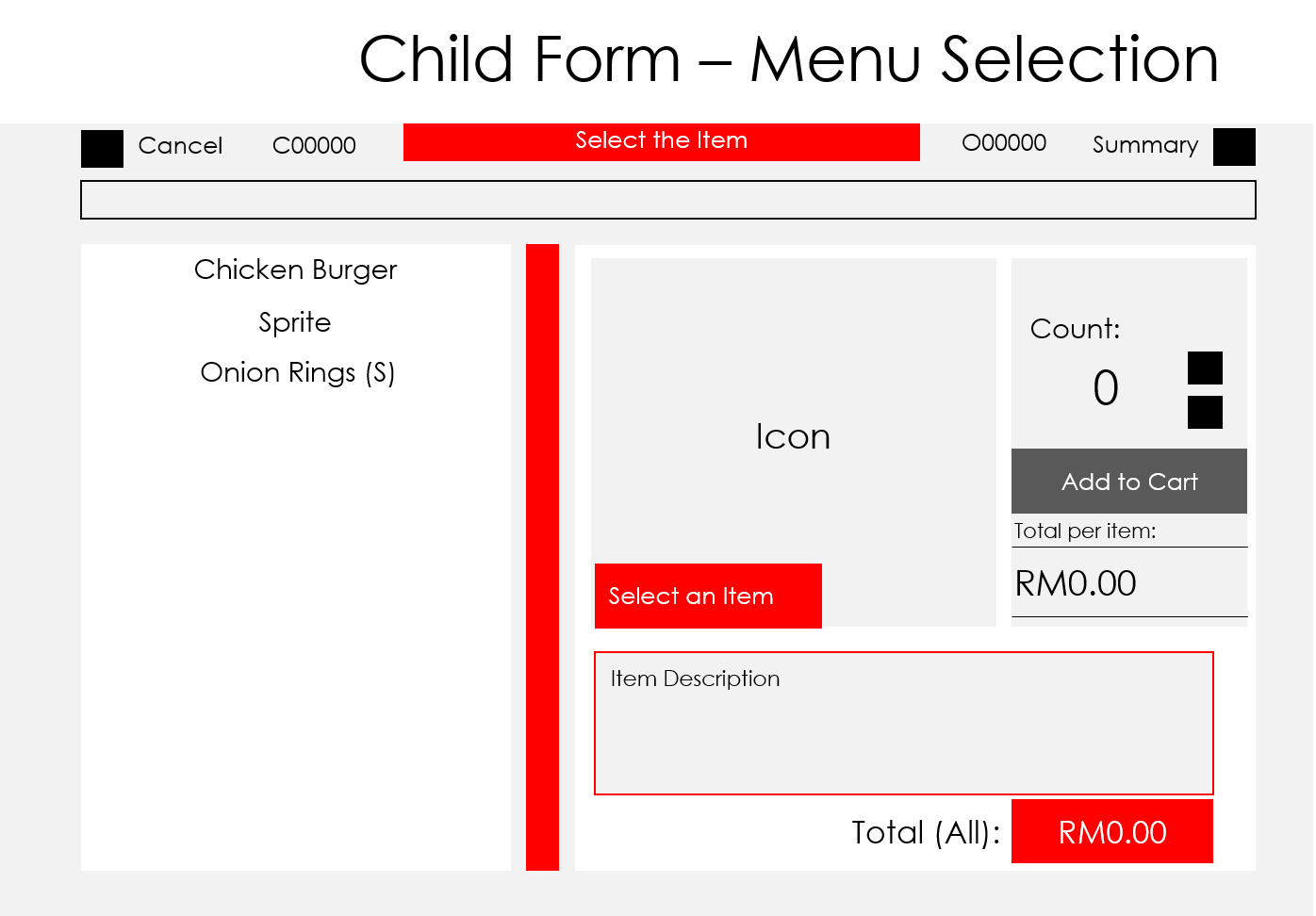


**Figure 5.1: Customer selection interface fetched result labelling.**

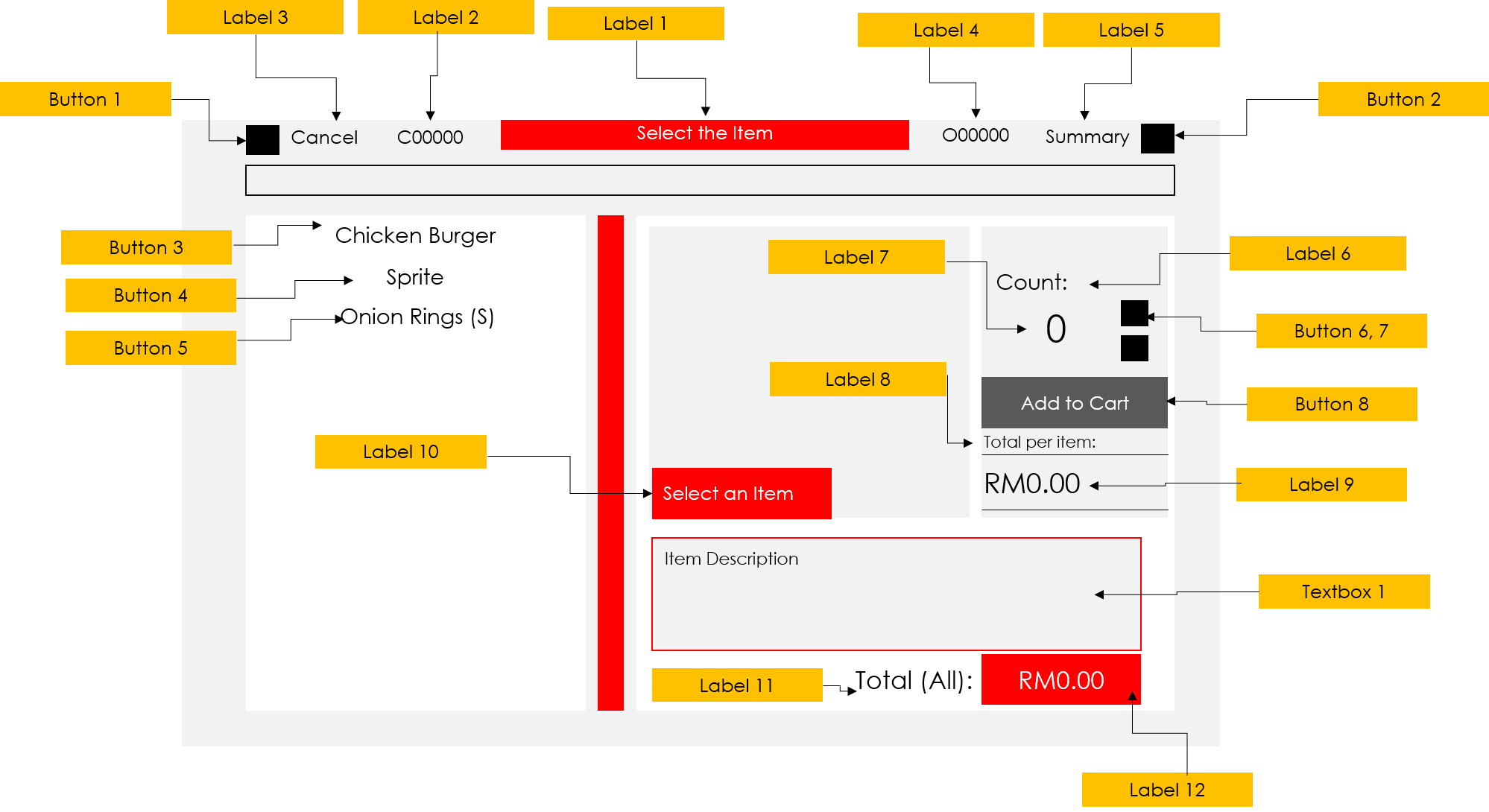
### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | customerselection | The form |
| Label 1 | lblHeader | Label displaying the header |
| Label 2 | lblTitle | Label displaying title |
| Text Box 1 | txtCustID | Textbox displaying Customer ID |
| Text Box 2 | txtCustName | Textbox displaying Customer Name |
| Text Box 3 | txtCustEmail | Textbox displaying Customer Email |
| Text Box 4 | txtCustPhone | Textbox displaying Customer Phone |
| Button 1 | btnProceed | Button to proceed to menu selection |
| Button 2 | btnReturn | Button to return to search |

## Part 6 – Menu Selection Interface



**Figure 6.1: Menu selection interface main form.**

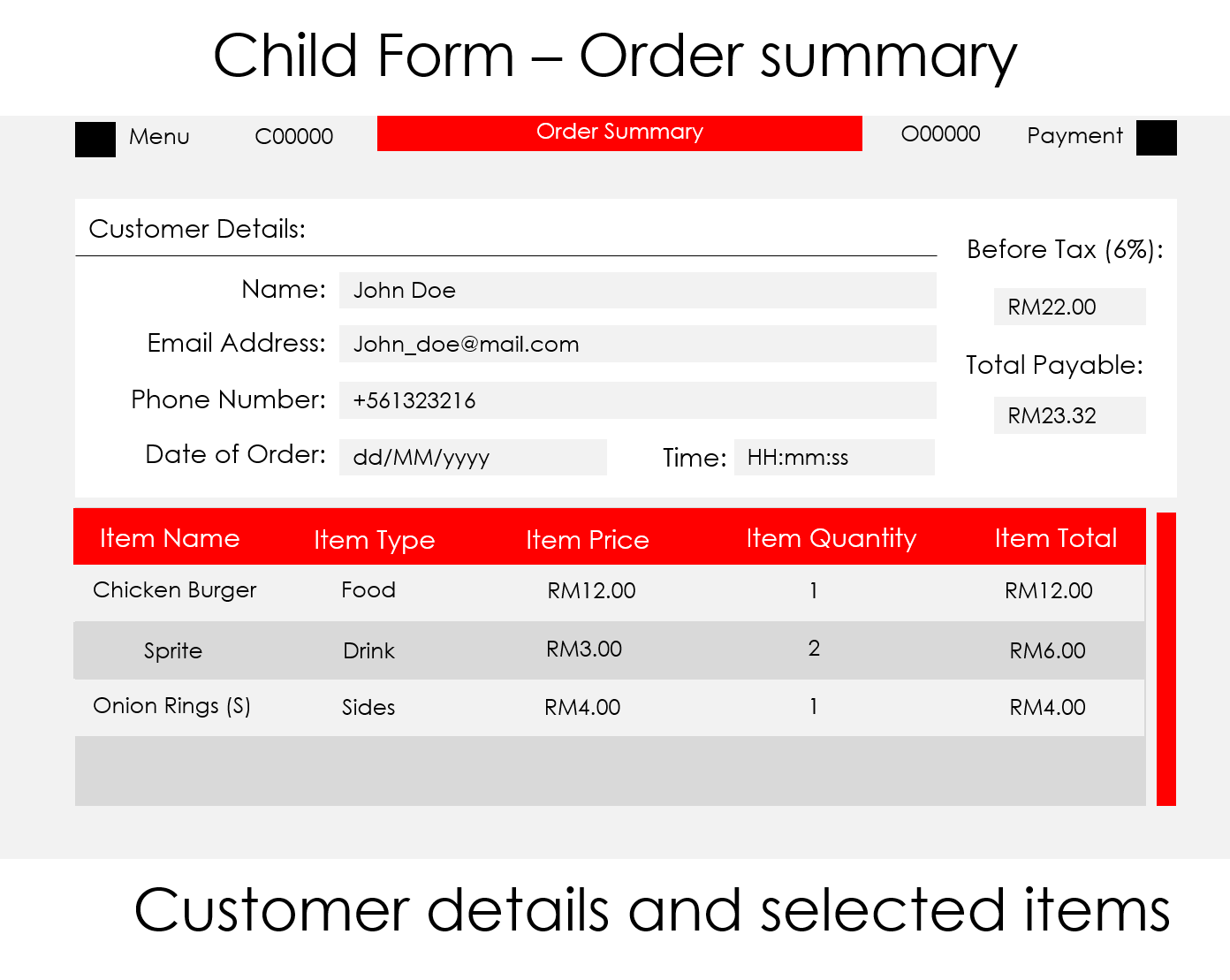


**Figure 6.2: Menu selection interface labelling.**

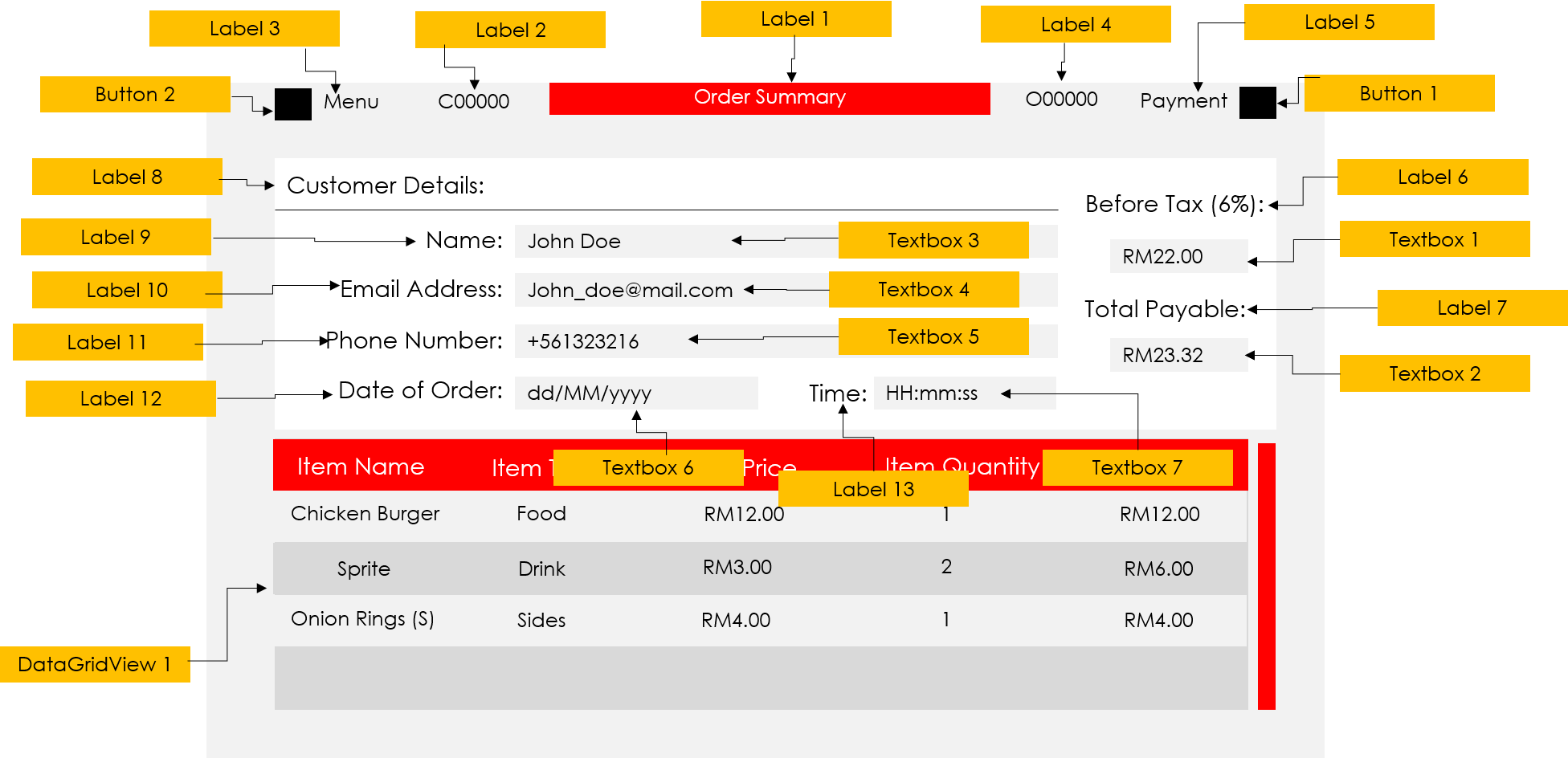
### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | menuselection | The form |
| Label 1 | lblHeader | Label displaying the header |
| Label 2 | lblCustID | Label displaying Customer ID |
| Label 3 | lblCancel | Label displaying Cancel title |
| Label 4 | lblOrdID | Label displaying Order ID |
| Label 5 | lblSummary | Label displaying Summary title |
| Label 6 | lblCount | Label displaying Count title |
| Label 7 | lblCountNum | Label displaying Count total |
| Label 8 | lblTotalItem | Label displaying Total Price Per Item title |
| Label 9 | lblTotalItemRM | Label displaying Total Price Per Item |
| Label 10 | lblItemPriceRM | Label displaying Price Per Item |
| Label 11 | lblTotalAll | Label displaying Total All title |
| Label 12 | lblTotalAllRM | Label displaying Total All |
| Button 1 | btnCancel | Button to cancel order |
| Button 2 | btnSummary | Button to go to order summary interface |
| Button 3 | btnChickBur | Button to load Chicken Burger information |
| Button 4 | btnSprite | Button to load Sprite information |
| Button 5 | btnOringsS | Button to load Onion Rings information |
| Button 6 | btnInc | Button to increment counter |
| Button 7 | btnDec | Button to decrement counter |
| Button 8 | btnAddCart | Button to add item to cart |
| Text Box 1 | txtDesc | Textbox displaying item description |

## Part 7 – Order Summary Interface



**Figure 7.1: Order summary interface main form.**

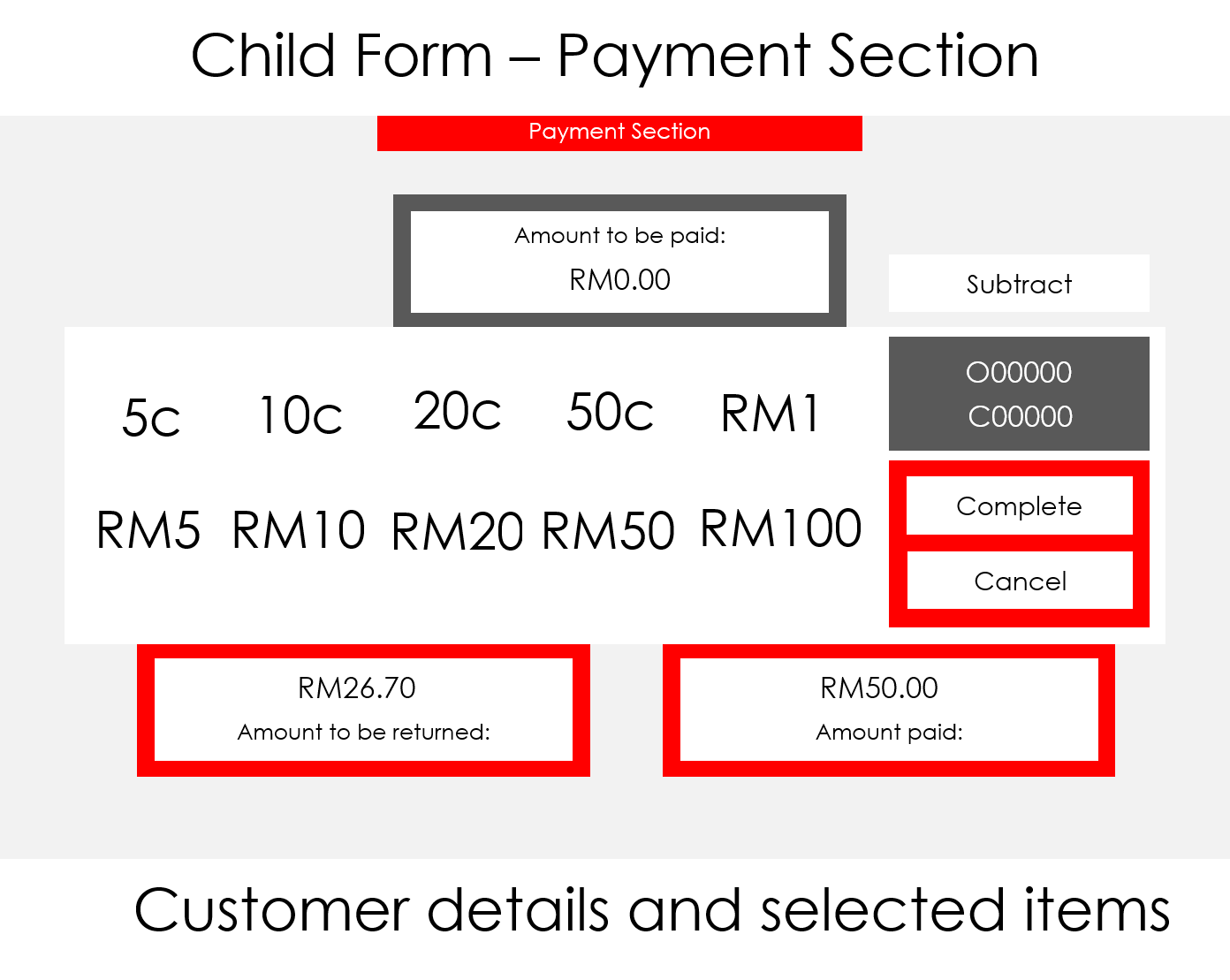


**Figure 7.2: Order summary interface labelling.**

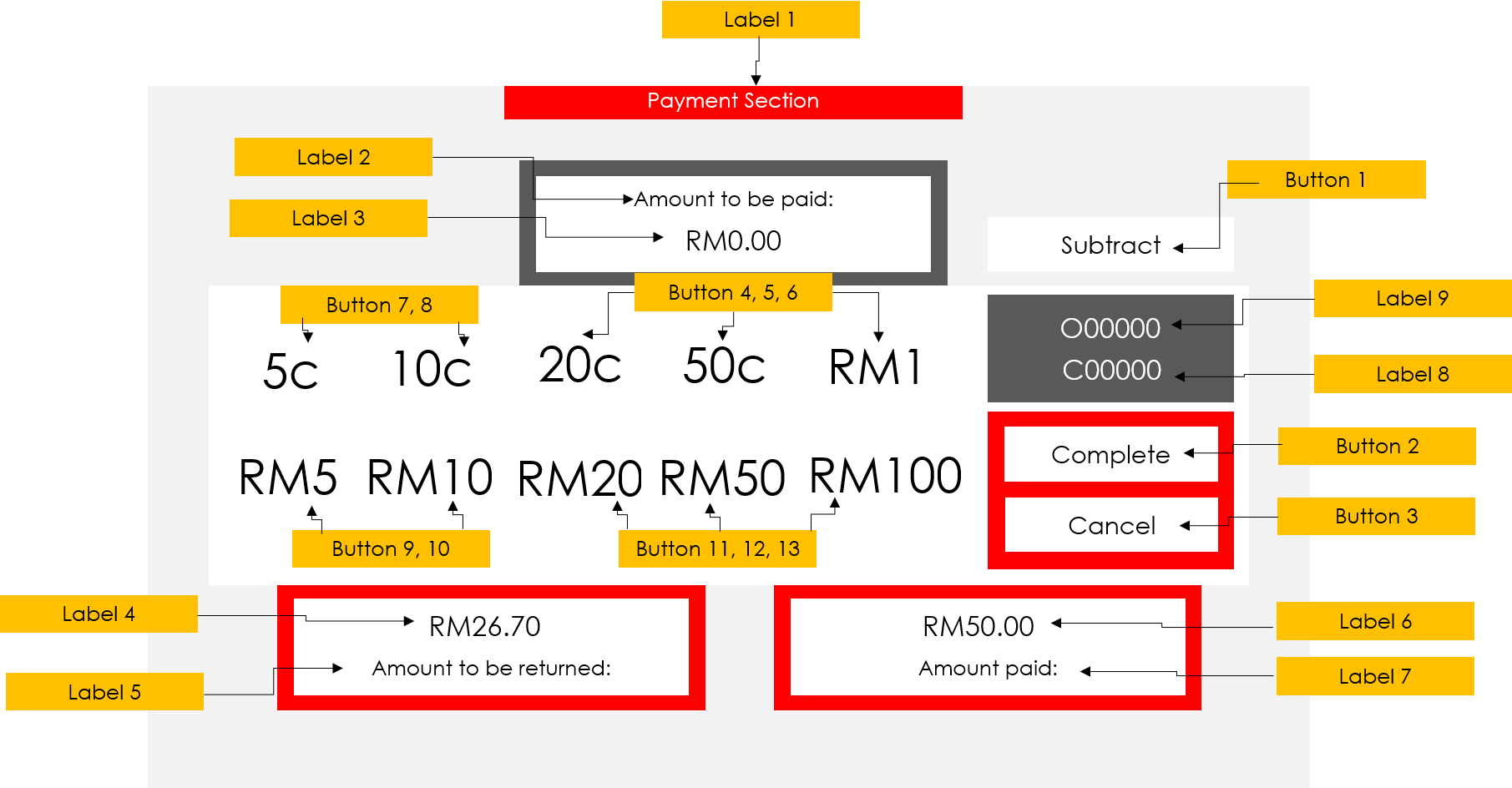
### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | ordersummary | The form |
| Label 1 | lblHeader | Label displaying the header |
| Label 2 | lblCustID | Label displaying Customer ID |
| Label 3 | lblMenu | Label displaying Menu title |
| Label 4 | lblOrdID | Label displaying Order ID |
| Label 5 | lblPayment | Label displaying Payment title |
| Label 6 | lblBeforeTax | Label displaying Before Tax title |
| Label 7 | lblTotalPayable | Label displaying Total Payable total |
| Label 8 | lblDetHeader | Label displaying Customer Details title |
| Label 9 | lblCustName | Label displaying Customer Name title |
| Label 10 | lblCustEmail | Label displaying Customer Email title |
| Label 11 | lblCustPhone | Label displaying Customer Phone title |
| Label 12 | lblDateOrd | Label displaying Date Order title |
| Label 13 | lblTimeOrd | Label displaying Time Order title |
| Text Box 1 | txtBeforeTax | Textbox displaying amount before tax |
| Text Box 2 | txtTotalPayable | Textbox displaying amount after tax |
| Text Box 3 | txtCustName | Textbox displaying Customer Name |
| Text Box 4 | txtCustEmail | Textbox displaying Customer Email |
| Text Box 5 | txtCustPhone | Textbox displaying Customer Phone |
| Text Box 6 | txtDateOrd | Textbox displaying Date Order |
| Text Box 7 | txtTimeOrd | Textbox displaying Time Order |
| Button 1 | btnPayment | Button to go to menu selection interface |
| Button 2 | btnMenu | Button to go to payment interface |
| DataGridView 1 | dgvSummary | Datagridview to load order details |

