



# System Specification For Poppel

17.10.2022

---

## System Development & Design

Compiled And Designed By:

Sinoyolo Qhashana (QHSSIN002)

Sinethemba Gomba (GMBSIN005)

**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: Sinethemba Gomba

Signed: S.Gomba Date: 17/10/2022

Full Name: Sinoyolo Qhashana

Signed: S.Qhashana Date: 17/10/2022

## Table Of Contents

1.	Introduction.....	4
1.1.	Overview of Specification.....	4
1.2.	Context & Scope of System Specification. ....	4
1.3.	Design Assumptions & Constraints.....	5
2.	User Interface & Dialogue Design.....	6
2.1.	Wireframe Diagram or Windows Navigation Document.....	6
2.2.	Screen Standards.....	9
2.3.	Detailed Screen Layout.....	9
3.	Design Sequence Diagrams.....	10
3.1.	Design Sequence Diagram 1 .....	10
3.2.	Design Sequence Diagram 2.....	11
4.	Design Class Diagrams.....	12
5.	Entity Relationship Diagram.....	13
6.	Report Design.....	14
6.1.	Report 1 .....	14
6.1.1.	Detailed Output Requirements.....	14
6.1.2.	Report Layout.....	16

7.	Input-Output Standards & Controls.....	17
7.1.	Formalised Outputs:.....	17
7.2.	Built-In Validation to Ensure Requirements are Met.....	17
7.6.	Input Integrity Controls.....	17
7.7.	Output Integrity Controls.....	18
8.	Implementation Plan.....	19
9.	Test Plan.....	21
9.1.	Test Environment.....	21
9.2.	Test Items.....	21
9.3.	Test Approaches.....	22
9.4.	Problem Tracking (Test Cases).....	23
9.5.	Test Schedule.....	24

# Introduction

## 1.1. Overview of Specification

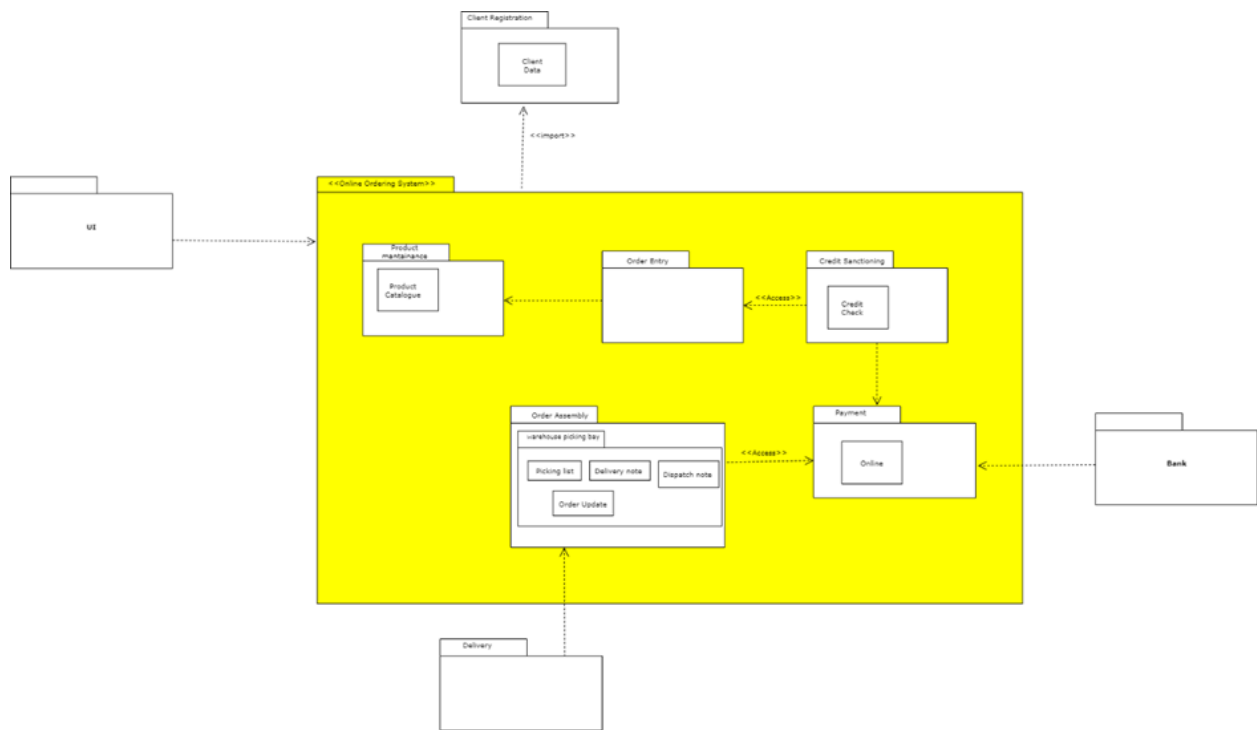
This document is about the documentation and the development of Poppel's Ordering System. The hierarchy of things starts with understanding the system's requirements and the output would be the implementation and deployment of the system, hence why we adopted this order of things. Our goal is first to provide context about the system we are currently developing for Poppel and incrementally show the adopted design and how we will achieve this through different UML diagrams.

The business has faced problems with manufacturing capacity and human error in manual processes and we were able to identify these problems in our previous document, the Business Case. We then went on to understand the user requirements for the development of the system, and we used the User Requirements Specification document. The project is currently in the design phase, and we are moving toward the implementation and deployment phase of the system.

