DEPARTMENT OF INFORMATION SYSTEMS

SYSTEMS DESIGN & DEVELOPMENT

SYSTEMS SPECIFICATION FOR POPPEL

TEAM MEMBERS

Student Number:DHLWAN001	*Student Name: Wandile Dhlamini*
Student Number:MTXTON001	*Student Name : Tonny Mtae*

PLAGIARISM DECLARATION

1. We know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. This Systems Specification is our own work.
3. We have not allowed, and will not allow, anyone to copy our work with the intention of passing it off as their own work.

Full Name: _____ Signed:_____ Date: ____/____/2016

Full Name: _____ Signed:_____ Date: ____/____/2016

TABLE OF CONTENTS

1. INTRODUCTION

1.1. OVERVIEW OF SPECIFICATION

The purpose of this this document is the outline the system specification on the improvement of the Poppel ordering system. We are going to improve the existing ordering system by adding some functionality on the system so that it can solve problems and meet the user requirements. Below are the steps that give a big picture of we are going to go about in meeting the user's requirements.

- Introduction
- User interface design
- Design sequence diagrams
- Design class diagrams
- ER diagrams
- Report designs
- Input and output standards and controls
- Implementation plan and
- Test cases

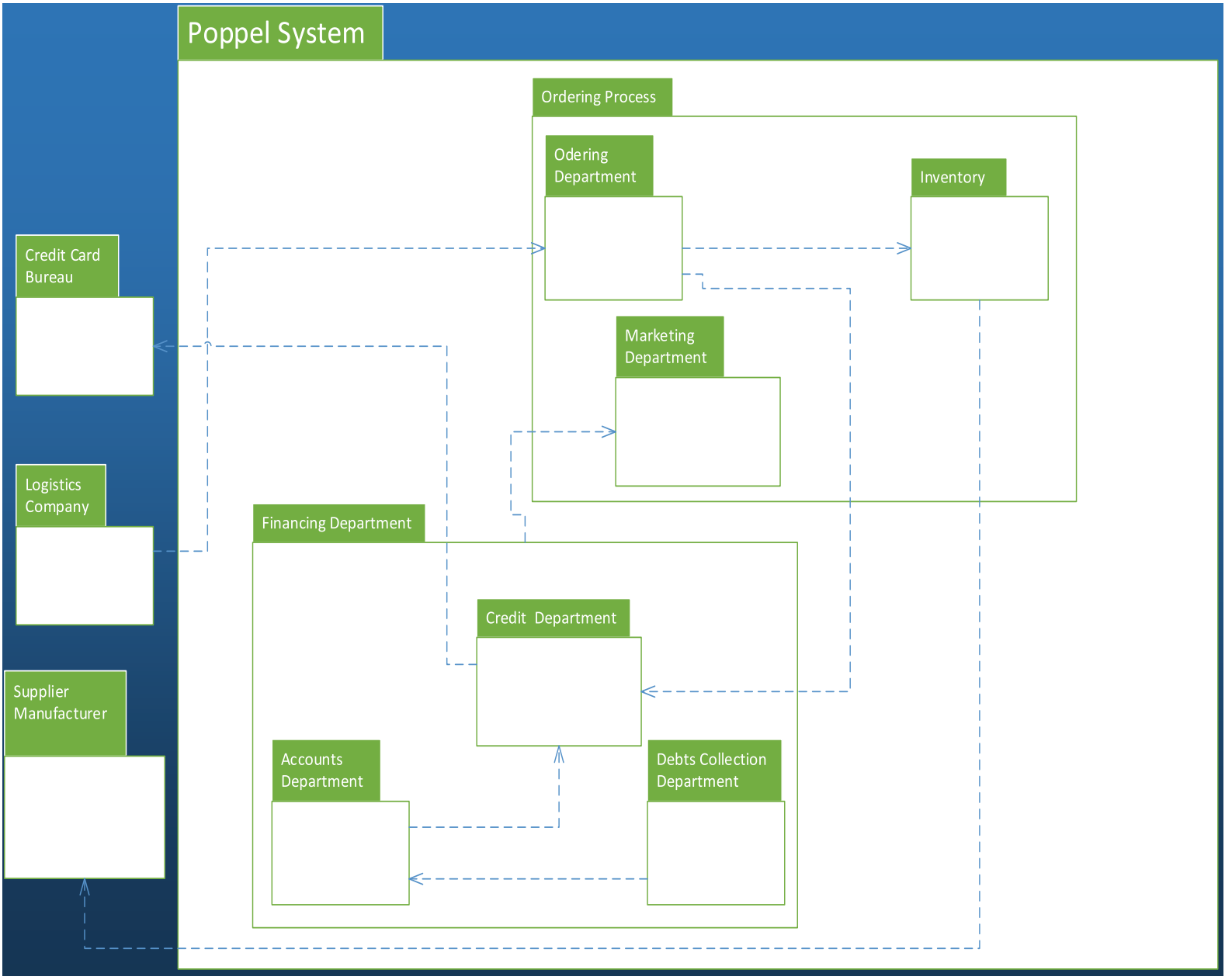
The reason for this changes in the system is because of the inefficiency ordering system that Poppel has currently.

1.2. CONTEXT & SCOPE OF SYSTEM SPECIFICATION

Poppel is a manufacturer and importer of confectionary and soft drinks and is based in Western Cape. The organisation has developed consistently in the previous years. However the current systems that the business is operating with are outdated. Some of the problems rising of this outdated system include:

- Decline in warehouse stock level and time consumption in capturing the stock into warehouse.
- Other problems include products incorrectly exposed to the wrong customers in the Product maintenance division.
- Ordering process consuming time in Order Entry division.
- In data capture and invoice generation division, orders are incorrectly captured resulting in inaccurate delivery note sent to customer. The orders are given code and this causes confusion. When picking orders, the order list is not put in categories resulting in some missing order items.

In this project we address some the issues mentioned above. We design a system that will create an order, do a stock count, update everything (stock, customer credit status), generate a list of expired products and also allows for the creation of a new customer if a customer is not registered in the system



Ordering subsystem being our focus, the above package diagram shows how this subsystem interacts with other subsystems in the Poppel system. The ordering system is being re-assessed inorder to boost the number of orders in the organisation.

1.3. DESIGN ASSUMPTIONS & CONSTRAINTS

The following constraints and assumptions were made when designing the system:

Constraints

- The user cannot record the product to the data source using the system.
- The user cannot delete a customer from the system.
- An employee cannot be recorded to the system.