## Part 8 – Payment Interface



**Figure 8.1: Payment interface main form.**

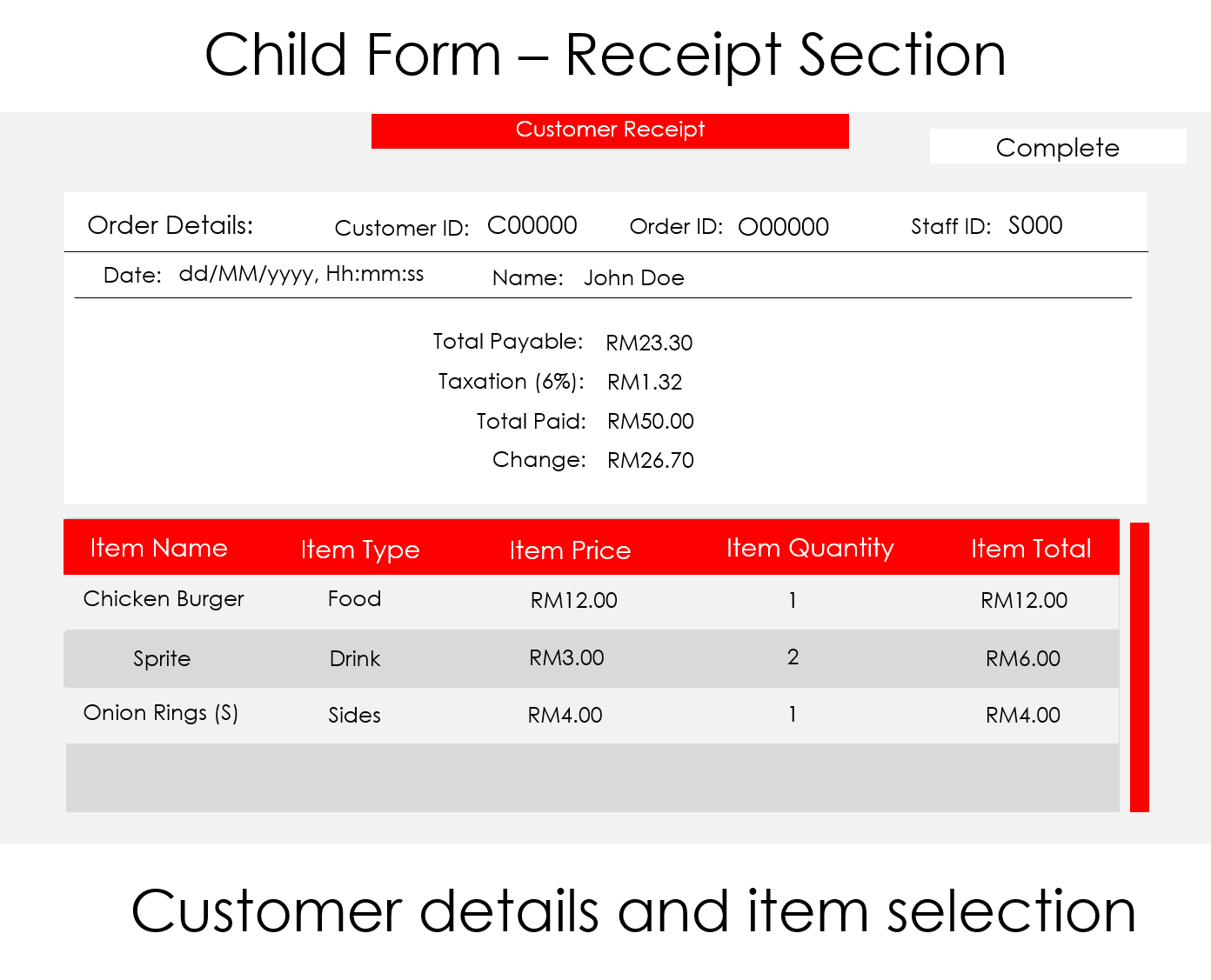


**Figure 8.2: Payment interface labelling.**

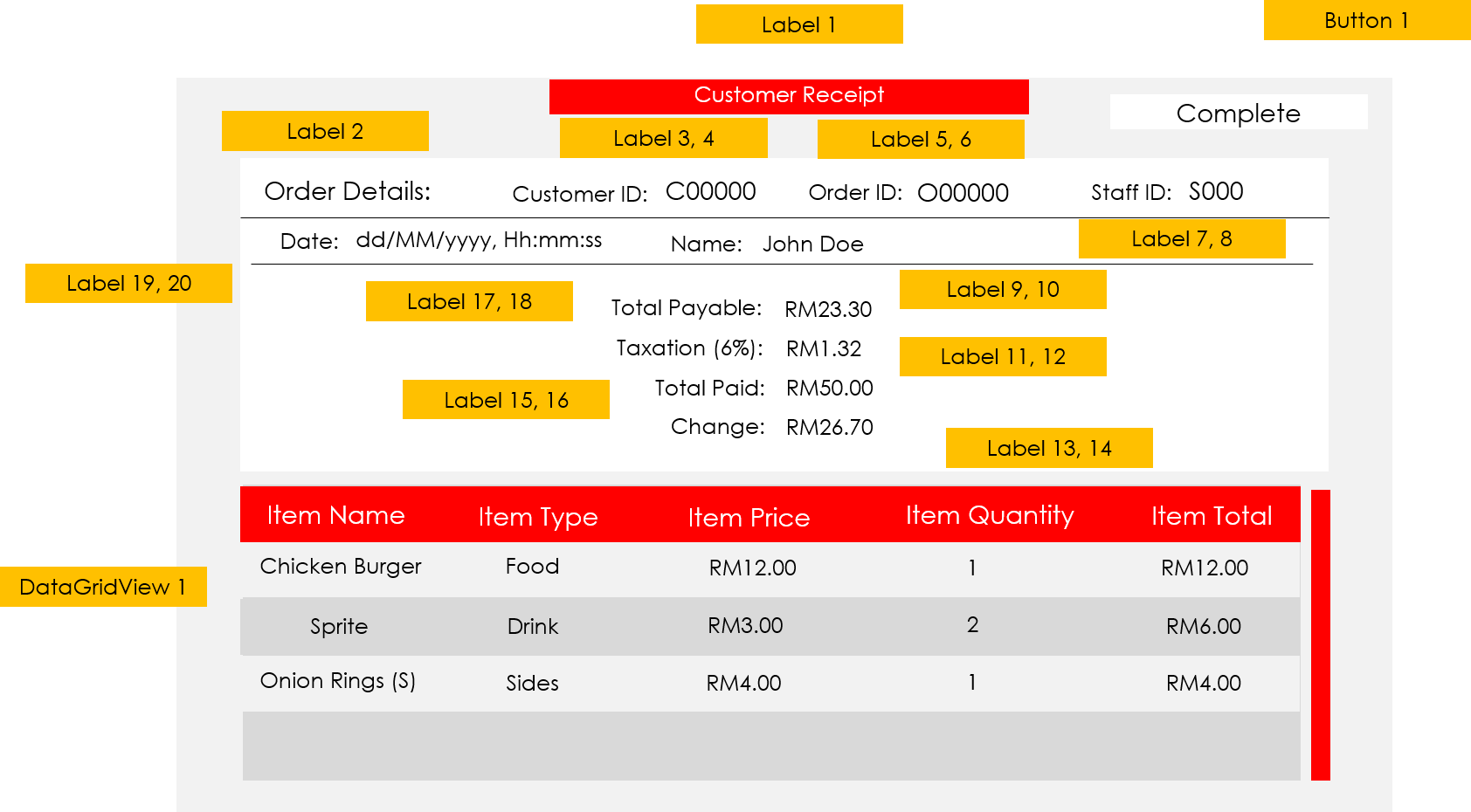
### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | payment | The form |
| Label 1 | lblHeader | Label displaying the header |
| Label 2 | lblToBePaid | Label displaying amount to be paid title |
| Label 3 | lblToBePaidRM | Label displaying amount to be paid |
| Label 4 | lblChange | Label displaying change amount title |
| Label 5 | lblChangeRM | Label displaying change amount |
| Label 6 | lblPaid | Label displaying paid amount title |
| Label 7 | lblPaidRM | Label displaying paid amount |
| Label 8 | lblCustID | Label displaying Customer ID |
| Label 9 | lblOrdID | Label displaying Order ID |
| Button 1 | btnSub | Button handles subtract value |
| Button 2 | btnComplete | Button to go to receipt interface |
| Button 3 | btnCancel | Button to cancel order |
| Button 4 | btn20c | Button to increment by 20 cent |
| Button 5 | btn50c | Button to increment by 50 cent |
| Button 6 | btnrm1 | Button to increment by RM 1 |
| Button 7 | btn5c | Button to increment by 5 cent |
| Button 8 | btn10c | Button to increment by 10 cent |
| Button 9 | btnrm5 | Button to increment by RM5 |
| Button 10 | btnrm10 | Button to increment by RM10 |
| Button 11 | btnrm20 | Button to increment by RM20 |
| Button 12 | btnrm50 | Button to increment by RM50 |
| Button 13 | btnrm100 | Button to increment by RM100 |

## Part 9 – Receipt Interface



**Figure 9.1: Receipt interface main form.**



**Figure 9.2: Receipt interface labelling.**

### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | payment | The form |
| Label 1 | lblHeader | Label displaying the header |
| Label 2 | lblOrdDet | Label displaying Order Detail title |
| Label 3 | lblCustIDTitle | Label displaying Customer ID title |
| Label 4 | lblCustID | Label displaying Customer ID |
| Label 5 | lblOrdIDTitle | Label displaying Order ID title |
| Label 6 | lblOrdID | Label displaying Order ID |
| Label 7 | lblStaIDTitle | Label displaying Staff ID title |
| Label 8 | lblStaID | Label displaying Staff ID |
| Label 9 | lblCustNameTitle | Label displaying Customer Name title |
| Label 10 | lblCustName | Label displaying Customer Name |
| Label 11 | lblTaxTitle | Label displaying Tax title |
| Label 12 | lblTax | Label displaying tax amount |
| Label 13 | lblChangeTitle | Label displaying Change title |
| Label 14 | lblChange | Label displaying change amount |
| Label 15 | lblPaidTitle | Label displaying Paid title |
| Label 16 | lblPaid | Label displaying paid amount |
| Label 17 | lblPayableTitle | Label displaying Payable title |
| Label 18 | lblPayable | Label displaying payable amount |
| Label 19 | lblDateTitle | Label displaying Date title |
| Label 20 | lblDate | Label displaying date ordered |
| Button 1 | btnComplete | Button to go to staff menu after ordering |
| DataGridView 1 | dgvDetails | Datagridview to show list of ordered items |

## Part 12 – Splash screen Interface





### Definition

|  |  |  |
| --- | --- | --- |
| Control | Control Name | Description |
| Form | splashscreen | The form |
| Picture Box 1 | pbxLoadingPic | Picture box to display the cafeteria logo |
| Label | lblLoading | Label to display loading captions |

# 3.0 USER MANUAL

## Usage Overview - Ordering

When an order is about to take place, the customer must specify to the staff whether they have registered before. If they have, the customer will need to give their customer ID number to the staff for search purpose. Once there is a result, the staff will start taking the customer’s order by going through food, drink and sides selection interfaces. Once items are selected, the staff will be brought to order summary interface where the list of ordered items will appear. The staff can the all the items selected, and double confirm with the customer before proceeding with payment. Once confirmed, the staff will be redirected to the payment interface. Here, the staff will ask for the customer to pay according to the total displayed. The total is calculated by adding tax and rounding it off to the nearest places to make it even. After receiving the payment, the system will display change amount if there is any. Lastly, the staff will be sent to the receipt details section for final check with the customer. Finally, the staff needs to press the ‘Complete’ button to finalise the order. The record of the sales will be stored inside the database for references.

## Usage Overview – Registering

If the customer has not registered, the staff must register them before being able to use to ordering system. First, the staff needs to navigate to the customer registration form. Next, the staff will have to fill in every fields as it is required to have all details set before the database can be updated. If there are any unfulfilled conditions, a warning popup will appear to ask the staff to fix the problems identified. Once all errors are dealt with, the staff can register the customer. The staff must inform the customer of their ID before using the system.

## Usage Overview – Modifying

If the customer record needs to be updated for any reason, the staff can do so by accessing the customer profile. First, the staff will need to search for the record by selecting any of the criteria provided. Then, the staff must key-in the keyword(s) based on the criteria selected. However, if the staff inserted an invalid keyword, the system will produce an error message asking the user to address the issue immediately. The same applies to when the staff searches for a deleted record. Once the staff finishes dealing with the error, the record will load into appropriate fields. To update the record, the staff must request for edit permission which can be accessed from the side panel. After requesting, the staff will have edit privilege and all the fields will become accessible for amendment. If the staff is done with the editing, they can apply the changes by accessing the side panel again.

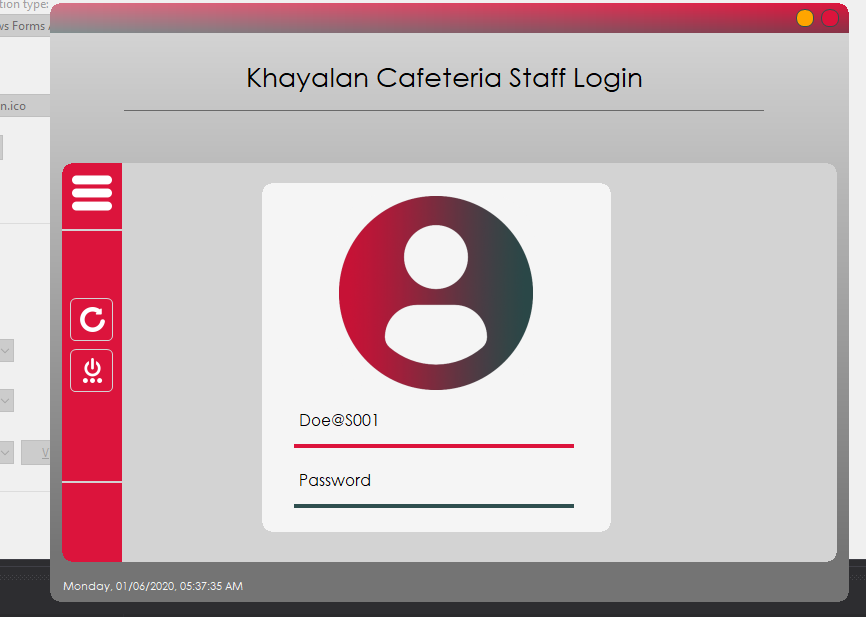
## Usage – Logging in as staff



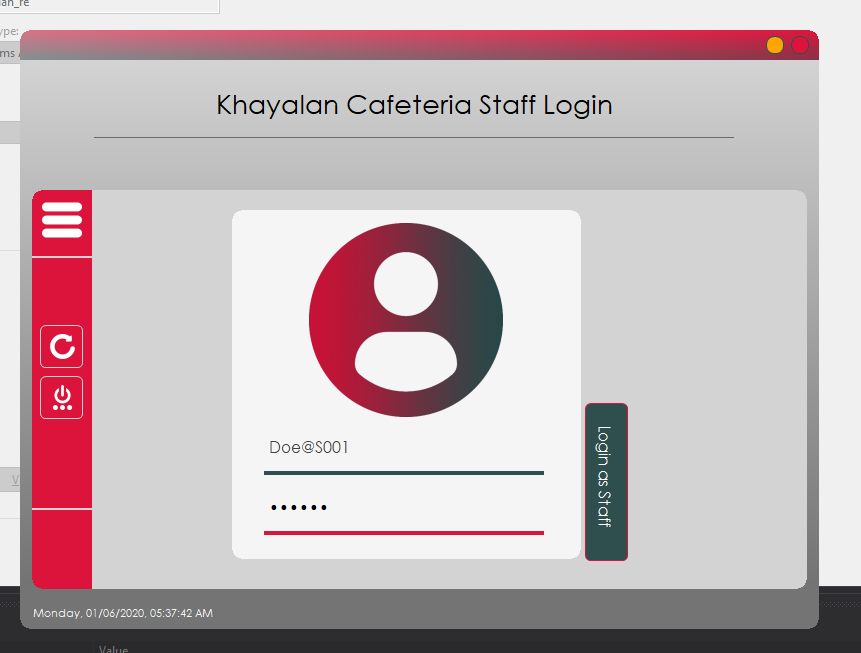
The splash screen will appear as soon as the application runs. It will act as a loading screen for necessary items to be loaded before the staff can use the system.



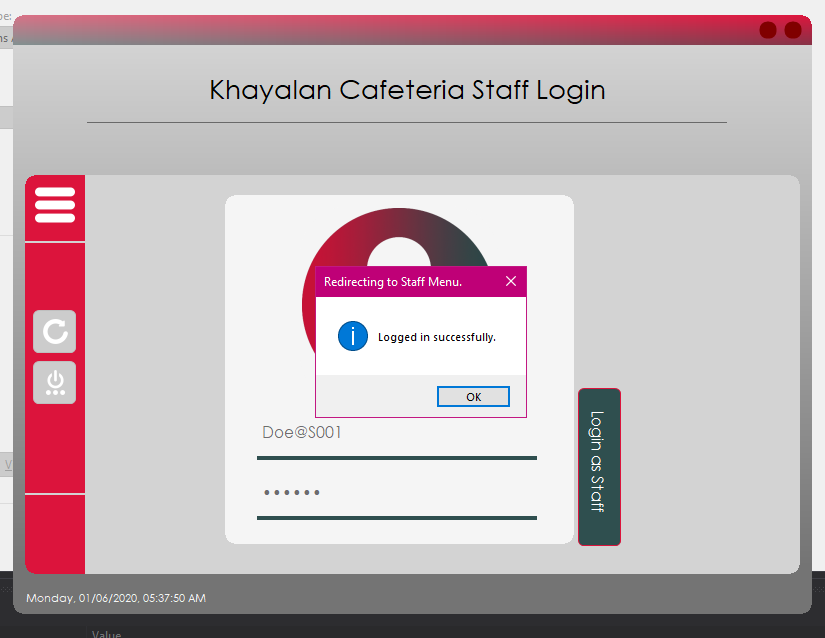
Now, the staff has arrived at the main login screen. Here, the staff can see several controls like a button to reset the form, close the application but not the login button. This is because the button is set to hidden when there is no user input within both textboxes.



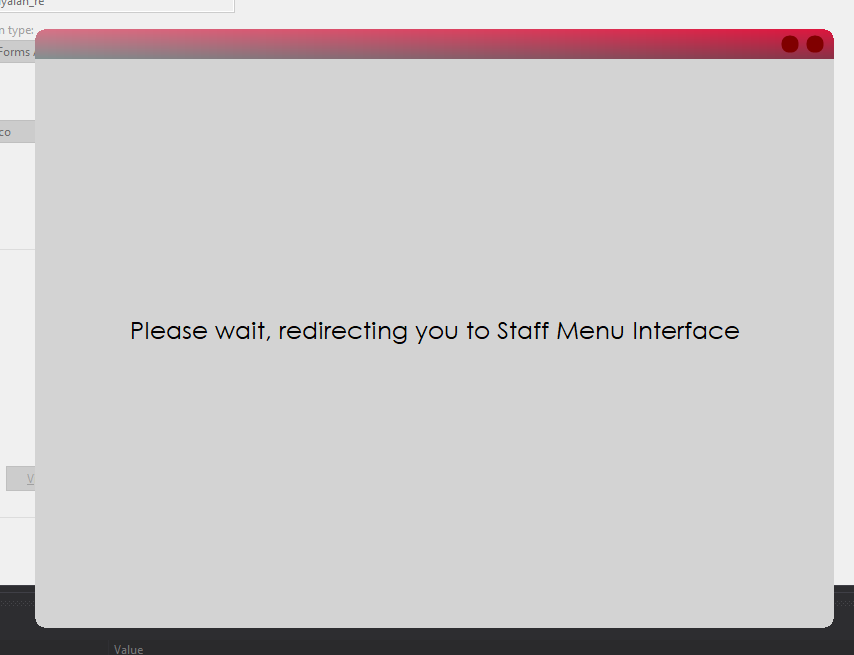
The staff can now start inserting the username first before the password.



After the username has been inserted, the staff can insert the password. As it is seen here, the button that handles user login appears. The staff can now proceed with login. Press the button to continue.



Once pressed, it depends on the credentials checking mechanism to validate whether the staff credentials inserted are valid or not. If the credentials are correct, the popup above can be seen. The staff must press on OK to continue with the whole login sequence. If, however, the staff inserted the wrong combination of username or password, another variant of this popup will appear. That error will prevent the user from logging in until a correct credentials are inserted and validated. In the end, the staff must possess a correct credentials before able to log into the system.

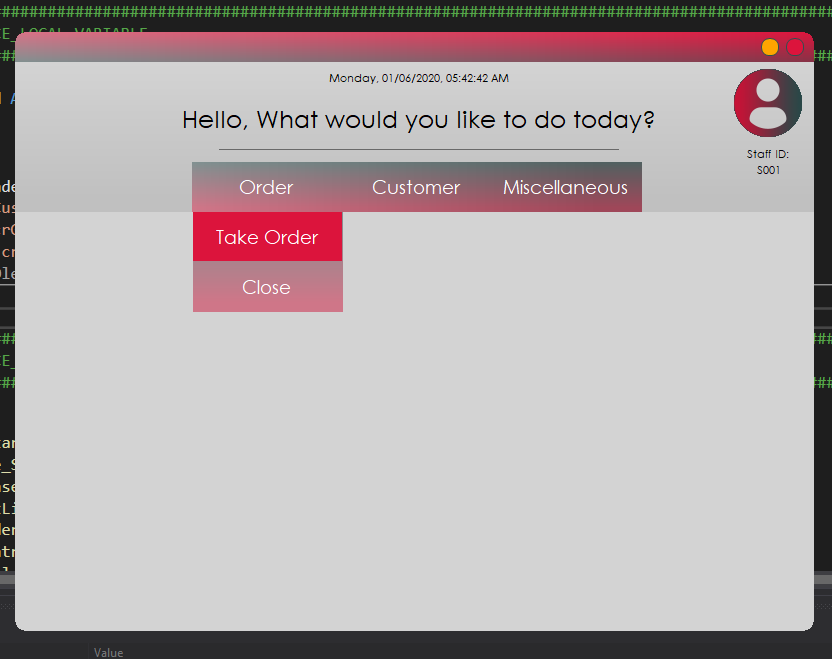


After the popup is dismissed, the loading screen to redirect the staff to the main menu appears. The loading sequence may take a while as it needs to load plenty of modules in the background. However, there is an error that can occur during this time. It is random and the only easy fix is to run the application again until the error does not happen.

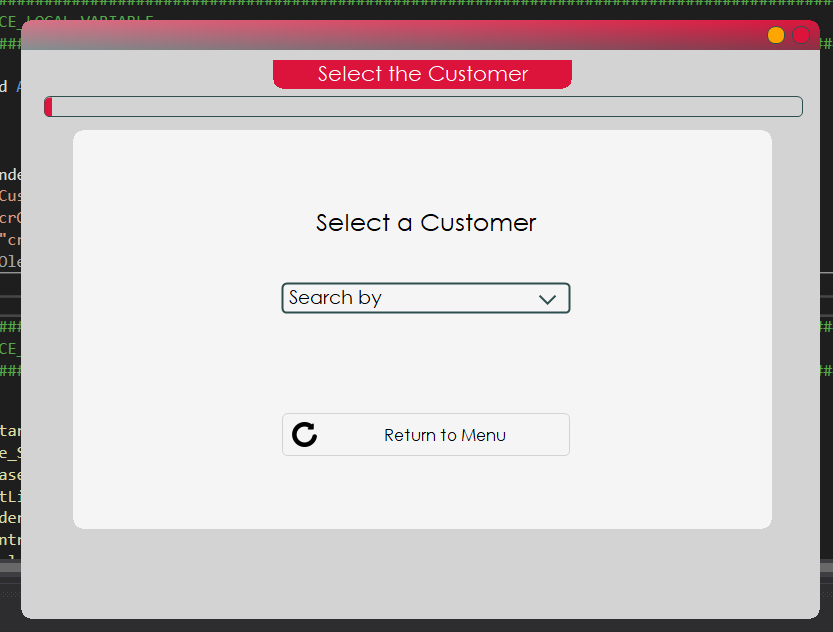


If the error mentioned previously does not occur, a final loading screen will play, and this menu will transition in. This is the staff main menu. The staff can access all the features this application has to offer here. Next, it will be about taking an order and completing it all the way until receipt section.

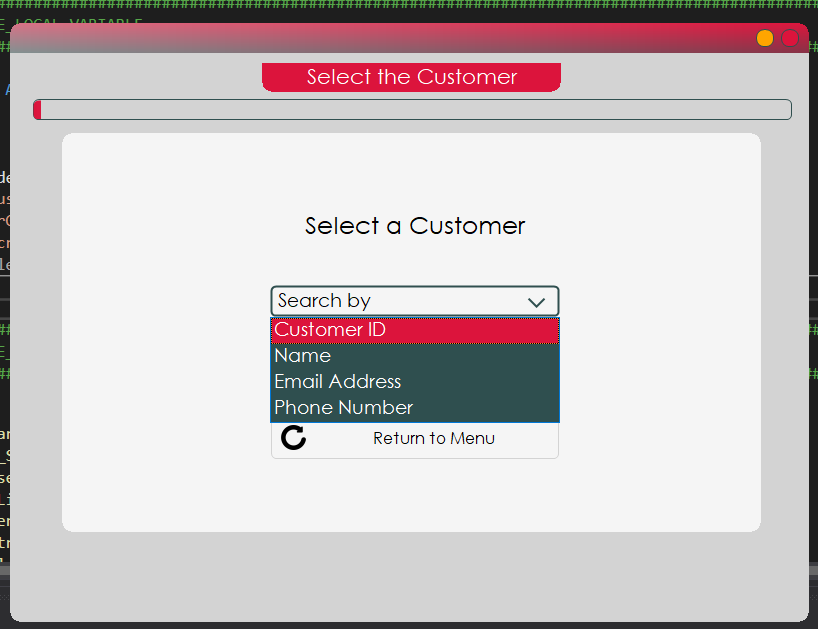
## Usage – Taking an order



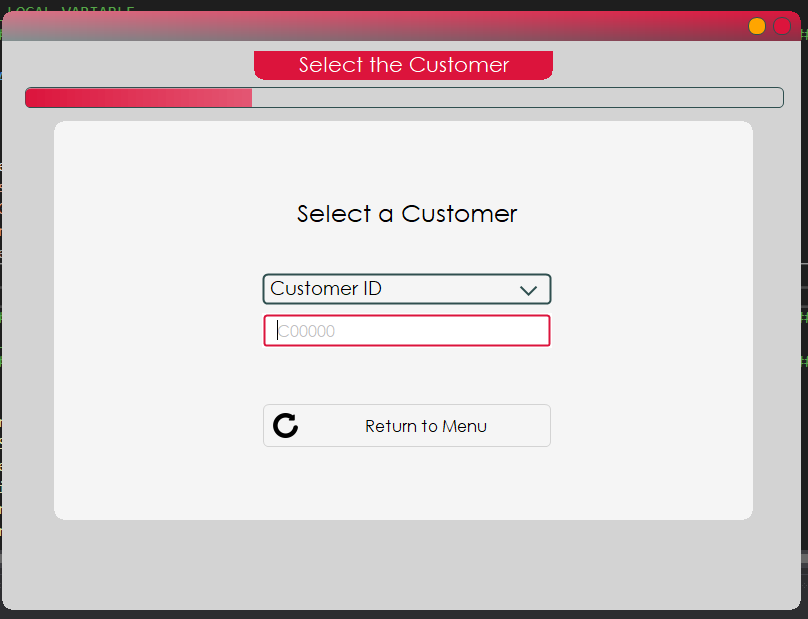
The staff can take an order for the customer through the staff menu. In order to access the menu selections, the staff must select a customer. To do that, press on ‘Order’ and click ‘Take Order’ button. This action will lead the staff to customer selection menu where the staff can select the customer.