## 1.2. Context & Scope of System Specification

Poppel is a business based in the Western Cape which manufactures and imports confectionary and soft drinks. It was started in 2003 by Bertus Van Heerden and Angus McClaren who are trying to revive the products that were popular when they were kids. The business has faced problems with both manufacturing capacity and human error in manual processes. To solve these issues, the business will build additional warehouses and hire an IT department to solve most of its technology-related problems by creating a Web interface. This will allow the business to reach more clients and simplify processes by making them more efficient, as well as improve the quality of the business. The aim of the project is ultimately to increase the business's customer pool and profit.

The objective of this project is to enable users to perform the ordering process in an application that enables easy access than a manual process that is prone to errors. This project will also contain the functional requirements that will be needed for the system to function effectively. Functional requirements are the basic functionalities which the end-user demands/requires from the system. These functions are mandatory and answer the question "What should the system do?". This includes actions such as the ability to create an account, the ability to checkout, etc. It will also consist of non-functional requirements which ensure the quality of the system. These include logical GUI layout, and fast and responsive application, which are crucial and intrinsic for the design as well.



The part of the system we are working on is the ordering system. The reason we chose to focus on this part of the system and base our scope here is because this part affects the entire system functionality. There have been problems inside Poppel where customers received incorrect orders and some experienced delays with their orders, and we have identified the root-cause of these issues and it lies in the ordering system hence we have fixated our development to configure it and better enhance its functionality to meet the customer's requirements.

### 1.3. Design Assumptions & Constraints

Assumptions	Constraints
The team will adopt the three-tier architecture model for the system design.	Budget: Funds may be limited
Scope – Guidelines stay the same	Risk - any unexpected occurrences that can affect the project

Resources- Team and materials will be available to complete the project	Project Quality - the project quality can be affected if other constraints are not managed effectively.
The design of the system will translate to the actual/implementation of the system to ensure that the business goals are met.	Time Management - the project needs to be done according to the timeline. Still, it may run over that timeline

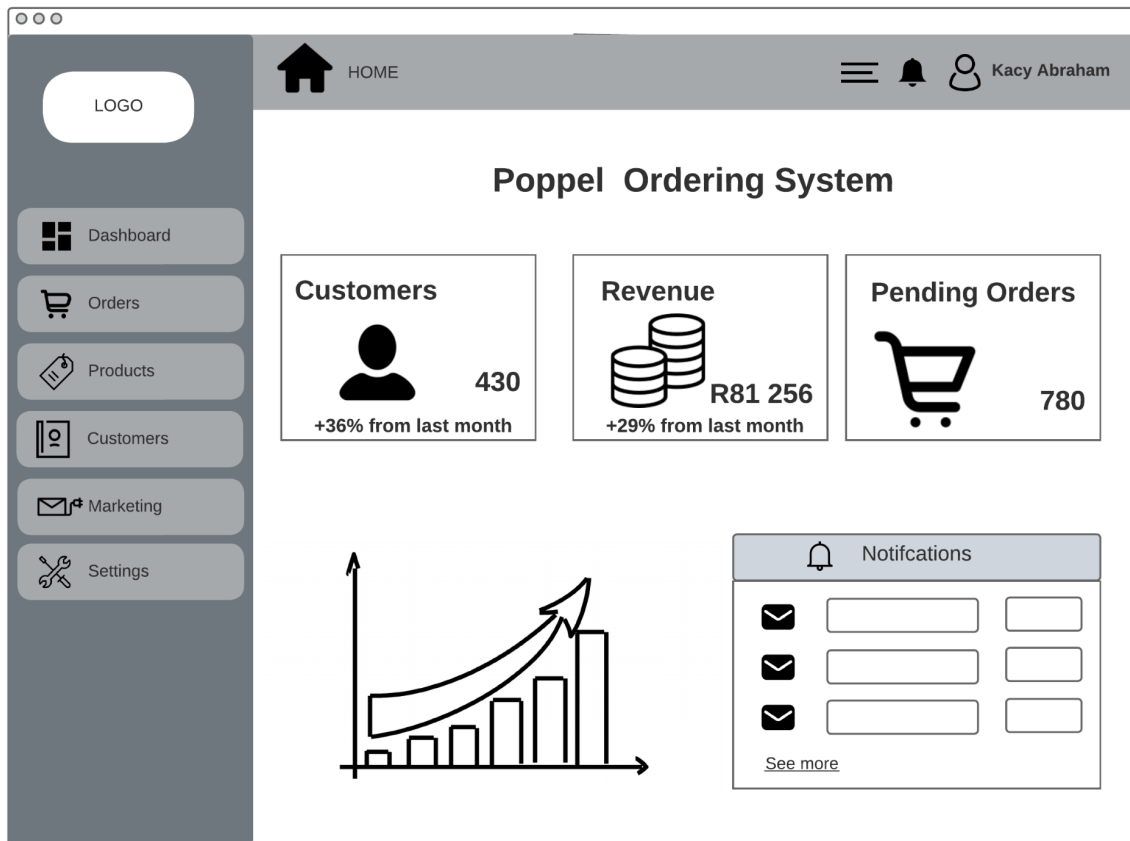
## User Interface & Dialogue Design

User interface (UI) design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. In this section, we will be illustrating how our User Interface will look by using Wireframes and Storyboarding to depict how the Dialogue design will also work.

In this section, we will design a wireframe of the system which will depict the homepage the user will see initially when they enter the system. We will then design a step-by-step procedure/ storyboard of how the user will interact with the system and its response. The storyboarding depicts a “happy day” scenario where the user will complete the set of tasks they want to do on the system.