Assumptions

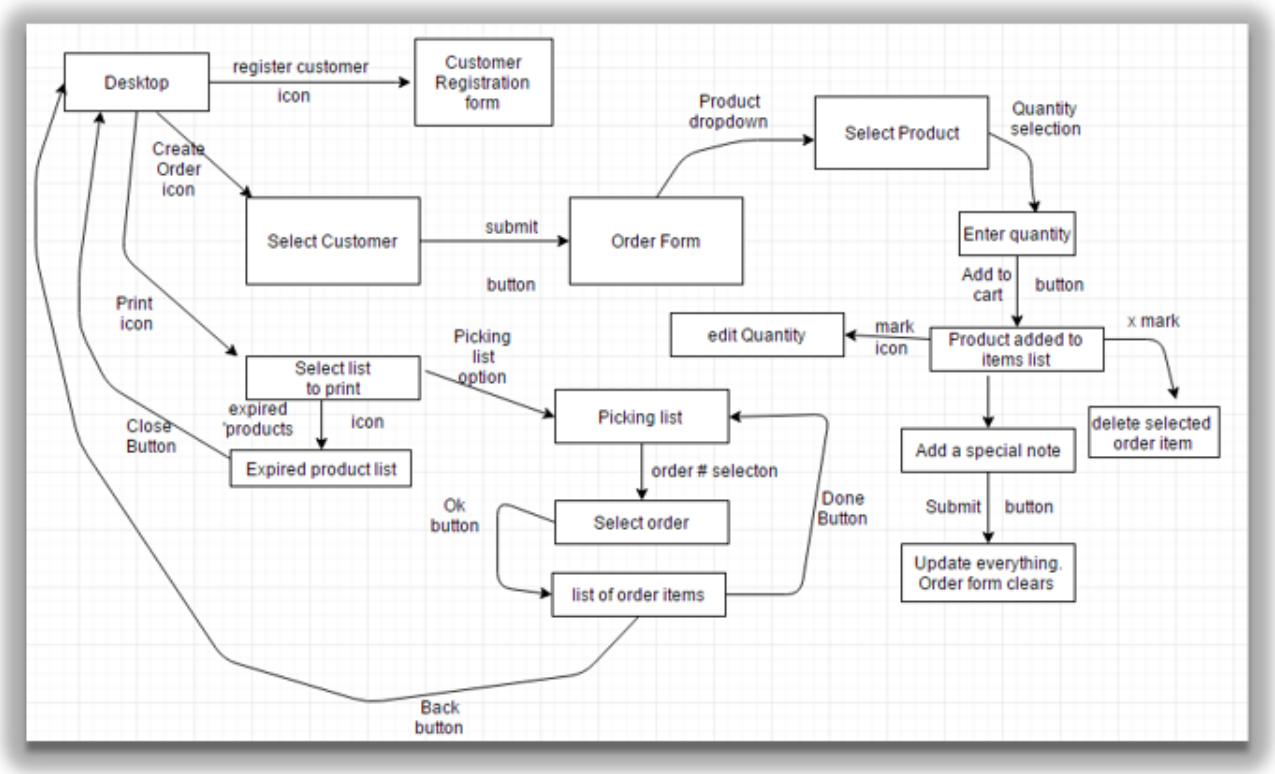
- A customer may only place an order once he/she is registered.
- There are only two types of users of the system (tele seller and -he picking clerk).
- Once an order has been placed, it cannot be cancelled.

2. USER INTERFACE & DIALOGUE DESIGN

In this section we are going to design the user interface that the user will interacting with. This is done by using wireframes. It is necessary for the wireframes to look neat and simple so that the user can get a big picture of what the upcoming system looks like. Graphic interface with too much information can confuse the users and very high chances it might fail to meet the user requirements. Below are some of designs that give a big picture of what the system will look like.

2.1. INTERFACE FLOW DIAGRAM

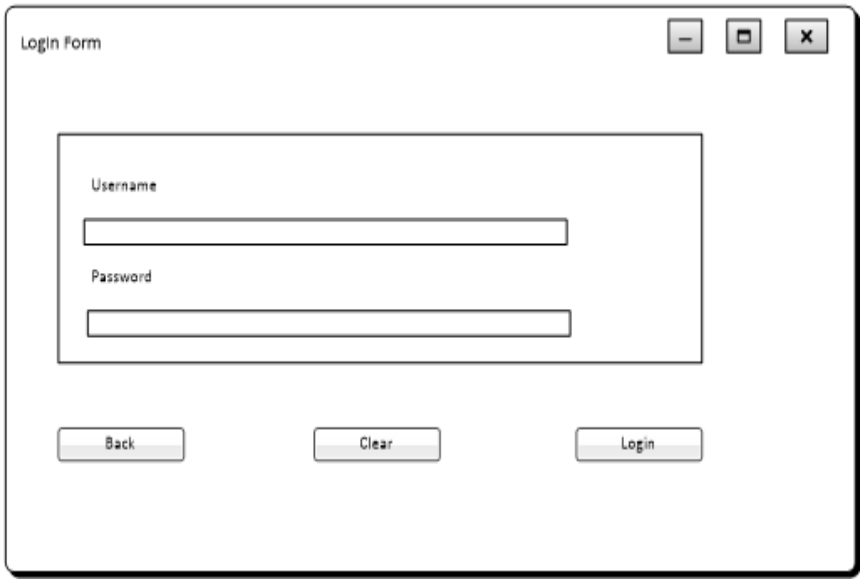
The interface flow diagram below show the sequence of activities when creating customer order and registering customer use cases. Moreover, it depicts how the interaction between the system and the user as well as the expected responses from the system.