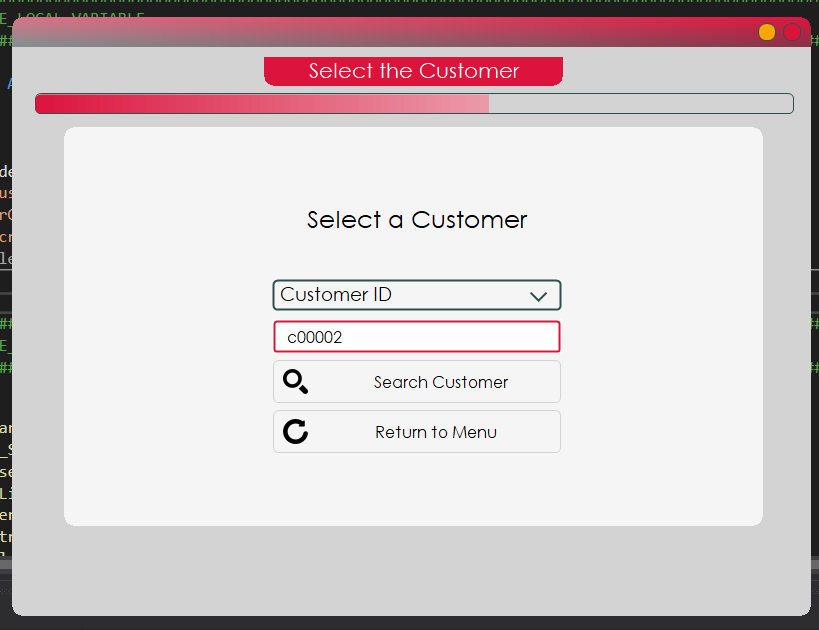
This is the customer selection interface. Here, the staff must use the dropdown box to choose a search criterion. Search criterion handles the way the system looks for a customer.



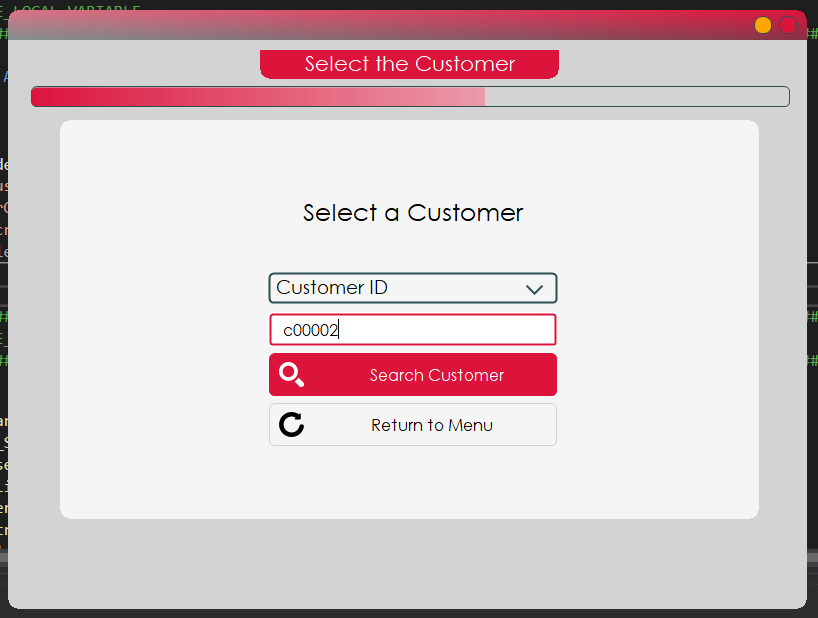
The dropdown box contains four options to choose from. The first is Customer ID which is the ID that is distributed to the customer right after registration and is also unique to every registered customer. Next is Customer Name and the way this works is by filling in the FULL NAME of the customer. If the staff inserted either one of the first or last name, the search function will not return any value and pop an error. After that, Customer Email Address, as the name suggests, searches the database by looking at email address. Although email addresses may be unique, sometimes it can have duplicates especially after the customer is deleted. The system cannot handle such event. Lastly, Customer Phone Number.



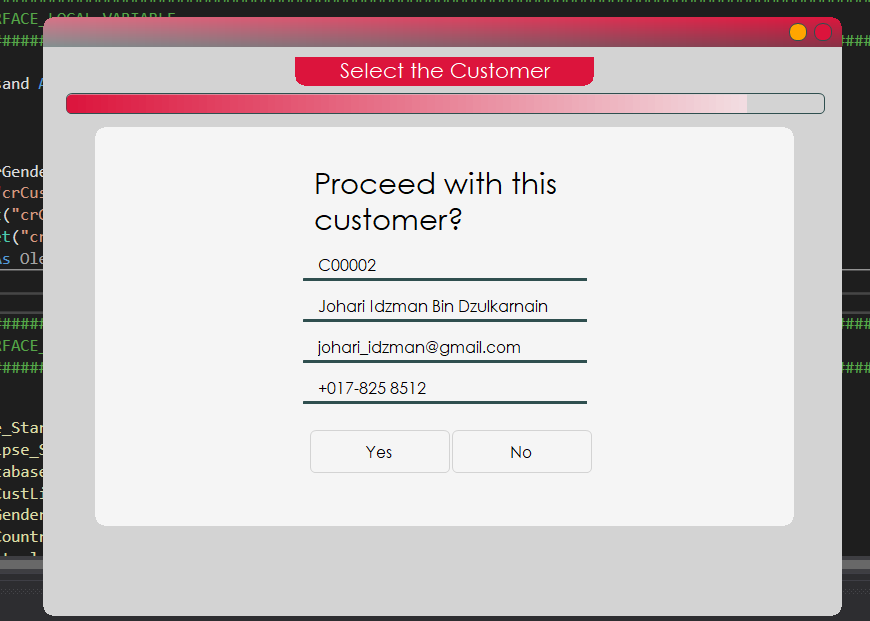
Once either of those four criteria is selected, the staff must insert a search keyword in order for the search button to appear.



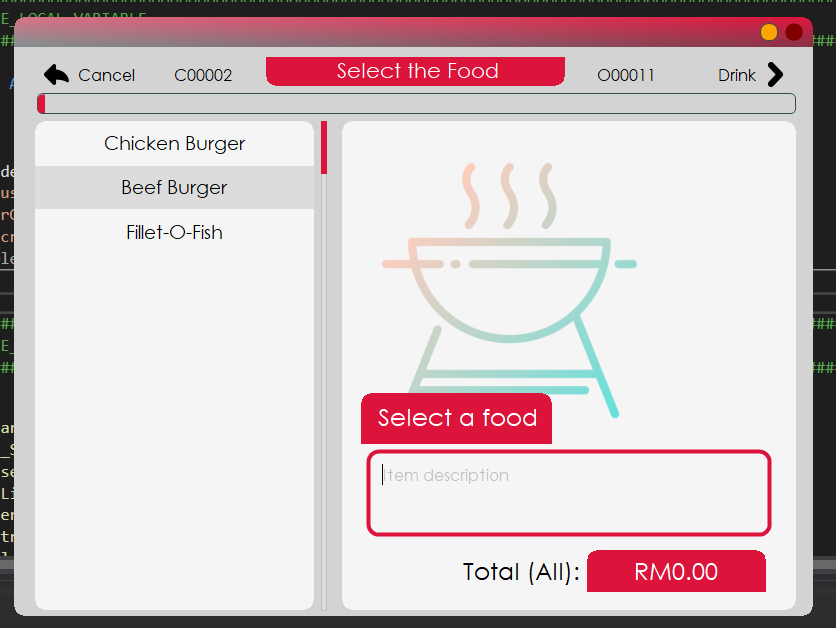
After inserting a keyword, the search button will appear, and this allows the staff to continue by pressing it. However, take note that if the record is not available inside the database either by deletion or purely absent, there will be an error and the staff needs to insert a valid search keyword for the search function to return valid result.



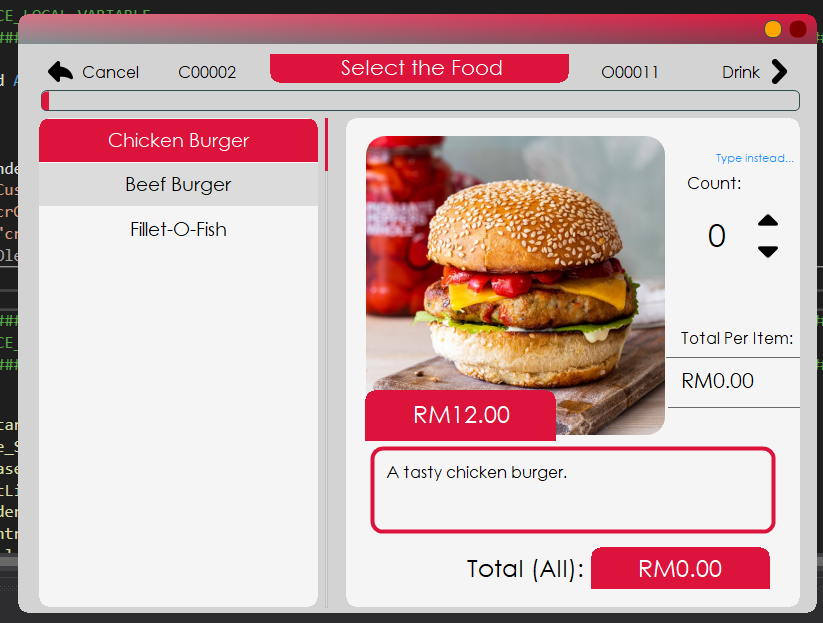
Now, the search button is highlighted to indicate a click.



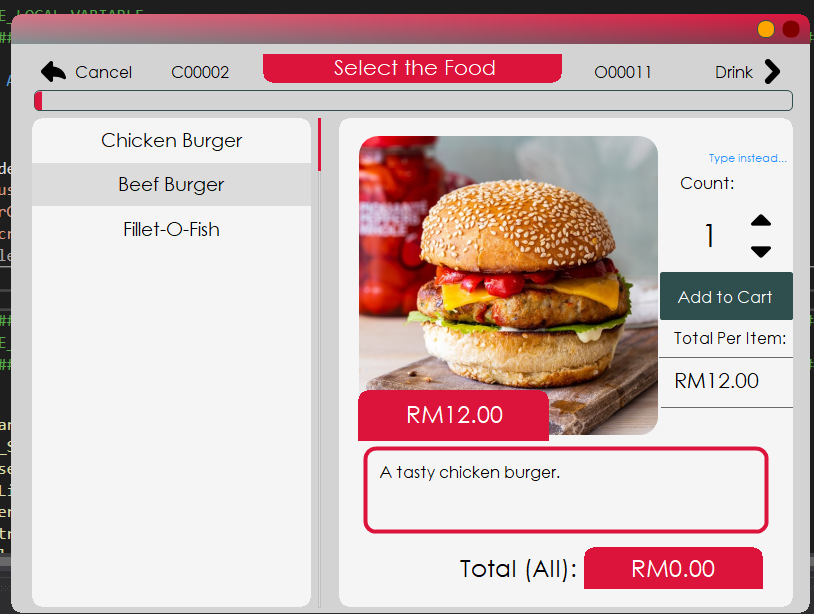
Here, an except of the customer’s record is fetched to allow staff to determine if this is the correct customer. If the staff got the wrong Customer ID, for example, these fields will help the staff in making sure they got the right record. Now, if the customer is correct, the staff may proceed by clicking ‘Yes’. Otherwise, click on ‘No’ to return to the search screen and redo steps prior to this.



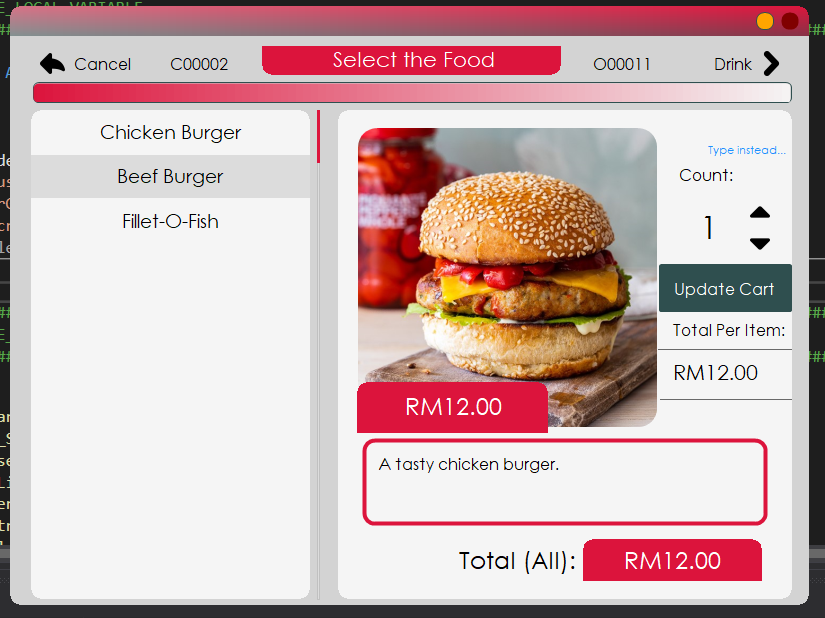
This is the food selection interface. It is the first menu selection that the staff needs to go through. Select on any of the food items on the list to display its information to the right side of the window.



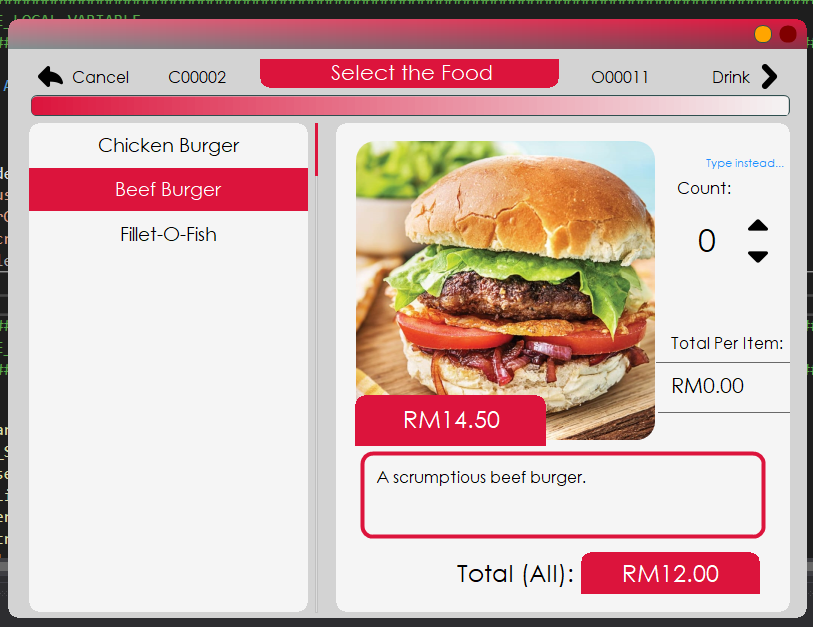
In the picture, chicken burger is selected as a demonstration. As it is shown, there is a panel to the right that transitions in right after the button is pressed. This panel stores the counter and the price per item label.



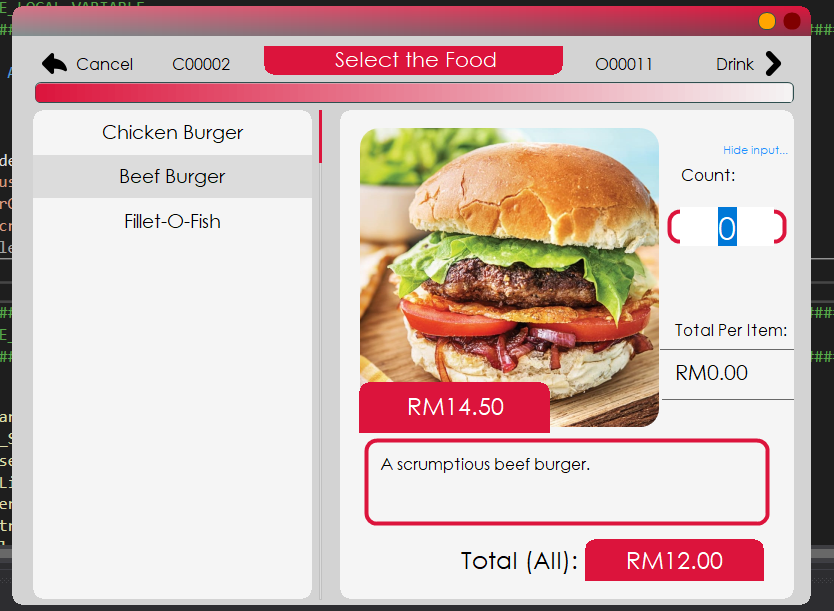
The counter can be incremented or decremented by pressing on either of the pointy arrows. For this occasion, the counter increments up in order to display the button which adds the number of items to cart.



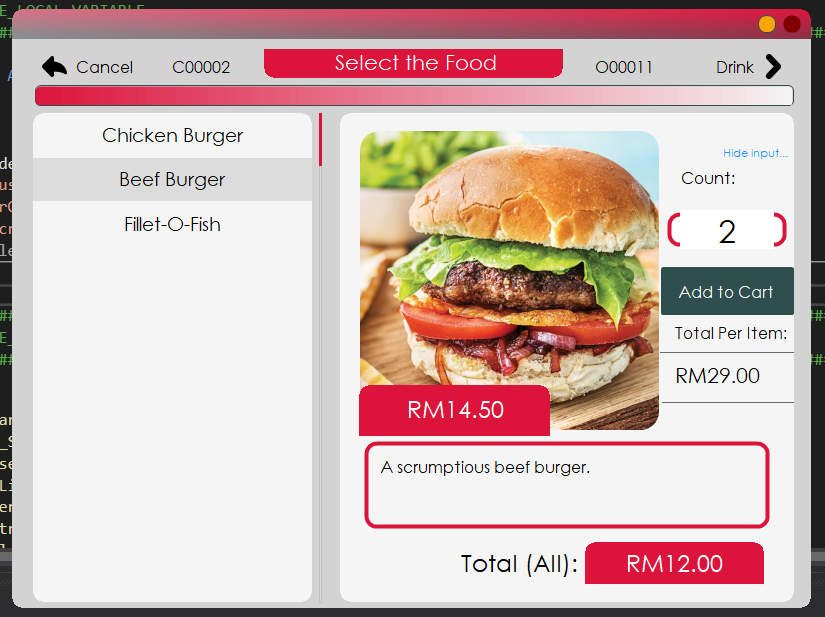
Once the button is pressed, it will switch to ‘Update Cart’ and this indicates that the item has successfully been placed into order. Also, notice that ‘Total (All)’ refreshed to display a value. This value is the total of all available items inside the cart together. It will refresh whenever the staff handles the counter.



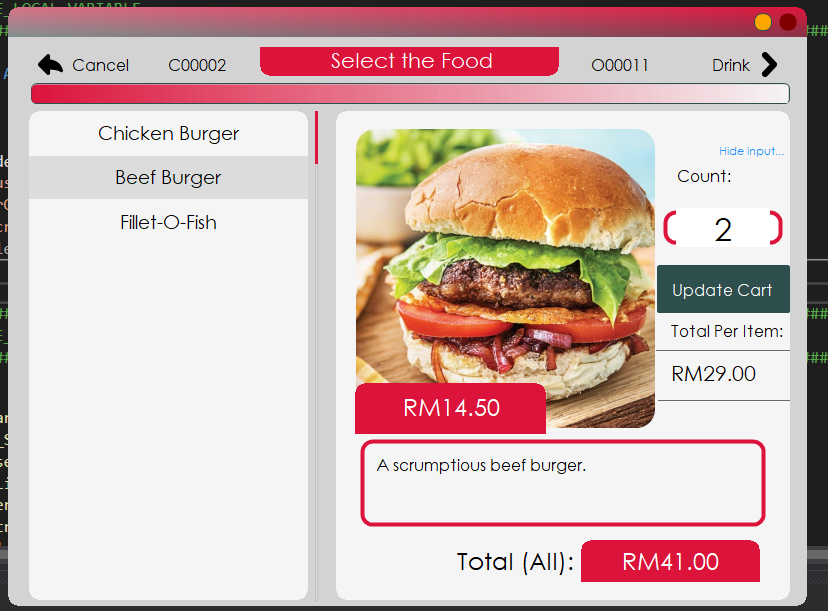
Now, notice that the ‘Total Per Item” switched to ‘RM0.00’ after another item is pressed from the list. This is because the total only handles value per item. If the staff were to switch back to chicken burger, the total will resume its last value, which is RM12.00.



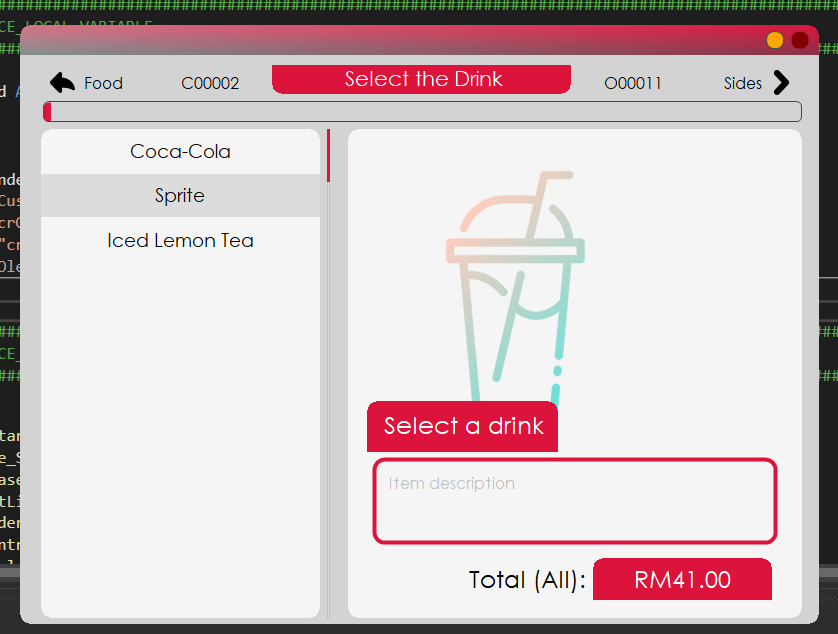
A manual input can be accessed by pressing on ‘Type instead…’ label above the counter. It will show this input box where the staff can manually insert the value they want. Be aware that if any characters besides numbers are inserted, it will pop an error and resets itself back to 0. This will also affect the total count.



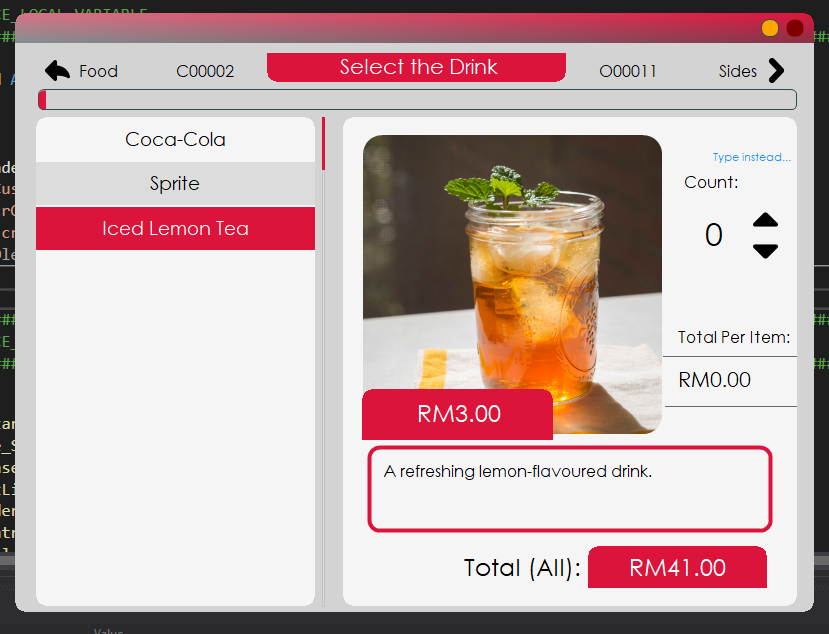
Here, the staff inserted two (2) manually and see that the total adds to return the price of RM29.00. This input refreshes the price every time it changes value so do bear that in mind.



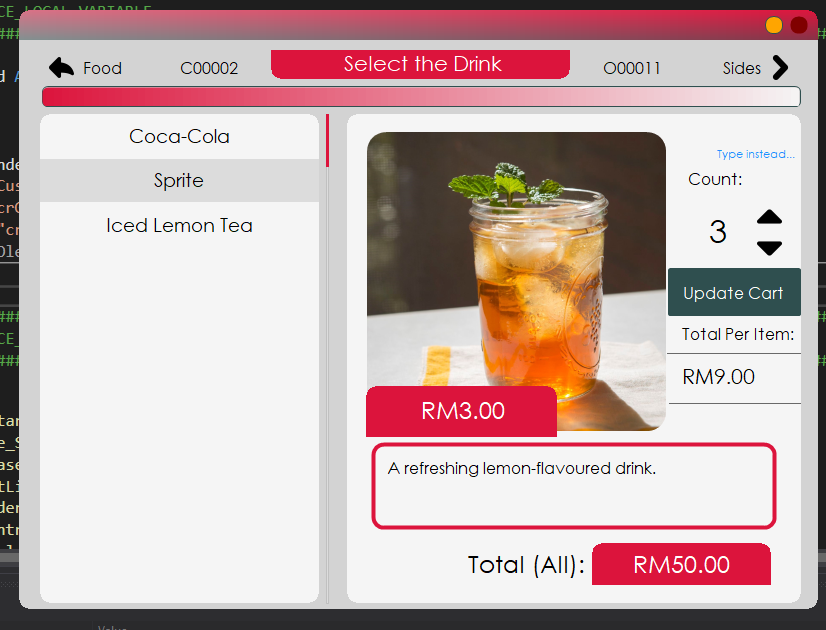
The add to cart button is pressed and the total for all items are updated accordingly. Previously it was at RM12.00 and now it adds up to RM41.00 to accommodate the extra 2 beef burgers.



