DEPARTMENT OF INFORMATION SYSTEMS

SYSTEMS DESIGN & DEVELOPMENT

SYSTEMS SPECIFICATION FOR POPPEL

TEAM MEMBERS

Student Number: CLNSIH001	*Student Name: Sihle Calana*
Student Number: KHNAHM006	*Student Name : Ahmed Khan*

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: ____/____/2020

Full Name: _____Ahmed Khan_____ Signed: _____A.K_____ Date: _10_/_05_/2020

TABLE OF CONTENTS

1. INTRODUCTION

1.1. OVERVIEW OF SPECIFICATION

The system specification documents the approach that will be used to develop an Order Processing system for Poppel. This will aid the marketing clerk in accepting orders and storing information on customers and their orders on a database as well as records of ingoing and outgoing inventory.
This document will detail how the user interface will look (using wireframes), the functions that occur in the background when the clerk interacts with the system (though sequence diagrams) and the information stored about objects like inventory, customers & orders (as displayed by the class diagram) as well as the relationship between these objects as shown in the entity relationship diagram.

1.2. CONTEXT & SCOPE OF SYSTEM SPECIFICATION

Poppel is a soft drink manufacturer who is looking to overhaul their aging systems. The aim for this project is to produce a piece of software that will help to smoothen the process of order placement between customers and clerks.

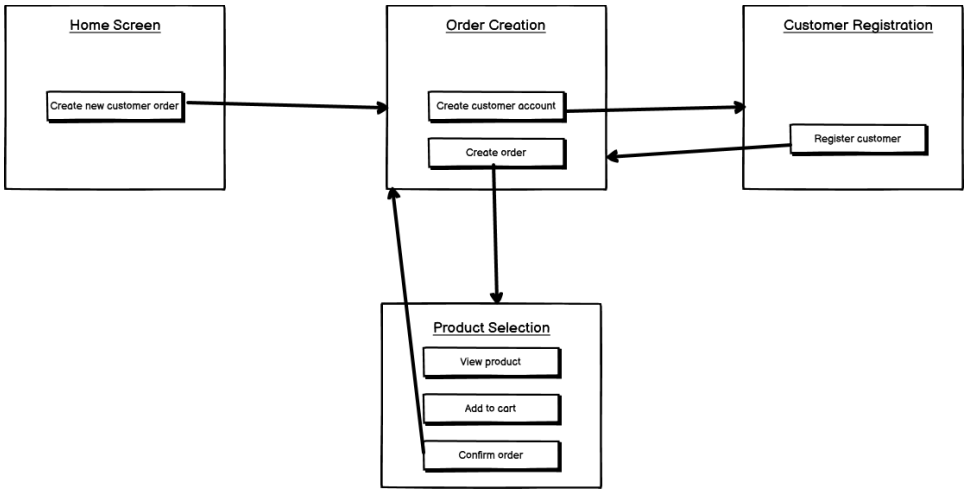
The requirements of this project are to develop an environment where are marketing clerk will be able to initiate an order on behalf of a customer. With a customer on the telephone, a clerk should be able to access that customers details using their unique customer registration number. The system should be able to check that the customer has a valid credit score, and if so, an order may be placed. The clerk will be able to view available items in the product catalogue and add them to the customers cart as he pleases, while the system will check the availability for each of these products. Once the customer is satisfied with the order, the clerk will be able to confirm it and an invoice will be sent out to the customer.

2. USER INTERFACE & DIALOGUE DESIGN

2.1. WIREFRAME DIAGRAM OR WINDOWS NAVIGATION DOCUMENT

The program will be made up of four separate forms as displayed in the windows navigation document below. The first window will act as a homepage where a new order can be initiated. This will also be the window that houses all the other forms. The second window will allow for a new order to be initiated. This is where a customer ID will be searched, and the customer's credit status will be checked. If Credit status is good, then the clerk will be able to create a new order. If the customer is not yet registered with poppel, the “Create customer account” button will open up the registration form. Once

the order is created, the clerk will be taken to the product selection form. Here the clerk will be able to view, add, and remove items to/from a cart before confirming the order. Once the order is complete the Clerk will be returned to the order creation screen where he can begin the next order.



