FACULTY OF ENGINEERING AIN SHAMS UNIVERSITY

CSE128: SOFTWARE ENGINEERING (1)

KRUSTY KRAB CASHIER SYSTEM

Submitted to:

Prof. Dr. Gamal Abdelshafy

Submitted by:

Abdelrahman Ibrahim ELGhamry	16P3043@eng.asu.edu.eg
Hossam ElDin Khaled Mohmed	16P3025@eng.asu.edu.eg
Abdelrahman Amr Issawi	16P6001@eng.asu.edu.eg
Fares Mohamed Ansary	16P8094@eng.asu.edu.eg
Amr Elsayed Ahmed	16P1116@eng.asu.edu.eg

ABSTRACT

The purpose of this study is to identify the consumer requirements and technical requirements that are part of our project, as applied to “Krusty Krab Cashier System”. The design of the system prepared based on accounting applications that has been built and implemented at the restaurants. Then The study is used as a reference for preparing and implementing the source code of the system.

The study methodology begins with collecting the user functional and non-functional requirements, the constraints on the services or functions offered by the system, the system reaction to particular inputs and how the system should behave in particular situations.

After analyzing all these requirements and specifying the exact user needs, we started the phase of designing the system modules and components by providing a preliminary design to all functional characteristics of the system. Then using it to provide our detailed design and our alternatives to deal with all expected problems.

After finishing the design process, we started the phase of distributing the system modules and components on our team developers, in order to start implementing, coding and testing them.

After testing and verifying that all components are working correctly, we started the phase of integrating these modules together to form our system.

By the end of integrating and testing the system on the required time, the system is delivered to the stakeholders. Where this stage is divided into two phases, the first one is Beta-testing, where a group of expected end-users starts testing the software and shaking out all bugs in the developed software, the second one is maintaining the system by taking care of all the founded bugs and delivering it to the customers.

Table of Contents

List of Figures

1. Introduction

1.1 Purpose

The purpose of this document is to represent all basic software engineering approaches to develop a software system.

This document is written for academic purposes, requested by Dr. Gamal Abdel Shafy.

1.2 List of Definitions

PC: Personal computer

HW: Hardware
SW: Software

PM: Person-Month
A: Organization constant
Size: Number of code lines in kilo (KLOC)
B: Effort reflector
M: Multiplier

RCPX: Product reliability and complexity
RUSE: The reuse required
PDIF: Platform difficulty
PREX: Personnel experience
PERS: Personnel capability
SCED: Required schedule
FCIL: Team support facilities

1.3 Scope

“Krusty Krab Fast Food Cashier System” provides an effortless way for the cashier to add new customers (for delivery service) or search for an old customer by his telephone number, select items, submit order, calculate total, receive payment and calculate change. In addition, the system prints invoice and send orders to the kitchen’s screen to be prepared.

1.4 Overview

In this document, the steps taken to develop “The Krusty Krab Fast Food Cashier System” are all presented in this order: system requirements, use case diagram, Narrative description, requirement validation, class model, state diagram, sequence diagram, detailed class diagram, user interface design, client-object related diagram, detailed design, testing. In addition to cost estimation and user guide.

2. General Description

2.1 Product Perspective

The client may use a kitchen screen system to automatically receive new submitted orders provided by this product, so that the chiefs can prepare the orders more efficiently.

Also, the client may use a printing system to print the invoices provided by this product.

2.2 General Capabilities

This product can handle all typical cashier's tasks, including; searching and adding customers, selecting items from the menus, making take away and delivery orders, calculating total and change amount, supporting credit cards payments, sending order to the kitchen and printing invoices.

2.3 General Constraints

- The system only supports the delivery and take away orders.
- The system needs a touch screen to display orders for the chief.
- The system needs a dot-jet printer for printing invoices.
- The system 's maximum number of different menu items in a single order is eleven.

2.4 User Characteristics

The system is only used by the cashier who selects items and confirms the order.

2.5 Environment Description

The cashier system pc must be connected to the kitchen's touch screen, the printer and the safe to work correctly.

2.6 Assumptions and Dependencies

The system relies on a file to read and store customers' info.

2.7 Other resources needed

The system saves/loads the customers information from "Customers.csv" file, this file is included inside the project folder.

3. System Requirements

3.1 Functional Requirements

1. The system must save customers' information (name, telephone, address) for delivery service.
2. The system must identify old customers by telephone number.
3. The system can add new customers to the saved customers list.
4. The system must contain all menu items.
5. The system must provide a way to select items from the menu.
6. The system must add the selected items to a new order.
7. The system must add the taxes.
8. The system must calculate the total fees.
9. The system must save the order when the cashier confirms it.
10. The system must send the order to the kitchen screen to be prepared.
11. The system must print a receipt to the customer.

3.2 Non-Functional Requirements

1. Develop with Java.
2. Rate of failure occurrence must not exceed 5 times per month.
3. In case of failure, system must restart within 30 minutes.
4. Required training for employees must not exceed 4 hours.
5. Event response time must not exceed 20 seconds.
6. Deliver in a month.
7. The system should conform to all applicable local and international laws.

4. Use-Case Diagram

Figure 1: Use Case Diagram

5. Narrative Description

5.1 Making Order

Use Case name:	Making order.
Related requirements:	Requirements (A1, A2, A3, A4, A5, A5.1, A5.1.1, A6, A7).
Goal in context:	Customer makes an order.
Precondition:	Customer Enters the restaurant.
Successful end condition:	Customer gets the order.
Failed end condition:	Customer doesn't get the order.
Primary Actor:	Cashier.
Secondary Actor:	Customer.
Trigger:	Customer asks the cashier to make a new order.
Include Case:	Select item, Calculate total.
Main Flow:	1. A new or existing customer calls or visits the restaurant. 2. Customer requests a new order.
Extension:	1. The customer didn't have enough money. 2. The system is down.

5.2 Select Item

Use Case name:	Select item.
Related requirements:	None.
Goal in context:	Item is selected.
Precondition:	Customer sees the menu.
Successful end condition:	Customer selects an item.
Failed end condition:	Customer didn't find the item he is checking for.
Primary Actor:	Customer.
Secondary Actor:	Cashier.
Trigger:	Customer asks the cashier for the menu.
Main Flow:	1. Customer checks the menu. 2. Customer selects some items. 3. Cashier adds the items to the order.
Extension:	The customer didn't find the item he wants.

5.3 Calculate Total

Use Case name:	Calculate total.
Related requirements:	None.
Goal in context:	Calculate the total.
Precondition:	Customer selects an item.
Successful end condition:	The total is calculated.
Failed end condition:	None.
Primary Actor:	System.
Secondary Actor:	None.
Trigger:	Item is selected.
Main Flow:	1. The system adds the selected items to calculate the total. 2. The system prints the total on the screen.
Extension:	The system is down.

5.4 Payment

Use case name:	Payment.
Related requirements:	Requirements (A5.1, A5.1.1).
Goal in context:	Customer pays the money.
Precondition:	The total amount is calculated.
Successful end condition:	Customer pays money or provides a credit card.
Failed end condition:	Customer doesn't pay or provide a credit card.
Primary Actor:	Customer.
Secondary Actor:	Cashier.
Trigger:	The system calculates the required amount.
Include Case:	Verify payment.
Main Flow:	1. The cashier tells the customers the total amount. 2. The customer pays in cash or provides a credit card.
Extension:	1. The customer didn't have enough money. 2. System is down.

5.5 Verify Payment

Use Case name:	Verify Payment.
Related requirements:	Requirements (A5.1.1).
Goal in context:	Verify customer's payment.
Precondition:	Customer pays the money.
Successful end condition:	Value of money given is equal to or greater than the total order value, or the credit card is valid.
Failed end condition:	Value of money payed is less than the required total, or the credit card is expired.
Primary Actor:	Cashier.
Secondary Actor:	Customer.
Trigger:	Customer pays for the order.
Include Case:	Calculate Change.
Main Flow:	1. The customer pays the money or provides his credit card. 2. The cashier verifies the money, or the credit card.
Extension:	1. The value of money is less than the required. 2. The credit card machine has an error.

5.6 Calculate Change

Use Case name:	Calculate Change.
Related requirements:	None.
Goal in context:	Change is calculated.
Precondition:	Payment is verified.
Successful end condition:	Customer gets the change.
Failed end condition:	The system calculates wrong amount.
Primary Actor:	System.
Secondary Actor:	None.
Trigger:	The cashier provides the system with the paid money.
Main Flow:	1. The cashier provides the system with the paid money. 2. The system calculates the change and prints it on the screen.
Extension:	1. System is down 2. The cashier enters a wrong amount.

5.7 Confirm Order

Use Case name:	Confirm Order.
Related requirements:	None.
Goal in context:	Confirm the order and send it to the screen.
Precondition:	Payment is verified.
Successful end condition:	Order is confirmed and sent to the screen.
Failed end condition:	Order is not confirmed.
Primary Actor:	Cashier.
Secondary Actor:	System.
Trigger:	Payment is verified.
Main Flow:	1. Payment is verified. 2. Cashier confirms the order.
Extension:	1. The system is down.

5.8 Save Order

Use Case name:	Save Order.
Related requirements:	None.
Goal in context:	The system saves the order.
Precondition:	Cashier confirms the order.
Successful end condition:	Order is saved.
Failed end condition:	Order is not saved.
Primary Actor:	Cashier.
Secondary Actor:	System.
Trigger:	The cashier confirms the order.
Main Flow:	1. The system saves the order.
Extension:	1. The system memory is full. 2. The system is down.

5.9 Print Invoice

Use Case name:	Print Invoice.
Related requirements:	None.
Goal in context:	Customer gets the invoice.
Precondition:	The order is saved.
Successful end condition:	Customer gets the invoice.
Failed end condition:	Customer doesn't get the invoice.
Primary Actor:	System.
Secondary Actor:	Customer.
Trigger:	Cashier confirms the order.
Main Flow:	1. The system prints the invoice. 2. Customer gets the invoice.
Extension:	1. The printer has an error.