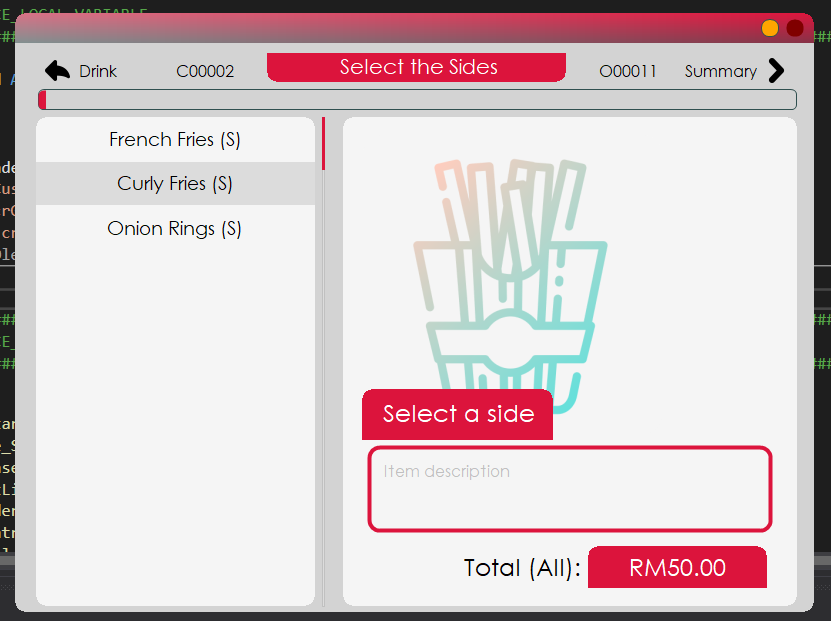
Moving on, this is the drink selection interface, the second menu selection interface the staff must go through. Overall, the design is like food selection interface and the procedure to use it. Select an item from the list to the left to display its information to the right.



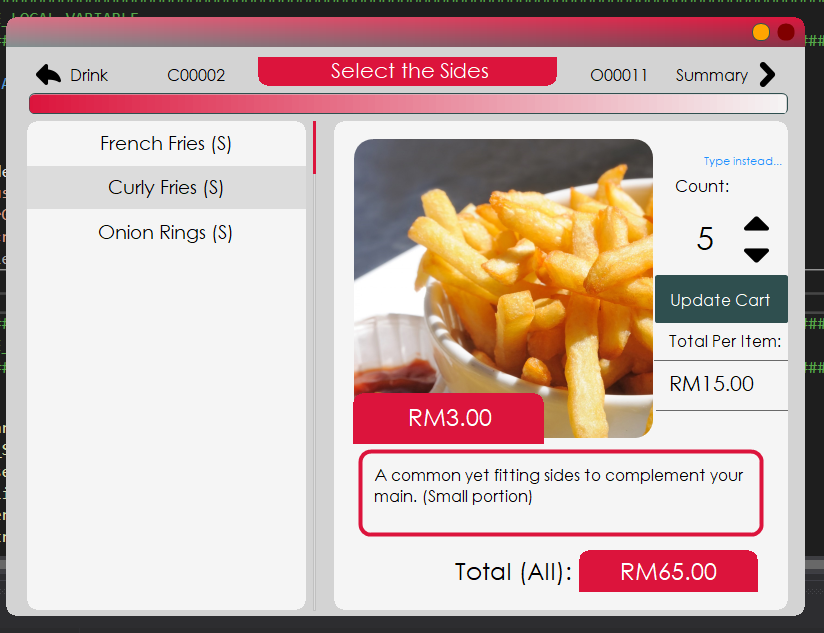
The staff selected an iced lemon tea. See that the total of all items did not change at all and is being carried over to this menu.



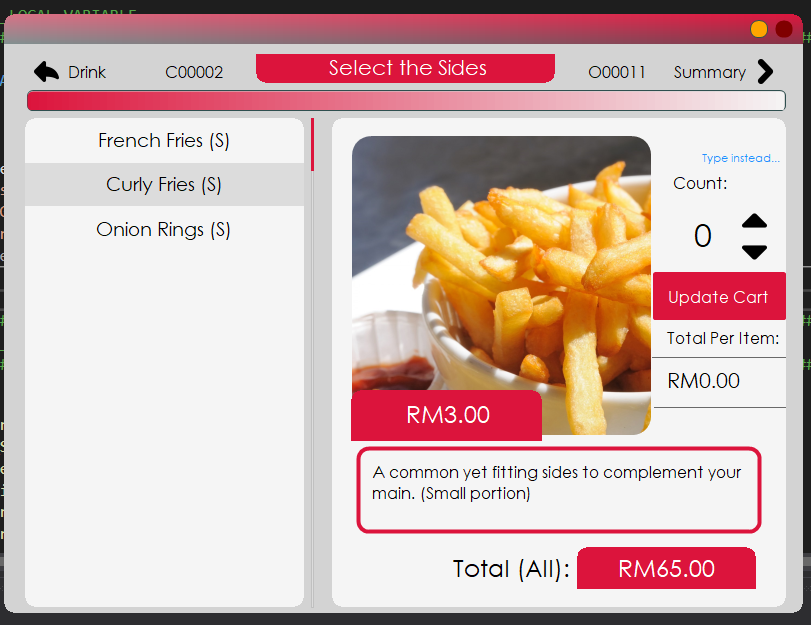
After selecting 3 iced lemon teas, the staff updates the cart and the total price is now reflected at RM50.00.



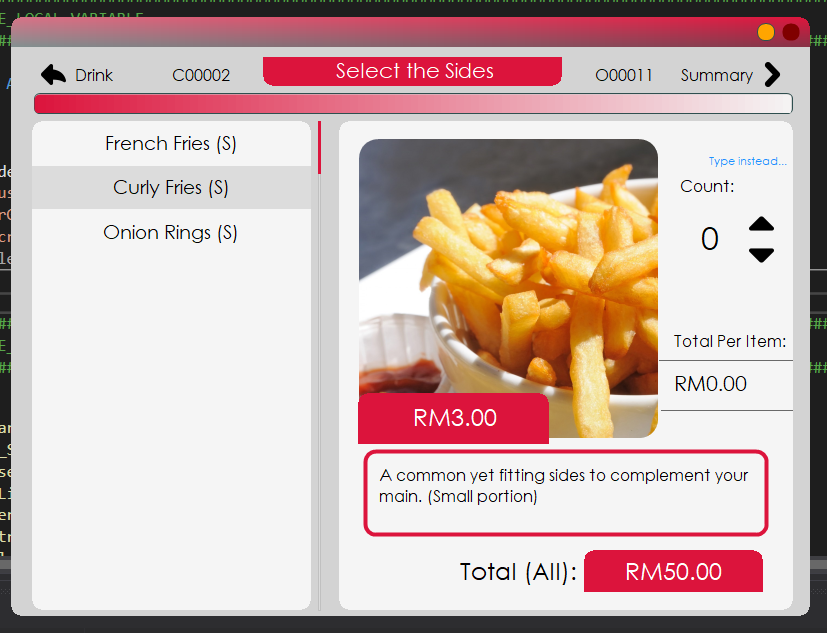
This is the final menu selection interface that Khayalan Cafeteria system has to offer. It handles the side dishes a customer might want to add to their main. The same procedure from before applies here.



From RM50.00 previously and due to the fries, the price rocketed to RM65.00. Now that the staff accidentally added 5 small French fries to the customer’s order, the staff can remove some or all of it by decrementing the count and pressing on update cart.



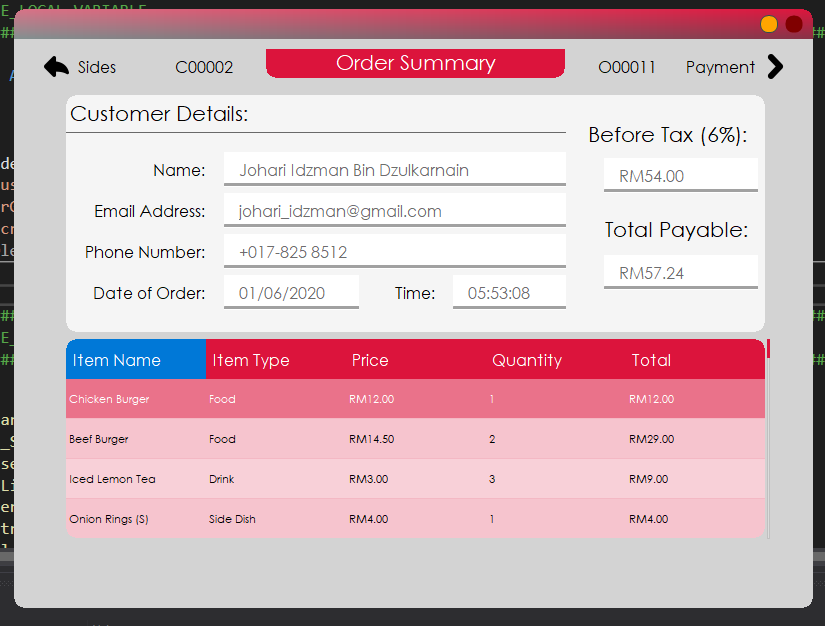
Once the counter is decremented all the way to 0, the staff can remove the item totally from the list by pressing on update.



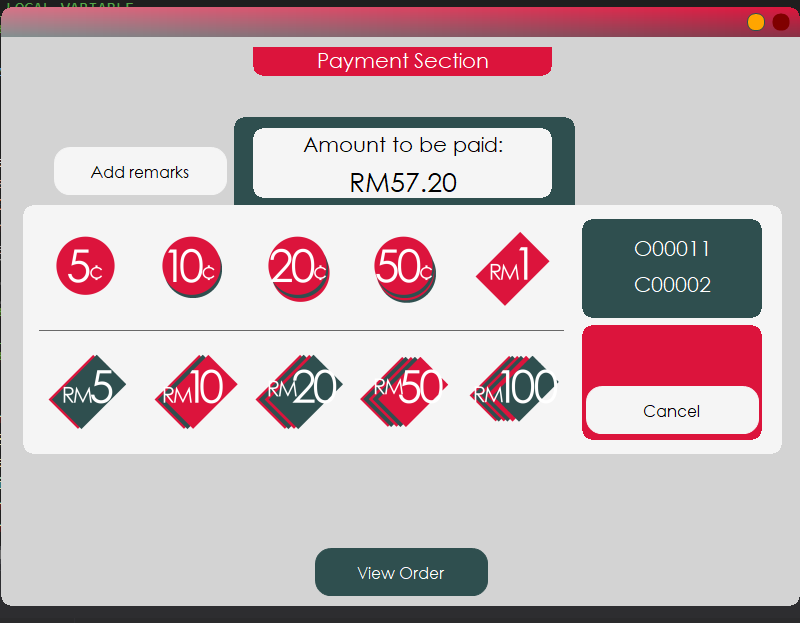
The item has been removed from the order because the button that handles the update has disappeared and the price of all items reset itself back to RM50.00.



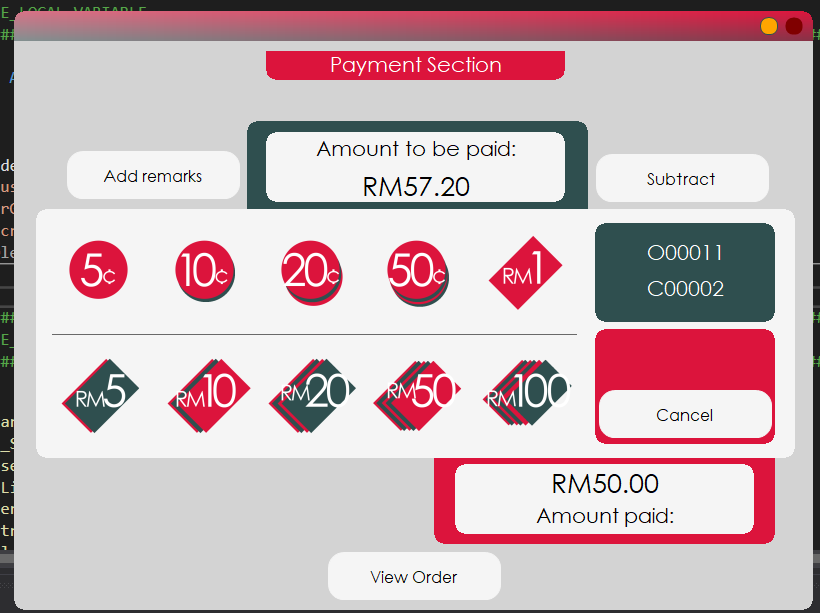
Now the staff selects a different item from the list. Now, it is time to display the summary of the orders. Click on Summary button to continue.



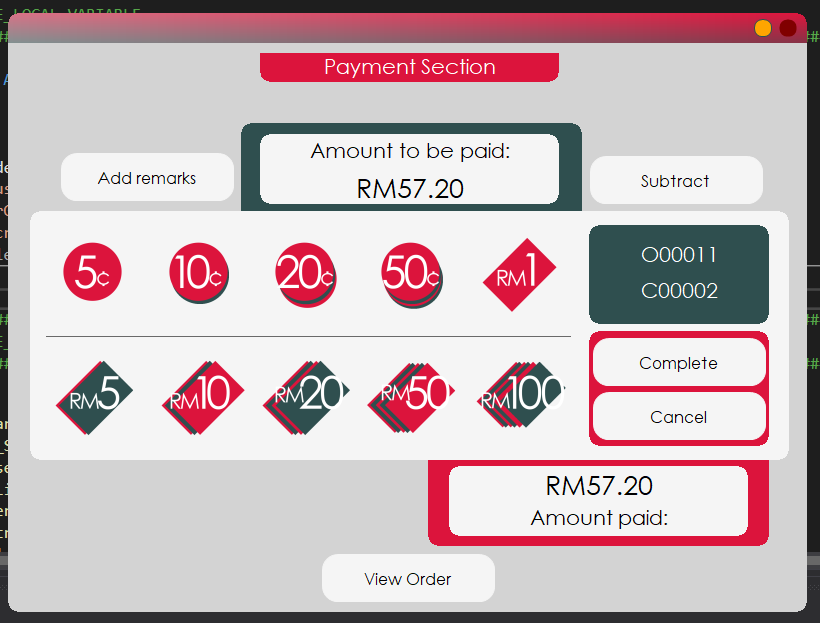
This is the order summary interface. It displays the customer details along with the total from the cart. Since most business require taxation, the final calculation is also inclusive of the said rate. At the bottom of the window, there is a list which displays all the selected item from the menu selection interfaces. This interface is purely for reference purpose. It is not interactable nor it is modifiable. The staff can return to the menus by selecting Sides at the top left or proceed to payment by pressing on Payment button.



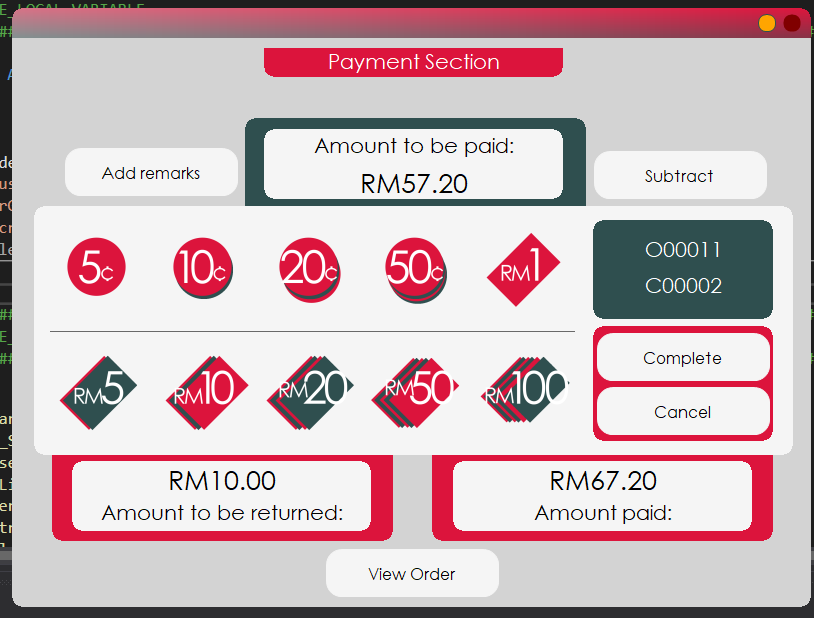
This is the payment interface. This is the interface that sets whether an order can be prepared or else. Also be alert with the total as it has been rounded off to the nearest decimal places. This is to ensure that the customer does not have to pay in odd notes. Notice that the button that completes the order is missing. That is because the button is set to hidden until the amount paid by the customer is exactly or more than the amount required to be paid. The buttons which has numbers on them from 5 sen all the way to RM100 handles the paid amount.



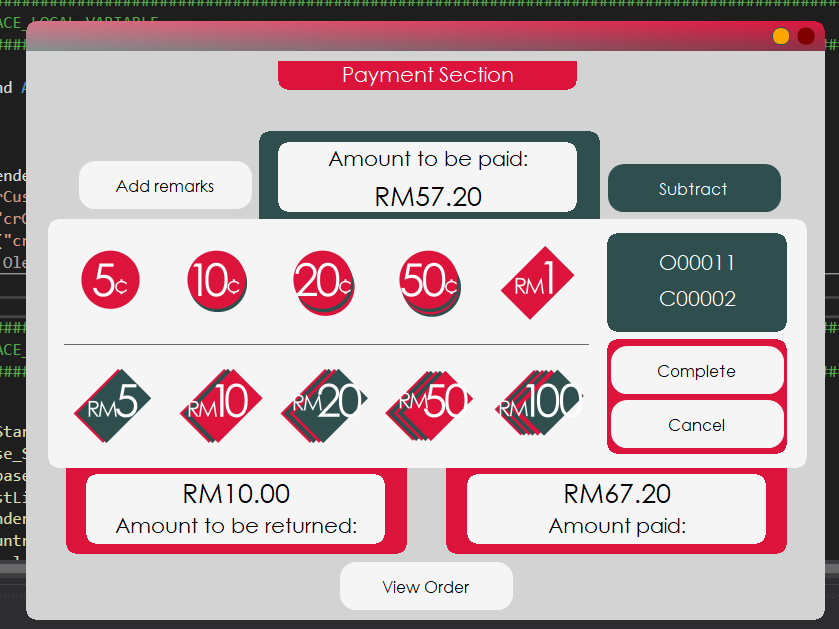
Since the required amount to be paid is RM57.20, the customer paid with RM50.00 note. Therefore, the staff can pick the RM50 button to add to the amount paid. Still, the button to confirm the order has not appeared due to the obvious condition – the amount paid is not tally with the required amount to be paid.



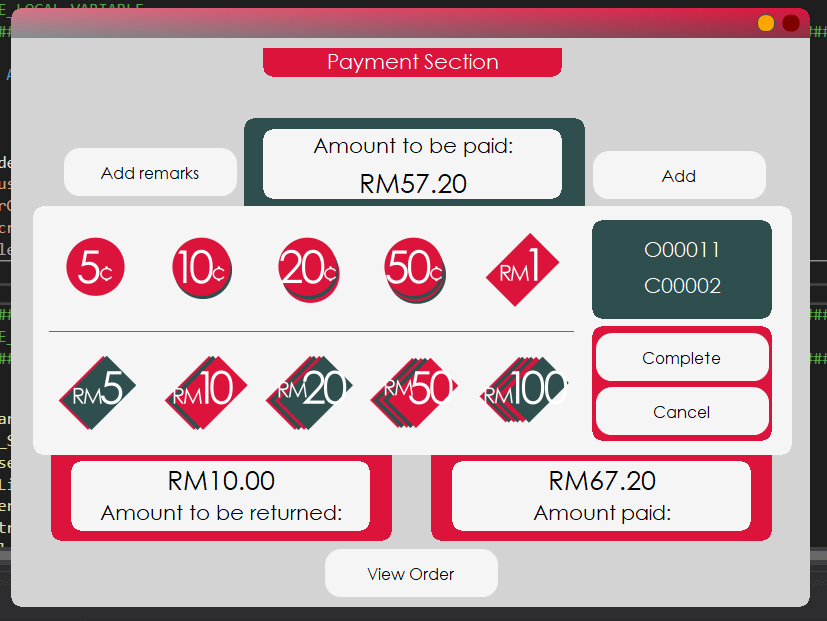
Now since the amount paid is exactly the amount to be paid, the button to confirm the order set itself to visible. The staff can complete the order simply by pressing on the button.



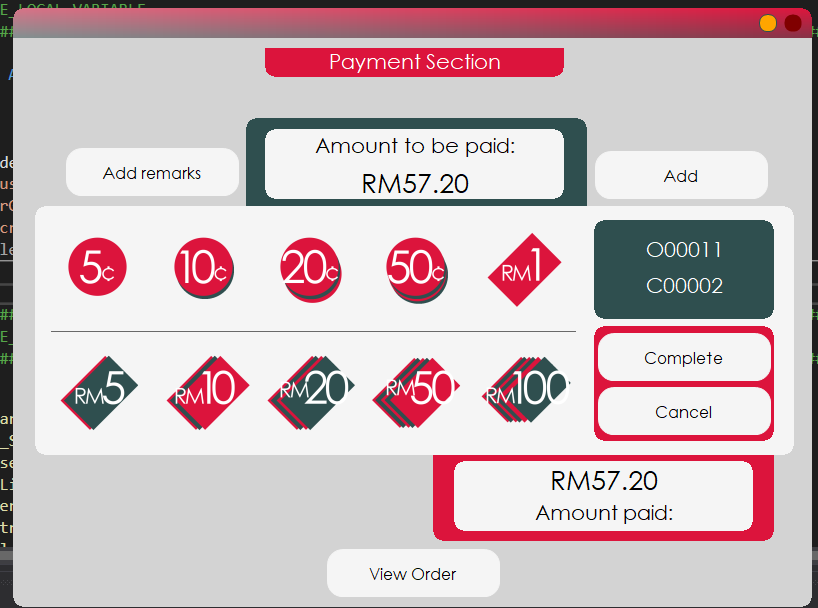
For this part, there is another section that displays itself. This is because the amount paid by the customer exceeded the total amount to be paid and therefore there is a change that needs to be returned to the customer. Hence, the section only displays itself if there is a change even as low as 1 sen if possible.



What if the staff accidentally pressed on any of the buttons while the customer did not even pay that much? There is a way to subtract the accidental add-ins. The button named Subtract handles the decrement of amount paid. It works by pressing on it to enter into subtraction mode and any of the money buttons will minus off from the paid amount instead of adding to it.



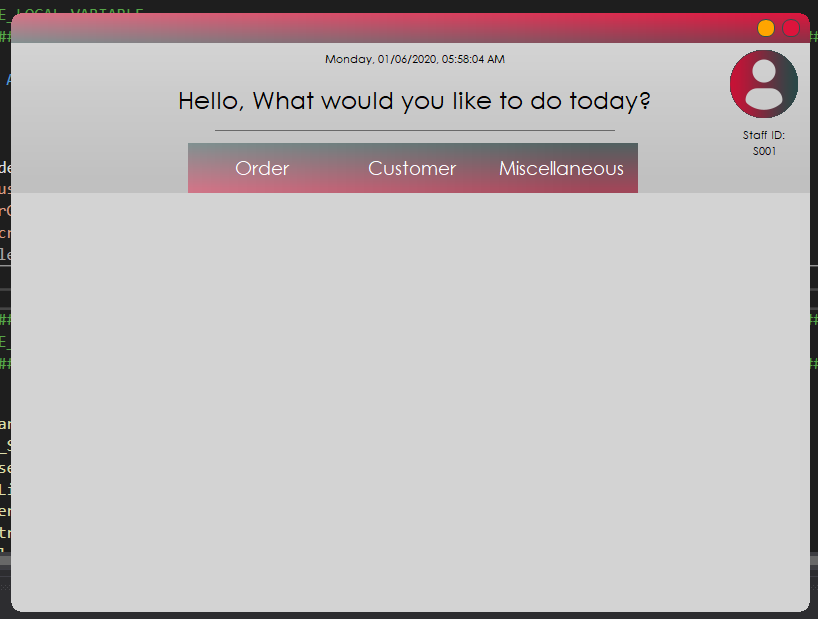
After the button is pressed, it will rename itself to Add. Therefore, this button works both ways, either to add or to subtract. Pressing the button will enter addition mode again.



Since the staff made a mistake earlier, the button RM10 subtracted that much amount from the amount paid and now it sets at RM57.20, exactly the amount to be paid, again. Notice too that the change section sets itself back to hidden as there is no change.

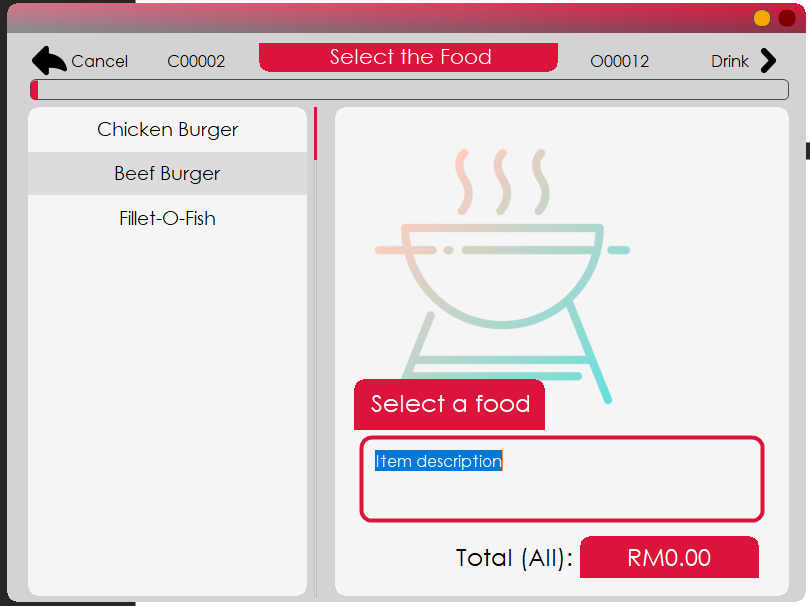


This is the last interface the staff will see when taking an order. It is the receipt of purchase after the payment has been completed. It shows the details of the purchase including the date of order, financial track and most importantly the ordered items. Once the staff is ready to end the sale, the button Complete on the top right corner will redirect the staff to the staff menu.