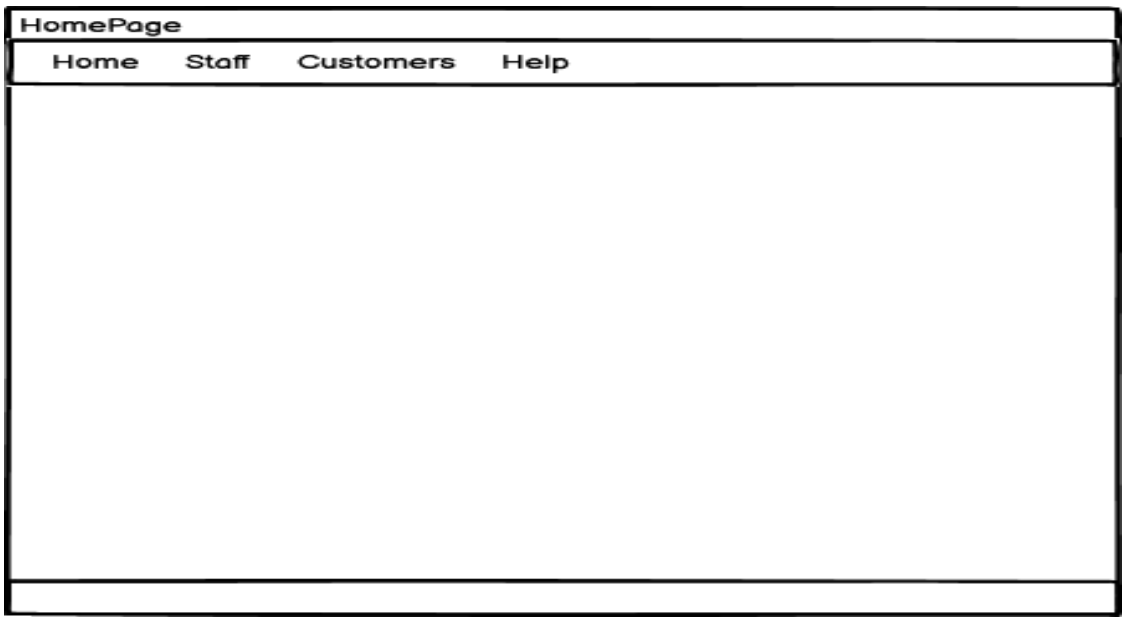
We felt that it was important to split each activity up into its own form in order to keep information on each page as simple and relevant as possible.

2.2. SCREEN STANDARDS

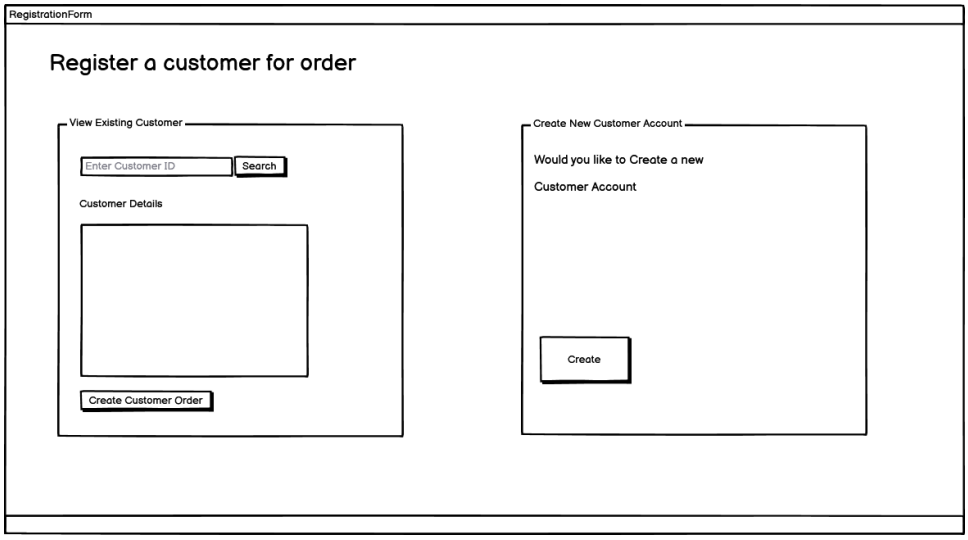
Each form will have the Poppel logo set in the top left corner. Underneath this will be the form heading, followed by whatever controls may exist in that form. When required, search bars will be used to find various customers and products. The bulk of the contents within each page will be displayed within a rich edit box aligned to the the left of the screen