5.10 Send Order to Kitchen

Use Case name:	Send Order to Kitchen.
Related requirements:	None.
Goal in context:	The kitchen's screen is updated by a new order.
Precondition:	The new order is saved.
Successful end condition:	The screen is updated.
Failed end condition:	The screen is not updated.
Primary Actor:	System.
Secondary Actor:	Screen.
Trigger:	A new order is saved.
Main Flow:	1. The cashier confirmed a new order. 2. The screen is updated with the new order.
Extension:	1. Connection error.

5.11 Search Old Customer

Use Case name:	Search Old Customer.
Related requirements:	None.
Goal in context:	Find the customer’s profile.
Precondition:	None.
Successful end condition:	Find the customer's data.
Failed end condition:	Didn't find the customer's data.
Primary Actor:	Cashier.
Secondary Actor:	Customer.
Trigger:	The customer is an old one.
Main Flow:	1. The cashier asks the customer for his phone number. 2. The cashier enters the phone number to the system. 3. The system searches for the customer.
Extension:	1. The system is down. 2. Cashier enters a wrong number.

5.12 Add Customer

Use Case name:	Add Customer.
Related requirements:	None.
Goal in context:	Adding a new customer.
Precondition:	Customer is not found the customers’ list.
Successful end condition:	A new customer is added.
Failed end condition:	Customer is not added.
Primary Actor:	Cashier.
Secondary Actor:	Customer.
Trigger:	Customer is not found the customers’ list.
Main Flow:	1. The cashier asks the customer for his name, phone number. and address. 2. The cashier adds the customer’s data.
Extension:	1. The memory is full. 2. The system is down.

6. Data Model

- The system saves/loads the customers information from “Customers.csv” file, this file is included inside the project folder.
- This file contains three attributes: Customer Name, Customers Phone Number and Customer Address.
- These three attributes are the definition of each customer.
- Any manipulation in this file may lead to lose all the customers data.

7. Requirements Validation

Requirements:

Req ID	Requirement
1.1	The system must save customers’ information (name, telephone, address) for delivery service.
1.2	The system must identify old customers by telephone number.
1.3	The system can add new customers to the saved customers list.
2.1	The system must contain all menu items.
2.2	The system must provide a way to select items from the menu.
2.3	The system must add the selected items to a new order.
2.4	The system must calculate the taxes.
2.5	The system must calculate the total fees.
3.1	The system must save the order when the cashier confirms it.
3.2	The system must send the order to the kitchen screen to be prepared.
4.1	The system must print a receipt to the customer.

Requirement traceability:

Req ID	1.1	1.2	1.3	2.1	2.2	2.3	2.4	2.5	3.1	3.2	4.1
1.1											
1.2	D										
1.3	D										
2.1											
2.2				D							
2.3				D							
2.4					D						
2.5					D						
3.1											
3.2									D		
4.1									D		

Figure 2: Requirement Traceability Matrix

8. Class Model

Concise Problem Definition:

Customer makes a delivery or a take away order, selects items from the menu and pays total fees, the cashier confirms the payment, and creates the order on the system, meanwhile the system calculates the fees, send the order to the kitchen and print invoice.

Informal strategy:

A customer requests to make an order either takeaway or delivery, in case of a delivery order the cashier retrieves the customer's data from the system controller, if not found the cashier adds a new customer to the system. The customer chooses the items and the cashier selects the items on system. Meanwhile the system calculates the fees. The customer can either pay with a credit card in that case the cashier verifies it, or in cash and in this case the cashier enters the paid amount to the system to calculate the change. The system saves the order and send it to the kitchen's screen, also the system should use the printer to print the invoice.

Formalized strategy:

A customer requests to make an order either takeaway or delivery, in case of a delivery order the cashier retrieves the customer's data from the system controller, if not found the cashier adds a new customer to the system. The customer chooses the items and the cashier selects the items on system. Meanwhile the system calculates the fees. The customer can either pay with a credit card in that case the cashier verifies it, or in cash and in this case the cashier enters the paid amount to the system to calculate the change. The system saves the order and send it to the kitchen's screen, also the system should use the printer to print the invoice.

The Nouns That Can Be Identified:

Customer, Order, Item, Cashier, Controller, System, Cash, Credit card, Kitchen's screen, Invoice, Data, Delivery, Printer and Take away.

Candidate Classes:

Customer, Order, Item, Controller, Kitchen's screen, Delivery.

Class Model:

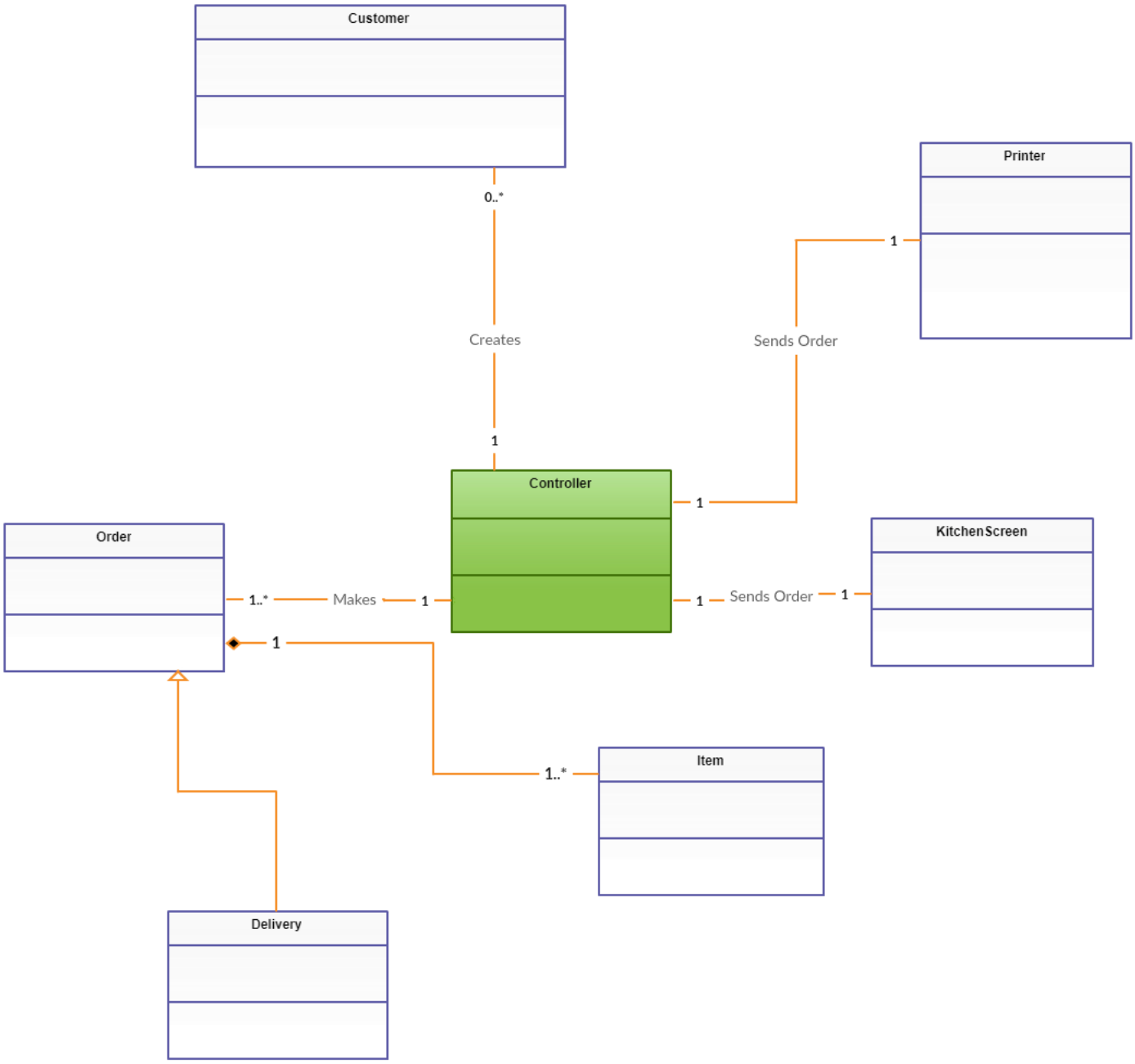

Figure 3: Class Model

9. State Diagram

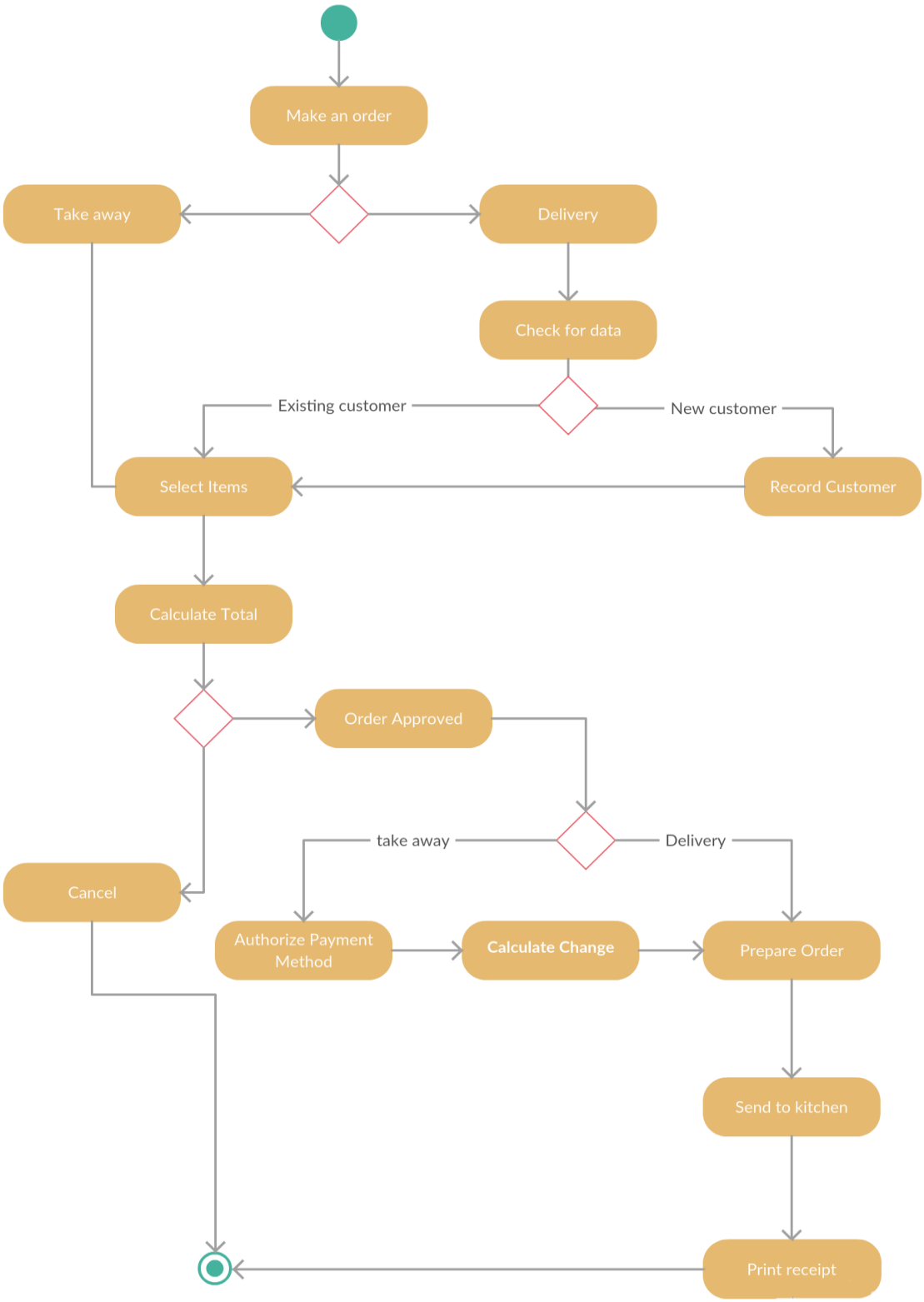

Figure 4: State Diagram

10. Sequence Diagram

10.1 Takeaway Paying in Cash

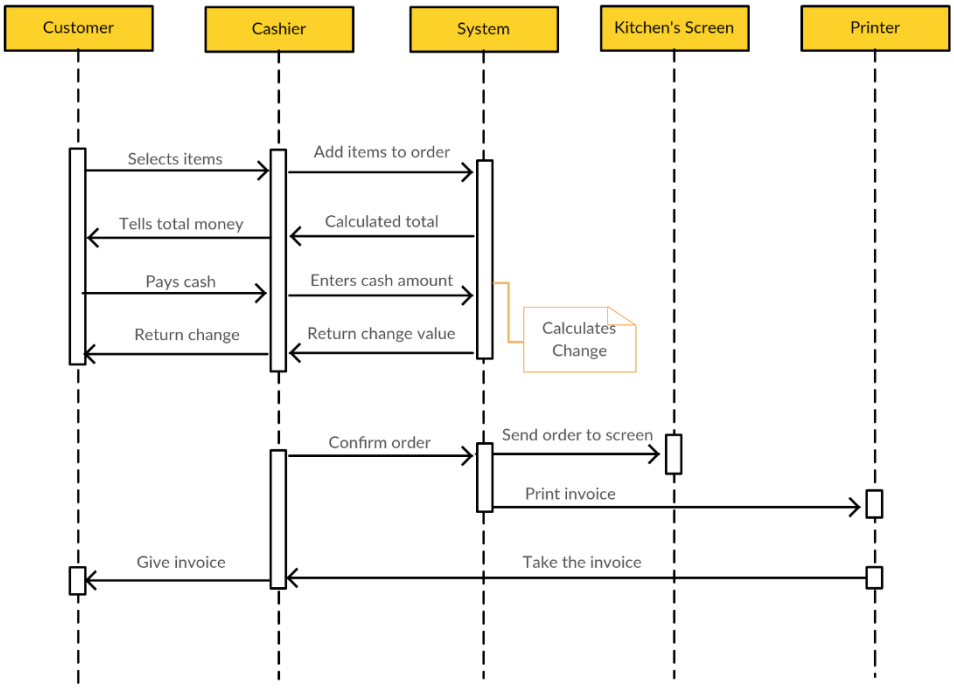

Figure 5: Sequence Diagram 1

10.2 Takeaway Paying with Credit Card

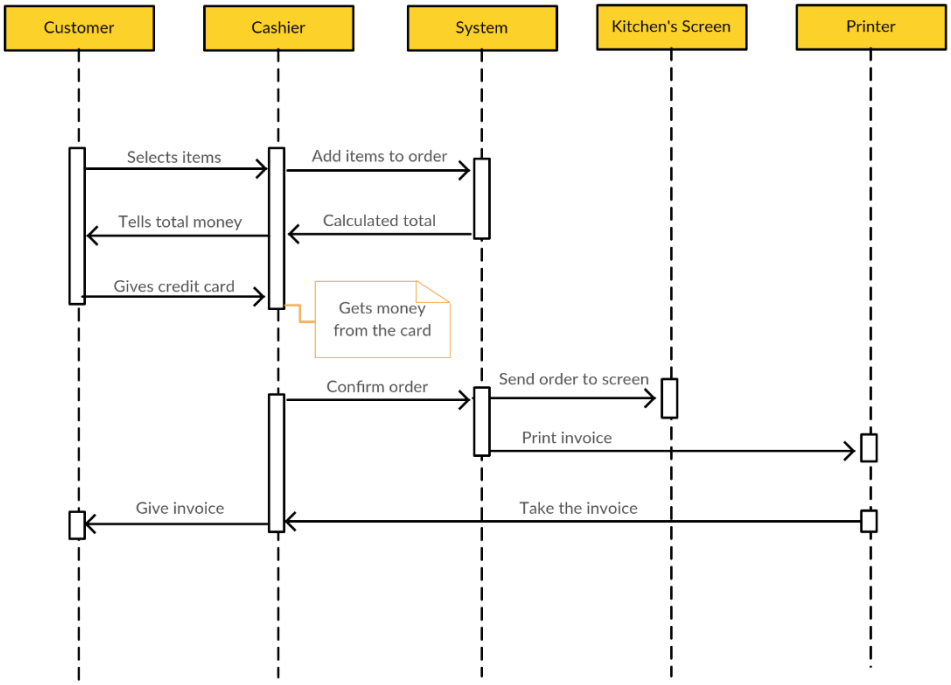

Figure 6: Sequence Diagram 2

10.3 Delivery Old Customer

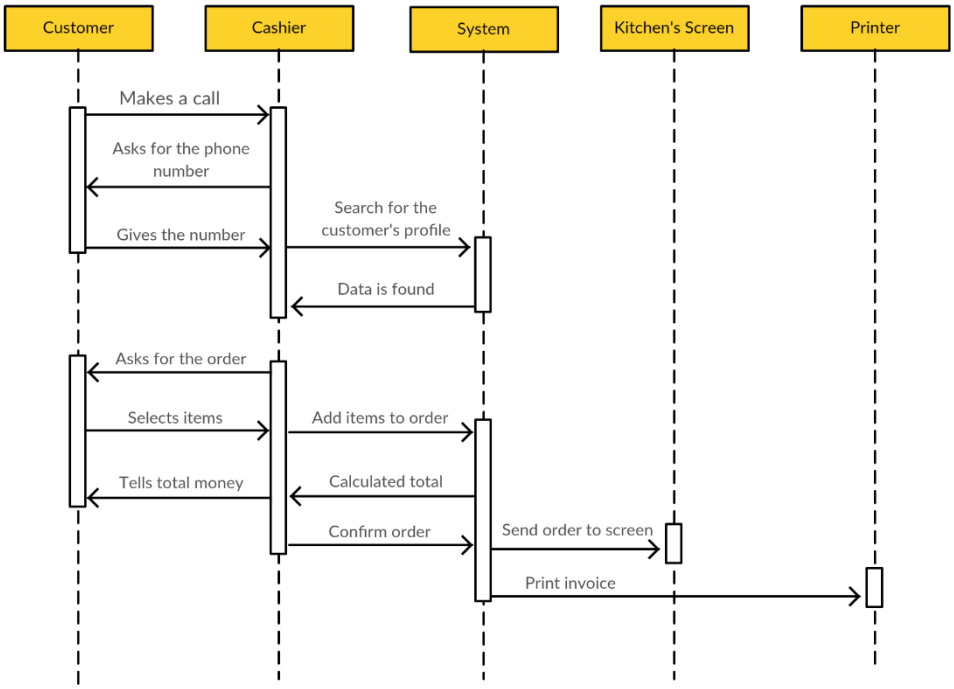

Figure 7: Sequence Diagram 3

10.4 Delivery New Customer

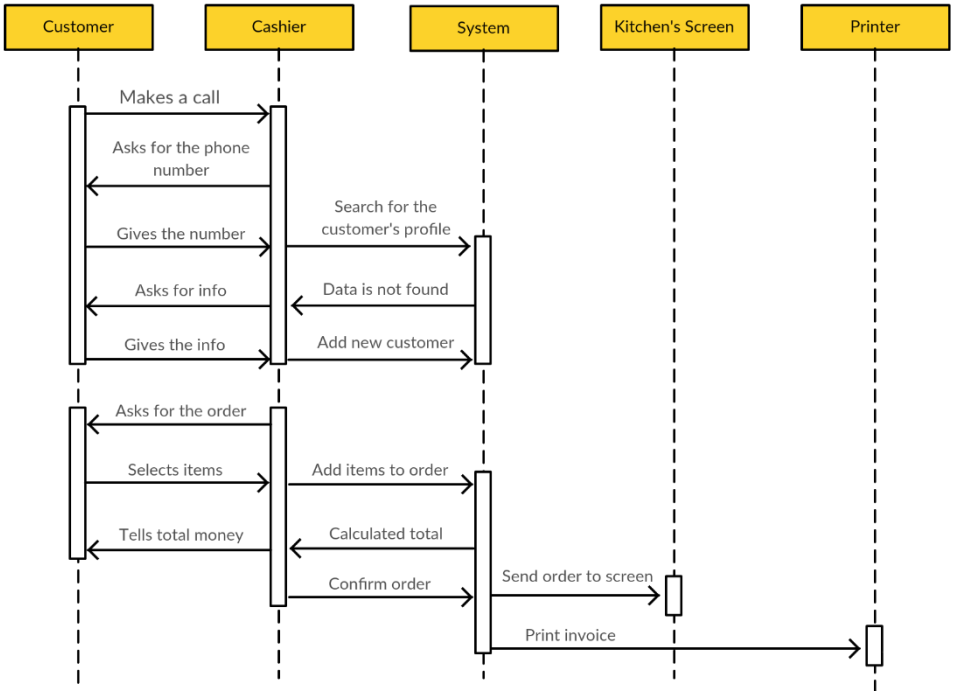

Figure 8: Sequence Diagram 4

11. Detailed Class Diagram

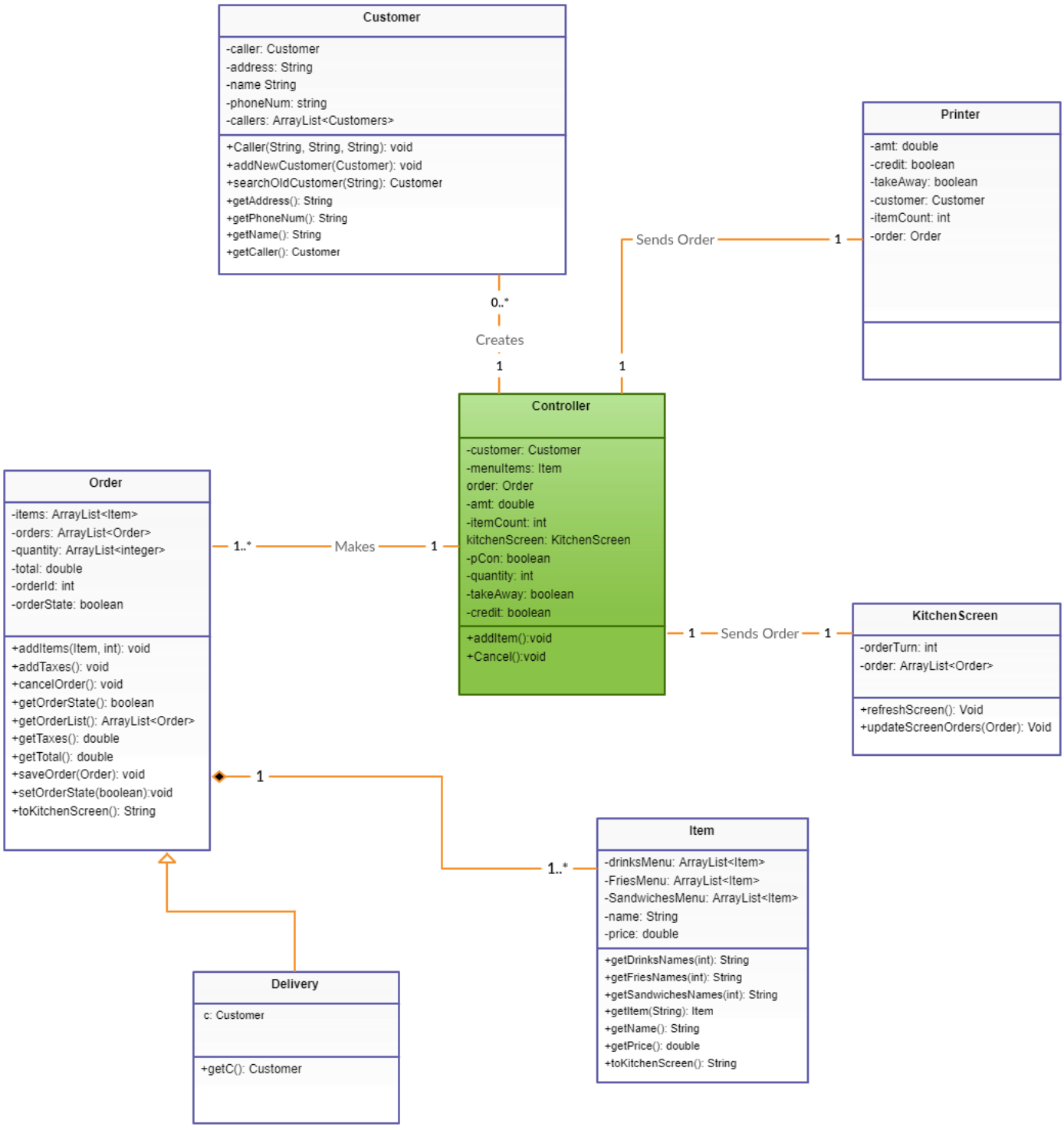

Figure 9: Detailed Class Diagram

User-Defined Data Types:

1. Customer:

- Attributes: name(String), phoneNumber(String), address(String), caller(reference to the recent caller) and callers(ArrayList holding references for all customers).
- Methods: addNewCustomer(Customer) void function that adds the new customer to data file.
  searchOldCustomer(String) searches for the user in the data file and returns a reference to him.
  Caller(String,String,String) void function that sets our caller's data.
  getAddress() String function that returns the customer's address.
  getPhoneNum() String function the returns the customer's phone number.
  getName() String function that returns the customer's name.
  getCaller() Customer function that returns the customer.

2. Order:
   - Attributes: total(double), orderId(int), orderState(boolean), quantity(ArrayList of integers), orders(ArrayList holding references for all orders), and items(ArrayList holding references for all items).
   - Methods: addItems(Item,int) void function that adds any quantity of an item to the list of items.
     addTaxes() void function that adds the taxes value to the total price.
     cancelOrder() void function that cancels the order.
     GetOrderState() boolean function returns whether the order is done or not yet.
     GetOrderList() function that returns list of orders.