After pressing on the button, the staff will be sent back to the main menu. Now, the staff can do other things or take a new order. Next guide will be on how to cancel an order.

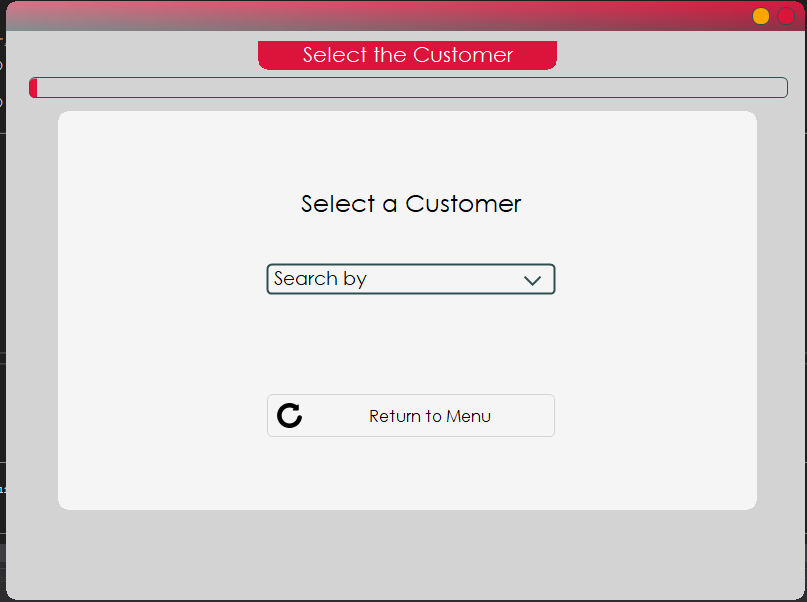
## Usage – Cancelling an order



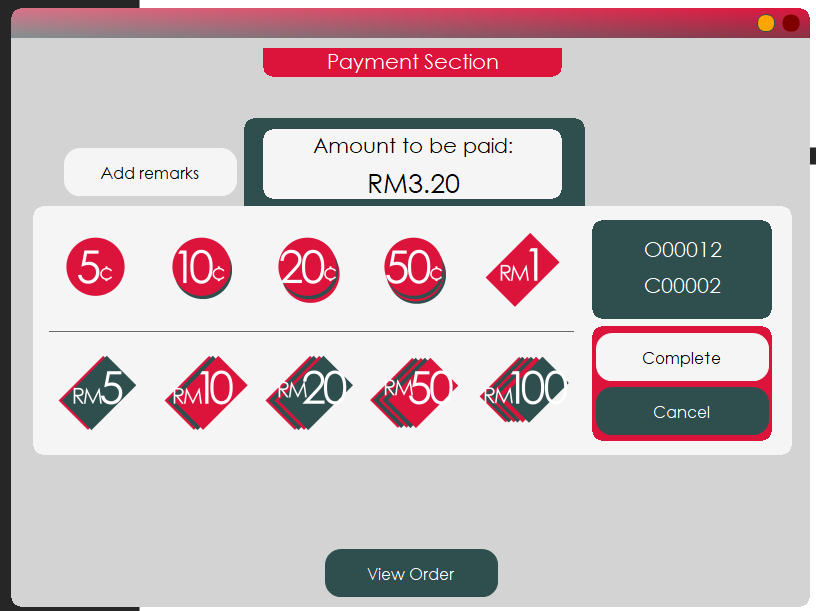
For this part, the order is going to be cancelled before the payment section is accessed, meaning that the staff cancelled the order before the customer pays. It can only be done by going back to food selection menu and press on the Cancel button.



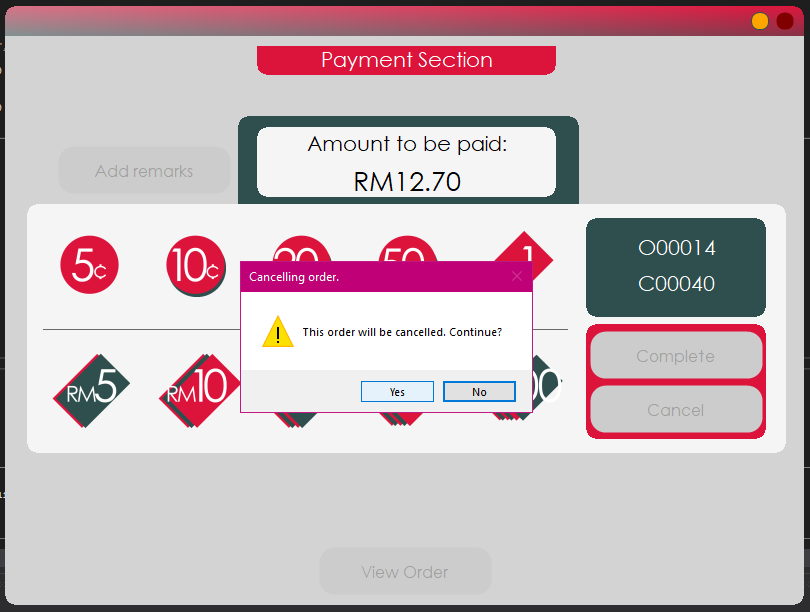
By doing so will pop a confirmation to the action. When the staff selects ‘Yes’, the order will be cancelled.



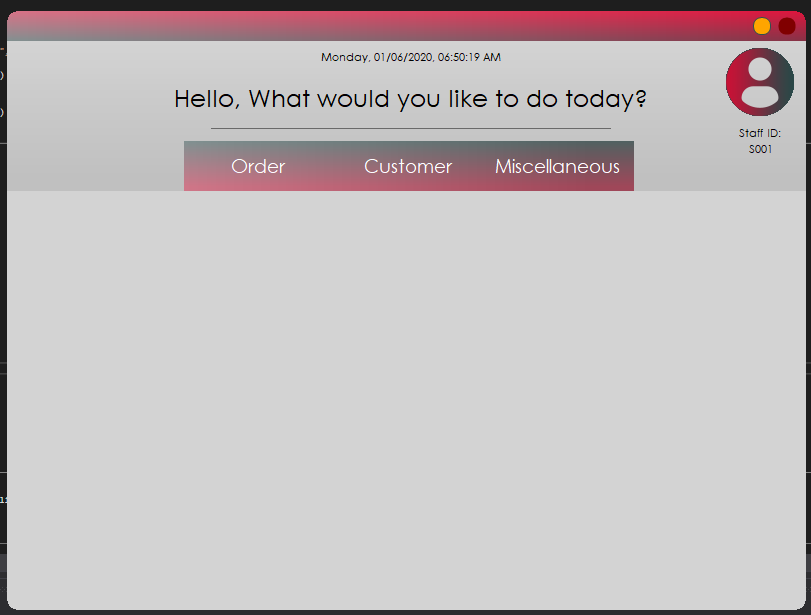
After confirming with a ‘Yes’, the staff will be redirected back to the customer selection interface.



Meanwhile, if the customer cancelled while paying, the staff will only have to press on the cancel button.

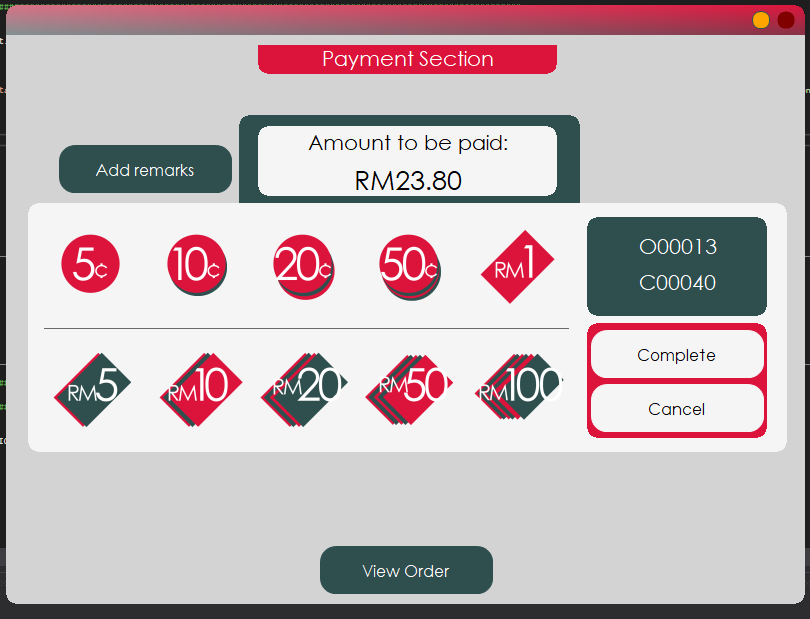


Once the button is pressed, it will pop a warning to ask the staff for confirmation. Press on ‘Yes’ to continue which will stop the ordering session and drop the order from the database.

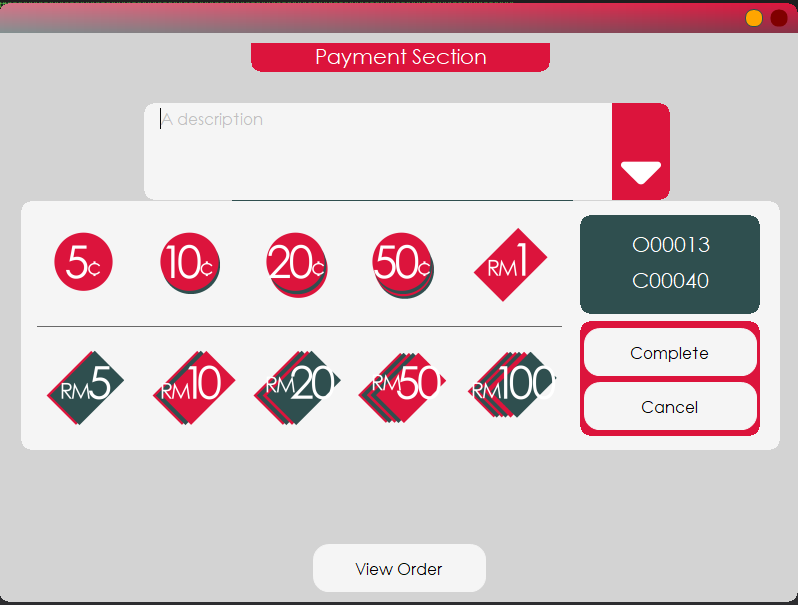


Once the button is pressed, the staff will be redirected back to the staff menu. Next will be on how to add a remark to the order.

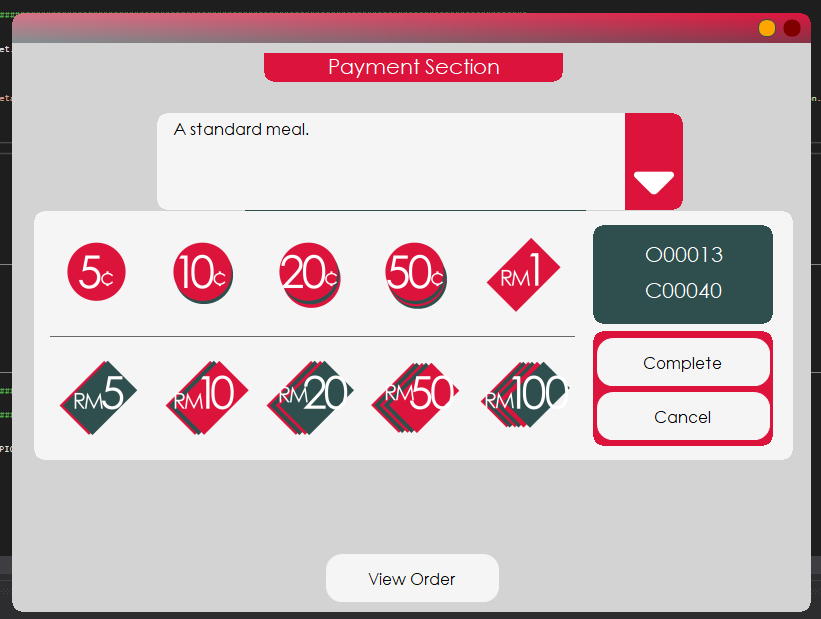
## Usage – Adding a remark to an order



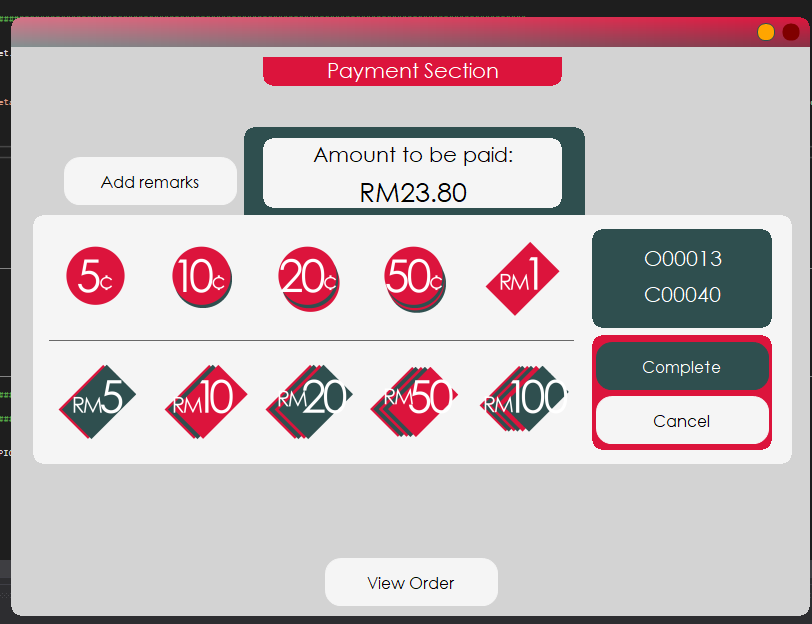
On the payment interface, there is a button on top left where it says Add remarks. This button allows the staff to add a description to the order so that when accessed inside the database, the staff will know exactly what the order is all about if written correctly.



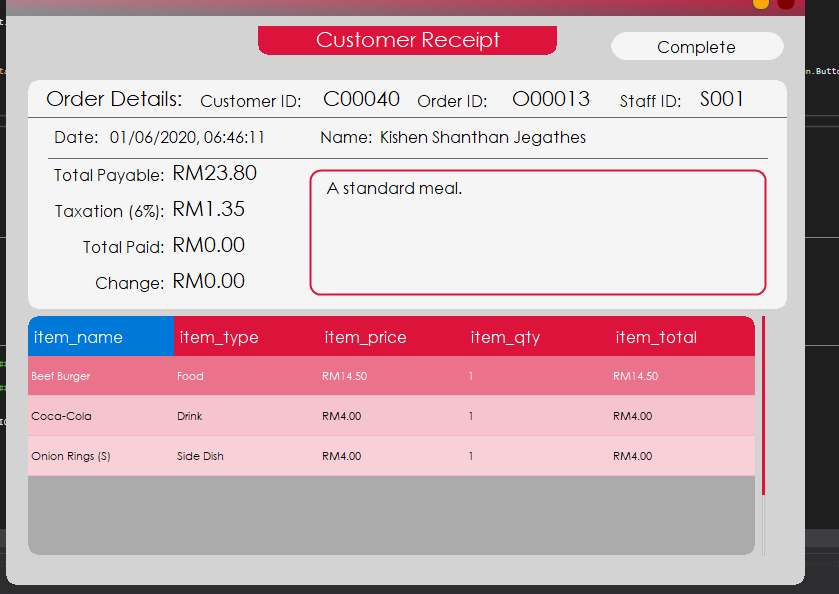
Once the button is pressed, a large, empty textbox appears, obscuring the amount to be paid display.



Insert a short description about the order to make it understandable in the future when accessing the records. Bear in mind that the maximum number of characters allowed is capped at 100 characters. Close the textbox by pressing on the white arrow.

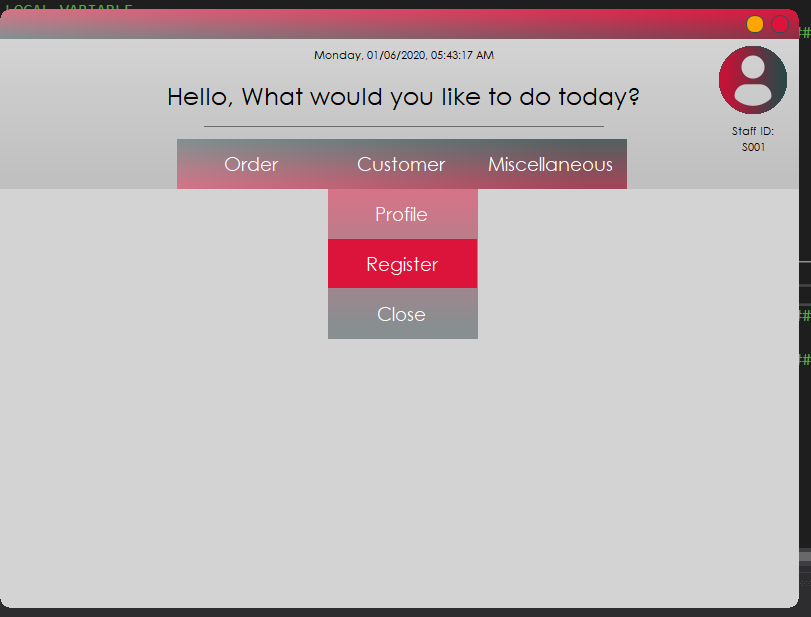


Complete the order like usual.

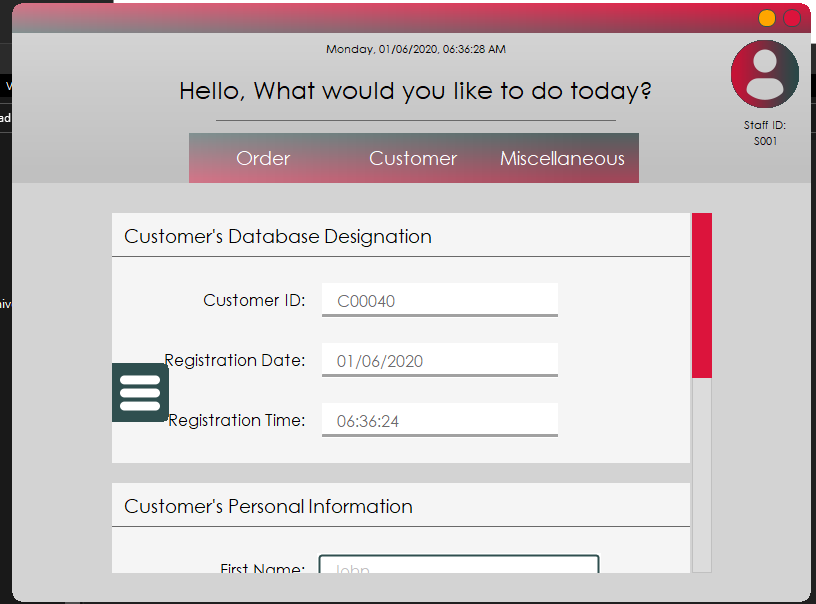


Now the description added earlier appears on the textbox inside customer receipt. Next will be on how to register a customer.

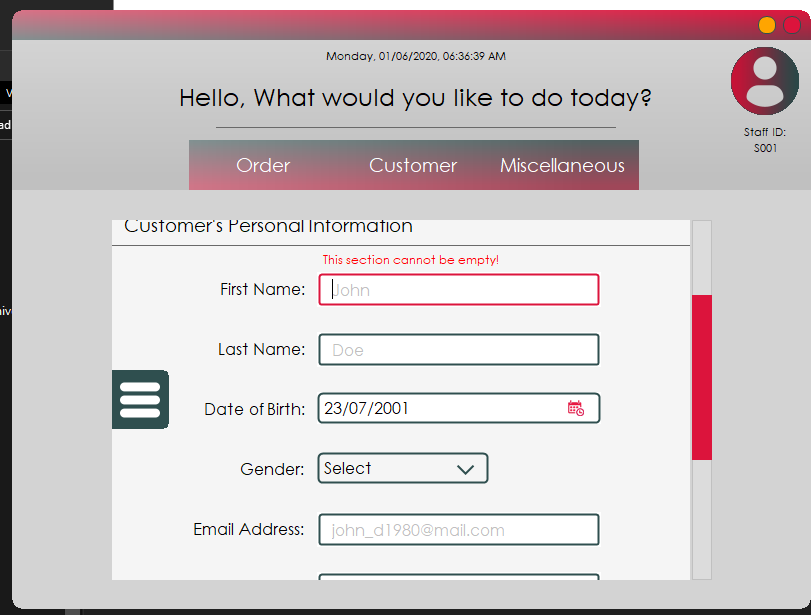
## Usage – Registering a customer



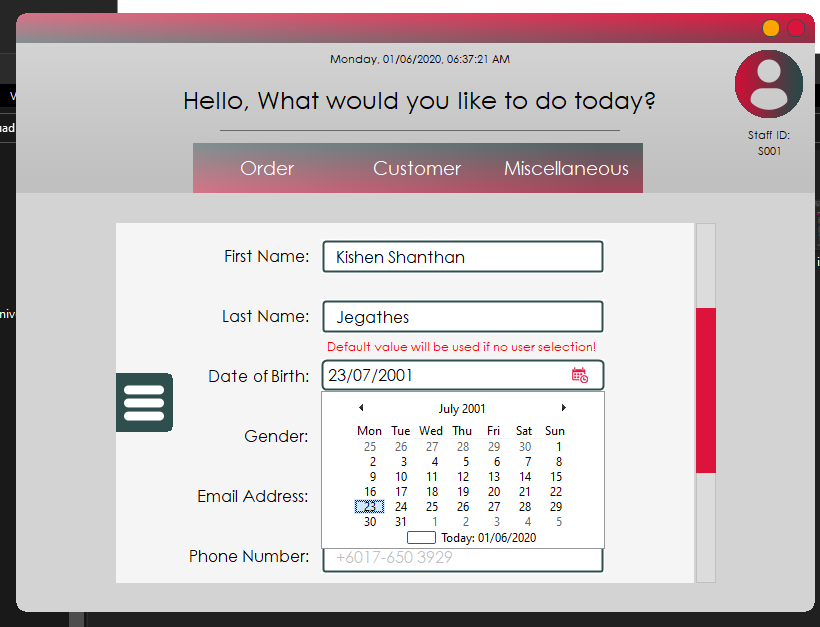
To register a customer, the staff can start by pressing on Customer and then Register button. This will cause the registration form to load just beneath the dropdown menu.



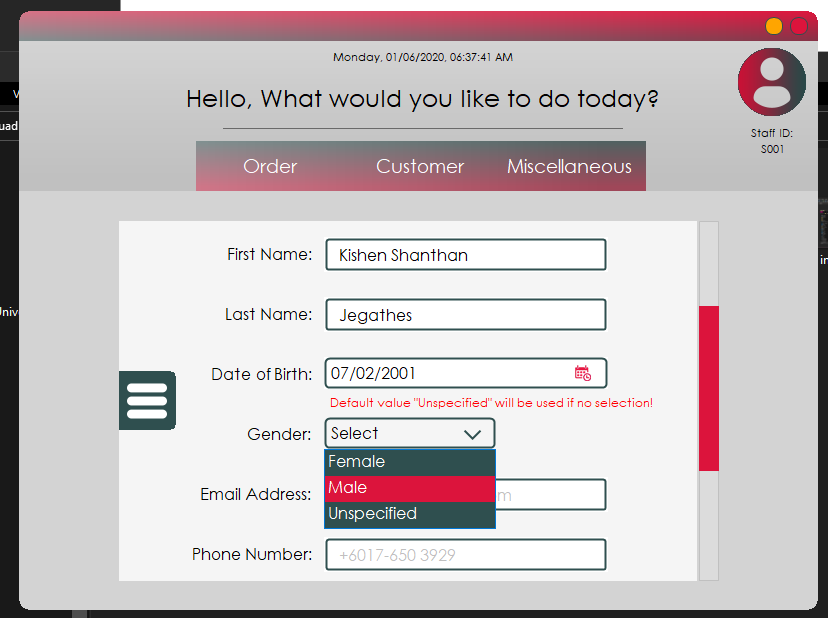
This is customer registration interface. It will handle the registration of a customer by saving the inserted details to the database which will be accessed later. Inside the Customer’s Database Designation tab, there are three textboxes that have its content autogenerated. The Customer ID is unique and therefore it will generate an unused ID for the next customer.



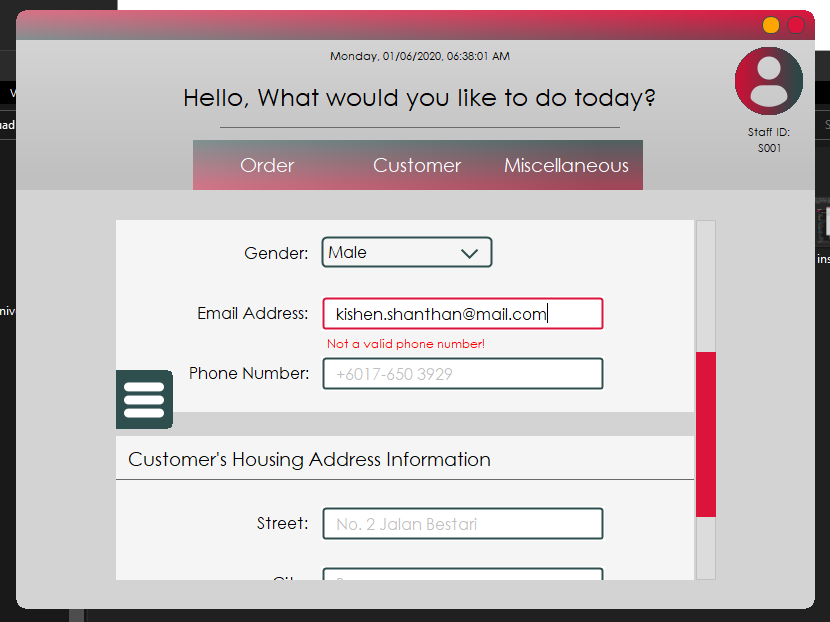
Scroll a bit down and the staff can see the mandatory fields highlighted in red whenever it is left empty. First and last name are considered as a must for human worker to identify their customers by name. An ID can only be easily understood by machine but not human. Therefore, it is a must for the names to become a mandatory field. These two fields only accept alphabets as input. Whenever characters besides alphabets are entered, it will display a warning label, but the character will not be inserted.