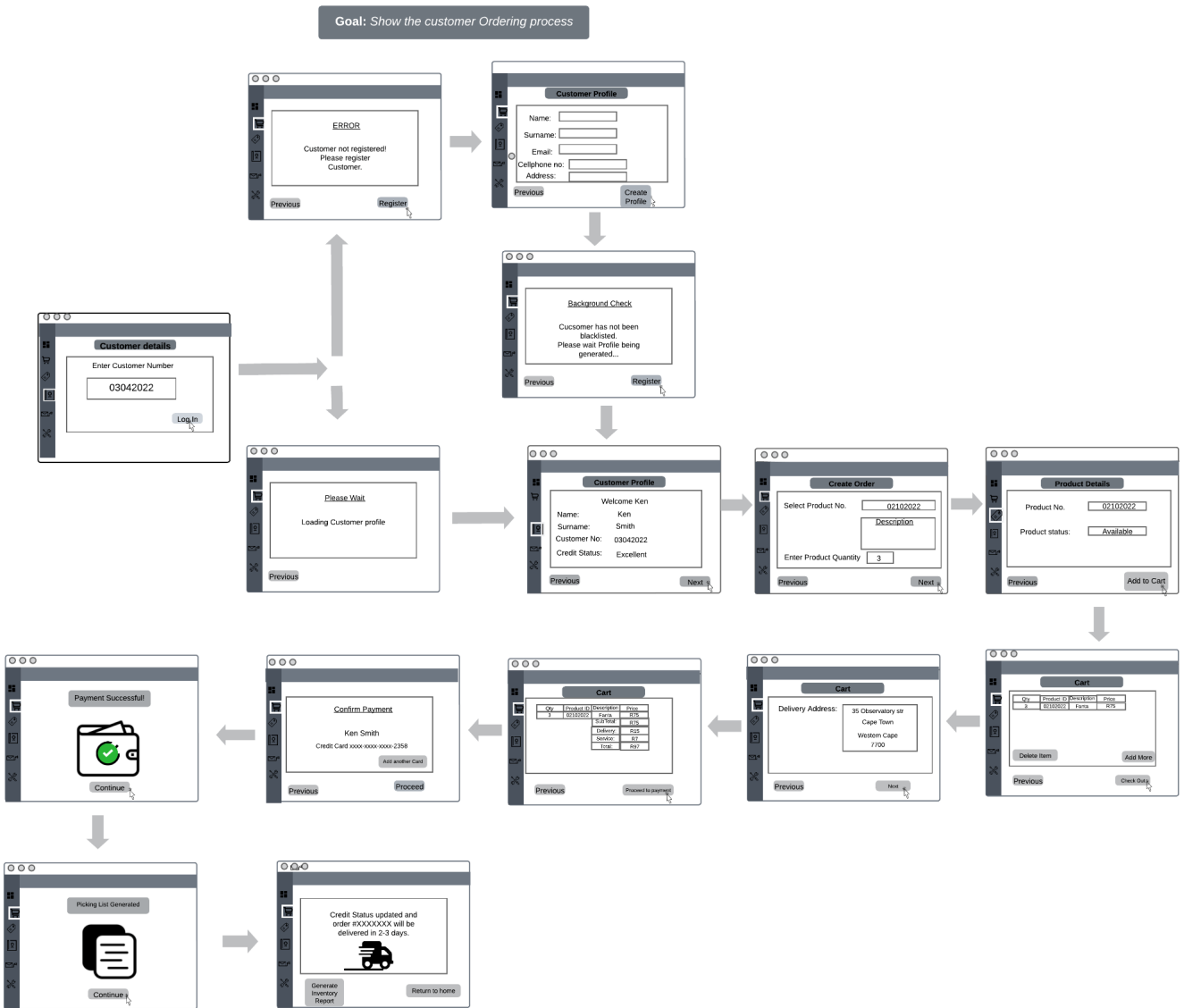
### 2.1 Wireframe Diagram or Windows Navigation Document

This section will focus on modeling the dialogue interaction between the user and the system. The first thing to do is to model the homepage of the system and how roughly it will look after the implementation phase. For now, our goal is to provide an idea of how the system will look in terms of where the application elements such as buttons, pictures, logo, icons, etc. will be placed. Below is the wireframe of the system we are trying to develop for Poppel.



From above, this is what the system's homepage will look like, with all these screen components. The following wireframe will illustrate how the user will interact with the system and how the system will respond to each click/action made. What we are modelling is a happy-day scenario where the user will fulfil the system request, e.g., making the order process and completing it.





The above picture depicts the interaction between the user and the system and how each response leads to the next. We have designed the screen and all its components this way because we feel that it is more informative and clear to show what they have to do and what each response means. This design is to ensure that we do not complicate this design, but rather simplify it to be user-friendly.

## 2.2 Screen Standards

The overall screen standards will entail the layout, colour and overall appearance. Our primary layout design is what we have depicted as per our wireframe for our homepage. Our primary colours will include black, and white, and our custom-made colour is written in Hexadecimal format (#2f1e68). Our overall appearance of the system will be that of a modern touch with minimal colours that will complement each other.