2.3. DETAILED SCREEN LAYOUT

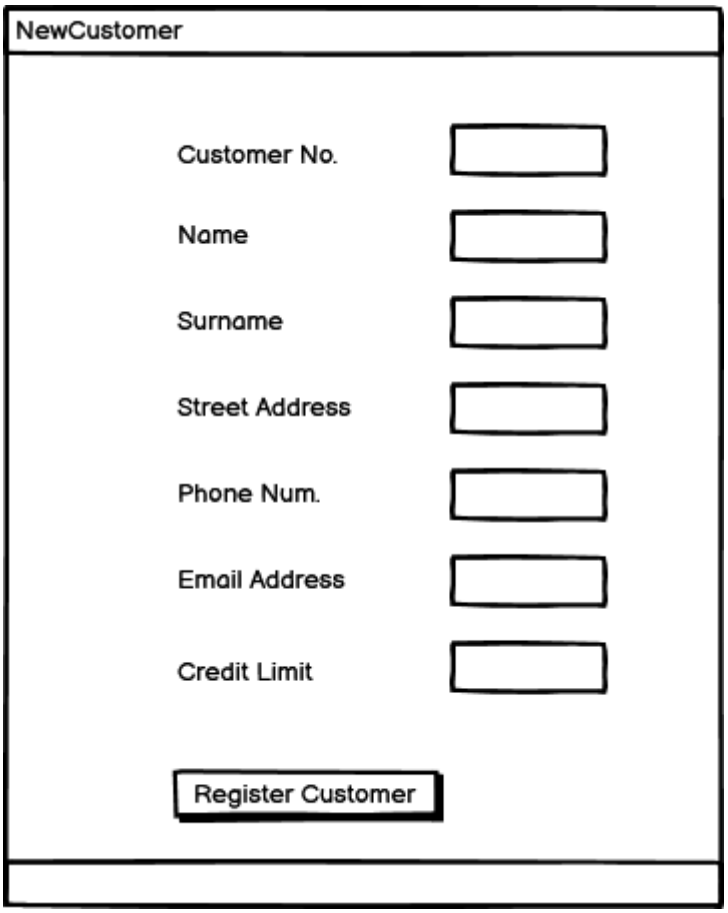
Main Screen



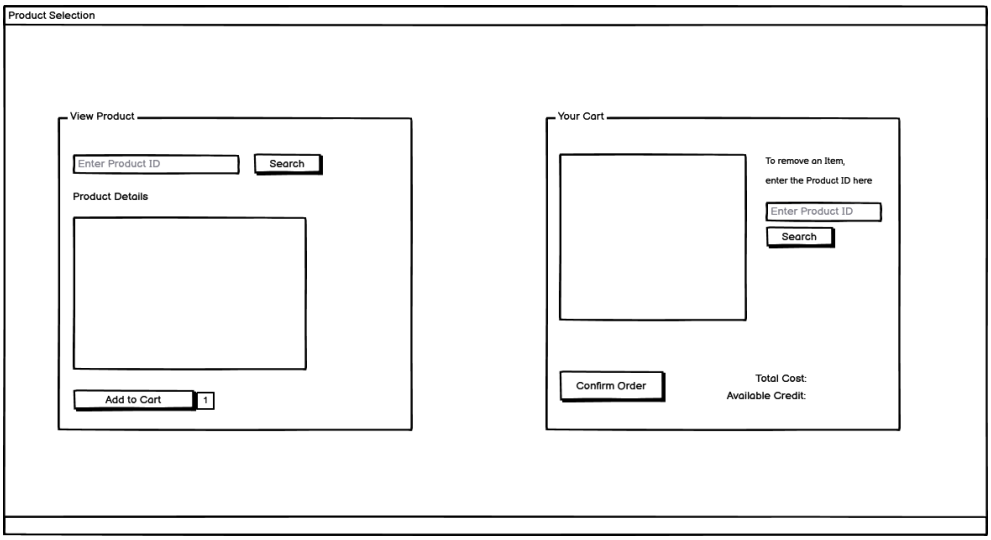
Create Order



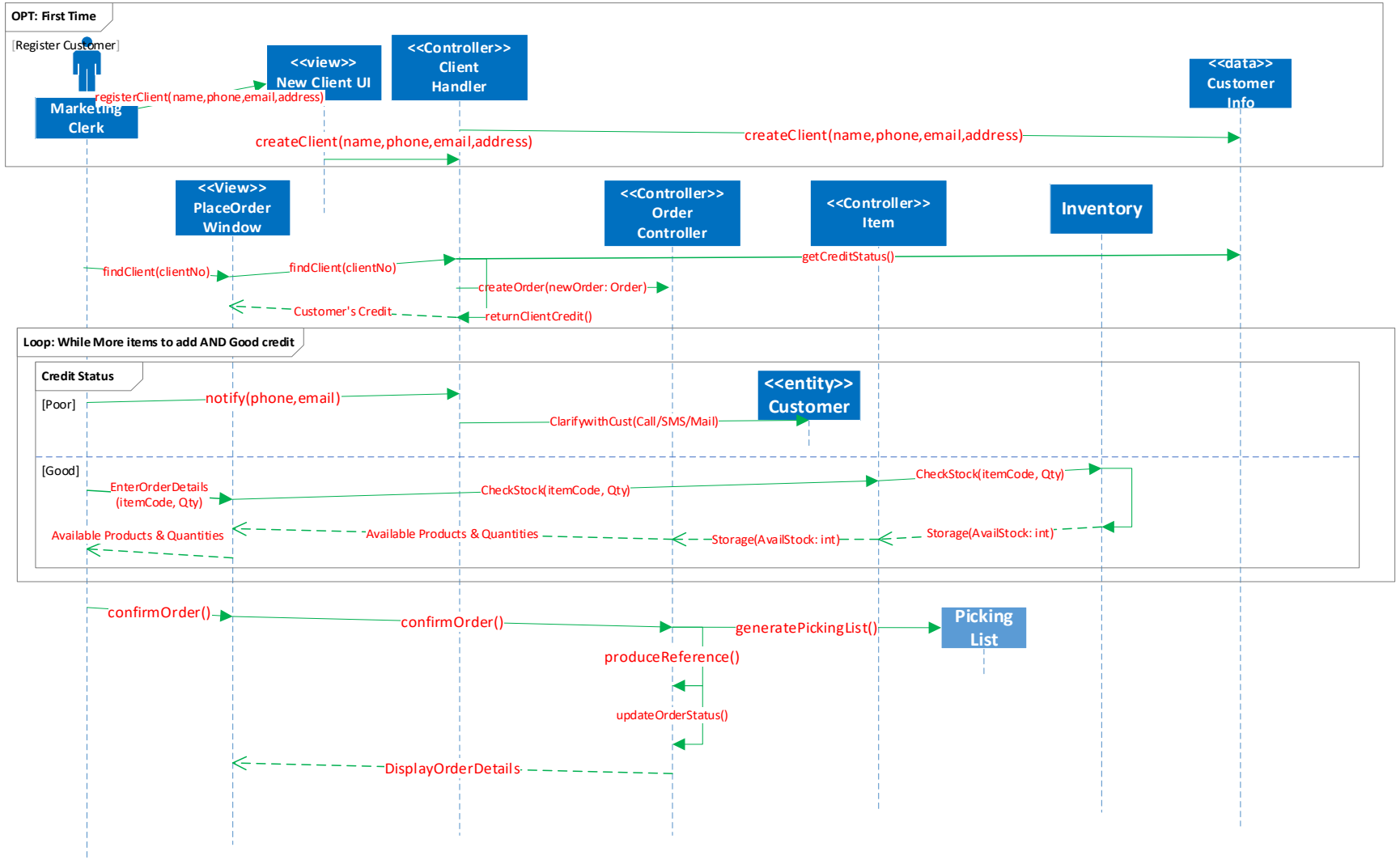