     GetTaxes() double function that returns the taxes value.
     GetTotal() double function that return the total.
     SaveOrder(order) void function that saves the order to the list.
     SetOrderState(boolean) void function that changes the order state.
     ToKitchenScreen() String function that returns order description (all items)..
3. Item:
   - Attributes: price(double), name(String), drinkMenu(ArrayList holding references for all items), friesMenu(ArrayList holding references for all items), and sandwichesMenu(ArrayList holding references for all items).
   - Methods:
     GetdrinksNames(int) String function returns the name of the specified drink.
     GetFriesNames(int) String function returns the name of the specified fries.
     GetSandwichesNames(int) String function returns the name of the specified Sandwich.
     getItem(String) String function the returns a specified item.
     getName() String function that returns the item's name.
     getPrice() double function that returns the item price.
     ToKitchenScreen() String function that returns item description (quantity and name).

4. Kitchen Screen:
   - Attributes: orderTurn(int), orders(ArrayList holding references for all orders).
   - Methods:

RefreshScreen() void function that refreshes the screen to display the added or removed orders.
UpdateScreenOrders(order) void function that adds order to the screen.

5. Delivery:
    - Attributes: C(Customer).
    - Methods:
      GetC() Function that returns a customer.

6. Printer:
    - Attributes: amt(double), credit(boolean), takeaway(boolean), customer(Customer), itemCount(int), order(Order).

7. Controller:
    - Attributes: customer(Customer), menuItems(Item), order(Order), itemCount(int), amt(double), PCon(boolean), quantity(int), kitchenScreen(KitchenScreen), credit(boolean), takeaway(boolean).
    - Methods: Additem() void function that adds item to the list.
      Cancel() void function that cancels an order.