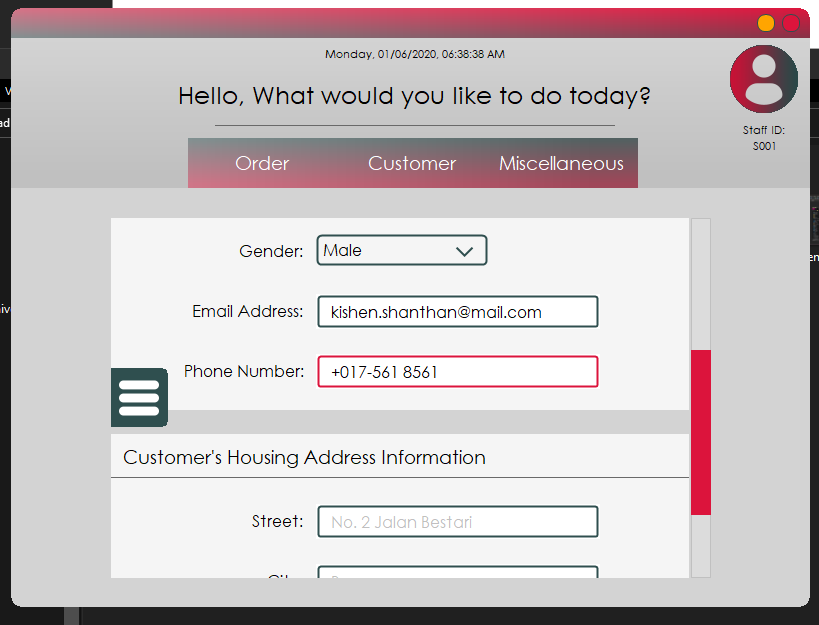
Next, the date of birth of the customer. This is not mandatory however do bear in mind that it will be defaulted to ‘23/07/2001’ when it is left untouched. Select the birth date of the customer and move on to the next part.



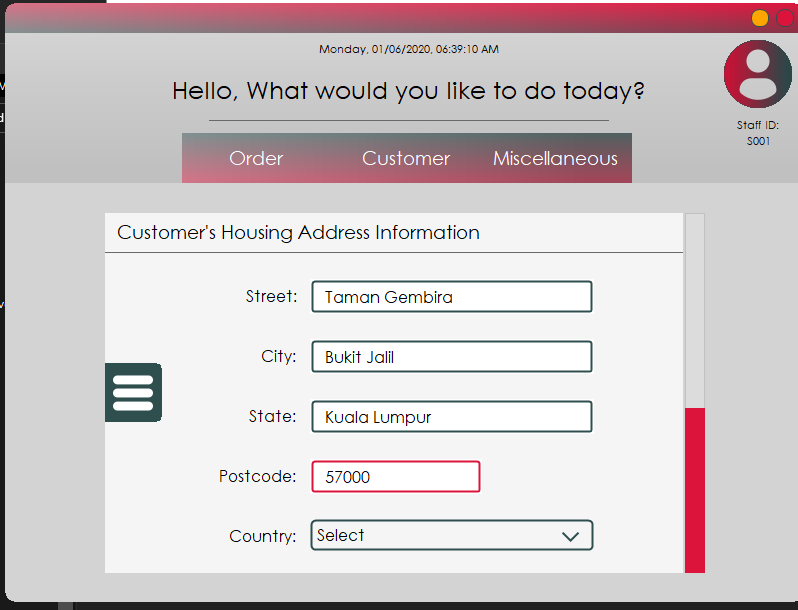
The gender dropdown box contains binary gender and an unspecified option. If the staff choose to ignore this box, it will be defaulted to ‘Unspecified’ as highlighted in the red warning label. Select a gender and move on to the next part.



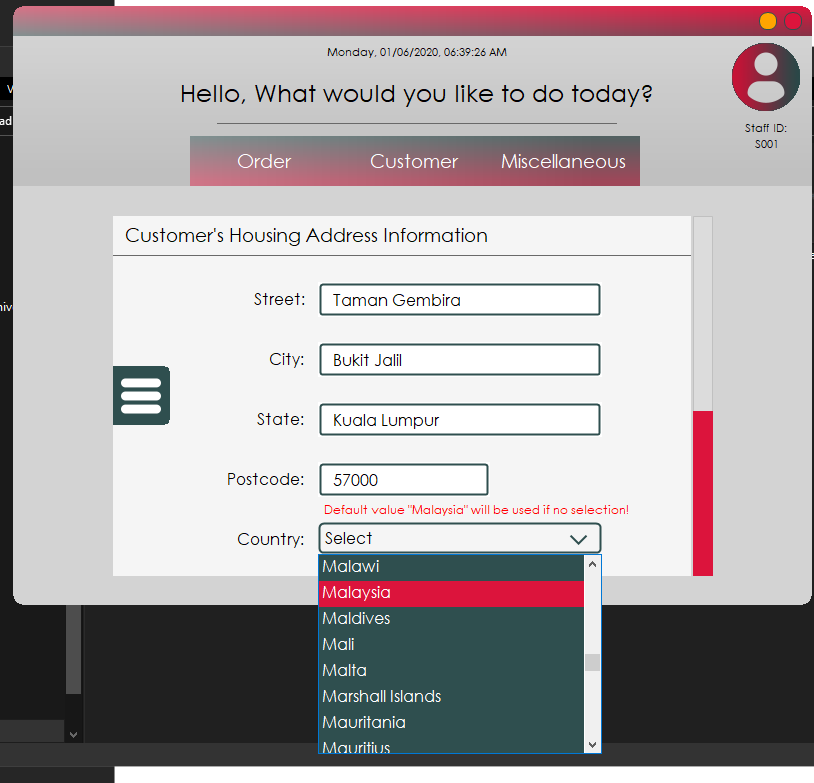
This field is strict as it only accepts email address formatted input. For the warning label to stop showing, the staff must insert a valid email address that has an alias, ‘@’, and a period, ‘.’, in that flow. Once the email is inserted, the staff can proceed to the next step.



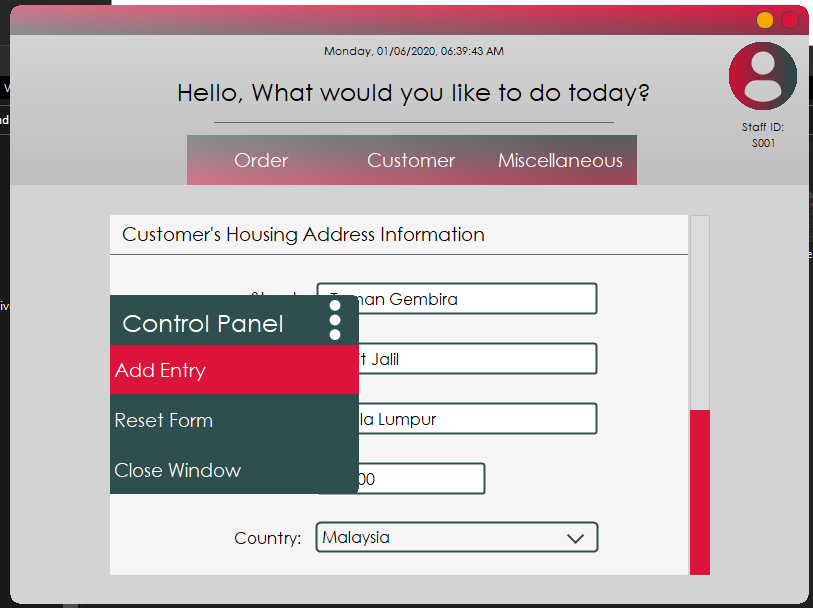
This field only accepts phone number in a specific format. The format used for this field is based on international set. Until the staff inserts a valid phone number, the persistent warning will not hide itself.



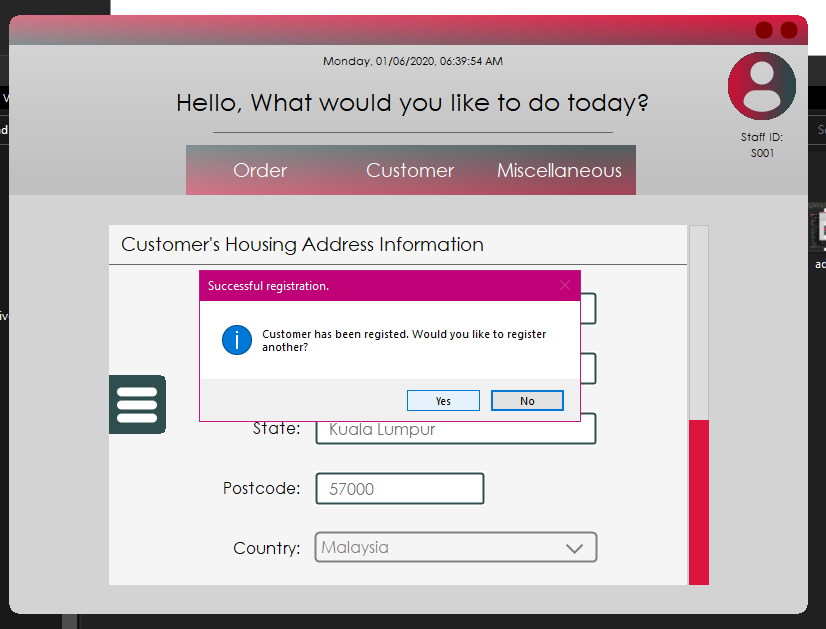
For the address part, the strictness level is not so much but it can never be left empty. Once all four fields are filled, the staff can move to the final step before adding the customer to database.



This is the country dropdown box. In this list contains all available countries around the world. If the staff leave this box empty, it will default itself to ‘Malaysia’. Select a country and the staff can move on to add the customer record to database.

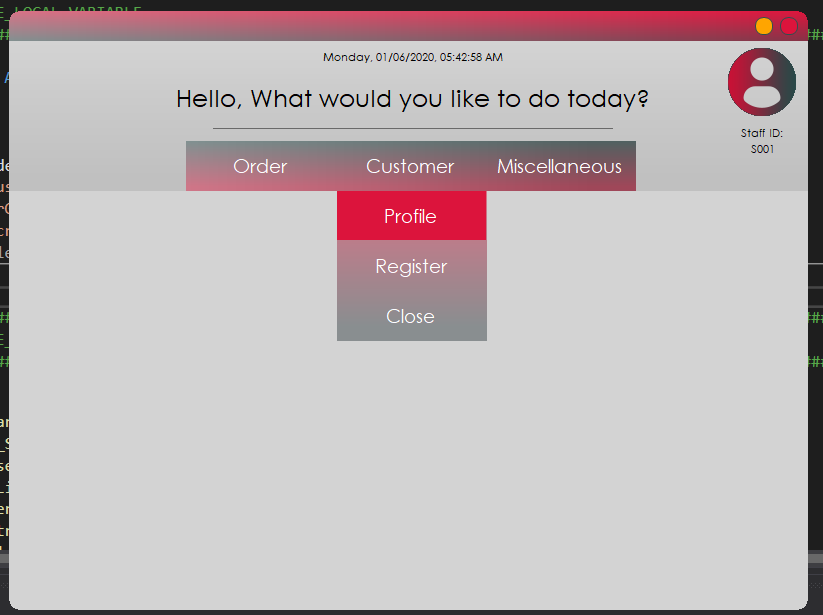


Extend the left side pane by pressing on the button. This panel will appear and to add the customer, select the first button which is the Add Entry.

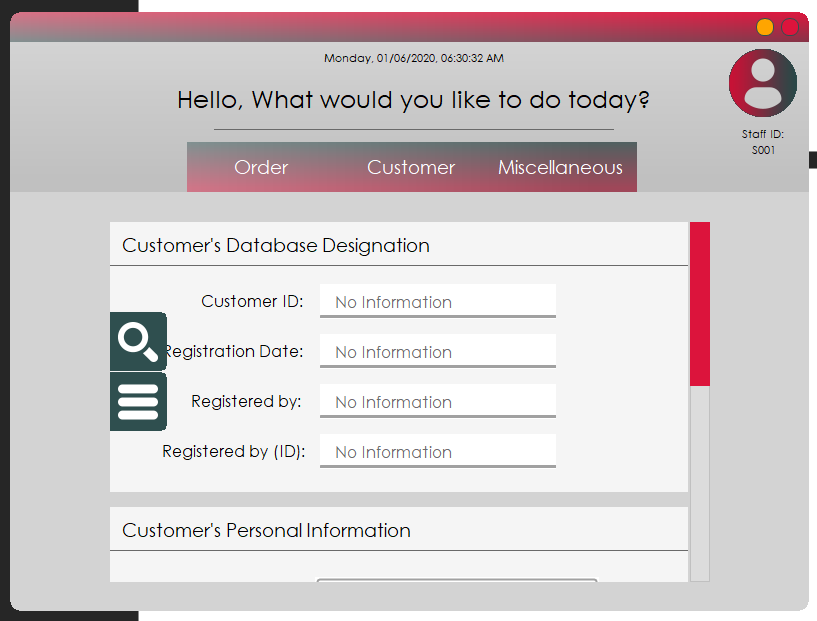


If there is no fault in the fields insertion, this popup will appear, and the customer has successfully been registered for the system to access to. The customer can now start ordering from the menu. Next will be on how to look up for a customer profile.

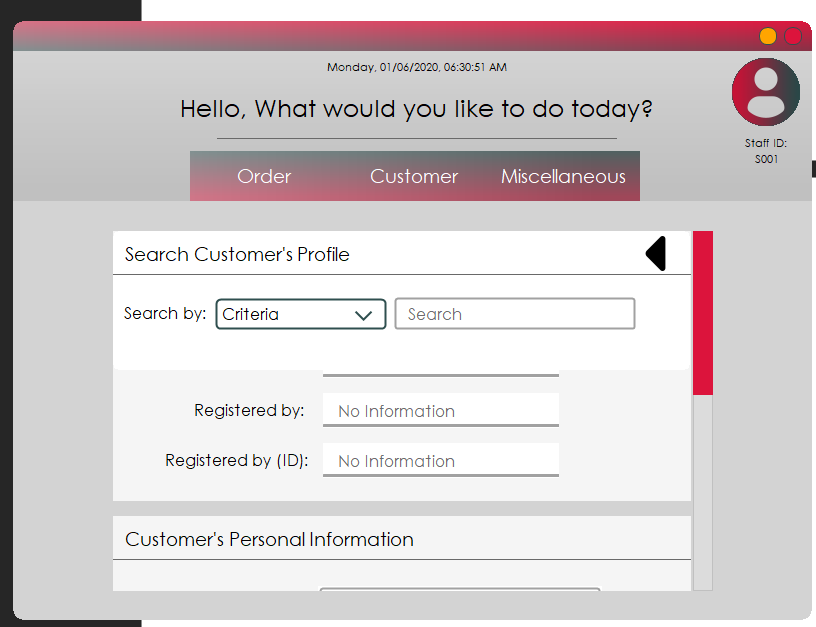
## Usage – Accessing a customer’s profile



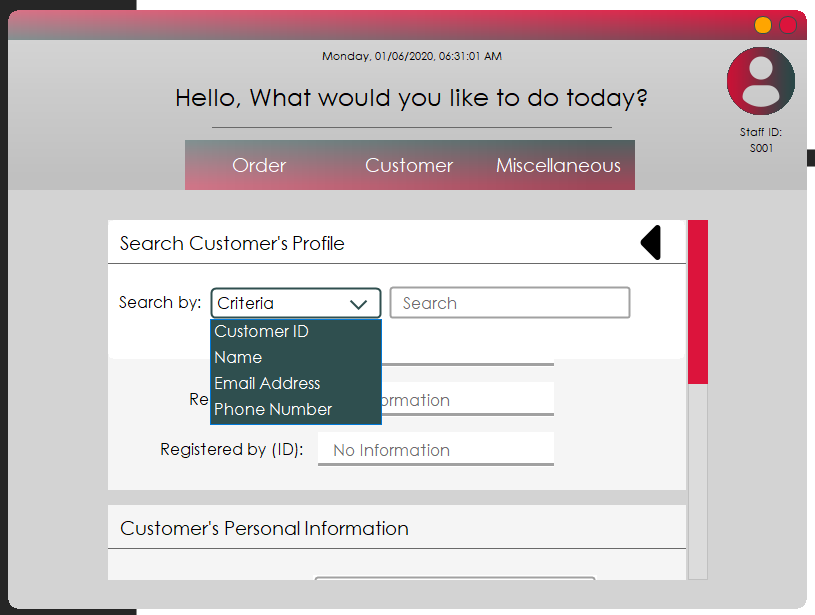
To access a customer’s profile, the staff must load the profile section by going to Customer and then press on Profile. It will load the profile interface underneath the dropdown menu.



This is the customer profile interface. There are two buttons on the left which handles how the form operates. To search, press on the first button.



The first button will extend this panel. To search for a record, the staff is required to select a criterion.



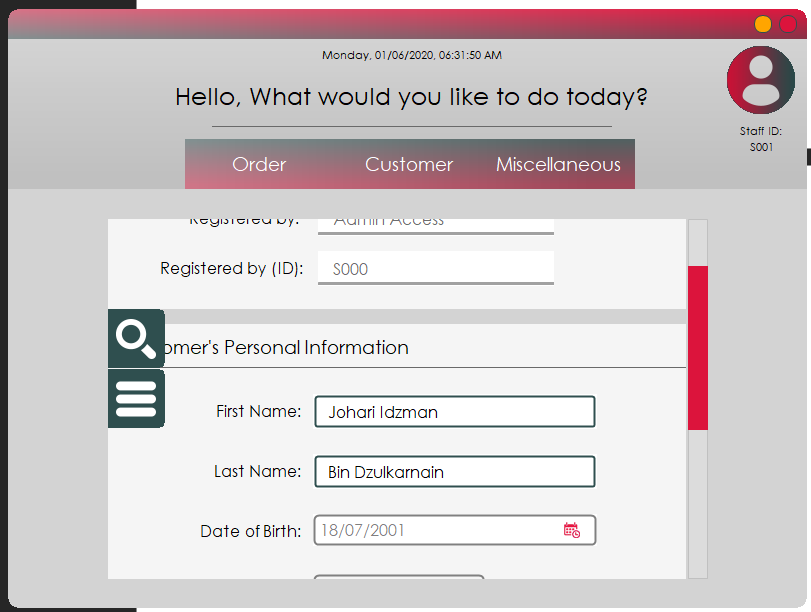
Once the staff pressed on the dropdown box, this list will appear. The criterion handles the search function differently. As the name of each criterion suggests, select the one that is most suitable for the search keyword. Note that Name requires a full name of the customer.



After selecting a criterion, the reset and search button appears.

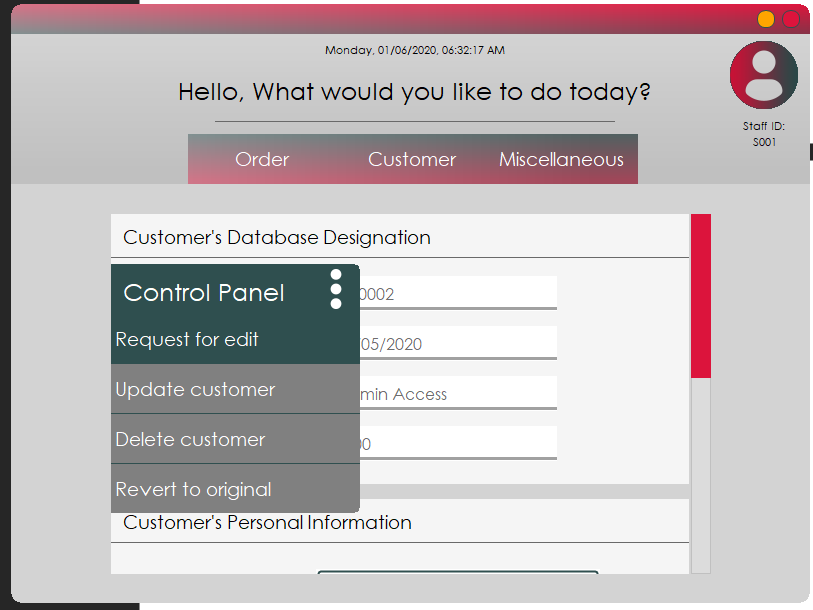


Insert a valid keyword into the search box and press on search to continue.

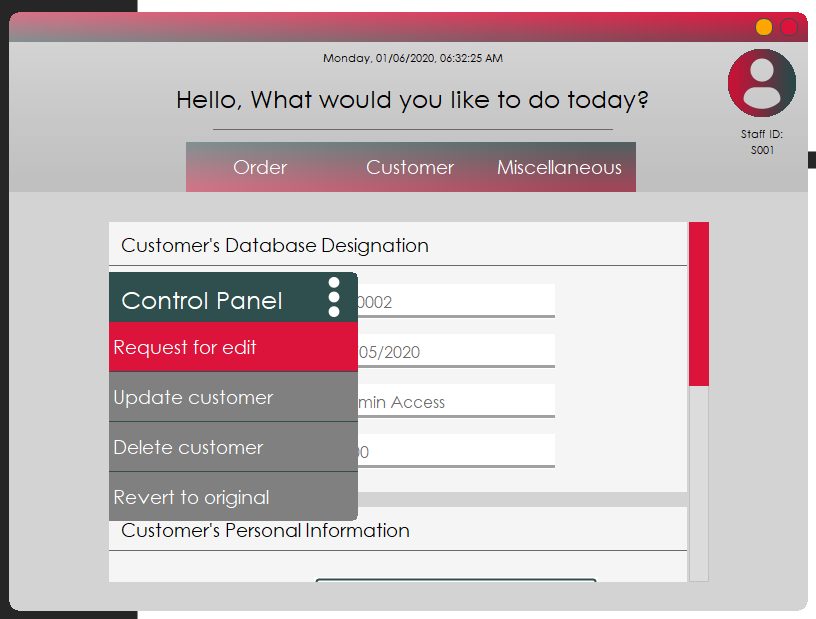


If the staff inserted an invalid or deleted customer information, a popup will appear to notify the user of the action. However, if the search returns a result, the customer record will load fully into designated fields. Next will be on how to modify a customer profile.

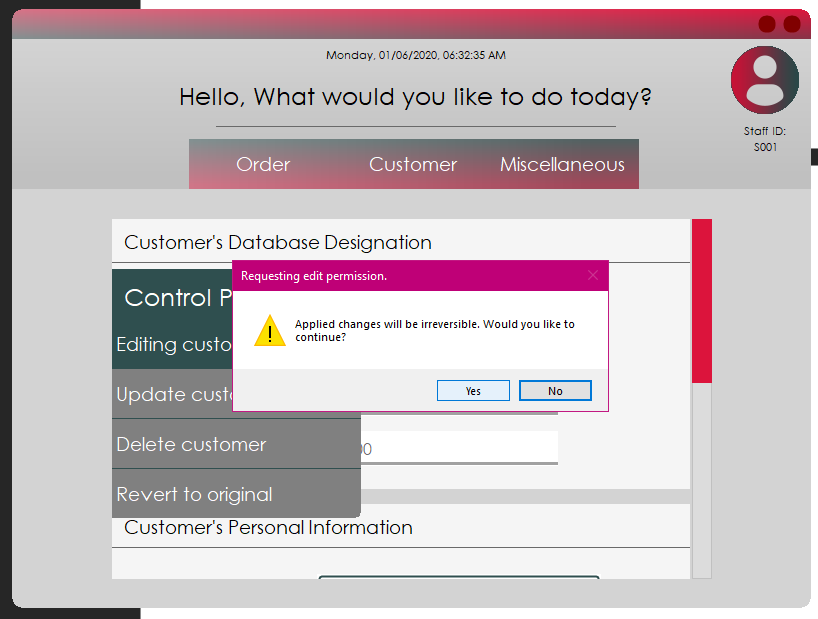
## Usage – Modifying a customer’s profile



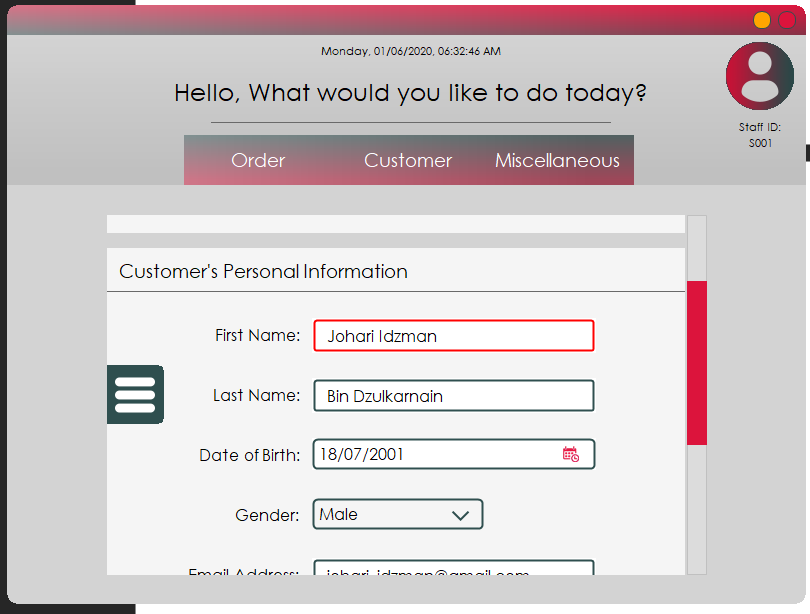
From the customer profile interface, select a customer and load it. Once loaded, press on the second button to extend this panel.