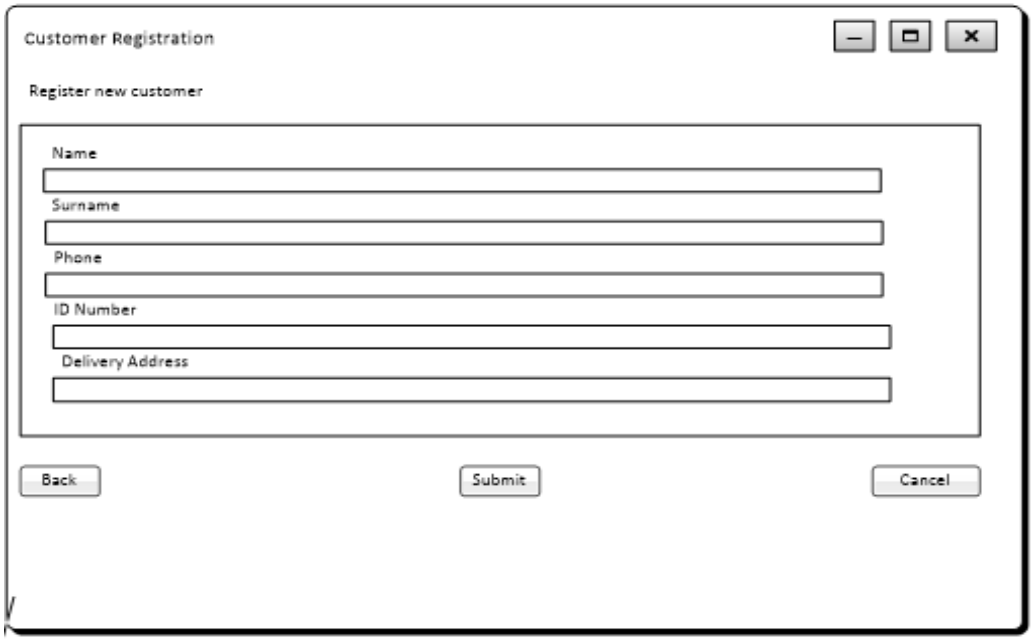
2.2. **WIREFRAME DIAGRAMS**

The wireframe diagrams below define the content and functionality of the Poppel System. These wireframe diagrams are useful for conveying the general structure and contents that are required for the Poppel system.

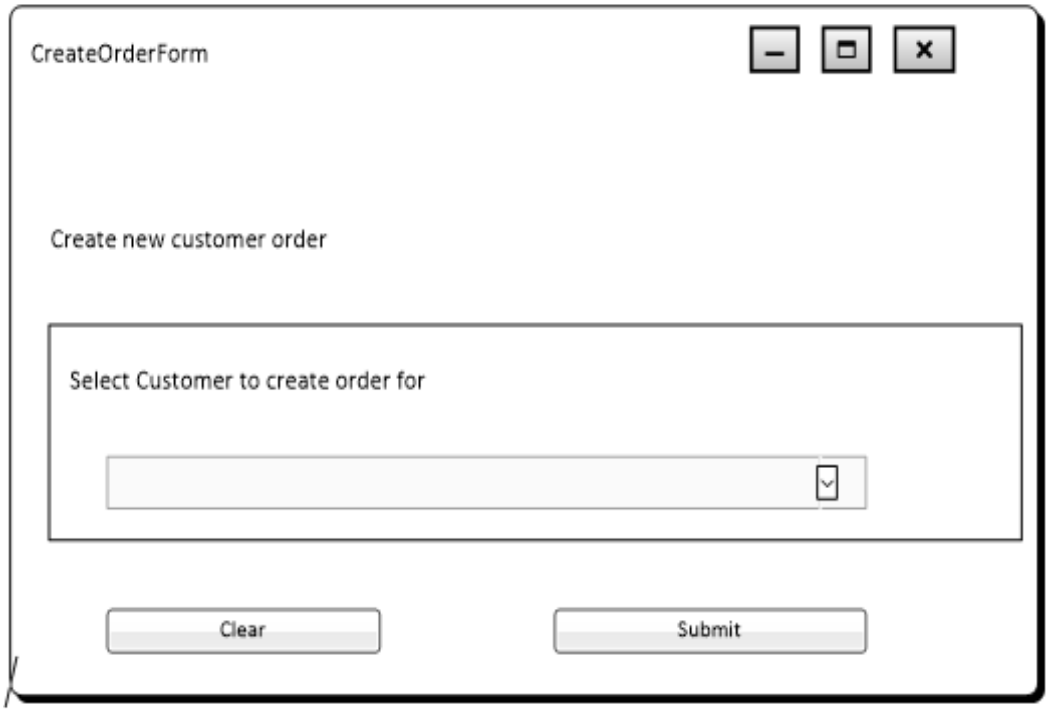
This is the login form of the system. It required a username and a password before anyone can login to the system.



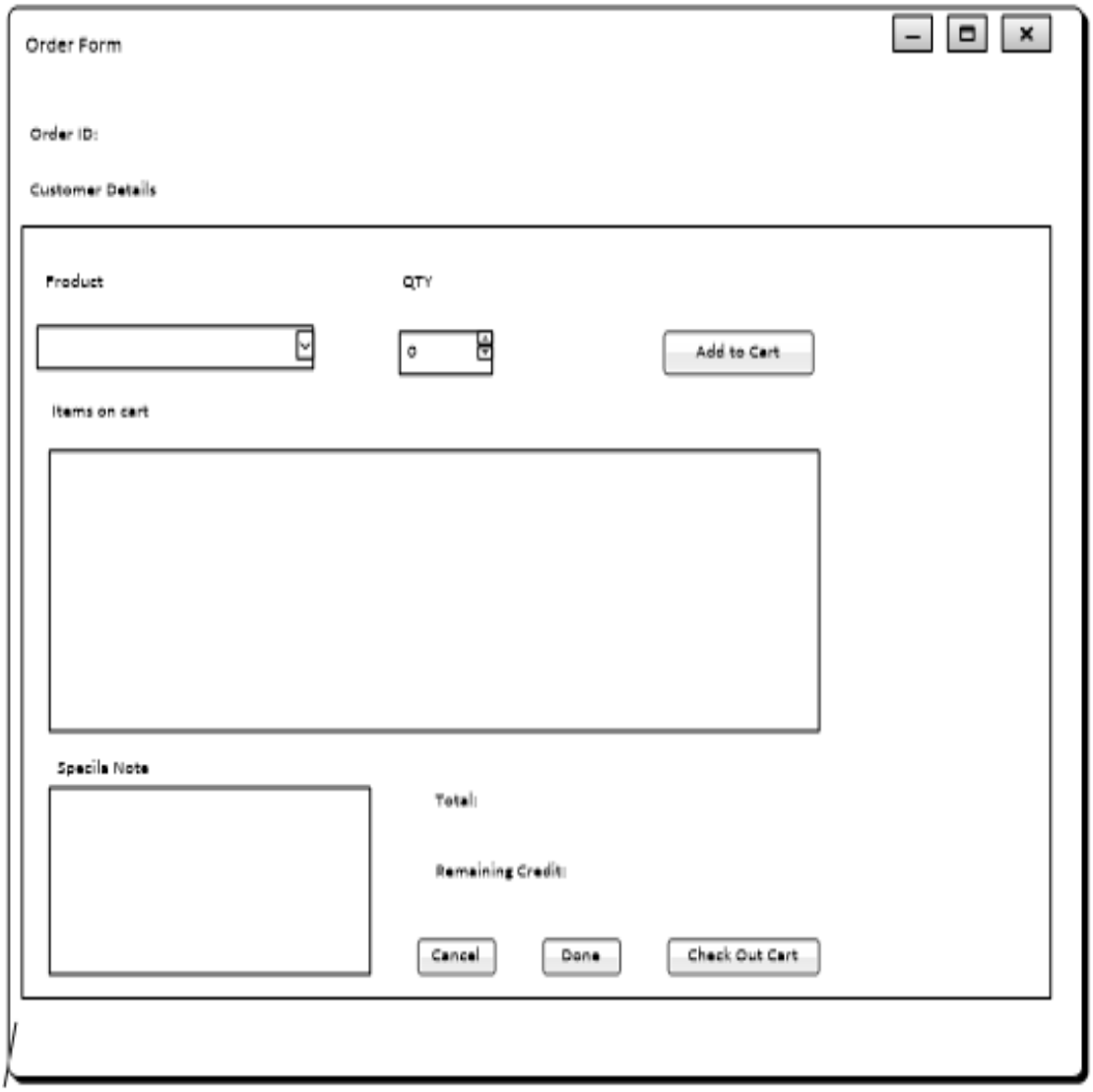
This is the customer registration form. The user of the system will enter a customer's details and then register the customer. After registering they click on the submit button in which a message box will show notifying them that their details are submitted correctly.