12. User Interface Design

Takeaway

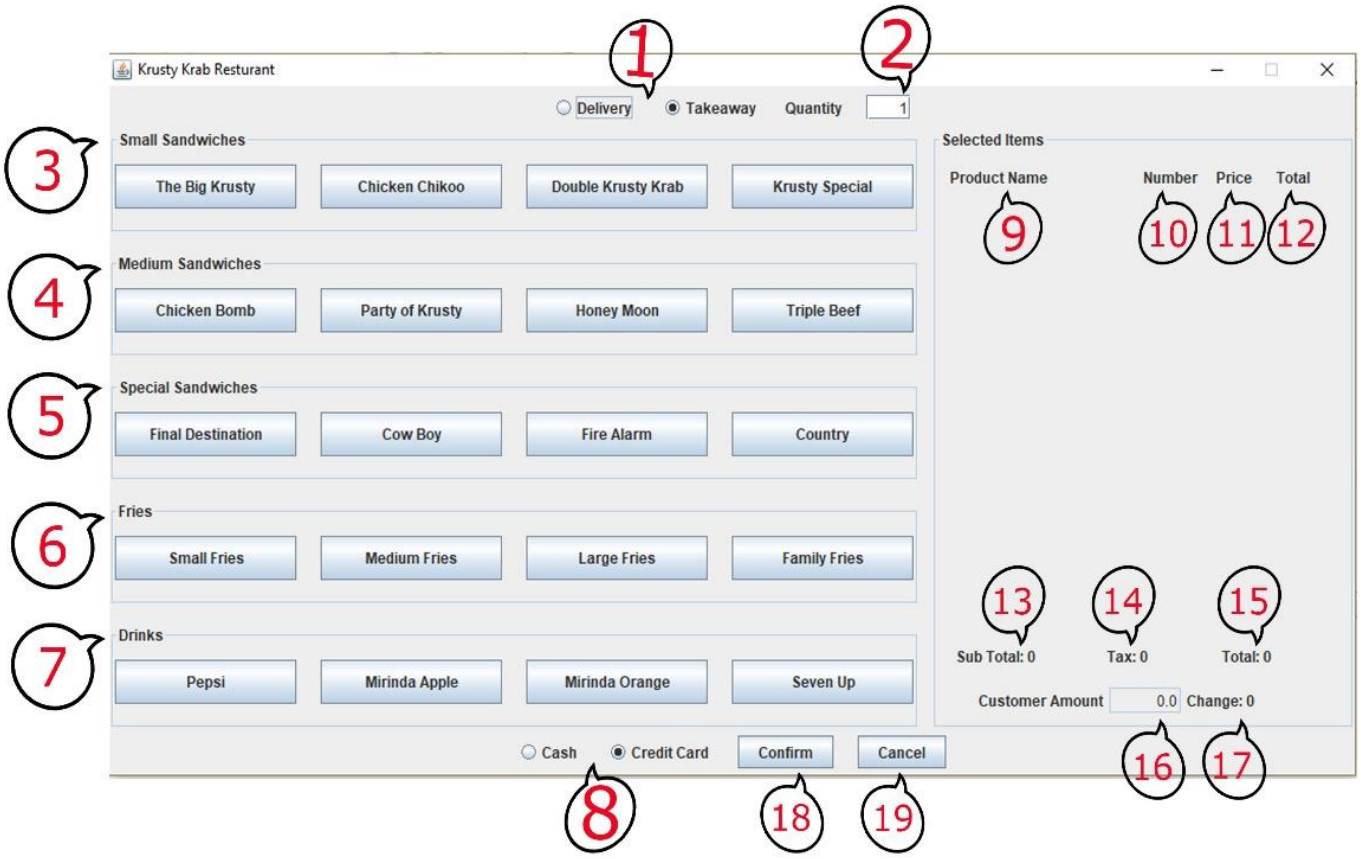

Figure 10: User Interface Main Window

1. Takeaway order type (1) is initially selected, changing the order type to delivery is covered in the next section.
2. The quantity (2) is initially set to one.
3. The menu items are divided into five groups; Small Sandwiches (3), Medium Sandwiches (4), Special Sandwiches (5), Fries (6) and Drinks (7).
4. If the user wants to set the quantity of any item, he should set it before selecting any menu item.
5. The user can only select 11 menu items regardless of their quantities, if the user tries to add the 12th item the system will pop the message shown and the user must click “OK” to continue.

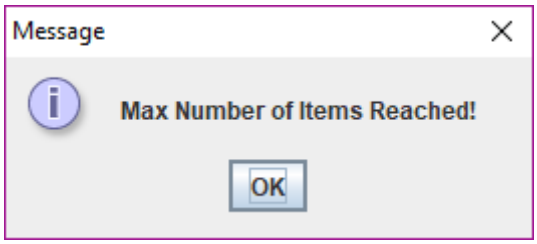

Figure 11: User Interface Message 1

6. Once an item is selected the system add it in the selected items section. Item's name (9), number (10), price (11) and total price (12) are also added. The system automatically updates the order's sub total (13), tax (14) and total (15).
7. Credit Card payment type (8) is initially selected, customer amount (16) is disabled.
8. If the user wants to the change payment to cash, he must click on Credit Card or Cash (8).
9. Once payment type is changed to cash customer amount (16) is now enabled, and the user must enter a number equal to or greater than the total (15), then press the keyboard "Enter" button.
10. The system calculates the change, change (17) is now updated.
11. If the user tries to enter a number less than the total (15), the system will pop the message shown and the user must click "OK".

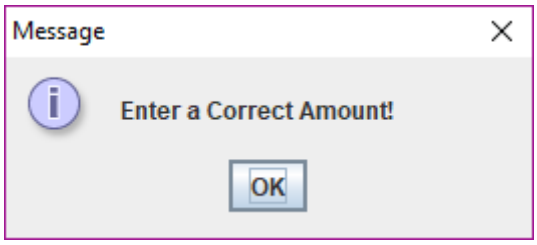

Figure 12: User Interface Message 2

12. To confirm the order, the user must click on Confirm (18), if the payment type is cash the user must enter a correct number in Customer Amount (16) and press the keyboard "Enter" key first, otherwise the system will pop the message shown and the user must click "OK".
13. If the user made any mistake, he can click on Cancel (19) and the system will go back to its initial state.

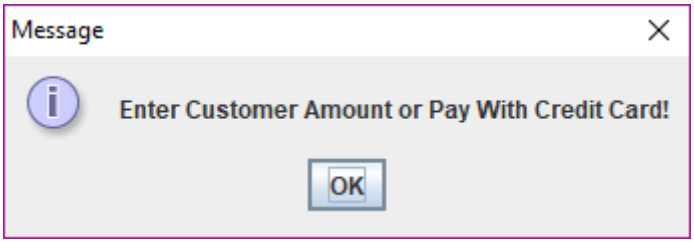

Figure 13: User Interface Message 3

14. When the order is successfully confirmed the system automatically sends the order to kitchen's screen and prints the invoice.

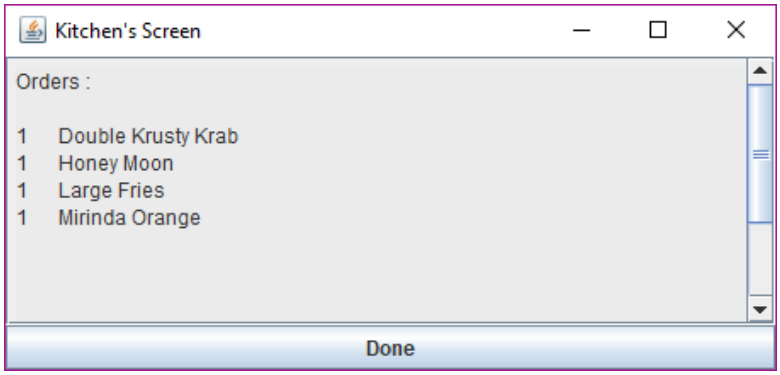

Figure 14: User Interface Kitchen's Screen

Figure 15: User Interface Invoice

15. When the order is done the user must click on the Kitchen's screen button (Done), to remove the order from the kitchen's screen.

Delivery

1. If the user wants to change the order type to delivery he must click on Delivery or Takeaway (1). Once delivery is selected the system will automatically open "Select Customer" window shown below.

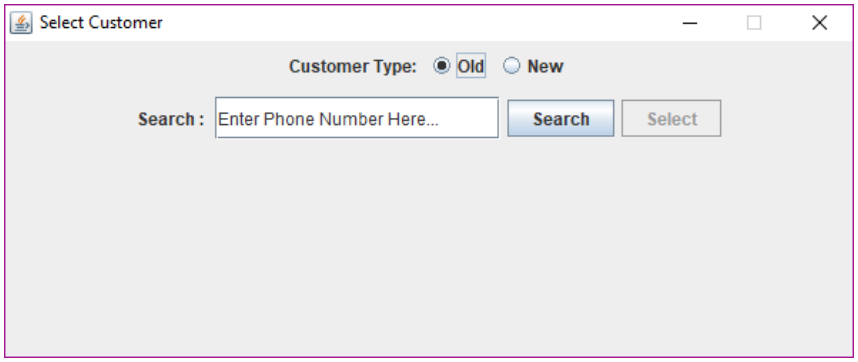

Figure 16: User Interface Select Customer -Old

2. Searching for an Old Customer is initially selected, the user must enter 11 number in "Enter Phone Number Here…" then click on the Search button.
3. If the system finds the customer, the system will show the customer's name and select button is now enabled so the user can select the customer. Otherwise, the system will show "Customer Not Found".
4. If the user wants to add a new customer, he must click on "New" and the system will provide other fields to enter the customer's data as shown below.
5. The user should enter customer's name, phone number and address, then click on "OK". The system will add the new customer to the customers list and automatically selects the new customer. If the user fails to enter any of the mentioned data and click "OK", the system will not proceed and will show "Enter valid data" message at the bottom.

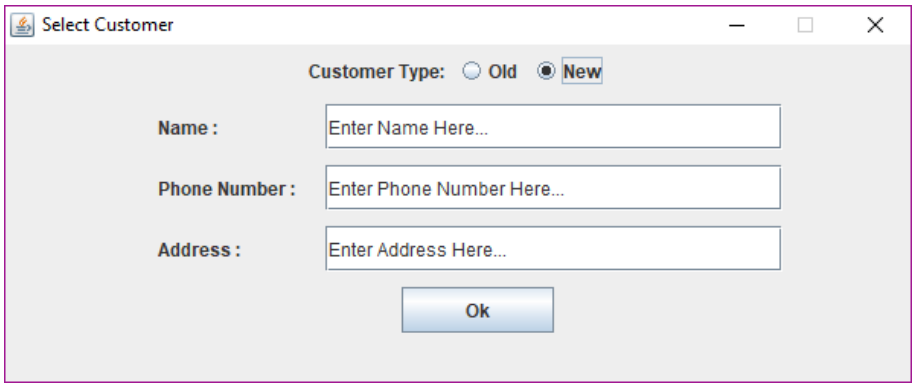

Figure 17: User Interface Select Customer -New

6. Once a customer is selected, the user can close this window and continue making the order. If the user closes this window without selecting or adding a new customer, the system will consider a customer with none values in its fields.