## 2.3 Detailed Screen Layout

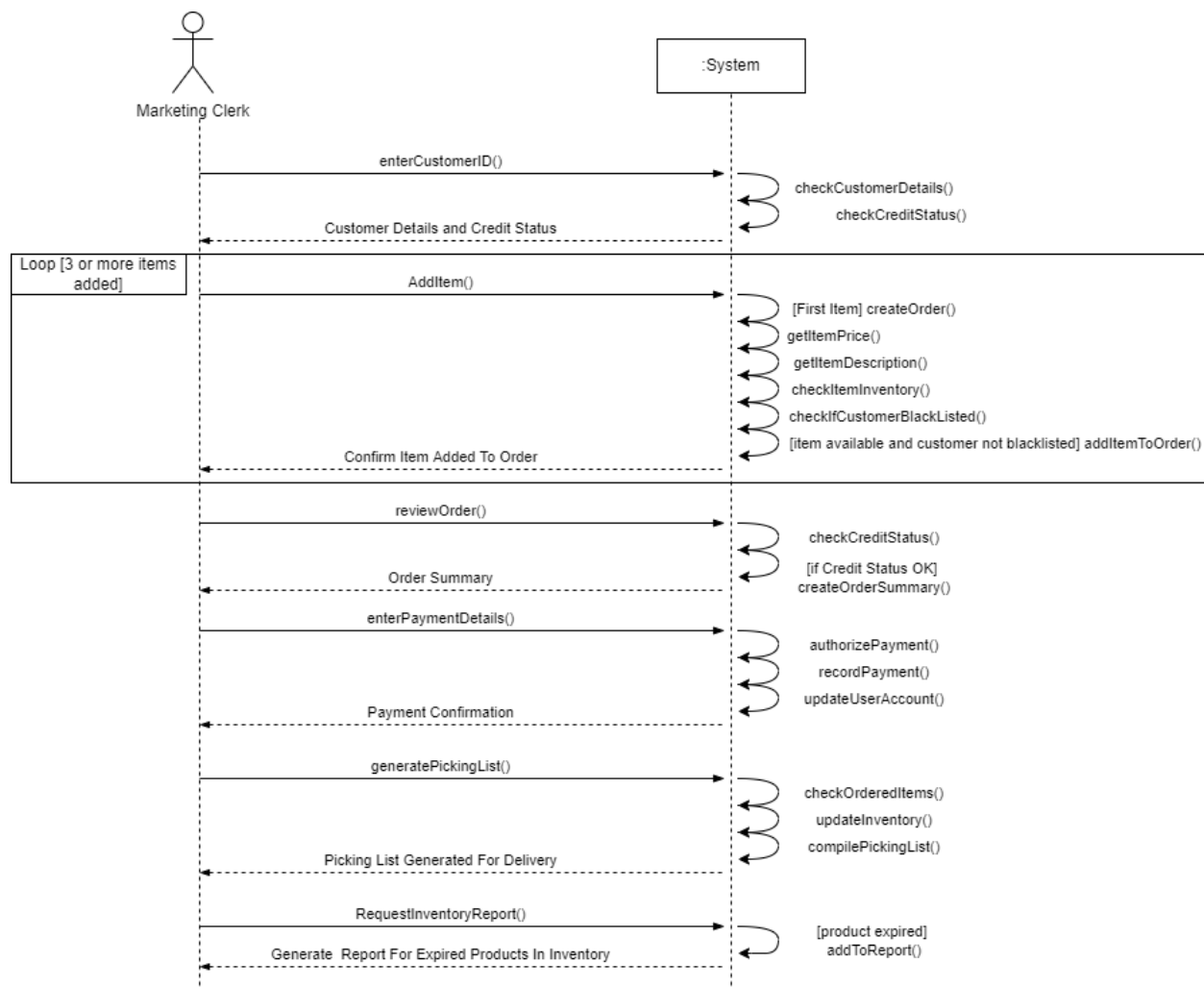
In this section, we would like to provide a realistic view or display of the system and how it would potentially look like. It is not the accurate version but rather something similar and close to it.



## Design Sequence Diagrams

### 3.1 System Sequence Diagram

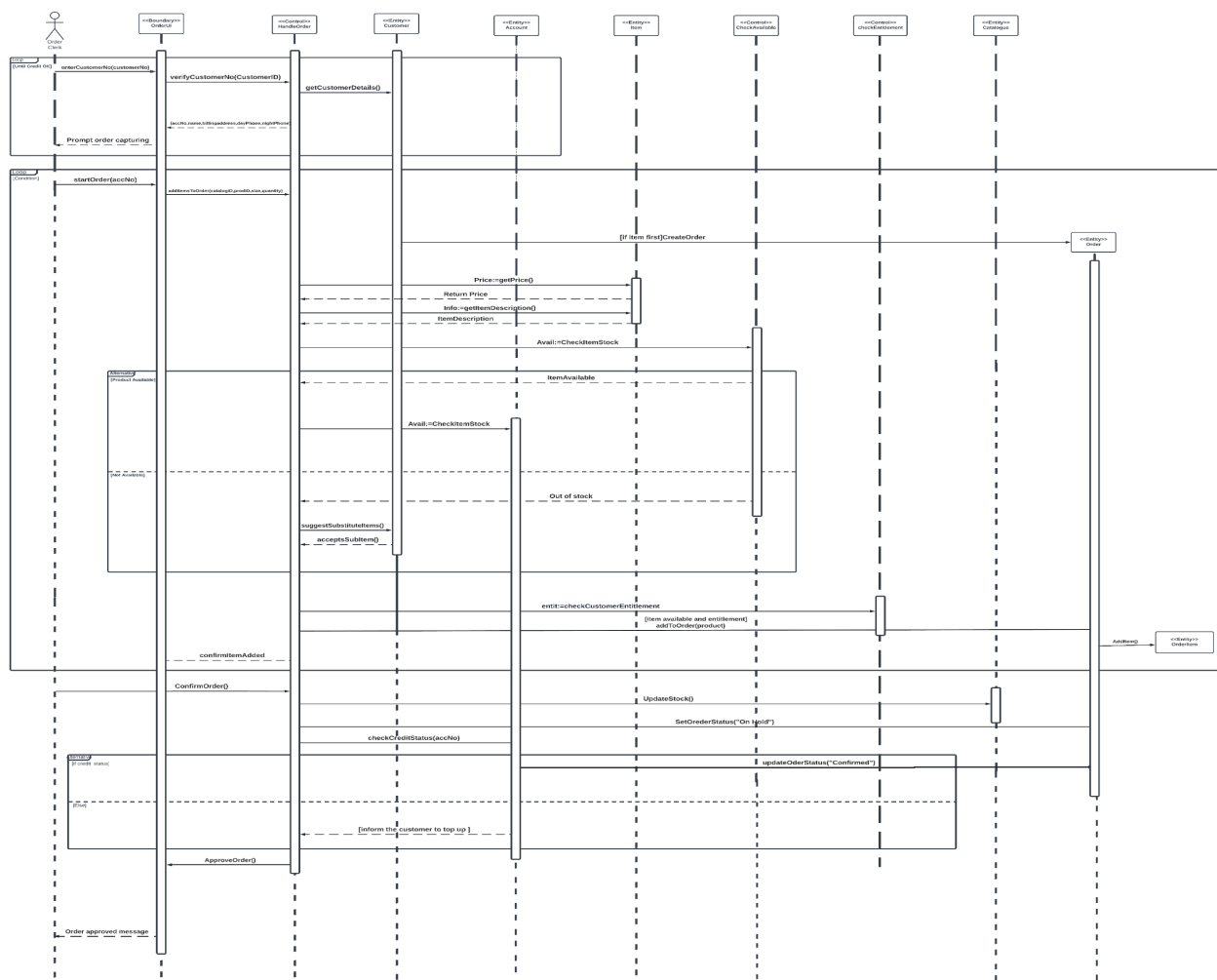
In the following diagram we have a systems sequence diagram which will depict how the marketing clerk will interact with the system in more detail. It shows the process interaction arranged in a time sequence between the actor and the system. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality of completing the order process for a customer. This is a high-level view of functionalities and a low-level will be depicted in our Detailed System Diagram.