Register Customer



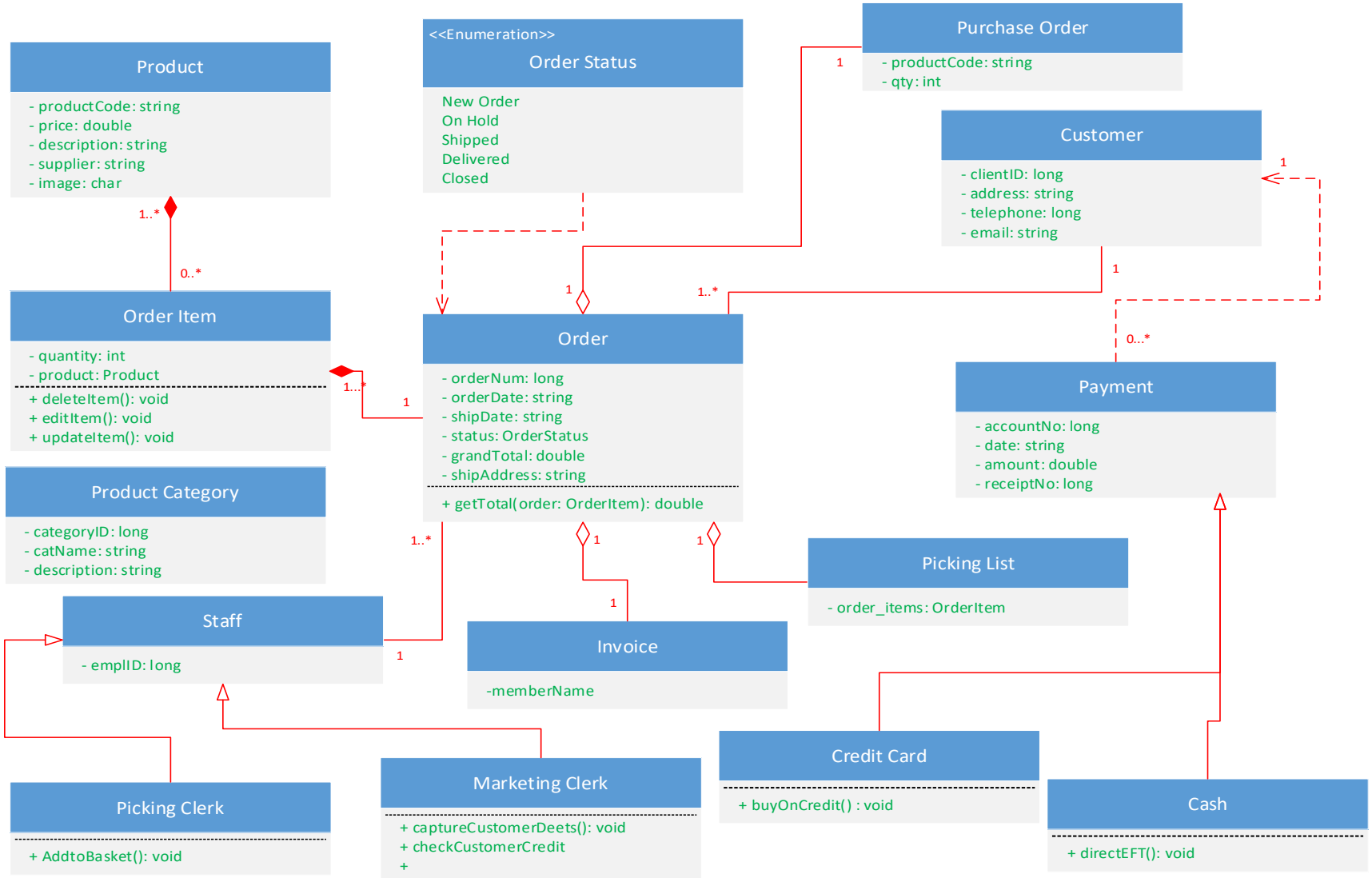
Product Catalogue



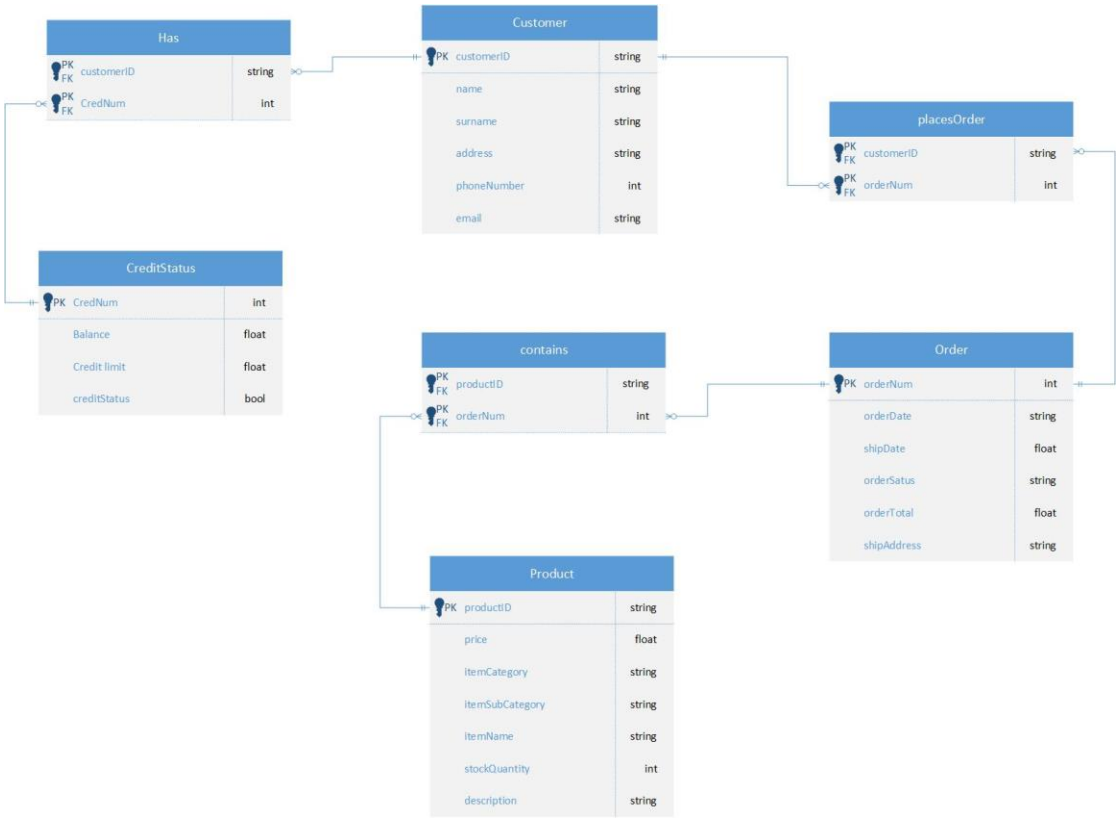
3. DESIGN SEQUENCE DIAGRAMS



4. DESIGN CLASS DIAGRAMS



5. ENTITY RELATIONSHIP DIAGRAM



6. REPORT DESIGN.

6.1. DETAILED OUTPUT REQUIREMENTS

Report type & ID: Operational report

Report Objectives: The objective of this report is to generate a list of items included in an order so that information may be relayed to the picking clerks with regards to which items they should pack