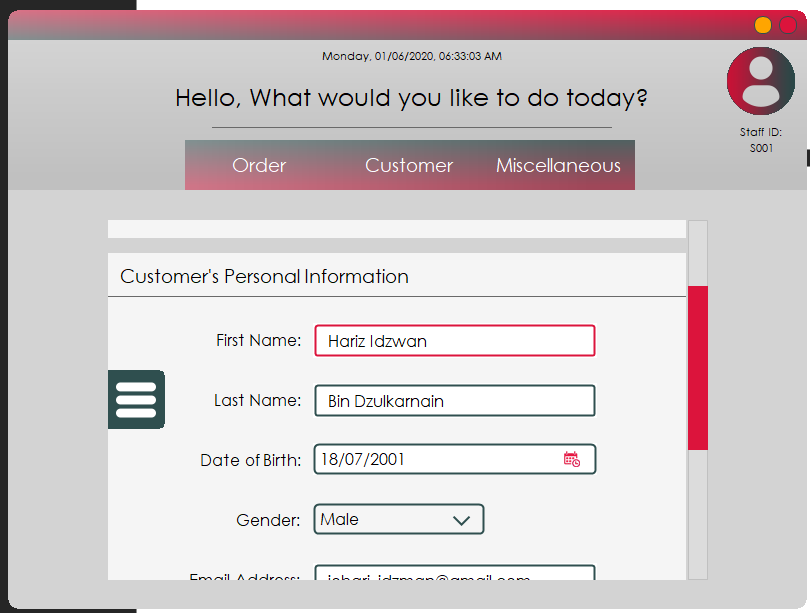
Press on request for edit to allow editing privileges to become enabled so that the records can be altered.



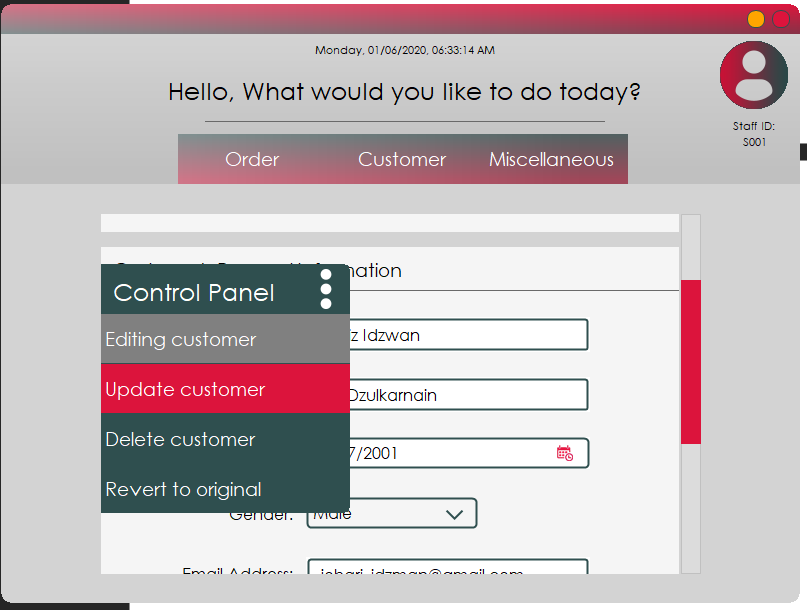
This popup will appear to notify the staff of irreversible effect any applied changes will have on the record. Press Yes to continue.



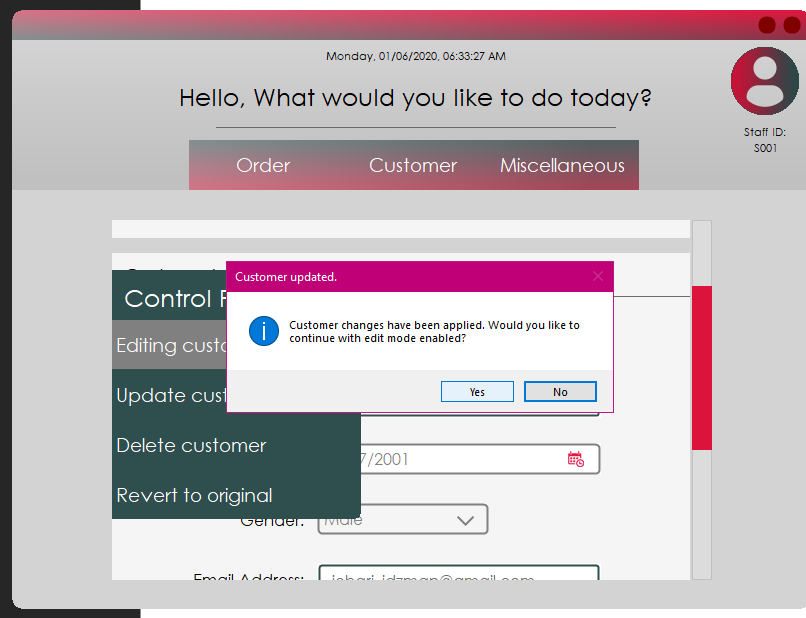
Now the form has unlocked the fields and ready to be edited.



Here, the name Johari Idzman is changed to Hariz Idzwan. For the staff to save the current change, the button on the left must be pressed to access the controls.

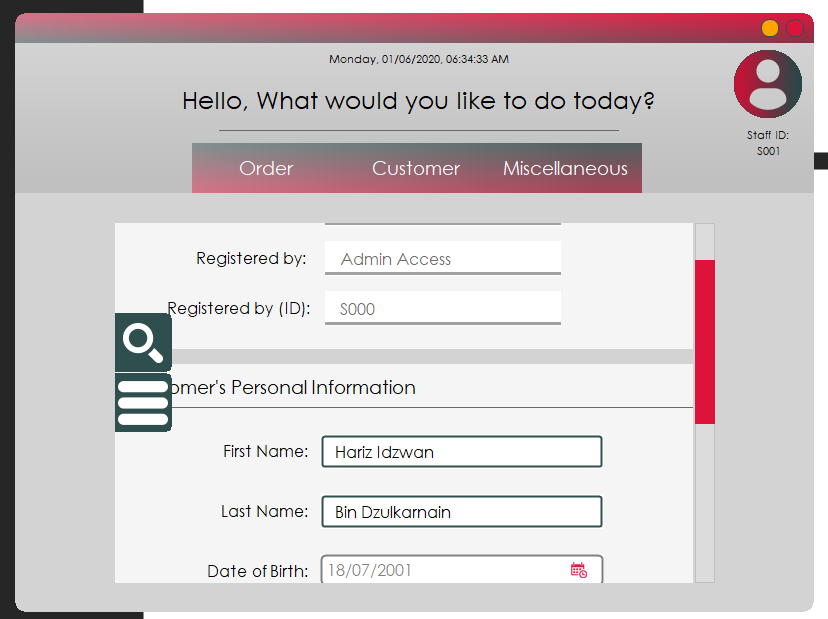


Now press on the update customer button to have the changes applied and be reflected into the database.

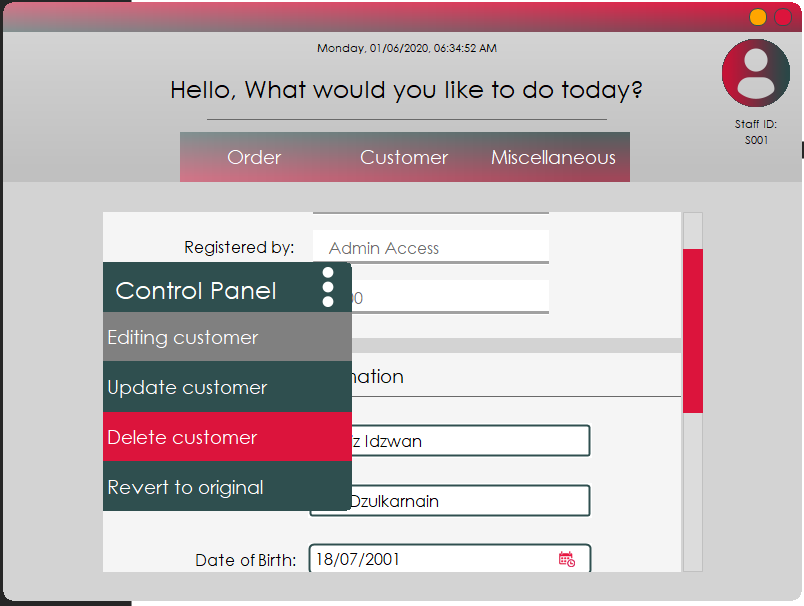


The popup appears to notify the staff that the record has successfully been updated. Pressing Yes to this popup will keep the edit mode privilege while No reverts the setting back to normal. Next will be on how to delete a customer profile.

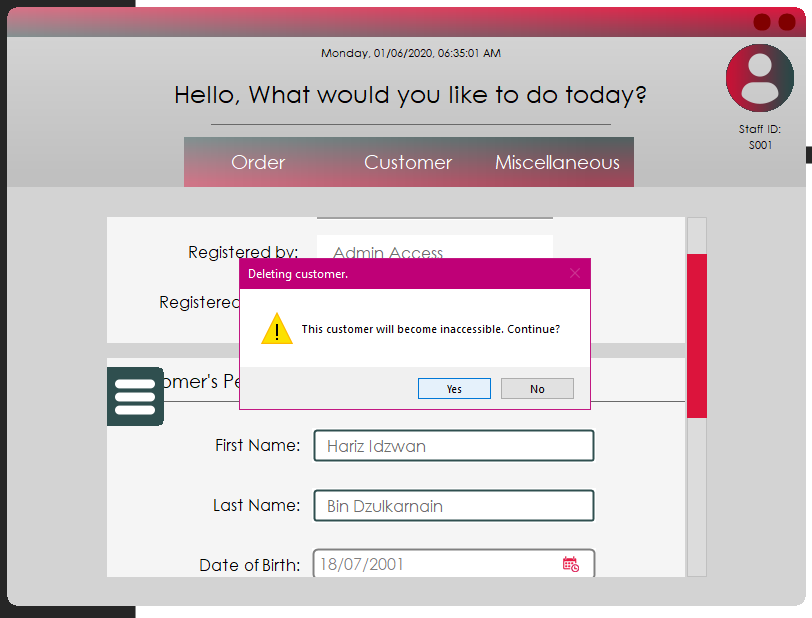
## Usage – Deleting a customer’s profile



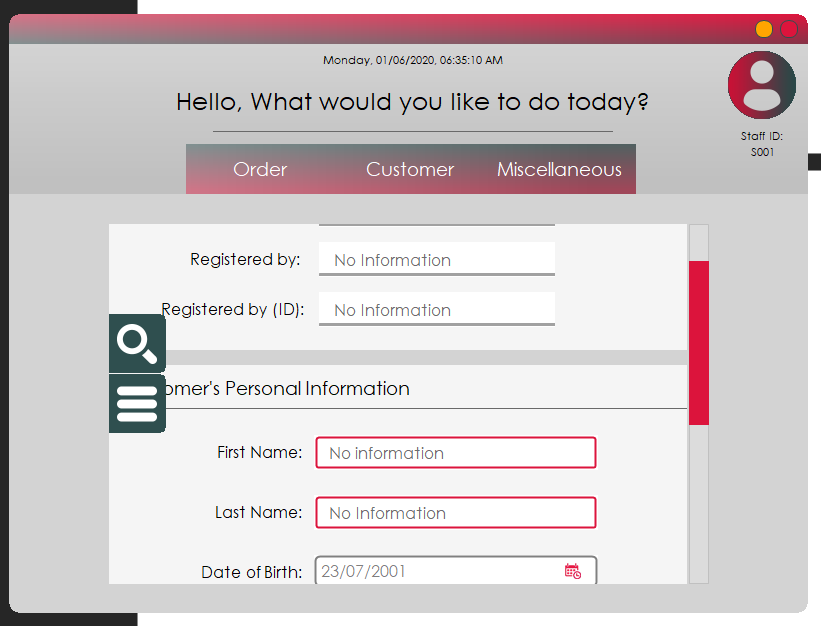
To delete a customer profile, search for the customer and let it load. Once loaded, press on the second button to request for edit permission.



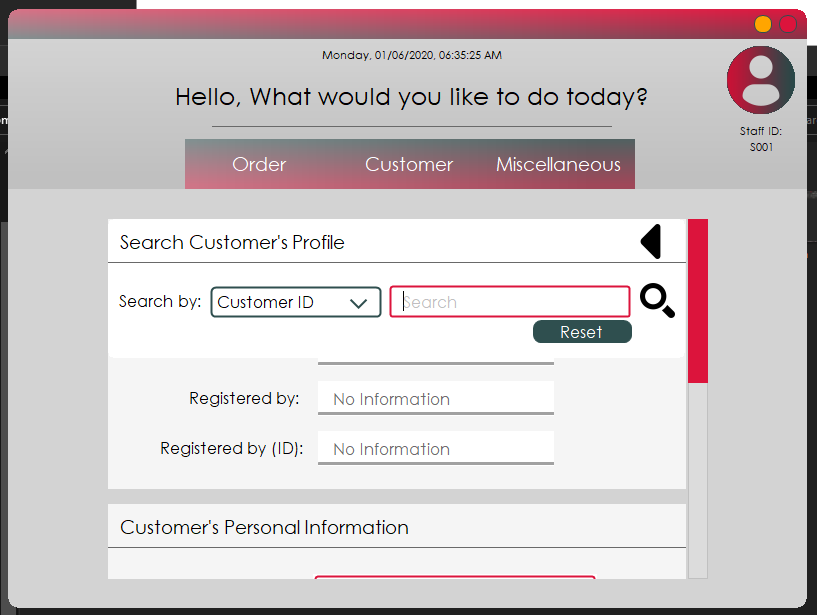
Once the edit permission is enabled, press on the delete customer button to proceed.



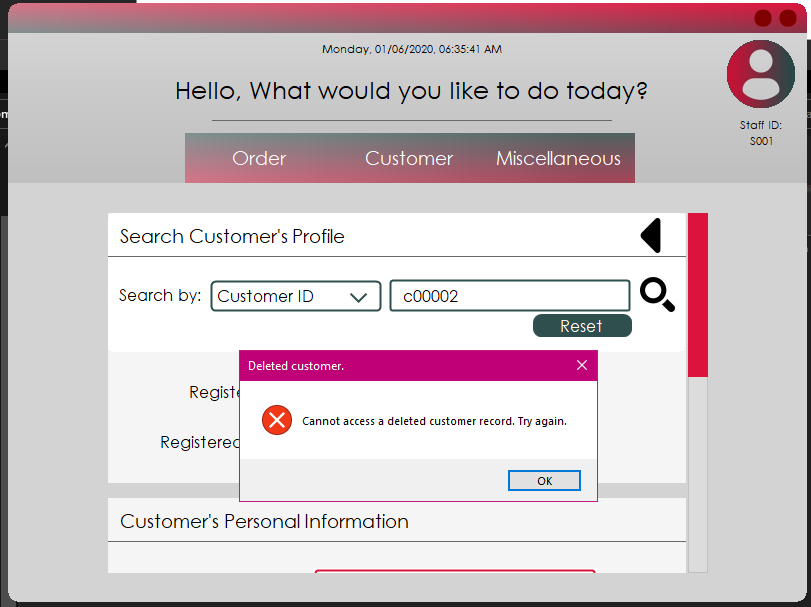
This popup will appear to ask for confirmation from the staff to avoid accidental record deletion. Press Yes to continue.



Once deleted, the form will reset itself and to check whether the record has been deleted, go and do a search using the same ID.

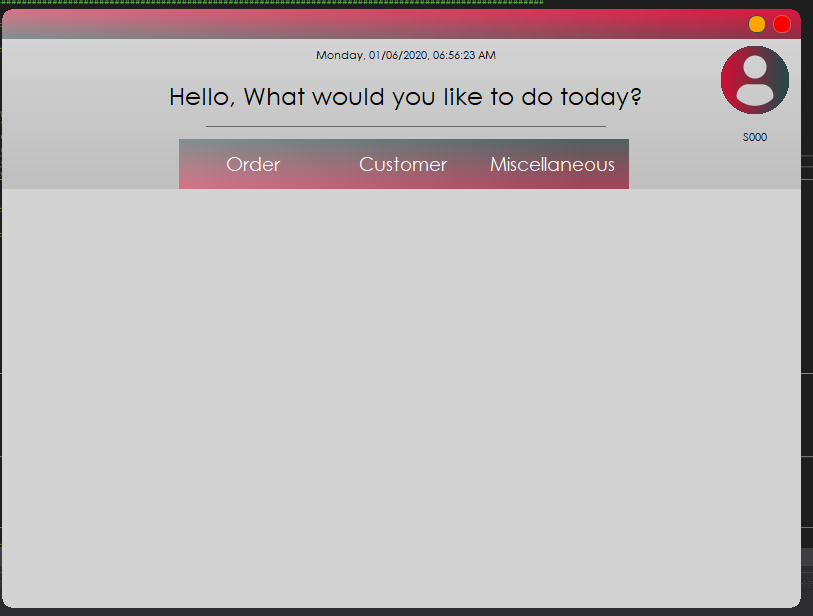


Once the panel extends, perform the procedure to load customer record.

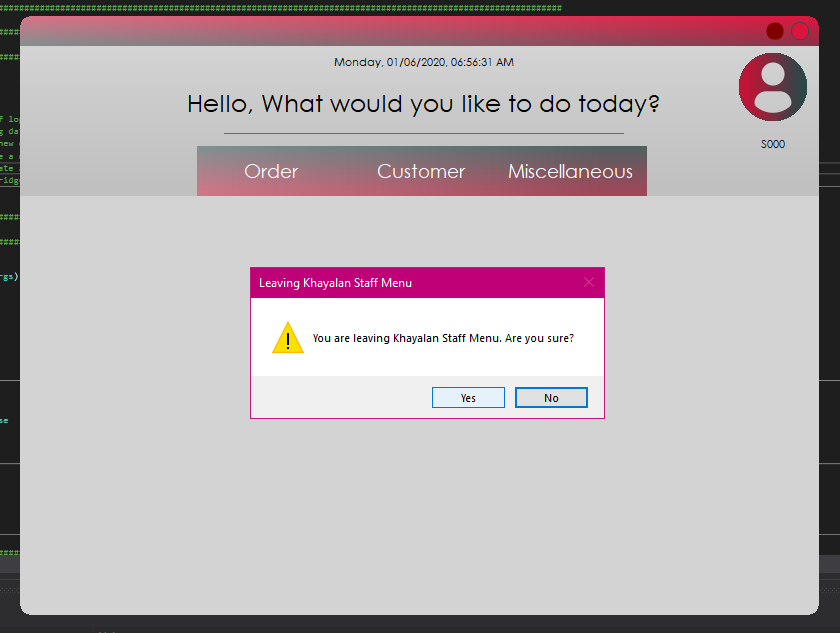


If this popup appears, the customer record has been deleted successfully. Next will be on how to exit the application.

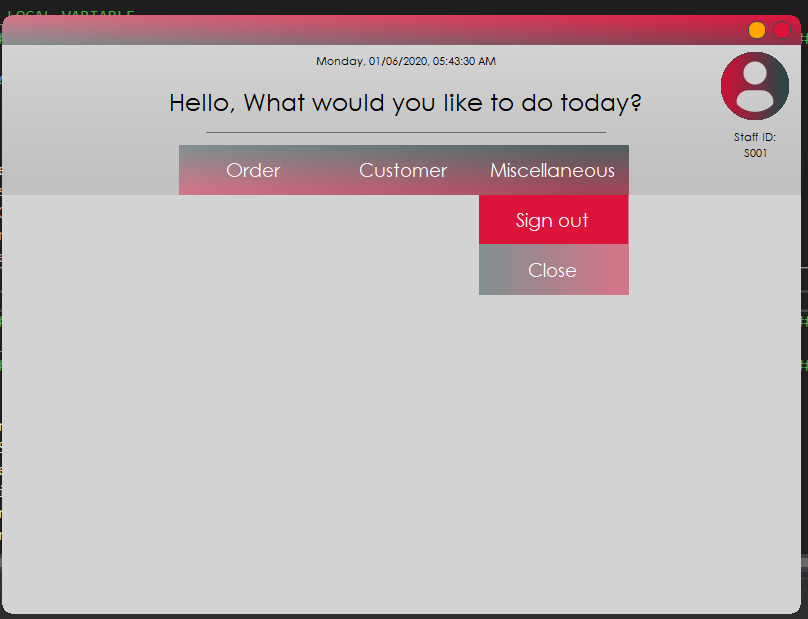
## Usage – Exiting the application



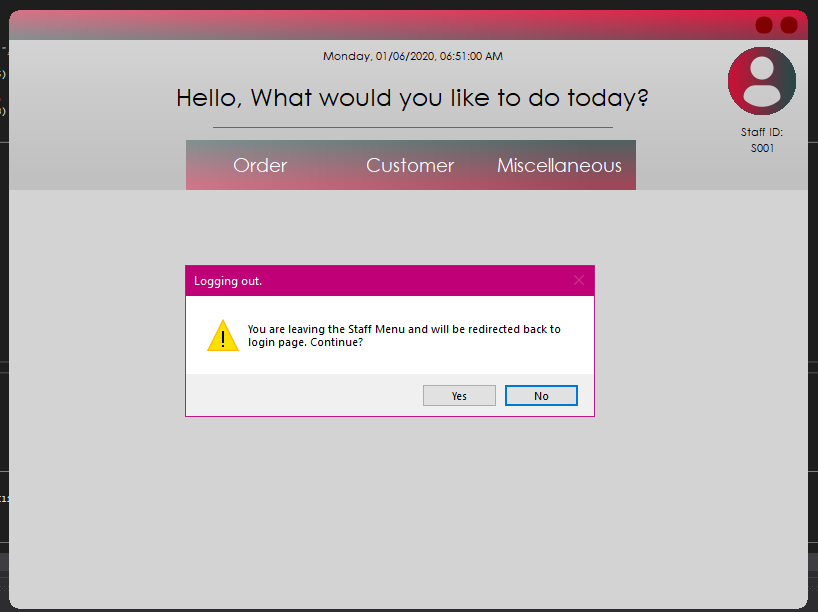
To close the application, the staff can either log out first or close straightaway from the staff menu interface. For the first method, the staff can close it by pressing on the top right corner button.



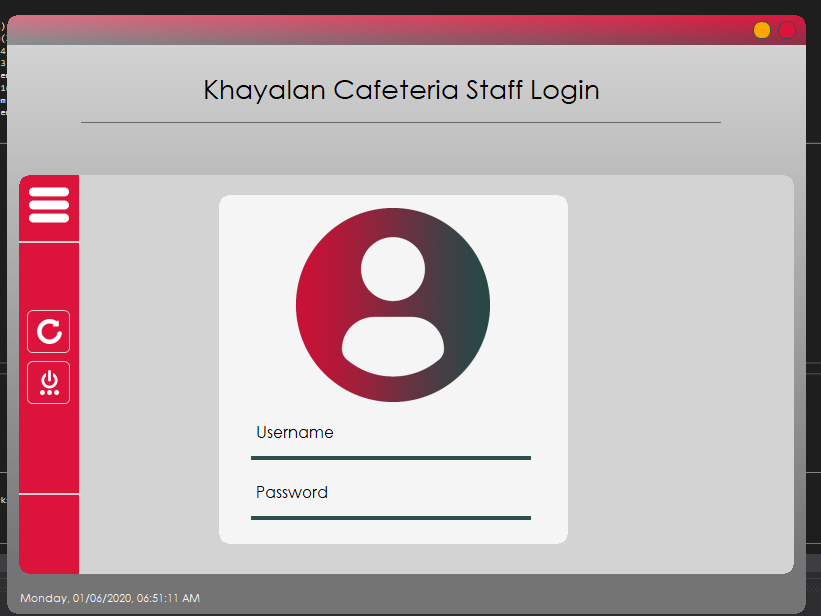
The button will display the confirmation popup. Press Yes to close the application.



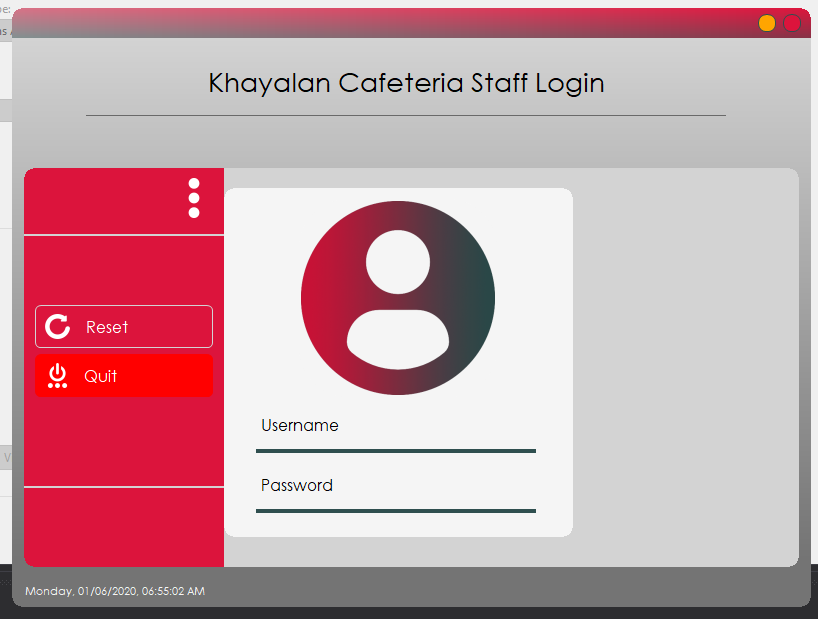
If the staff prefers signing out, the staff can go to Miscellaneous and press on Sign Out to be redirected back to the login page.



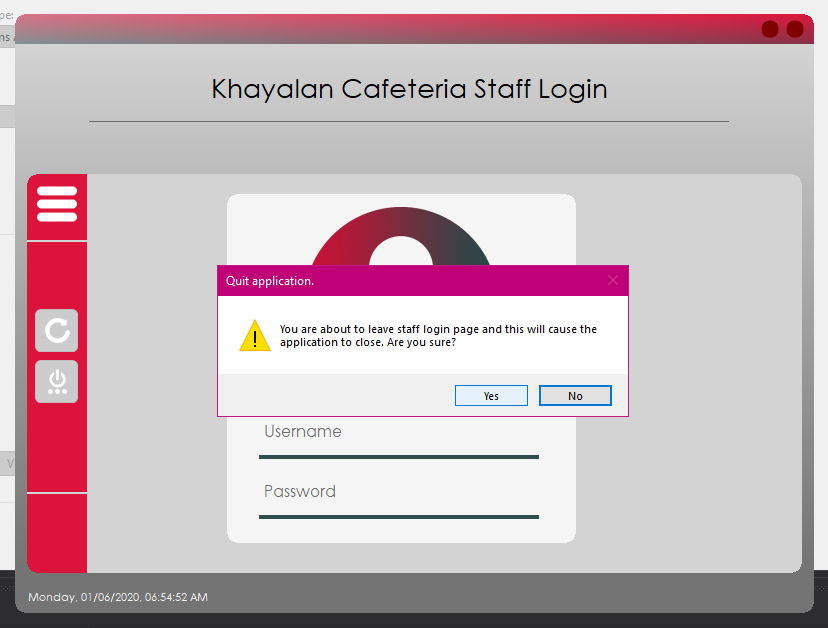
A confirmation popup will appear and press Yes to continue.