## 3.2 Detailed Design Sequence Diagram

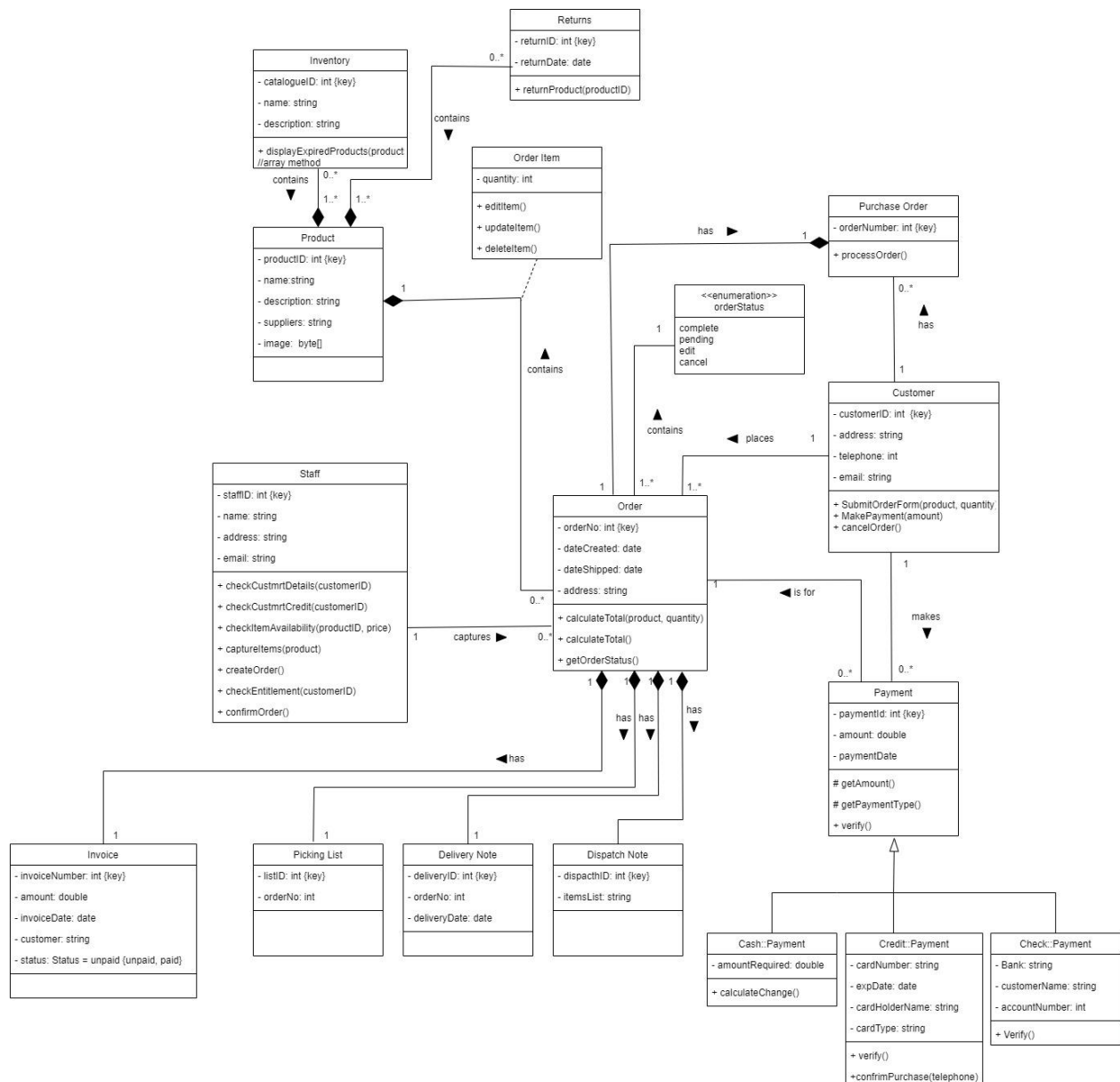
The following diagram is a sequence diagram to illustrate in detail how the customer and the system interact. The assumptions made in this diagram are as follows:

- The first assumption is that the customer is registered. No need for registration
- The second assumption is that the customer fulfils the order, pays for it and if it's out of stock, they opt for a substitute item
- The third assumption is that the customer is entitled to the orders and that his credit status is positive
- The fourth assumption is that there is a database that houses all the system's catalogue and items that the company has
- The fifth assumption is that the customer does not cancel the order and it gets delivered successfully.