This is the create order form. An order can only be created for a registered customer

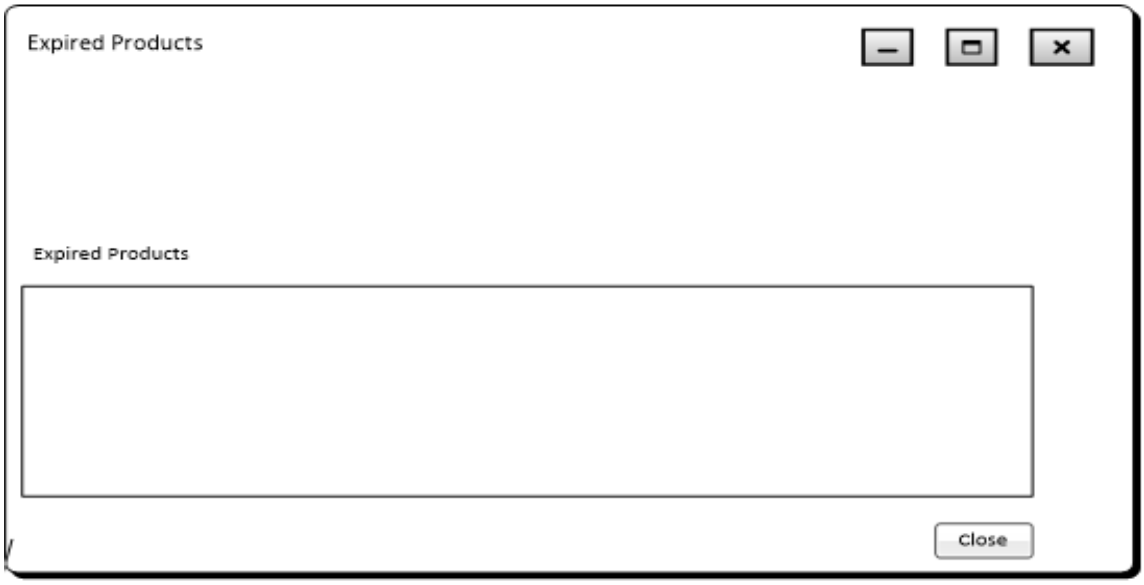


The order form will allow a user to select an item, the quantity of the item and then add to their order cart. A user can cancel an order item or increase or decrease the quantity of an item they have ordered.



The Picking List will shows the orders that have been made and that need to be delivered. After each list is picked, it is removed from the list. This will ensure that orders are not mixed and that there are no shortages in each order.

The expired products form will show a list of all expired products from the products table. This will help management make better purchasing decisions before buying certain products



2.3. SCREEN STANDARDS

Our screen standards take into consideration the following;

- Enabling shortcuts to users.
- Enable easy reverse of actions.
- Reducing short-term memory load.
- Giving informative feedback.
- Providing consistency.
- Provide error handling tool.

2.4. DETAILED SCREEN LAYOUT

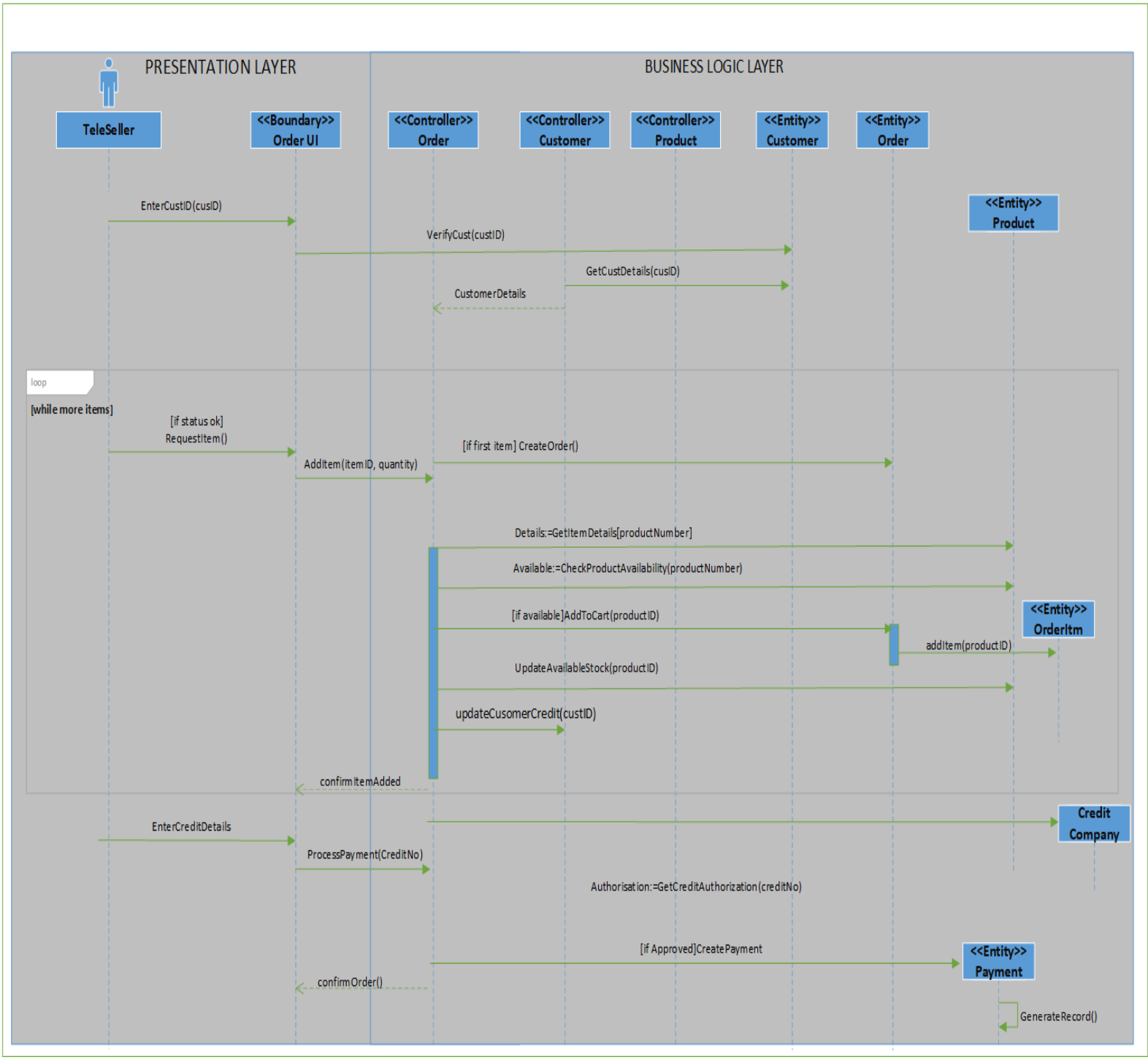
The screenshots below show the final Poppel system. They bring life to the wireframes explained above.