13. Client-Object Relation Diagram

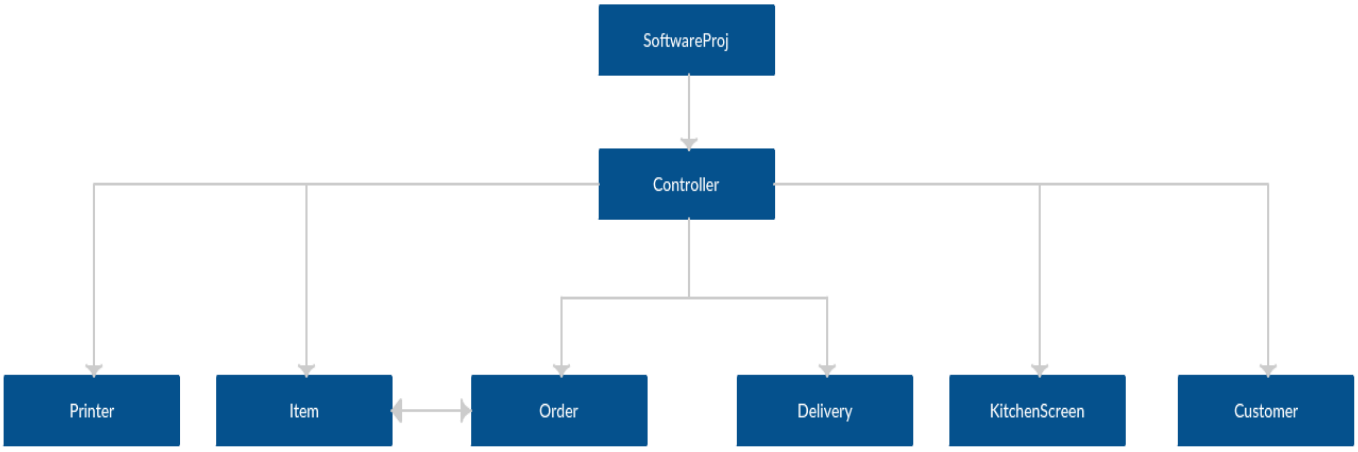

Figure 18 Client-Object Diagram

14. Detailed Diagram

```
IF delivery THEN
   GET data FROM Customer
   IF not exist THEN
      GET Customer's information
    DO order
    IF order is confirmed THEN
       save order to system, send to Kitchen's screen and print invoice
ELSE IF take Away THEN
       DO order
       IF cash THEN
          DO verify
          IF there is change THEN
             return change to customer
          IF verified and order confirmed THEN
              save order to system, send to Kitchen's screen and print invoice
       ELSE IF credit THEN
             DO verify
             IF verified and order confirmed THEN
                 save order to system, send to Kitchen's screen and print invoice
END IF
```

15. Testing

Test Case 1

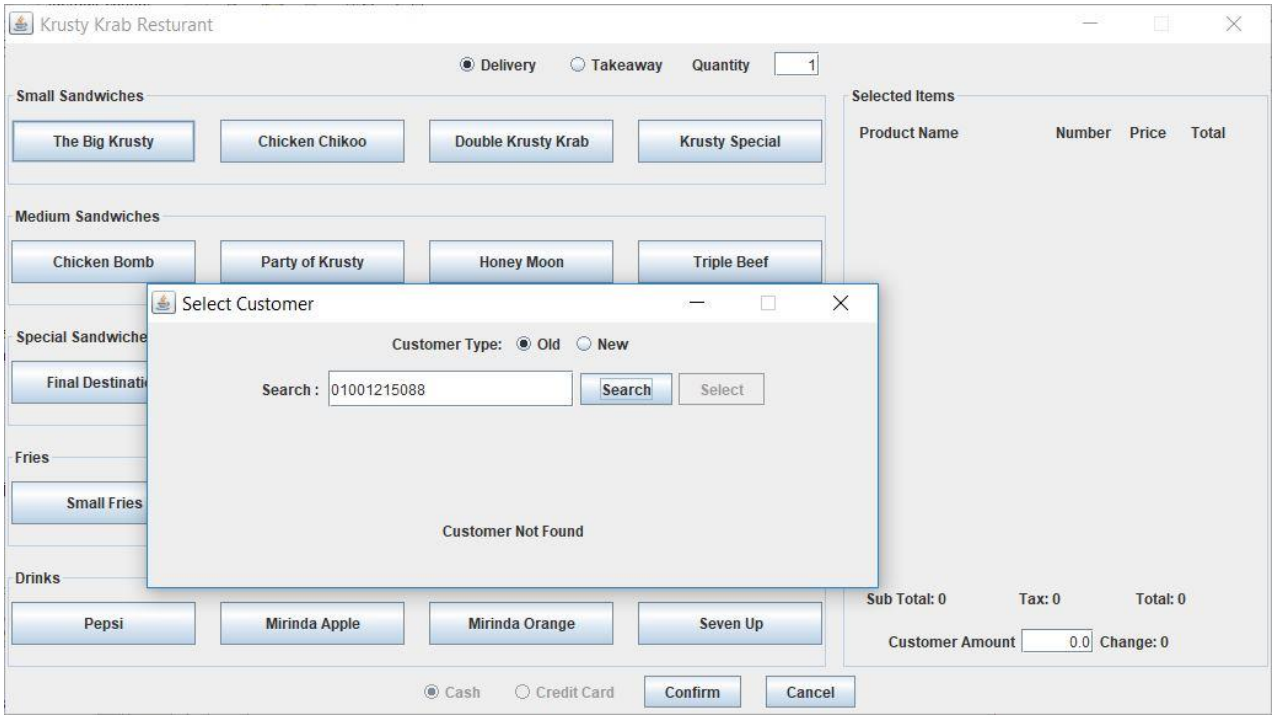

Figure 19: Test Case 1.1

Making a delivery order and the caller phone number is not found.
While searching for a non-existing customer, the system gives a message to the cashier that this customer is not found, and the Select button is disabled.
However, if the cashier wants to proceed by closing the customer selection window, the system behaves normally, the invoice is printed successfully and the invoice (Name, Number, Address) fields are “None”.
capThe order is also sent to the kitchen and kitchen’s screen is updated with the new order.

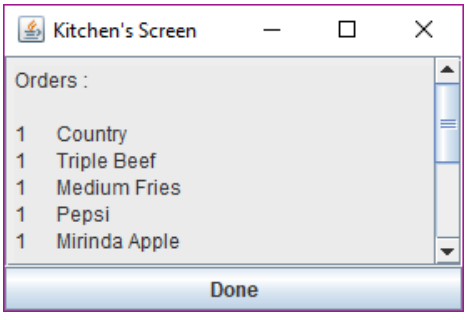

Figure 21: Test Case 1.3

Figure 20: Test Case 1.2

Test Case 2

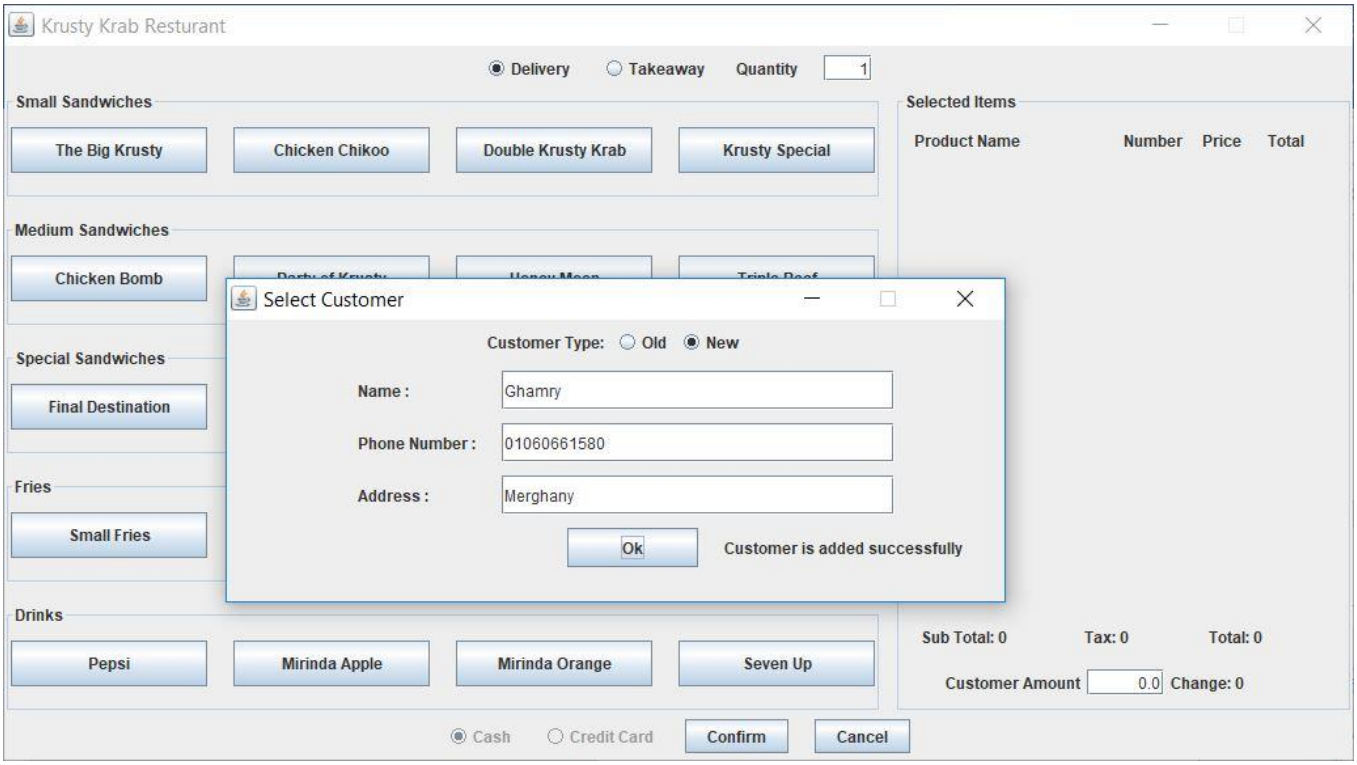

Figure 22: Test Case 2.1

Making a delivery order by adding a new customer to the system.