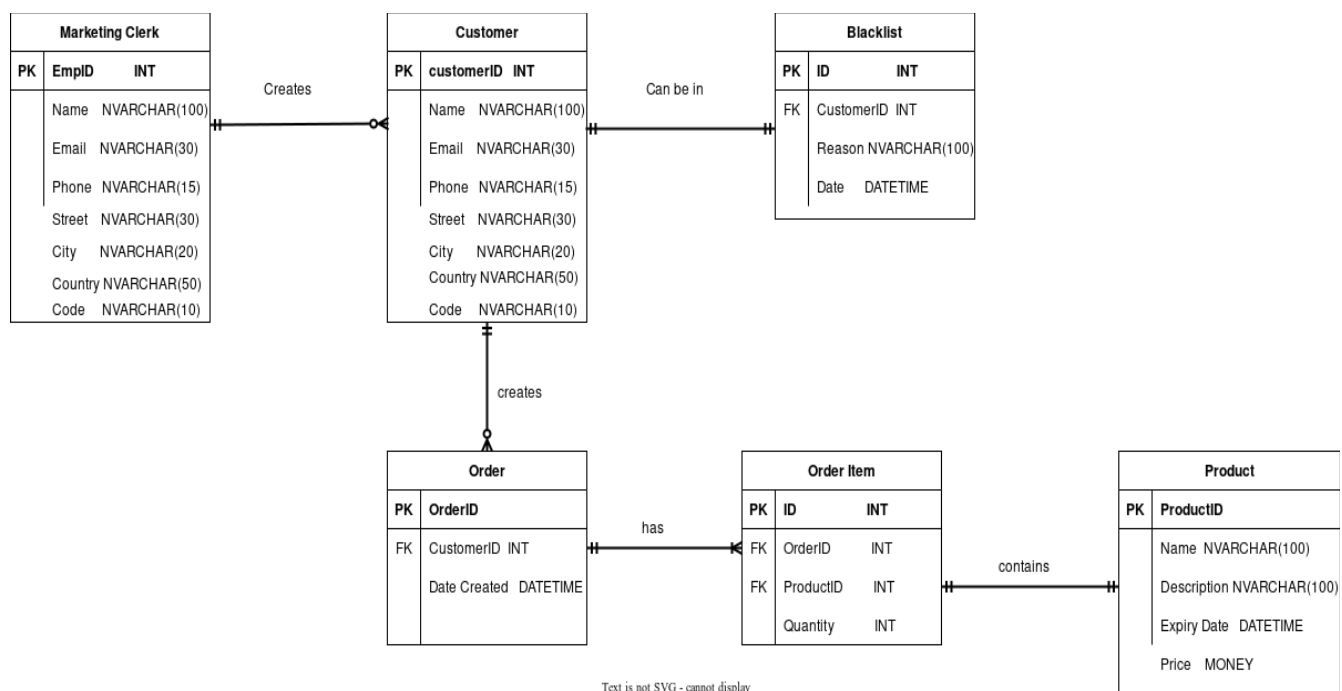
## Design Class Diagram

This section focuses on the **full** detailed class diagram that shows the system utilities within our scope. This shows each class will interact with the other, the associations involved, the relationships, etc. Each class is named according to its functionality and its attributes and methods as well as access is stated below in the diagram. This is to map a road map to the implementation phase and ensure that we deliver the solution required by the scope defined in the project guidelines.



## Entity Relationship Diagram

In this section, we focus on conceptualising and showcasing our database design and what will be in our database and project. We have provided all the key attributes and their data types as well as their relationship with each other. This database has been normalised to the 3rd Normal form. This means that there is no repeating or redundancy such as transitive dependency.



## Report Design

### 6.1 Report 1

#### 6.1.1 Detailed Output Requirements

This report will be generated by the system to identify all expired products in inventory. This report will provide relevant and meaningful information about the stock that the business has. This can help with the business decisions (*e.g. business executives*) or execute their daily tasks (*e.g. warehouse agents*).

<b>Report type &amp; ID</b>	Name -Expired Products Inventory report in September ID – PP-S214
<b>Report Objectives</b>	The goal of this report is to track all the expired products in the inventory.
<b>Audience</b>	Marketing manager Stock Control Supervisor
<b>Layout</b>	Table format
<b>Report Content</b>	Product name, Stock Ordered, Stock Sold, No. of Expired Products
<b>Selection</b>	Select all products that have expired from the inventory.