- This is the home page of the system. The user logs in as either a picking clerk or as tele-seller.
  Screen shot 1
- This is the customer registration form. The tele-seller will enter the customer details and submit them to the customer database. A customer will then get a customer number which is generated automatically by the system.
  Screen shot 2
- This is the form where the tele-seller will search for a customer before the customer can place an order.
  Screen shot 3
- This is the place an order form. It allows for tele-seller to create an order for a customer. It also allows for the option of adding a special note to each order that is being placed.
  Screen shot 4
- This form allows for the picking clerk to select whether they want to generate a picking list or an expired products list.
  Screen shot 5

3. DESIGN SEQUENCE DIAGRAMS

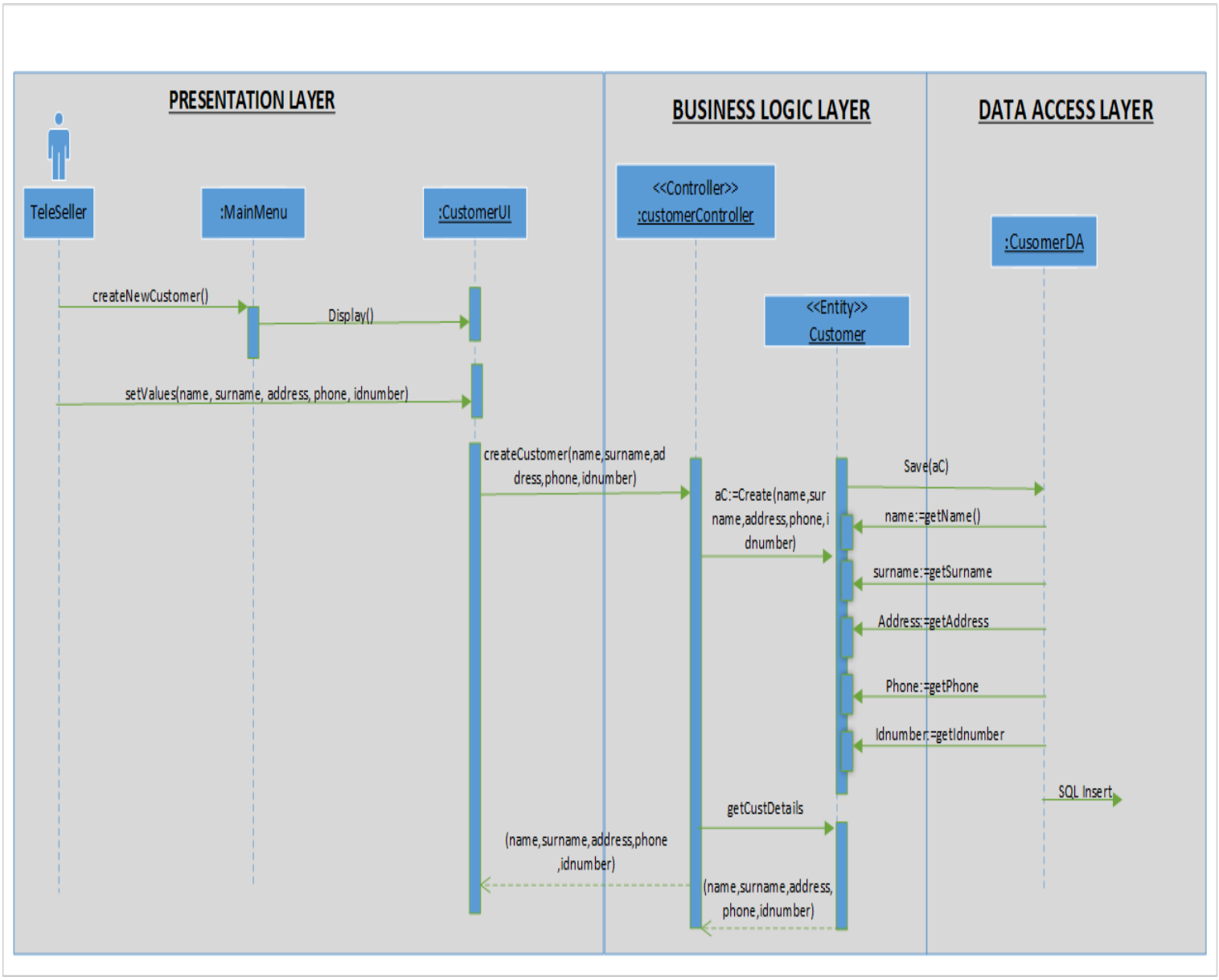
3.1. Order Creation Sequence Diagram

This sequence diagram depicts interaction between the system and tele-seller when he/she is placing an order. First, the tele-seller first enters the customer's ID into the system, thus the Order user interface. Secondly, the user interface sends the customer ID to the order controller class to check if the customer does exists. Then controller directs the task for checking the existence of the customer to the database. If the customer exists it requests for customer details and return them to the tele-seller. If the customer has a good status with the business, the clerks then initiates an order. The order controller direct the message to the order entity class to set up an order. The picking clerk enters a product ID, to add to order, the order controller first checks on the product DB if that particular product is available and it adds to cart if the product is available. This process loops until there is more item to add. Finally the system notifies the tele-seller of the order that has successfully being made.