While making a delivery order for a new customer, the cashier enters the new customer's information and press the Ok button.

After the items are selected and the order is confirmed, the order is sent to the printer and an invoice is printed, containing the new customers information.

The order is also sent to the kitchen and kitchen's screen is updated with the new order.

Figure 23: Test Case 2.2

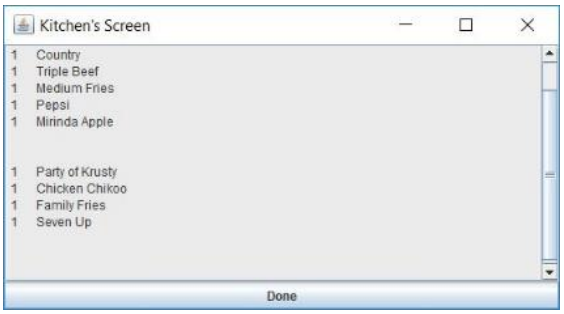

Figure 24: Test Case 2.3

Test Case 3

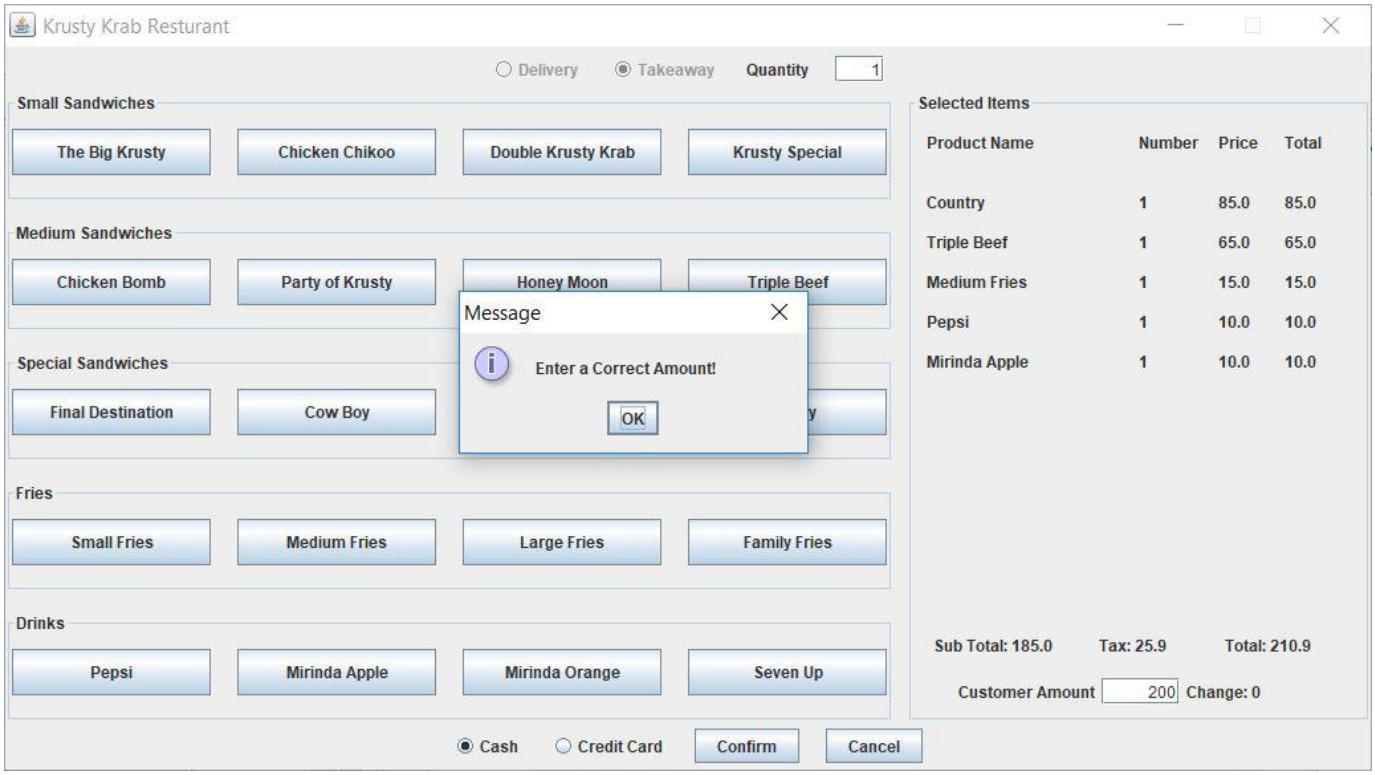

Figure 25: Test Case 3

Making a takeaway order and paying less than the total order amount.

While making a takeaway order, the cashier entered the customer paid amount less than the order total price.

The system stops the order-making procedure and returns a warning message.

The system will not confirm the order, till the cashier enters a correct amount.

Test Case 4

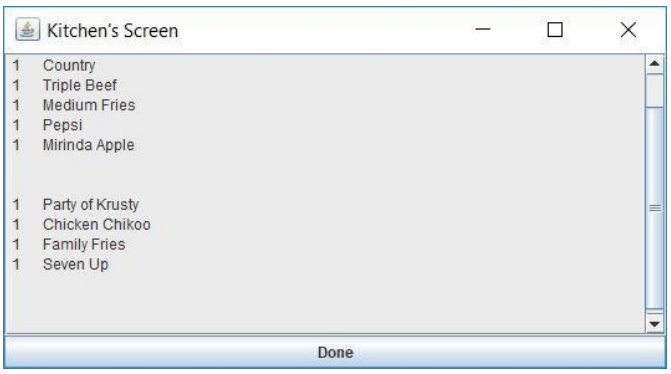

Figure 26: Test Case 4.1

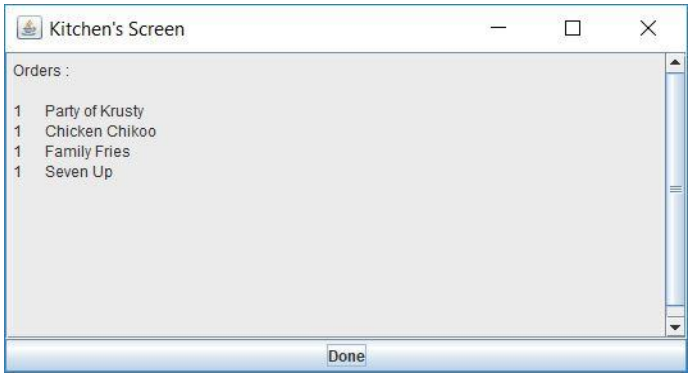

Figure 27: Test Case 4.2

Using the done button to remove from the kitchen's screen an order that has been successfully prepared.

When the chief press the done button, the order that is successfully prepared disappears from the kitchen screen.

Test Case 5

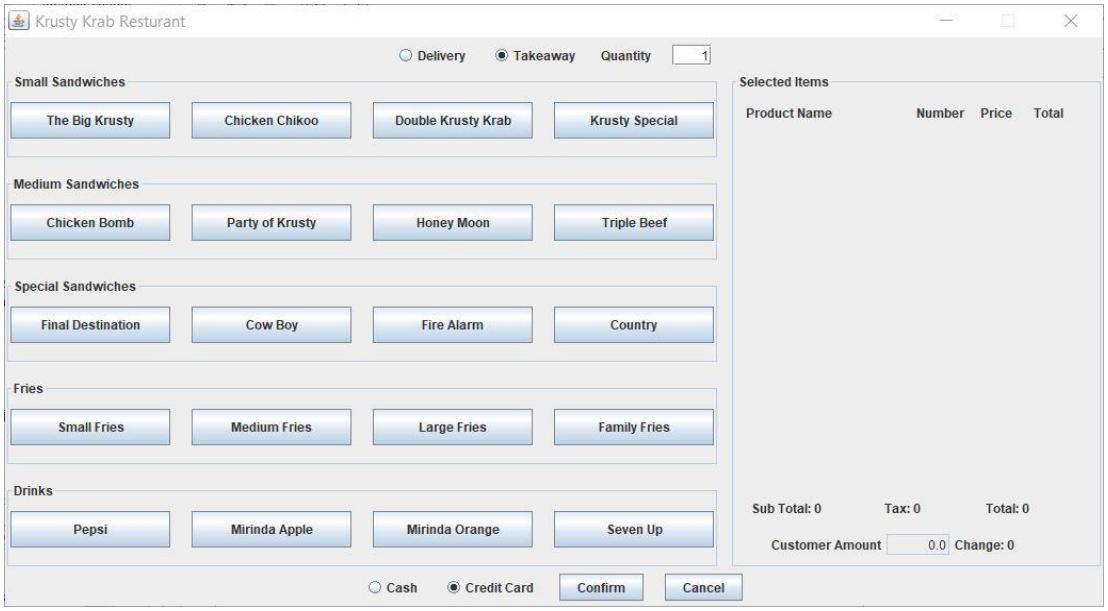

Figure 26: Test Case 5

When no items are selected, and the Confirm button is pressed accidently.

The system state remains as it is, waiting for the cashier to add items.

Also, we note than when Credit Card is selected, the customer amount area is disabled.