<b>Sequence</b>	Descending order of Products with the most expired .
<b>Grouping / Summarization</b>	Group order Products for example.
<b>Media to be used</b>	Electronic

<b>Frequency, Timing, Delivery</b>	Monthly
<b>Distribution</b>	Management
<b>Privacy, Security, &amp; Integrity Constraints</b>	Available to only executive management and marketing department



## 6.1.2 Report Layout

**Category:** Expired Products

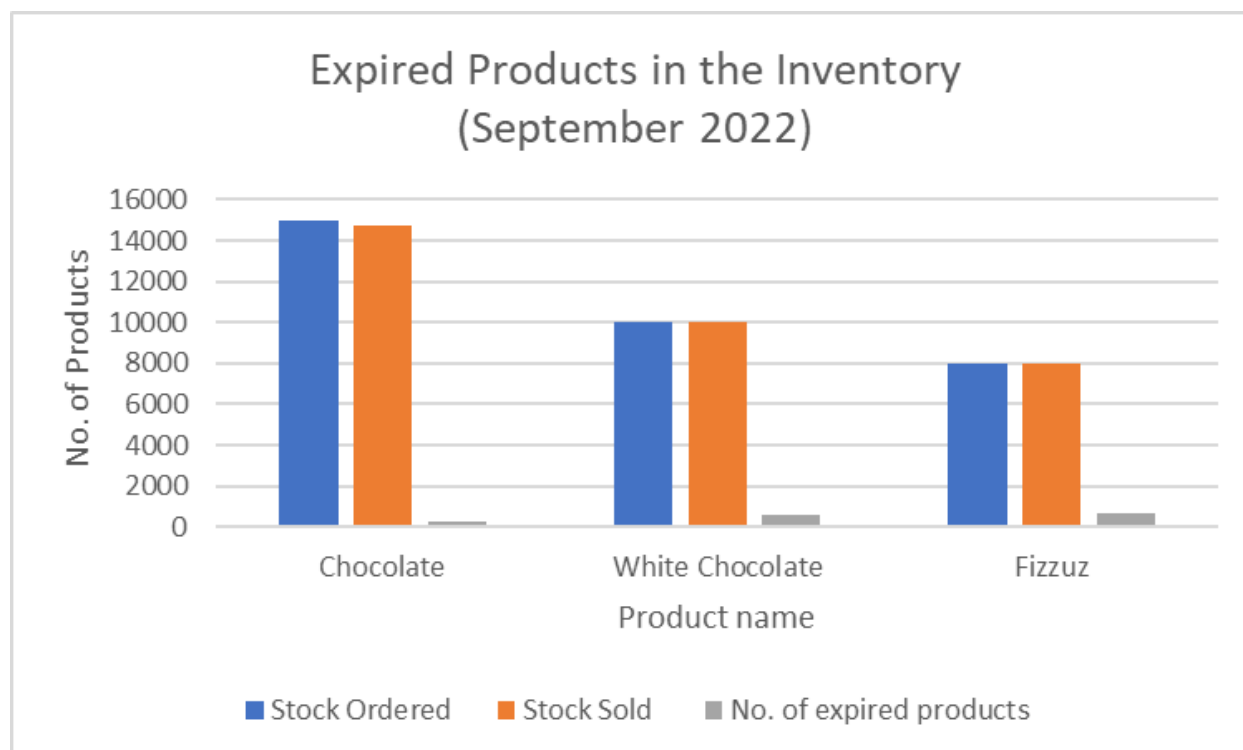
**Year:** 2022

**Month:** September

**Date of Report:** September 2022

Product Name	Stock Ordered	Stock Sold	No. of expired products
Chocolate	15000	14700	300
White Chocolate	10000	9994	560
Fizzuz	8000	7993	690

### Track of stock and the number of Expired products



## Input-Output Standards & Controls

### 7.1 Formalised Outputs

Warning and confirmation messages will be displayed throughout the system. An option to print the inventory system will also be made available for the Marketing Clerk to help with the business decisions. Message boxes are in place to guide the user on what type of data should be entered, the type of selections, etc. We also made the system highlight where the data might be missing and this is shown after the message box output.


### 7.2 BUILT-IN VALIDATION TO ENSURE REQUIREMENTS ARE MET

There are message boxes and validation, as well as exception-handling functionalities built into our system. The message boxes describe the error and provide more details in terms of how to proceed with the task. Data Validation ensures that the required data type is entered into the textbox and also the required credentials are entered. E.g. On our login page, you have to use a predefined username and password and it's "admin". Our system handles exceptions which ensure that the application does not crash when an invalid input is entered. And lastly, we have implemented a class for Validation to validate that the entered data is the desired one.

### 7.3 Input Integrity Controls

We have used 4 input integrity controls to ensure the quality of our data. These are:

**Field Combination Controls:** review various combinations of fields to ensure that the correct data is entered.



**Value Limit Controls:** check numeric fields to make sure that the amount entered is reasonable.

**Completeness Controls:** ensure that all the necessary fields are completed. This check can be executed as an input occurs so that, depending on which fields are entered, additional required fields must be entered.

**Data Validation Controls:** ensure that numeric fields that contain codes are correct. Other data validation can be done online against internal tables or files.

## 7.4 Output Integrity Controls

The output integrity controls we have used in our system are:

**Destination Controls:** The output of the system is channelled to the right people, in our case, the marking clerk. They can also print the picking list and send it to the designated stakeholder.

The appropriate information is present in all outputs, e.g. Textboxes, listviews, etc.

## Implementation Plan

### Poppel Ordering System Implementation Plan

An Implementation Plan template provides business leaders and managers with a framework to create a summary of the strategy to be used to make a solution possible for a piloted project. An implementation plan is informed by a pilot program and should include most of the same information contained in the pilot but adjusted to take into consideration the outcome of the pilot.

The Implementation Plan walks business leaders and managers through the process of developing a strategy which can be rolled out after the successful completion of a pilot program. The implementation plan should take into consideration cost savings when implementing the full-scale initiative (beyond the pilot). Prioritisation matrices can also be used to narrow down the focus of the project and develop a more specific implementation plan. Questions that need to be asked when developing the Implementation Plan include:

- Where & how do we use the implementation plan?
- What are the barriers to the implementation of this plan?
- What opportunities are there to improve from the initial pilot?