PRESENTATION LAYER
BUSINESS LOGIC LAYER
TeleSeller
<<Boundary>> Order UI
<<Controller>> Order
<<Controller>> Customer
<<Controller>> Product
<<Entity>> Customer
<<Entity>> Order
<<Entity>> Product
EnterCustID(cusID)
VerifyCust(custID)
GetCustDetails(cusID)
CustomerDetails
loop
[while more items]
[if status ok] RequestItem()
AddItem(itemID, quantity)
[if first item] CreateOrder()
Details:=GetItemDetails[productNumber]
Available:=CheckProductAvailability(productNumber)
[if available]AddToCart(productID)
<<Entity>> OrderItm
addItem(productID)
UpdateAvailableStock(productID)
updateCusomerCredit(custID)
confirmItemAdded
EnterCreditDetails
Credit Company
ProcessPayment(CreditNo)
Authorisation:=GetCreditAuthorization(creditNo)
[if Approved]CreatePayment
<<Entity>> Payment
confirmOrder()
GenerateRecord()

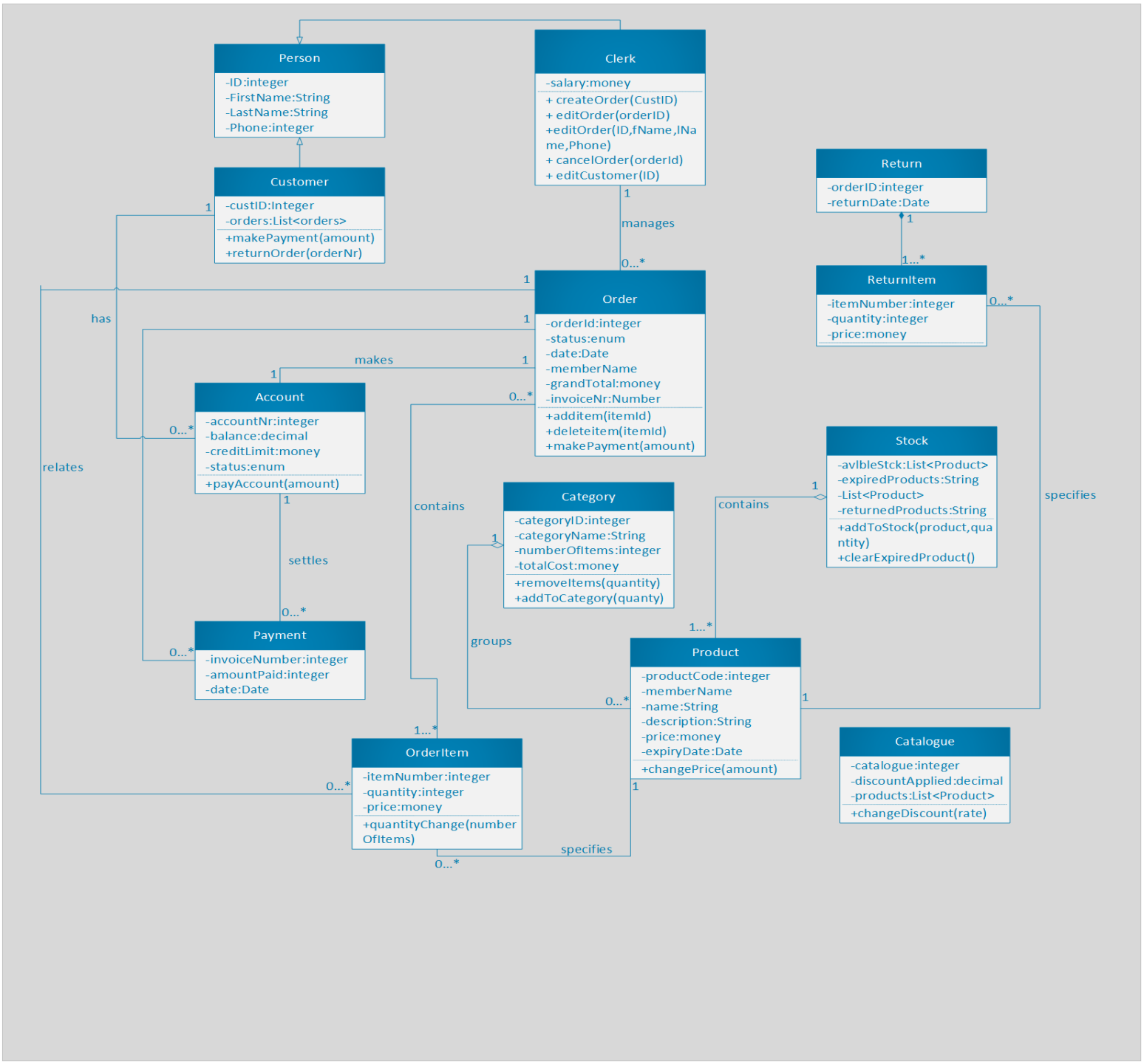
3.2. Customer Registration Sequence Diagram

This sequence diagram shows the interaction between the system and tele-seller when he/she is registering a new customer. First, the tele-seller goes to the MDI Parent/ Main Menu to click create customer button. Then customer user interface displays where the tele seller fills in the customer details. The task for creating customer handled by the customer controller and then directed to the customer class for task completion. Finally the customer details are saved to the data base.