Audience: Picking clerks

Report Content: Content will include information regarding details of each product within an order. This will include the order number, and for each item on the list: The items name, product code, category, and sub-category so that it might be easier for the clerks to find and identify it

Layout: A list of purchased items will be displayed in rows

Selection: The data displayed will be any relevant information that may help the clerk to find the item in the warehouse

Sequence: Items within the order will be organized in the order in which they were added to the cart

Grouping / Summarization: Picking lists will be groups based on priority of when they need to be delivered

Comparison: select_order

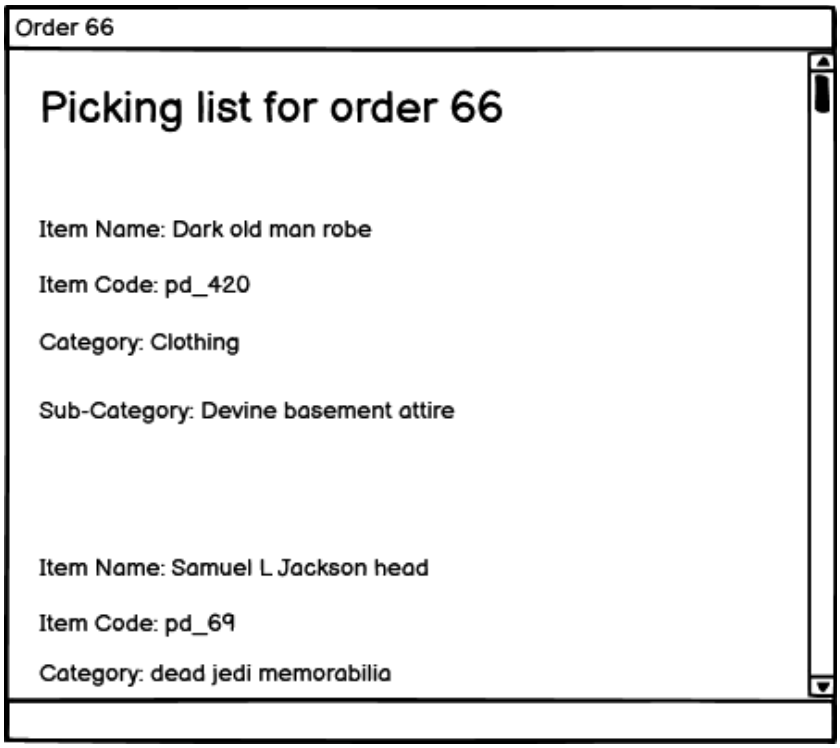
Media to be used: Reports will be generated digitally and be presented to the clerks electronically, or may be printed out physically.

Frequency, Timing, Delivery: Reports will be generated upon the confirmation of an order by the marketing clerk

Distribution: Reports will be delivered to clerks via the internal poppel system

Privacy, security & integrity requirements: Reports should only be available to the picking clerks, the clerk who placed the order... and anybody higher up in the company hierarchy

6.2. REPORT LAYOUT



7. *INPUT-OUTPUT STANDARDS & CONTROLS*

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator. Any additional information may be added to this section and may be organized according to whatever structure best presents the operator input and output designs. Depending on the particular nature of the project, it may be appropriate to repeat these sections at both the subsystem and design module levels. Additional information may be added to the subsections if the suggested lists are inadequate to describe the project inputs and outputs
Developers of sensitive State systems are required to develop specifications for the following minimum levels of control:

- *security to restrict access of critical data items to only those access types required by users*
- *Application audit trails to dynamically audit retrieval access to designated critical data*
- *Standard Tables to be used or requested for validating data fields*
- *Verification processes for additions, deletions, or updates of critical data*

7.1. FORMALISED OUTPUTS:

The system uses message boxes you output information to the user. This is prevalent after each user interaction with the system as each user action will have a system response (opening the correct window or producing confirmation messages). The system also creates a picking list that is stored in the bin directory

7.2. BUILT-IN VALIDATION TO ENSURE REQUIREMENTS ARE MET

To ensure that the customers with bad credit were unable to place orders the place order button was disabled to prevent it. This is just one example of how the system is kept accurate.

7.3. INPUT INTEGRITY CONTROLS

If user entered data of the incorrect type, errors would be caught, and the user would be notified. This prevented the system from crashing. Input validation has been included on each of the text inputs within our system, in order to help minimize garbage in, garbage out scenarios.

There are also various buttons within our forms that are locked out until a certain condition is met. This is in order to help guide the user away from controls that he should not be handling and helps preventing human error.

7.4. OUTPUT INTEGRITY CONTROLS

The system is designed to wait for an order to be confirmed before it commits any potential changes to the database

8. IMPLEMENTATION PLAN

Build Object Classes - To be completed by 11 October

- The first step towards implementation will be the development of the class entities which will form the building blocks of this project

Build database classes – To be completed by 17 October

- The second step will be the development of the database classes. These will store information regarding the Customer and Product entities.

Build UI – To be completed by 23 October

- Next will be the constructing of windows forms in order to create the UIs depicted in the wireframes above

Create functionality between the various layers – To be completed by 29 October

- The next stage will be the development of functionality within the entire program. All layers previously created must be able to link together and communicate with each other in order to present the experience outlined in the system specification

Bug testing - To be completed by 16 November

- The final stage will be the testing of what should be a virtually completed project. The date 4 November is the due date outlined in the specification document however this may be subject to change

Submit

- The completed project will be submitted on the due date

9. TEST PLAN

The plan is to subject the system to a series of tests that emphasize unconventional scenarios that the system may encounter. The aim is to throw these scnearios at our system in the hopes that it may uncover one or two errors before an unconventional user might bump into *them.*

9.1. **TEST ITEMS**

We will be conducting various tests in order to see how our system reacts in response to various scenarios such as: incorrect user input validation (unit testing), insufficient customer credit (integration testing), insufficient stock (integration testing), and customer account creation (integration testing), as well as the reversal of accidental inputs (system testing).

9.2. **PROBLEM TRACKING (TEST CASES)**

ID	Test Scenario	User action	Test data	Expected results	Test Result comments
1	Order capture for invalid(unregistered) customers (not on file)	Customer is not a valid customer as he/she does not have a customer number.	Not applicable	No systems response.	
2	Order capture for invalid customers (incorrect ID format)	Enter customer ID to start order creation	*(Some invalid customer ID)* Mksldmnal	System presents an error message indicated that the customer does not exist, and prompts the user to create a new customer account	
3	Order capture for customers in an unacceptable credit state.	Capture the details of the requested items and their quantities into the system	ML0021 30 *(product code and quantity)*	Return client_creditStatus Should return **on hold**	This would indicate to the marketing clerk that the customer has poor credit status and outstanding payments
4	Order acceptance if the credit limit will be exceeded.	Attempt to accept order	n/a	Throw an error with an error message containing CustCredit	This will inform the user whether the credit limit of R1 500 has been exceeded or not.
5	A product is out of stock	Request a product that is out of stock	n/a	System will indicate that item is out of stock and will not allow the add to cart button to be clicked.	
6	Order acceptance if a customer requests more items than there is available stock	Request an irrationally high number of a item in stock	Item Quantity: 300000	Once the final item from the order is reserved, the system will send out an alert to say that the stock has been depleted	