Project Name	Poppel Ordering System	Estimated Time To Complete	17/10/2022	
#1. Organise the Implementation Effort				
Task	Responsible	Date Started	Date Completed	Comments
Creating Database Tables	Software Developers	2022/09/15	2022/09/17	The implementation of a database was successful and the referencing of primary and foreign keys was successful
Creating Classes & Controllers	Software Developers	2022/09/18	2022/09/21	The implementation of classes to form the blueprint of objects

				was successful.
<b>UI Design</b>	Software Developers	2022/09/22	2022/10/26	This implementation was very difficult to create because a lot of things had to be considered. For example, colours, the types of panels, the types of buttons, and the type of boxes, e.g. Listbox vs Combobox. In the end, we were able to standardise the UI and ensure that each form is related to the other.
<b>UIX</b>	Software Developers	2022/09/27	2022/09/30	This task required designing the experience of the user and we have included things like the animation to ensure that the application is appealing.
<b>Data Validation</b>	Software Developers	2022/10/1	2022/10/03	In This task, we ensured that the data validation standards are being adhered to.
<b>Testing</b>	Software Developers	2022/10/04	2022/10/17	Testing was done and we were able to see how the system responds to different favourable and unfavourable inputs. We then handled the unfavourable inputs to ensure that it responds positively to errors.

## Test Plan

This section of the project focuses on testing the working system of Poppel. It has been created and developed. Now, we would like to find out some of the errors that may exist in the system and potential new or undiscovered errors.

### 9.1 Test Environment

The minimum software required is Visual Studio 2022, which is the IDE we have used to develop the application. The Hardware for the software to run on a PC/Laptop must have at least:

- Intel Core
- Minimum 2GB Ram
- At least 1GB Free Storage
- Internet Speed of 5MBPS
- Windows OS

### 9.2 Test Items

#### Features to be tested:

- The system's ability to create a customer
- The system's ability to create a new order
- Adding an item to the cart.
- Can the system remove an item before the customer pays for it.
- Would the system be compatible with the different devices that it would be used on.
- How the system can tolerate/ handle exceptions

#### Features not to be tested:

- Theme Selector of the system-this is not necessary and is out of scope, our main goal is to design with user friendly colours for our system.

- Salutation Selection- we will use the users name and surname
- Check if Captcha is working -this not a necessity as we do not expect bots in our system (no bot would place an order).

### 9.3 Test Approaches

We will first do **Unit Testing**. We would like to ensure that each component can function properly when tested individually before it can be tested with other components. The goal is to ensure that no component performs its function incorrectly while in isolation.

We will then move to **Integration Testing**. We want to ensure that the components are able to function with each other effectively. That means receiving data from one component to the other and that data must be used correctly.

We will also do **System and Stress Testing**. This is to ensure that functional and non-functional requirements are met.

## 9.4 Problem Tracking (Test Cases)

We have tracked the problems and test cases using the following diagram. This helped us know which component of the system had errors and we had to re-engineer it again.

Poppel Ordering System Problem Tracking				
TEST CASES	Test Pass (04/10/2022)	Test Pass (09/10/2022)	Test Pass (17/10/2022)	Bug ID
<b>Hardware Functionalities</b>				
Internet	PASS	PASS	PASS	
Ram	PASS	PASS	PASS	
Space	PASS	PASS	PASS	
OS	PASS	PASS	PASS	
<b>Software Functionalities</b>				
Create a Customer	FAIL	PASS	PASS	11
Create an Order	FAIL	FAIL	PASS	21, 32
Print Picking List	FAIL	FAIL	PASS	41, 43
Check Blacklisted Customers	FAIL	FAIL	FAIL	35, 37, 39
Display Orders	FAIL	PASS	PASS	9
Print Expired Products	PASS	PASS	PASS	
Login App	PASS	PASS	PASS	
Handle Exception	FAIL	FAIL	PASS	50, 51
Exit Application	PASS	PASS	PASS	



## 9.5 Test Schedule

The Test schedule was coordinated with the development schedule exactly on the time frame from the 04/10/2022 - 17/10/2022. This time frame was assumed in the development schedule and we had estimated that testing all these would take some time hence we have dedicated enough time and effort to ensure that the system was working properly. This Test Schedule was part of the implementation schedule and every date aligns with the specification

Poppel Ordering System Problem Tracking			
Task Name	Start Date	End Date	Duration
Create a Customer			
<b>Unit testing</b>	2022/10/04	2022/10/04	1 day
<b>Integration testing</b>	2022/10/04	2022/10/05	2 days
Create an Order			
<b>Unit testing</b>	2022/10/05	2022/10/05	1 day
<b>Integration testing</b>	2022/10/05	2022/10/07	2 days
Print Picking List			
<b>Integration testing</b>	2022/10/08	2022/10/08	1 day
Check Blacklisted Customers			
<b>Integration testing</b>	2022/10/09	2022/10/10	1 day
Display Orders			
<b>Unit testing</b>	2022/10/11	2022/10/11	1 day
<b>Integration</b>	2022/10/11	2022/10/11	1 day
Print Expired Products			
<b>Integration</b>	2022/10/12	2022/10/12	1 day



Login App			
<b>Unit testing</b>	2022/10/13	2022/10/13	1 day
<b>Integration</b>	2022/10/13	2022/10/13	1 day
Handle Exception			
System Stress	2022/10/14	2022/10/16	2 days
Exit Application			
<b>Integration</b>	2022/10/16	2022/10/17	1 day