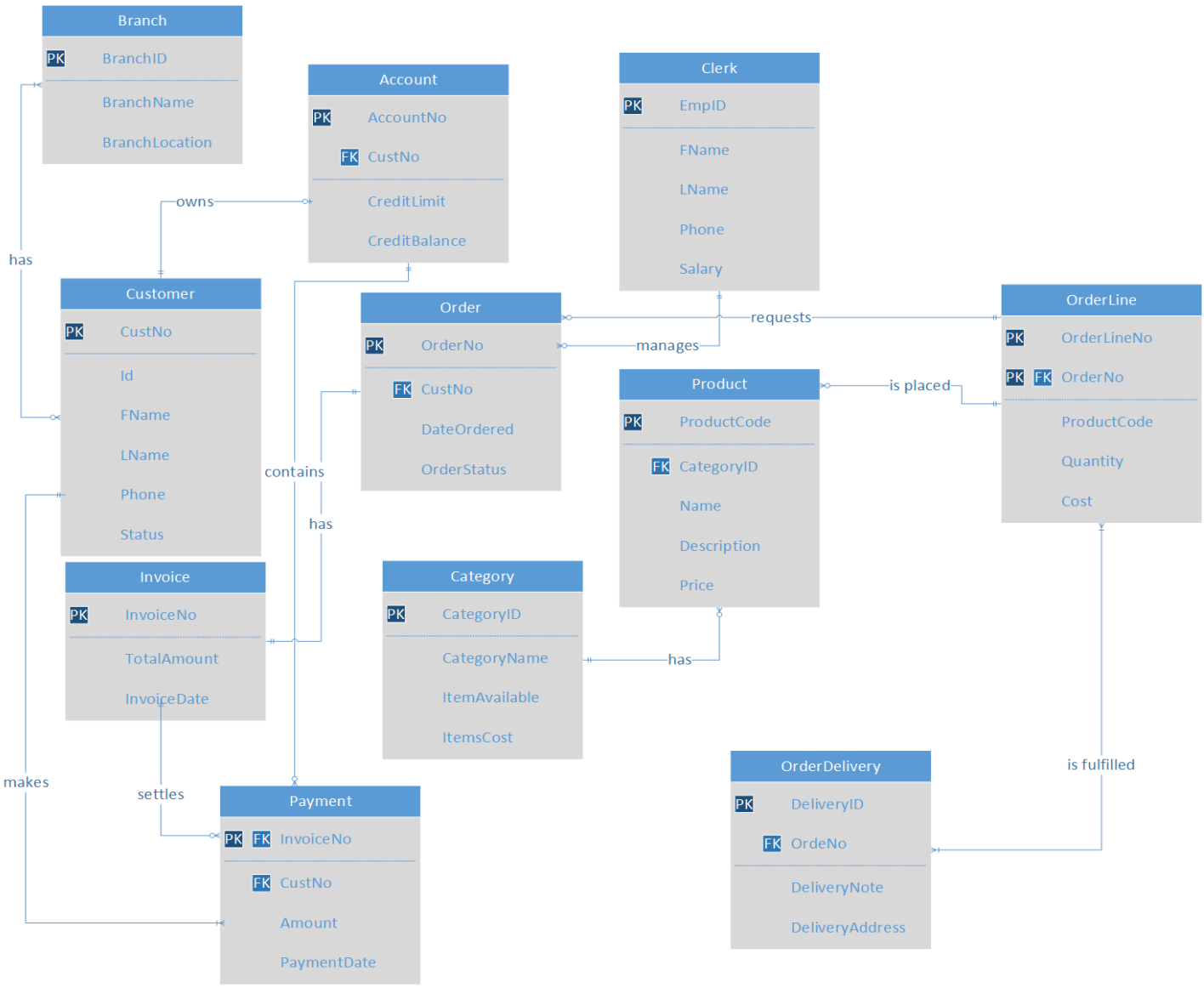


4. DESIGN CLASS DIAGRAMS

- Customer class inherits from person class where all personal information of each person with their specific ID are initially registered.
- Order class contain at least one item.
- Clerk maintains customer and catalogue details and prepares the order.
- Stock contains all the products in the warehouse and the system checks for item availability before adding an item to an order.
- Payment class is used when the outstanding amount for an order is paid for.
- Catalogue class- contains all the items and prices of these items in the warehouse.



5. ENTITY RELATIONSHIP DIAGRAM



Classes are important for elements in the project to interact well. Each class needs a primary key as key identifier of the class. This class uniquely identifies records in the database and more over it helps to extract and maintain data easily.

6. REPORT DESIGN

6.1. Detailed Report of Expired Products

This report will focus on stock flow in the warehouse that have expired. This will be done by printing out the expired products, the quantity of stock that has expired and for each expired product with its date of expiry. The objective is to keep track of the performance of the business, and to highlight products that are not in demand. This detailed report is automatically generated by the system and used by the Sales Manager or Marketing Clerk in making business decision. This report will help the business to see what kind of products are not high demand, so that they can come up with new strategies of improving sales. Moreover, with this report the business will be able to save more space in the warehouse them reduce the production of the items that are not moving on the shelves, this will reduce wastage in the business.

6.1.1. *Detailed Output Requirements*

Output type & ID: Poppel Detailed Report of expired products in stock.
Report Objectives: Emphasize on the products that are rarely bought in stock in different season and as result end up to waste. This will ensure production of those products is regulated during its low season or an alternative advertise these product about to expire so as to boost the rate at which they are bought.
Audience: Chief Executive Officer, Sales Manager, Chief Financial Officer, Marketing Clerk and store managers
Content: Product ID, Product description, Quantity in stock, Expiry date.
Layout: Columns
Selection: Click the expired products button under the MDI Parent to display
Sequence: Ascending by Date
Grouping/Summarisation: Group by product ID, group by product category
Comparison: If date is greater than expired date then product has expired. (SELECT * EXPIRED PRODUCTS FROM PRODUCT)
Media to be used: Word, Excel and printed report.
Frequency, Time, Delivery: Email report generated monthly basis.
Distribution: Marketing Clerk, Sales Manage and store managers
Privacy, security & integrity requirements: Report is available to Sales Manage, and Marketing Clerk, store manager and top stake holders only.

6.2. Summary Report of Best Selling Products at a Given Time

This report focuses on the amount of products sold in the market in monthly basis and yearly basis. Moreover it will summarize the amount of product still on hand, and determine revenue that is generated in Poppel financial year. This report is to be used by the Chief Executive Officers, Chief Financial Officer and Marketing Manager including the clerks. This will help them understand how the business performance for each division of the products and make key decision based on the trend. This report will also help them know what season during the year a certain product is on high demand so that they can produce more to gain profit and not loss.

6.2.1. Detailed Output Requirements

Output type & ID: Poppel summary report for most-selling products and monthly performance.
Report Objectives: Emphasize on products demanded in different season and highlight the trends of the stock in and out of the firm. This helps to make sure that there is sufficient stock for a particular product when there is high demand.
Audience: *:* Chief Executive Officer, Sales Manager, Chief Financial Officer, Marketing Clerk and store managers
Content: Product ID, Product description, Quantity in stock, Expiry date.
Layout: Columns, bars and graph.
Selection: All high-selling products
Sequence: Ascending product ID
Grouping/Summarisation: Group by product ID, group by product category
Comparison: If there is high sale of certain products at a particular season than other products within the same season. (SELECT MOSTSELLING PRODUCTS FROM PRODUCTS)
Media to be used: Word, Excel and printed report.
Frequency, Time, Delivery: Email report generated monthly and quarterly basis.
Distribution: Marketing Clerk, Sales Manage and store managers.
Privacy, security & integrity requirements: Report is available to Sales Manage, and Marketing Clerk, store manager and top stake holders.

7. INPUT-OUTPUT STANDARDS & CONTROLS

7.1. FORMALISED OUTPUTS:

Message boxes shall pop up whenever the user is going to submit, edit, check out, gives a wrong username or password and when the user is confirming with the picking list.

7.2. BUILT-IN VALIDATION TO ENSURE REQUIREMENTS ARE MET

We used try catch statements to make sure that we mitigate the error when the program is running and also ensure the user requirements are properly met. Example when the user enters a string for a phone number, an error message will appear and this because the try catch statements ensures the input entered is valid, thus with integer data type.

7.3. INPUT INTEGRITY CONTROLS

- **Data Validation Controls**: This has being implemented by making sure the Phone Number and ID numbers for product, customer, Clerk are in numeric values.