Once the user is redirected back to the login page, there are two ways but with the same outcome of closing the application. The top right button works essentially the same as the one found in staff menu. The button found inside the side panel is not an exception.



Press the button to extend the panel and press on Quit.



The confirmation popup will appear and press Yes to proceed with the exit. This is the end of the user manual guide to use the Khayalan Cafeteria system.

# 4.0 TEST PLAN AND TEST CASES

## 1.0 Splash Screen Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Startup timer | To see whether it loads the staff login interface during the 4th second. | It loads staff login interface as it is intended to | Similar to expected result | None |
| 2 | Introductory transition | To see whether it plays the initial startup transition | It plays the transition as it should | Similar to expected result | None |
| 3 | Hide form | To see whether the form hides itself after the 9th second where the timer will stop. | It hides itself | Similar to expected result | None |
| 4 | Bring staff login interface to front | To see whether the form brings the loaded staff login interface to front before setting itself hidden | It brings the form forward and proceeds to hide itself | Similar to expected result | None |
| 5 | Loading caption | To see whether the loading caption changes itself according to specified text at specified second | It changes itself according to specified text at specified second | Similar to expected result | There is a bit of delay due to background loading |

## 2.0 Staff Login Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Credentials checker | To see whether it allows user to enter after correct credentials are inserted | It popups a messagebox to notify of successful login | Similar to expected result | None |
| 2 | Credentials checker | To see whether it prevents user from entering after wrong credentials are inserted | It popups an error and ask the user to check the input | Similar to expected result | None |
| 3 | Extend side pane button | To see whether the button extends the side pane once pressed | It extends the side pane | Similar to expected result | None |
| 4 | Retract side pane button | To see whether the button retracts the side pane once pressed | It retracts the side pane | Similar to expected result | None |
| 5 | Reset form button | To see whether it resets the textboxes | It clears the textboxes. | Similar to expected result | None |
| 6 | Quit application button | To see whether a popup to confirm user action displays | It displays user popup | Similar to expected result | None |
| 7 | Quit application button | To see whether the Yes button exits the application | It exits the application | Similar to expected result | None |
| 8 | Quit application button | To see whether the No button cancels the action | It hides the popup and run the application normally | Similar to expected result | None |
| 9 | Show login button | To see whether the login button is displayed when both textboxes are filled | It shows the login button when both textboxes are filled | Similar to expected result | None |
| 10 | Hide login button | To see whether the login button hides itself when one or both textboxes are empty | It hides itself when either or both textboxes are empty | Similar to expected result | None |
| 11 | Forward Staff ID of logged in user | To see whether the login sequence loads Staff ID from the form to staff menu interface | It displays staff ID under the user icon in staff menu interface | Similar to expected result | None |

## 3.0 Staff Main Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Quit application button | To see whether it popups a messagebox to confirm user action | It popups a messagebox to ask for confirmation | Similar to expected result | None |
| 2 | Quit application button | To see whether the ‘Yes’ button exits the application | It exits the application | Similar to expected result | None |
| 3 | Quit application button | To see whether the ‘No’ button hides the popup and run the application normally | It hides the popup and runs the application normally | Similar to expected result | None |
| 4 | Sign out button | To see whether it popups messagebox to confirm user action | It popups a messagebox to ask for confirmation | Similar to expected result | None |
| 5 | Sign out button | To see whether the ‘Yes’ button hides the form and redirects user to staff login interface | It hides the form and redirects user to staff login interface | Similar to expected result | None |
| 6 | Sign out button | To see whether the ‘No’ button hides the popup and run the application normally | It hides the popup and runs the application normally | Similar to expected result | None |
| 7 | Order button | To see whether it extends the order panel | It extends the order panel | Similar to expected result | None |
| 8 | Customer button | To see whether it extends the customer panel | It extends the customer panel | Similar to expected result | None |
| 9 | Miscellaneous button | To see whether it extends the miscellaneous panel | It extends the miscellaneous panel | Similar to expected result | None |
| 10 | Close button | To see whether all close buttons inside each panel closes respective panels | It hides respective panels only | Similar to expected result | None |
| 11 | Hide opened buttons | To see whether when a button is pressed, preceding panels that are opened is closed | It closes the opened panels first before opening the recently pressed button | Similar to expected result | None |
| 12 | Take Order button | To see whether it loads customer selection interface within the form with transition | It loads customer selection interface inside the form with transition | Similar to expected result | None |
| 13 | Profile button | To see whether it loads the customer profile interface within the form | It loads customer profile interface inside the form | Similar to expected result | None |
| 14 | Register button | To see whether it loads the customer registration interface within the form | It loads customer registration interface inside the form | Similar to expected result | None |
| 15 | Clock | To see whether the clock functions properly by displaying the correct time | It displays the correct time | Similar to expected result | There might be some freezing event while loading form |

## 4.0 Customer Registration Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Empty fields checker | To see whether new entries are rejected when required fields are left empty | It displays a popup asking user to insert into required fields by displaying warning labels on top of each required fields | Similar to expected result | None |
| 2 | Invalid formatting checker | To see whether new entries are rejected when required fields are inserted with invalid format | It displays a popup asking user to fix the formatting by displaying warning labels on top of each invalid formatting fields | Similar to expected result | None |
| 3 | Add entry button | To see whether the customer record is updated to database and display a success popup | It updates the database with new row and display a popup to notify of successful registration | Similar to expected result | None |
| 4 | Show side pane button | To see whether the side pane will extend when the button is pressed | It extends the pane when the button is pressed | Similar to expected result | None |
| 5 | Hide side pane button | To see whether the side pane will retract when the button is pressed | It retracts the pane when the button is pressed | Similar to expected result | None |
| 6 | First and Last name validation | To see whether user can insert only alphabetical characters and not symbols and numerals | It displays warning labels whenever non-alphabetical characters are pressed and prevent them from being inserted | Similar to expected result | None |
| 7 | Date of Birth validation | To see whether a warning label displays to notify user of default value when left unchanged | It displays the warning label to set default to ‘23/07/2001’ | Similar to expected result | None |
| 8 | Gender validation | To see whether a warning label displays to notify user of default value when left unselected | It displays the warning label to set default to ‘Unselected’ | Similar to expected result | None |
| 9 | Email validation | To see whether a persistent warning label to indicate wrong email address formatting displays | It displays the warning label until the correct format is inserted | Similar to expected result | None |
| 10 | Phone number validation | To see whether a persistent warning label to indicate wrong phone number formatting displays | It displays the warning label until the correct format is inserted | Similar to expected result | None |
| 11 | Street, City and State validation | To see whether a label appears on each fields of empty fields | It displays the warning label when mouse cursor is within the field | Similar to expected result | None |
| 12 | Postcode validation | To see whether a persistent label appears when the inserted string is below required length | It displays the warning label when the input is less than 4 characters | Similar to expected result | None |
| 13 | Country validation | To see whether a warning label displays to notify user of default value when left unselected | It displays the warning label to set to default to ‘Malaysia’ | Similar to expected result | None |
| 14 | Reset form button | To see whether it clears all fields off inputs and reset the form to original state | It clears each fields and hides remaining warning labels before retracting side pane | Similar to expected result | None |
| 15 | Customer ID incrementation | To see whether Customer ID will increment automatically to prevent ID duplication | It increments the value by 1 every time a customer is registered | Similar to expected result | None |
| 16 | Date and Time generation | To see whether date and time is displayed inside the textbox | It gets the current date and time and display it inside respective textboxes | Similar to expected result | None |
| 17 | Close window button | To see whether the form will close itself when the button is pressed | It will close the form and reset the form to its original condition | Similar to expected result | None |

## 5.0 Customer Profile Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Empty fields checker | To see whether new entries are rejected when required fields are left empty | It displays a popup asking user to insert into required fields by displaying warning labels on top of each required fields | Similar to expected result | None |
| 2 | Invalid formatting checker | To see whether new entries are rejected when required fields are inserted with invalid format | It displays a popup asking user to fix the formatting by displaying warning labels on top of each invalid formatting fields | Similar to expected result | None |
| 3 | Update customer button | To see whether the customer record is updated to database and display a success popup | It updates the database of the specified customer with new records and display success notification | Similar to expected result | None |
| 4 | Show side pane button | To see whether the side pane will extend when the button is pressed | It extends the pane when the button is pressed | Similar to expected result | None |
| 5 | Hide side pane button | To see whether the side pane will retract when the button is pressed | It retracts the pane when the button is pressed | Similar to expected result | None |
| 6 | First and Last name validation | To see whether user can insert only alphabetical characters and not symbols and numerals | It displays warning labels whenever non-alphabetical characters are pressed and prevent them from being inserted | Similar to expected result | None |
| 7 | Date of Birth validation | To see whether a warning label displays to notify user of default value when left unchanged | It displays the warning label to set default to ‘23/07/2001’ | Similar to expected result | None |
| 8 | Gender validation | To see whether a warning label displays to notify user of default value when left unselected | It displays the warning label to set default to ‘Unselected’ | Similar to expected result | None |
| 9 | Email validation | To see whether a persistent warning label to indicate wrong email address formatting displays | It displays the warning label until the correct format is inserted | Similar to expected result | None |
| 10 | Phone number validation | To see whether a persistent warning label to indicate wrong phone number formatting displays | It displays the warning label until the correct format is inserted | Similar to expected result | None |
| 11 | Street, City and State validation | To see whether a label appears on each fields of empty fields | It displays the warning label when mouse cursor is within the field | Similar to expected result | None |
| 12 | Postcode validation | To see whether a persistent label appears when the inserted string is below required length | It displays the warning label when the input is less than 4 characters | Similar to expected result | None |
| 13 | Country validation | To see whether a warning label displays to notify user of default value when left unselected | It displays the warning label to set to default to ‘Malaysia’ | Similar to expected result | None |
| 14 | Revert to original | To see whether the form resets itself to original | It clears each fields and hides remaining warning labels before retracting side and search pane | Similar to expected result | None |
| 15 | Request for edit button | To see whether a popup notification displays to confirm user action | It displays a popup asking the user for confirmation | Similar to expected result | None |
| 16 | Request for edit button | To see whether ‘Yes’ allows user to edit the customer details | It enables edit-bound controls and also disables read-only states for every field | Similar to expected result | None |
| 17 | Request for edit button | To see whether ‘No’ returns the form to view only mode | It retracts side pane and displays each field as read-only | Similar to expected result | None |
| 18 | Open search pane button | To see whether the search pane extends itself when the button is pressed | It extends itself and show all search controls | Similar to expected result | None |
| 19 | Close search pane button | To see whether the search pane retracts itself when the button is pressed | It retracts itself | Similar to expected result | None |
| 20 | Displays search button | To see whether the search button displays after selecting from criteria selection | It shows itself when the user selects search criteria | Similar to expected result | None |
| 21 | Displays reset button | To see whether the search button displays after selecting from criteria selection | It shows itself when the user selects search criteria | Similar to expected result | None |
| 22 | Displays information based on criteria | To see whether the information fetched is based on the selected criteria | It loads each field with the records from database according to criteria keyword | Similar to expected result | Name requires a full string of First and Last name combined. Only the first search result for any similar name will be displayed |
| 23 | Accessing deleted record checker | To see whether the system will prevent loading a customer labelled as ‘Deleted’ | It will popup an error to notify the user of deleted customer | Similar to expected result | None |

## 6.0 Customer Selection Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Return to menu button | To see whether the user will be redirected back to staff menu when the button is pressed | It will play a transition and show staff menu interface | Similar to expected result | None |
| 2 | Criteria selection combo box | To see whether the selection made by the user will display the search box | It will immediately display the search input box | Similar to expected result | None |
| 3 | Search textbox | To see whether the search button is displayed only when the input box is not empty | It will detect for inputted characters real-time and show or hide the search button accordingly | Similar to expected result | None |
| 4 | Search customer button | To see whether a panel with fetched customer fields loads | It loads with transition a panel that has textboxes which displays fetched customer records | Similar to expected result | None |
| 5 | Invalid search checker | To see whether an invalid input displays error message | It popups an error to notify user of invalid search | Similar to expected result | None |
| 6 | Deleted record  Checker | To see whether the system prevents from loading customer labelled as ‘Deleted’ | It popups an error notifying of deleted customer | Similar to expected result | None |
| 7 | Progress bar | To see whether the progress bar updates itself immediately after user perform certain action | It increments by part during certain actions such as selecting criteria, inputting search keywords and pressing search customer button | Similar to expected result | None |
| 8 | Proceed with selected customer | To see whether the ‘Yes’ button loads the food selection interface | It loads the food selection interface | Similar to expected result | None |
| 9 | Don’t proceed with selected customer | To see whether the ‘No’ button return the user to the search screen | It hides and resets the fields and return the user back to search | Similar to expected result | None |

## 7.0 Food Selection Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Specific item functionality | To see whether each item has its own count and total price per item when selected | It will display the count and total per item respective to the item button pressed | Similar to expected result | None |
| 2 | Item button | To see whether the button will load relevant information | It will load the image, price and last count of the item | Similar to expected result | None |
| 3 | Count | To see whether count displays the add to cart button when its value is not 0 | It will show or hide the add to cart button respective to the count value | Similar to expected result | None |
| 4 | Add to Cart button | To see whether the total price updates itself and store the selected item inside database | It stores the item inside the database along with its name, type, price, quantity and total price before updating the total inside the form | Similar to expected result | None |
| 5 | Update Cart button | To see whether the button displays itself only when there is a record of the respective item inside database | It shows itself only when there is a record of the item inside database | Similar to expected result | None |
| 6 | Manual count textbox | To see whether the textbox will update the total count dynamically | It updates the count whenever the textbox refreshes with new value | Similar to expected result | None |
| 7 | Cancel order button | To see whether the button will clear the ongoing order and send the user back to customer selection interface | It clears the temporary order table and resets the order before sending the user to customer selection interface | Similar to expected result | None |

## 8.0 Drink Selection Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Specific item functionality | To see whether each item has its own count and total price per item when selected | It will display the count and total per item respective to the item button pressed | Similar to expected result | None |
| 2 | Item button | To see whether the button will load relevant information | It will load the image, price and last count of the item | Similar to expected result | None |
| 3 | Count | To see whether count displays the add to cart button when its value is not 0 | It will show or hide the add to cart button respective to the count value | Similar to expected result | None |
| 4 | Add to Cart button | To see whether the total price updates itself and store the selected item inside database | It stores the item inside the database along with its name, type, price, quantity and total price before updating the total inside the form | Similar to expected result | None |
| 5 | Update Cart button | To see whether the button displays itself only when there is a record of the respective item inside database | It shows itself only when there is a record of the item inside database | Similar to expected result | None |
| 6 | Manual count textbox | To see whether the textbox will update the total count dynamically | It updates the count whenever the textbox refreshes with new value | Similar to expected result | None |

## 9.0 Sides Selection Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Specific item functionality | To see whether each item has its own count and total price per item when selected | It will display the count and total per item respective to the item button pressed | Similar to expected result | None |
| 2 | Item button | To see whether the button will load relevant information | It will load the image, price and last count of the item | Similar to expected result | None |
| 3 | Count | To see whether count displays the add to cart button when its value is not 0 | It will show or hide the add to cart button respective to the count value | Similar to expected result | None |
| 4 | Add to Cart button | To see whether the total price updates itself and store the selected item inside database | It stores the item inside the database along with its name, type, price, quantity and total price before updating the total inside the form | Similar to expected result | None |
| 5 | Update Cart button | To see whether the button displays itself only when there is a record of the respective item inside database | It shows itself only when there is a record of the item inside database | Similar to expected result | None |
| 6 | Manual count textbox | To see whether the textbox will update the total count dynamically | It updates the count whenever the textbox refreshes with new value | Similar to expected result | None |
| 7 | To Order Summary button | To see whether the form loads order summary interface when there is no selection | It will popup an error asking the user to select an item before resuming | Similar to expected result | None |

## 10.0 Order Summary Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Fetch order information from database | To see whether the customer and order record is displayed based on the ID | It will display the current order and the customer record based on its ID | Similar to expected result | None |
| 2 | Display selected item datagridview | To see whether the datagridview only displays the selected item along with its required information | It will only display the selected items along with its price, quantity, type and total price | Similar to expected result | None |
| 3 | Tax inclusion | To see whether the total payable displays value after taxation and without rounding it off to the nearest place | It will display the total payable inclusive of tax | Similar to expected result | None |
| 4 | Date and Time of order textboxes | To see whether the texboxes display the date and time of when the form loads | It updates the date and time whenever the summary order is loaded | Similar to expected result | None |

## 11.0 Payment Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Amount to be paid | To see whether the amount to be paid displays the total after tax and rounding off | It will display the total inclusive of tax and rounded off to the nearest place | Similar to expected result | None |
| 2 | The payment notes buttons | To see whether the buttons handle the correct amount of increments based on its predefined values | It will increment the total paid based on the value it is set to | Similar to expected result | None |
| 3 | Complete button | To see whether the complete button stays hidden until the amount paid equals or greater than the amount to be paid | It will display itself when the total paid is equal or greater than the amount to be paid | Similar to expected result | None |
| 4 | View order button | To see whether the button displays the datagridview of the current order | It shows the datagridview of the current order list in the middle of the screen | Similar to expected result | None |
| 5 | Add remarks button | To see whether the button displays a textbox for description purposes | It shows the textbox above the payment notes buttons | Similar to expected result | None |
| 6 | Amount to be returned | To see whether the amount to be returned only appears when there is value to be returned | It will only be displayed when it is more than RM0.00 | Similar to expected result | None |
| 7 | Subtract button | To see whether the button removes the paid amount by the value it is predefined to | It will subtract off paid amount by the value it is set to | Similar to expected result | None |
| 8 | Add button | To see whether the button resumes the operation of adding to paid amount | It will switch from subtract to addition | Similar to expected result | None |
| 9 | Cancel button | To see whether there is a popup to ask user for confirmation | It will display a popup to confirm with the user for cancellation | Similar to expected result | None |
| 10 | Cancel button | To see whether the ‘Yes’ button cancels the order and redirects user to staff menu interface | It will clear the ongoing order from the database and returns the user to staff menu interface | Similar to expected result | None |
| 11 | Cancel button | To see whether the ‘No’ button resumes the order process | It will hide the popup and continue the ordering process | Similar to expected result | None |

## 12.0 Receipt Interface

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| Test Case | Function Name | Test Objective | Expected Result | Actual Result | Remarks |
| - | Name of the function to be tested | The objective to the test conducted | The expected outcome before the function is processed | The actual outcome after the function is processed | Brief description to the test conducted |
| 1 | Customer details | To see whether the fetched customer information is according to the ongoing order | It will display the correct customer ID and the name | Similar to expected result | None |
| 2 | Order details | To see whether it displays the correct details of the completed order | It will display the correct order ID, date of order, total payable, tax rate, total paid and change returned | Similar to expected result | None |
| 3 | Item list datagridview | To see whether the datagridview displays only the selected item from the current order process | It will display only the items that has been added or updated to cart | Similar to expected result | None |
| 4 | Complete button | To see whether the button will complete the order and redirects the user to staff menu interface | It completes the order and return the user to staff menu interface | Similar to expected result | None |

# 5.0 CRITICAL ASSESSMENT

In a nutshell, the assignment was done to teach students the ability to program an application based on specific criteria. This was done to simulate real-world condition whereby the client will be defining the programmer’s do’s and don’ts. Although it was not easy to deliver a working application without bugs, the finished application can do the required tasks lined by the assignment criteria such as register and editing a customer’s record, placing an order for a customer and complete the order through a payment system. There are no additional features implemented throughout the programming session as most of it were too buggy to be troubleshooted and went beyond the allocated time consumption to finish this application.

The perks of using this system is that the menu system is robust as it stores its selected list inside a specially reserved temporary table in the database. This method helps the application knows which item from the menu selection list is selected as the item is identified not by the name but the unique one-time ID. It also enables specific item updating so that no new rows are created when the same item is added again. These are applied to the temporary cart before the order is finalised. Therefore, when an order is to set as complete, the system will transfer the records from the temporary storage and store inside the actual order table as reference along with unique order ID.

The next strength would be the customer registration interface. Since Microsoft Access is a bit problematic when it comes to Autonumbering feature whereby the sequence can never restart from 1 once the records are deleted, the Customer ID is tailored in a way that it uses Short Text data type instead. However, the ID is made so that it can be incremented although it is a string type by utilising an array function inside VB.Net. The Customer ID will always increment itself when it finds a record with the same ID from the database.

However, there is one bug that the programmer deems as ‘Random Bug’ due to its nature to randomly occur without any signs. It happens whenever the application loads from staff menu interface. Due to this random error, the programmer had spent so much time and effort into overcoming the matter. Unfortunately, there is no available fix to this problem until the application is finished. The random error pops up constantly at first but the exception will stop after few attempts. A simple yet unethical band-aid to this problem would be to repeatedly attempt to start the application after it crashes. It will load with no problem when there is no error and the application can be used as it is intended to.

The key weakness of this application is that it lacks the staff management system. The system does not have a way for staff to be registered other than manually inserting a new row inside the database. This is due to the time limit of the assignment and numerous troubleshooting that led to many ideas to become scrapped. It was planned to be added but time did not allow it to happen. To overcome the mismanagement of time, the programmer could create a strict Gantt chart that specifically allocates the amount of time required for each development phases so that many promising features can be implemented on time with less worry or bug.

Finally, this application had so much potentials, but the unstable loading sequence render it unappealing from time to time as applications must be reliable especially for business like cafeteria. The constant flow of customer into the premise will overload and crash the system. It is not recommended for future project to have eye-candy user-interface and transitions in place first before the core programmes. Focus on the fundamental aspects of the application before working out on the beautification.

# 6.0 REFERENCES

Youtube, 2016. Smooth Photoshop like Splash screen , C# Vb.Net - 現代設計用戶界面. [Online] Available at: <https://www.youtube.com/watch?v=I70OdofAZu0> [Accessed 30 April 2020]

Bunifu Framework, n.d. Bunifu DataGridView. [Online] Available at: <https://docs.bunifuframework.com/en/articles/2820601-bunifu-datagridview> [Accessed 30 May 2020]

Bunifu Framework, n.d. Bunifu Button. [Online] Available at: <https://docs.bunifuframework.com/en/articles/2240720-bunifu-button> [Accessed 3 May 2020]

Bunifu Framework, n.d. Bunifu Date Picker. [Online] Available at: <https://docs.bunifuframework.com/en/articles/2261426-bunifu-date-picker> [Accessed 15 May 2020]

Bunifu Framework, n.d. Bunifu Progress Bar. [Online] Available at: <https://docs.bunifuframework.com/en/articles/2264679-bunifu-progress-bar> [Accessed 23 May 2020]

Bunifu Framework, n.d. Bunifu ScrollBars. [Online] Available at: <https://docs.bunifuframework.com/en/articles/2270593-bunifu-scrollbars> [Accessed 12 May 2020]

Bunifu Framework, n.d. Bunifu Transition. [Online] Available at: <https://docs.bunifuframework.com/en/articles/2273086-bunifu-transition> [Accessed 15 April 2020]

Bunifu Framework, n.d. Bunifu Elipse. [Online] Available at: <https://docs.bunifuframework.com/en/articles/2272900-bunifu-ellipse> [Accessed 3 May 2020]

Stack Overflow, 2012. A fast way to delete all rows of a datatable at once. [Online] Available at: <https://stackoverflow.com/questions/11103181/a-fast-way-to-delete-all-rows-of-a-datatable-at-once/11103246> [Accessed 24 May 2020]

Stack Overflow, 2010. VB.NET: Clear DataGridView. [Online] Available at: <https://stackoverflow.com/questions/2217175/vb-net-clear-datagridview> [Accessed 24 May 2020]

Stack Overflow, 2009. How do you strip a character out of a column in SQL Server? [Online] Available at: <https://stackoverflow.com/questions/983417/how-do-you-strip-a-character-out-of-a-column-in-sql-server> [Accessed 24 May 2020]

Stack Overflow, 2014. Undefined function 'Replace' in expression , Replace alternative? [Online] Available at: <https://stackoverflow.com/questions/24908724/undefined-function-replace-in-expression-replace-alternative> [Accessed 24 May 2020]

Stack Overflow, 2014. VB.Net Checking if DataSet has rows or not [Online] Available at: <https://stackoverflow.com/questions/22033605/vb-net-checking-if-dataset-has-rows-or-not> [Accessed 25 May 2020]

Stack Overflow, 2012. Dynamic SQL generation for the UpdateCommand is not supported against a SelectCommand that does not return any key column information [Online] Available at: <https://stackoverflow.com/questions/14045901/dynamic-sql-generation-for-the-updatecommand-is-not-supported-against-a-selectco> [Accessed 25 May 2020]

Stack Overflow, 2012. what should I do in “Data type mismatch in criteria expression” [Online] Available at: <https://stackoverflow.com/questions/11087698/what-should-i-do-in-data-type-mismatch-in-criteria-expression> [Accessed 25 May 2020]

Stack Overflow, 2012. Get a textbox to accept only characters in vb.net. [Online] Available at: <https://stackoverflow.com/questions/12633780/get-a-textbox-to-accept-only-characters-in-vb-net> [Accessed 15 May 2020]

Microsoft, n.d. Math.Round Method. [Online] Available at: <https://docs.microsoft.com/en-us/dotnet/api/system.math.round?view=netcore-3.1> [Accessed 27 May 2020]

Microsoft, n.d. Addhandler Statement. [Online] Available at: <https://docs.microsoft.com/en-us/dotnet/visual-basic/language-reference/statements/addhandler-statement> [Accessed 8 May 2020]

DotNetPerls, n.d. VB.NET Event Example: AddHandler, RaiseEvent. [Online] Available at: <https://www.dotnetperls.com/event-vbnet> [Accessed 8 May 2020]

Microsoft, n.d. Sendkeys.Send(String) Method. [Online] Available at: <https://docs.microsoft.com/en-us/dotnet/api/system.windows.forms.sendkeys.send?view=netcore-3.1> [Accessed 12 May 2020]

Microsoft, 2015. How to make the label rotated in vb.net. [Online] Available at: <https://social.msdn.microsoft.com/Forums/vstudio/en-US/f2ec0dd0-5764-43e3-bddc-0b69ad2a5f44/how-to-make-the-label-rotated-in-vbnet?forum=vbgeneral> [Accessed 30 April 2020]

CodeProject, 2012. C# - Form inside a Panel. [Online] Available at: <https://www.codeproject.com/Questions/414107/Csharp-Form-inside-a-Panel> [Accessed 2 April]