16. Estimated Project Cost

According to the COCOMO II

M = PERS × RCPX × RUSE × PDIF × PREX × FCIL × SCED

PERS=1 RCPX= 1.1 RUSE= 0.92 PDIF= 1 PREX= 1.2
FCIL=1 SCED= 1.15

M = 1 x 1.1 x 0.9 x 1 x 1.2 x 1 x 1.15 = 1.396

PM = A × Size^{B} × M

A= 2.94 Size= 1.158 B= 1.15 M= 1.396

PM = 2.94 x $1.158^{1.15}$ × 1.396 = 4.858 Person-Month

Components Contribution

HW:
5 laptops x 15,000 = 75,000LE

SW:
Microsoft Visio 2016=4500LE
Adobe Photoshop CC= 1000/month = 3000LE

Salaries + Social & Insurance:
10,000 x 5=50,000LE

Building rent + electricity:
(5000+700) x 1 =5,700LE

Total estimated cost = 138,200LE

17. User Guide

17.1 Main Frame

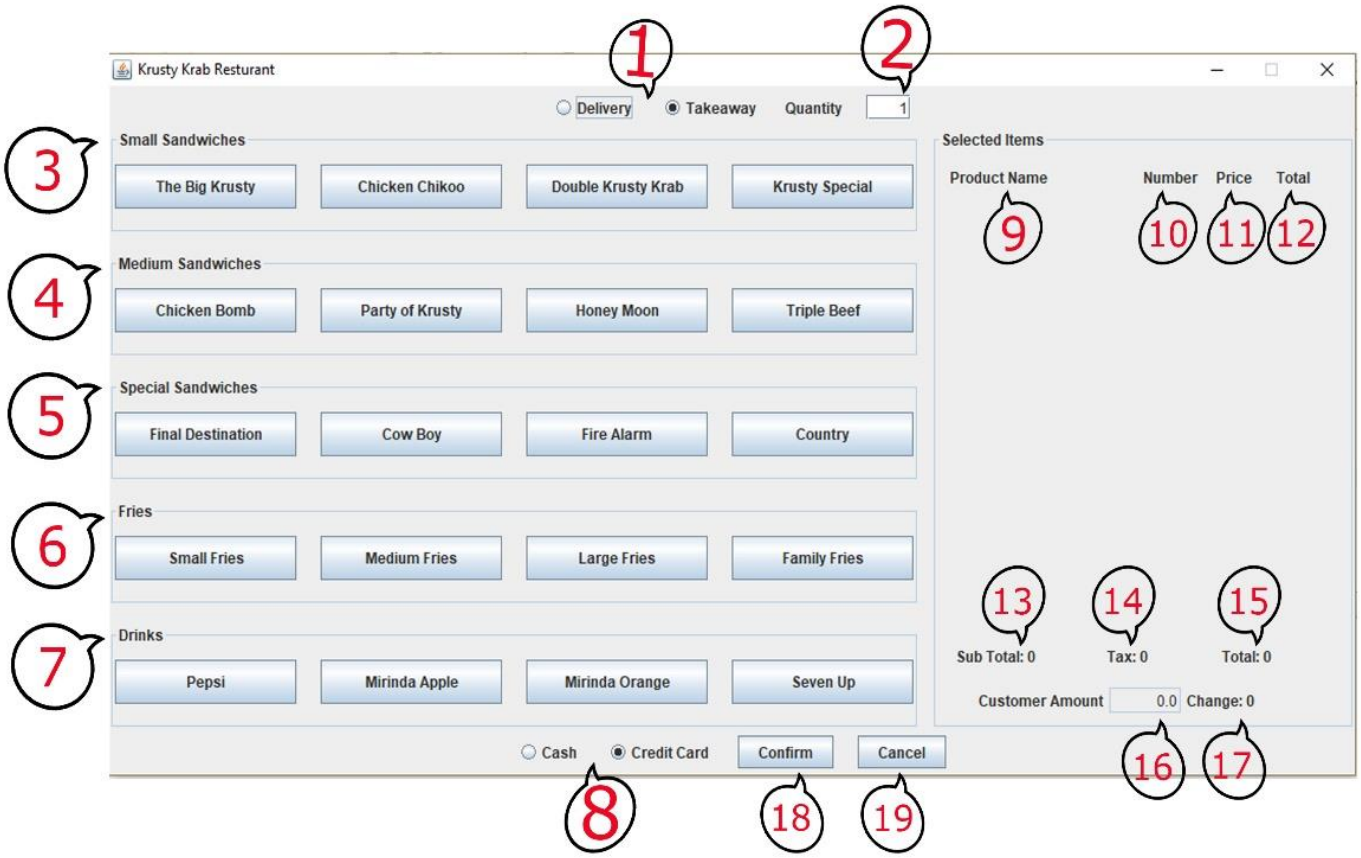

Figure 27: User Guide -Main Frame

1. Choose the order type.
2. Choose the item quantity.
3. Choose from small sandwiches.
4. Choose from medium sandwiches.
5. Choose from special sandwiches.
6. Choose the fries size.
7. Choose the drink.
8. Payment method.
9. The product name.
10. The product quantity.
11. The product price.
12. Total price (quantity*price).
13. Price without taxes.
14. Taxes value.
15. Total price after adding extra fees.
16. Customer paid money value.
17. Change required.
18. Confirm the order.
19. Cancel the order.

17.2 Old Customer Frame

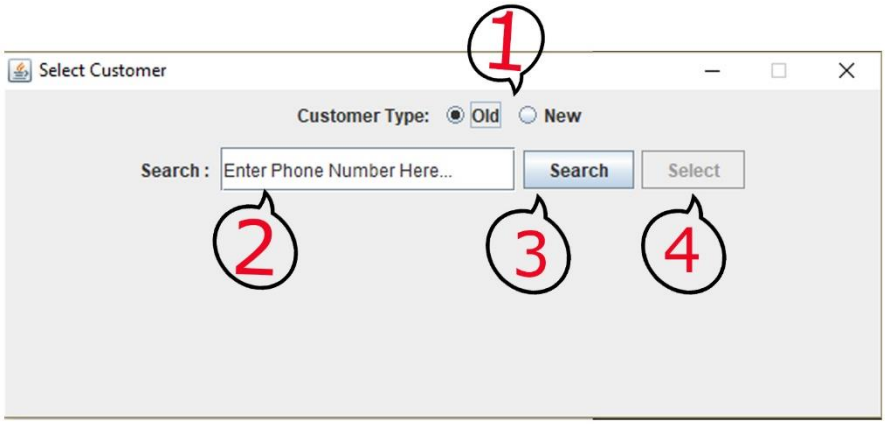

Figure 28: User Guide -Old Customer Frame

1. Choose whether the customer is old or new (old is selected).
2. Enter the customer phone number.
3. Search for the number (the text area only accepts numbers).
4. Select the customer number (if exists).

17.3 New Customer Frame

Figure 29: User Guide -New Customer Frame

1. Choose whether the customer is old or new (new is selected).
2. Enter the customer name.
3. Enter the customer phone number (the text area only accepts numbers).
4. Enter the customer address.
5. Save the customer data.

17.4 Kitchen Screen Frame

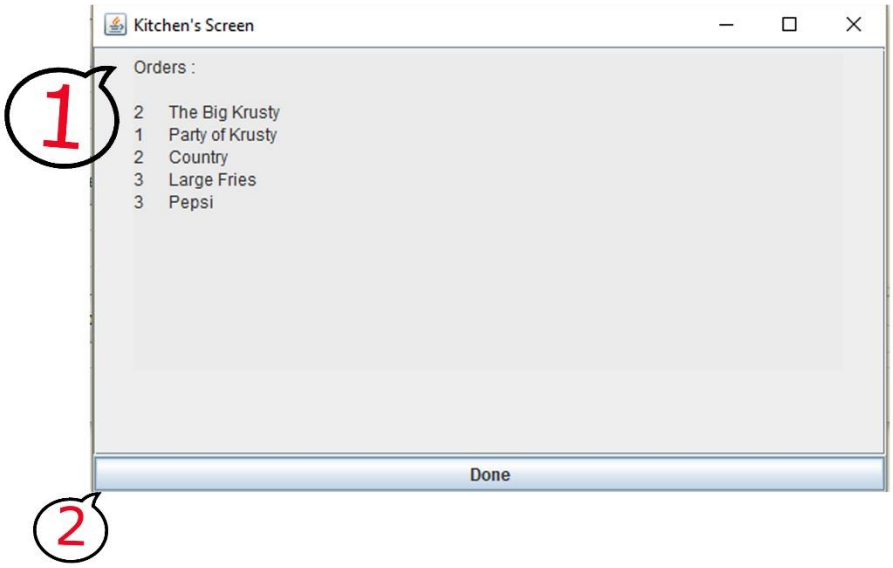

Figure 30: User Guide -Kitchen's Screen.

1. Confirmed orders.
2. Clicked when order is finished (remove the oldest order added from the screen).

17.5 Invoice Frame

1. Order ID.
2. Order type (delivery or takeaway).
3. The product name.
4. The product quantity.
5. The product price.
6. Total price (quantity*price).
7. Price without taxes.
8. Taxes value.
9. Total price after adding taxes.
10. Payment method (cash or credit).

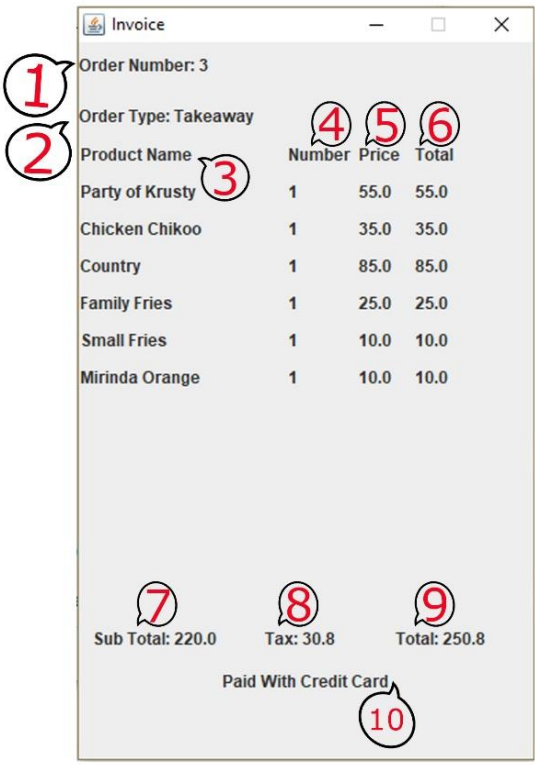

Figure 31: User Guide -Invoice