- **Value Limit controls**: This is implemented by ensuring there is a restriction of 10 number for phone number. More over the ID numbers is restricted to 6 numbers.

- **Completeness Controls**: This is basically to ensure that there are no missing components assembled. Example the registration form will not be perform the submit functionality unless all the fields are filled.

7.4. OUTPUT INTEGRITY CONTROLS

- **Destination Controls:**

7.5. SECURITY INTEGRITY CONTROLS

8. IMPLEMENTATION PLAN

The table below shows the implementation plan of the Poppel system. The task are schedules between the two group members.

Task	Person Responsible	Due date
1. Design GUI	Tonny	
2. Business Layer Classes	Wandile	
3. Database classes	Wandile	
4. Integrate the classes	Tonny	
5. Report Writing	Tonny and Wandile	

9. TEST PLAN

9.1. TEST ENVIRONMENT

Hardware Requirements

- 1.6 GHz or faster processor
- 1 GB of RAM (1.5 GB if running on a virtual machine)
- 5 GB of available hard disk space
- 5400 RPM hard drive
- DirectX 9-capable video card running at 1024 x 768 or higher display resolution

Additional Requirements
On Windows 8.1 and Windows Server 2012 R2, KB2883200 (available through Windows Update) is required.

Supported Operating Systems

- Windows 7 SP1 (x86 and x64)
- Windows 8 (x86 and x64)
- Windows 8.1 (x86 and x64)
- Windows Server 2008 R2 SP1 (x64)
- Windows Server 2012 (x64)
- Windows Server 2012 R2 (x64)

9.2. TEST ITEMS

9.3. TEST APPROACHES

9.4. PROBLEM TRACKING (TEST CASES)

Provide a description of all the features to be tested

TEST CASE 1

OUTCOMES: Unsuccessful login with specific error messages; back button works; user can only perform view catalogue when not logged in.
TASKS:

- Check parent form only allows for performing catalogue viewing, and no other tasks
    - Check that "Viewing Catalogue" menu item presents listview of the available products
- Check that error is presented when incorrect employee number is entered
  [Convention is capital *EMP* followed by three digits; e.g.: EMP003]
- Check that error is presented when incorrect password is entered
  [For sake of testing, password is standardised to *0123456*]
- Check that error is presented if no login details are entered into employee number nor password textboxes
    - Open the Login form and click on "Login" button without entering any details
- Check that back button returns user to "Poppel" [parent] form

TEST CASE 2

OUTCOME: successful login
TASKS:

- Check that user is able to login successfully with correct details

TEST CASE 3

OUTCOMES: unsuccessful registration of new customer; user is informed of unsuccessful registration
PRE-CONDITIONS: user has successfully logged in
TASKS:

- Check that error is presented when any textbox field is unfilled when attempting to click "Submit"
- Check that error is presented when the phone number entered on the phone textbox does not begin with the digit 0 and is
- Check that error is presented when the phone number entered on the phone textbox not 10 digits long
  [assumption that customers have local South African contact number]
- Check that "Cancel" Button clears all textbox fields

TEST CASE 4

OUTCOME: successful registration of new customer; customer is assigned a customer ID; new customer details are captured in database;
PRE-CONDITIONS: user has successfully logged in
TASKS:

- Check that after clicking "Submit" button the user is informed of successful entry
- Check in database that customer ID is assigned and captured

[Convention is first letter of first name + first three letters of their last name + first 6 digits of ID; e.g.: *ASMI940423*]

TEST CASE 5

OUTCOME: program successfully handles unusual name lengths [eg Angela L and Wendy Wu]
PRE-CONDITIONS: user has successfully logged in
TASKS:

- Check that customers with unusual name lengths Customer ID follows the convention

TEST CASE 6

OUTCOME: successful selection of customer for creating a new order
PRE-CONDITIONS: user is logged in
TASKS:

- Check that user is not able to create an order for a non-registered customer
- Check that user is able to select an existing registered customer

TEST CASE 7

OUTCOME: successfully create a new order for a customer
PRE-CONDITIONS: user is logged in; customer is existing and registered
TASKS:

- Check that Order ID is correctly assigned
  Convention for Order ID is ORDER and a four digit number, beginning at 1000; as more orders are added, the Order ID increases by 1; e.g.: *ORDER 1009*]
- Check that selected customer details appear at top of Order Form
- Check that the rest of the form's functionality is activated after product is selected from combo box
- Check that "Add to Cart" Button adds correct item (with correct Order Item ID, Product ID, Product Name and Quantity) to the Cart Listview
  [Convention for Order Item ID is *ITEM* and a four digit number, beginning at 1000; as more order items are added, the Order Item ID increases by 1; e.g.: *ITEM 1004*]
  [Convention for Product ID is PRO and a four digit number, beginning at 1000; e.g.: *PRO001*]
- Check that Total and Remaining Credit Balance is updated correctly as items are added
- Check that "Check Out Cart" button requires user to confirm order

- Check if available stock is updated in the Products Table in the database after adding an order

TEST CASE 8

OUTCOME: user is able to interact with ListView
PRE-CONDITIONs: user is logged in; customer is existing and registered; user is busy creating a new order
TASKS:

- Check that user is able to select item in Cart Listview
- Check that user is able to delete selected item from Cart Listview

TEST CASE 9

OUTCOME: unsuccessful placement of order

TASK:

- With items added in the Cart Listview, check that when user clicks the window's exit button, a confirmation of cancellation is prompted
- With items added in the Cart Listview, check that when the user clicks the "Ba

TEST CASE 10

OUTCOME: user unable to add items to cart beyond their credit limit
PRE-CONDITIONS:
TASKS:

- Check that user is notified and prevented from attempting to add items to cart which total beyond the "available credit balance"

TEST CASE 11

OUTCOME: generation of picking list ready for printing
PRE-CONDITIONS: order has been successfully placed
TASKS:

- Check that the picking list displays access to most recently created order
- Check that all other functions of the picking list form are enabled only after the "OK" button is clicked
- Check that clicking "OK" without an order ID selected generates an error message
- After clicking "OK", check that Clerk ID, Order Date, Order ID, Customer ID, Customer Name and Delivery Address are correct
- Check that all products ordered are listed in the Picking Listview
- Check that an accurate print preview is generated
- Check that selecting "print" will connect to network printers

TEST CASE 12

OUTCOME: successful sales reporting -- user is able to see correct purchase history and inventory available
PRE-CONDITIONS: at minimum, one order has been placed
TASKS:

- Check that user is able to select Sales Report in the toolstrip
- Check that system reflects purchase history when "sales report" button is selected
- Check that system reflects inventory when the item is selected
- Check that graphics reflect the correct quantities
- Check that graphics' colour key is correct