Design Plan

ThreadCount

Developed for **Nicholas duchon**

CMSC 495

Jennifer Prizeman, ian Chrisman, Justin Jennings, edward caro, Shawn thompson, colin crowley

2016

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision #** | **Revision Date** | **Document Name** | **Changes** |
| 0.1 | October 31, 2016 | Project Plan | Initial Project Plan Created |
| 0.2 | November 2, 2016 | [Project](mailto:jennifer.prizeman@gmail.com) Plan | Edits made |
| [0.3](mailto:jennifer.prizeman@gmail.com) | [November 5, 2016](mailto:jennifer.prizeman@gmail.com) | [Project](mailto:jennifer.prizeman@gmail.com) Plan | Edits made |
| [0.4](mailto:jennifer.prizeman@gmail.com) | [November 6, 2016](mailto:jennifer.prizeman@gmail.com) | [Project](mailto:jennifer.prizeman@gmail.com) Plan | Edits made / Project Submitted |
| 0.5 | November 10, 2016 | User Guide & Test Plan | Initial User Guide & Test Plan Created |
| 0.6 | November 11, 2016 | User Guide & Test Plan | Edits made |
| 0.7 | November 13, 2016 | User Guide & Test Plan | Post-peer review Edits/ User Guide & Test Plan Submitted |
| 0.8 | November 15, 2016 | Project Design | Initial Project Design Created |
| 0.9 | November 16, 2016 | Project Design | Edits made |
| 1.0 | November 19, 2016 | Project Design | Edits made |
| 1.1 | November 20, 2016 | Project Design | User Guide updated  Edits made / Project Submitted |

Table of Contents

[1. Requirements Specification 6](#_Toc467404041)

[1.1 Purpose, Scope, and Objectives 6](#_Toc467404042)

[1.2 Project Background 6](#_Toc467404043)

[1.3 User Profile 6](#_Toc467404044)

[1.4 Assumptions and Constraints 7](#_Toc467404045)

[1.5 Project Deliverables 7](#_Toc467404046)

[1.6 Project Outline 7](#_Toc467404047)

[1.6.1 UML Diagram 8](#_Toc467404048)

[1.6.2 User Scenarios 8](#_Toc467404049)

[2. System Specification 9](#_Toc467404050)

[2.1 Hardware 9](#_Toc467404051)

[2.2 Software 9](#_Toc467404052)

[3. Project Plan 9](#_Toc467404053)

[3.1 Key Milestones 9](#_Toc467404054)

[3.2 Key Roles 11](#_Toc467404055)

[3.3 Upcoming Tasks and Schedule 11](#_Toc467404056)

[3.4 Test Plan and Error Handling 11](#_Toc467404057)

[4. User Guide 12](#_Toc467404058)

[4.1 Main Screen 12](#_Toc467404059)

[4.2 Customer 13](#_Toc467404060)

[4.3 Catalog 14](#_Toc467404061)

[4.4 Sale 15](#_Toc467404062)

[4.5 Reports 16](#_Toc467404063)

[4.6 Inventory 17](#_Toc467404064)

[5. Test Plan 18](#_Toc467404065)

[5.1 Test Team 18](#_Toc467404066)

[5.1.1 Lead Tester Responsibilities (Expanded) 18](#_Toc467404067)

[5.2 Test Environment 18](#_Toc467404068)

[5.3 Test Acceptance Criteria 18](#_Toc467404069)

[5.4 Test Deliverables 19](#_Toc467404070)

[5.4.1 Phase 1 Testing 19](#_Toc467404071)

[5.5 GUI Test Plan 19](#_Toc467404072)

[5.5.1 Scope 20](#_Toc467404073)

[5.5.2 GUI Test Cases 20](#_Toc467404074)

[5.6 Database Test Plan 20](#_Toc467404075)

[5.6.1 Scope 21](#_Toc467404076)

[5.6.2 Database Test Types 21](#_Toc467404077)

[5.6.3 Database Test Approach 21](#_Toc467404078)

[5.6.4 Database Test Activities and Schedule 21](#_Toc467404079)

[5.6.5 Risks 22](#_Toc467404080)

[6. Design 23](#_Toc467404081)

[6.1 Overall Approach 23](#_Toc467404082)

[6.1.1 Workflow 23](#_Toc467404083)

[6.2 Program Structure 24](#_Toc467404084)

[6.2.1 UML Diagram 24](#_Toc467404085)

[6.2.2 Classes, Methods, Fields, & Interfaces 24](#_Toc467404086)

[6.2.3 Class Design 25](#_Toc467404087)

[6.3 User Interface 25](#_Toc467404088)

[6.3.1 Customer module flow chart 25](#_Toc467404089)

[6.3.2 Catalog module flow chart 25](#_Toc467404090)

[6.3.3 Sales module flow chart 26](#_Toc467404091)

[6.3.4 Reports module flow chart 26](#_Toc467404092)

[6.3.5 Inventory module flow chart 26](#_Toc467404093)

[6.4 GUI 27](#_Toc467404094)

[6.4.1 GUI Workflow 27](#_Toc467404095)

[6.4.2 GUI Design 27](#_Toc467404096)

[6.4.3 GUI Components 27](#_Toc467404097)

[6.4.4 GUI Tabs 28](#_Toc467404098)

[6.5 Database 29](#_Toc467404099)

[6.5.1 Sample SQL Schemas 29](#_Toc467404100)

[6.5.2 SQL Queries 29](#_Toc467404101)

[6.5.3 Connectivity 30](#_Toc467404102)

[6.5.4 Database Components 30](#_Toc467404103)

[6.5.5 File Import/Export Mechanism 30](#_Toc467404104)

[6.8 Design Phases 30](#_Toc467404105)

[6.8.1 Phase 1 (Due Nov 27th) 30](#_Toc467404106)

[6.8.2 Phase 2 (Due Dec 4th) 30](#_Toc467404107)

[6.8.3 Phase 3 (Due Dec 11th) 30](#_Toc467404108)

# 1. Requirements Specification

This section will list, in detail, the project objectives, product background, user profile, assumptions, constraints, deliverables, and product outline.

## Purpose, Scope, and Objectives

The objective of the ThreadCount program is to provide software to increase the efficiency of personnel, particularly those who manage small clothing boutiques. By providing a software application to manage inventory, sales, and customer information, the program will save time and effort for the user. Inventory and balance will be adjusted in the database each time the user enters a transaction. The user interface will enable the application user to display various reports about sales, customer, and inventory data. These reports will give the salesperson pertinent information, such as their current inventory and shipment information.

## Project Background

The direct sales model is making a comeback. With social media being widely available, the sales and marketing opportunities to sell products to friends and acquaintances has exploded. This has made it easier now than ever to start and run a home business. This is a particularly attractive idea for stay at home parents seeking to supplement family income and have fun at the same time. However, these homes are rarely equipped with business class IT infrastructure or software to help run the business. While inexpensive software exists for generic tasks such as accounting and taxes, these new small business owners lack the ability to properly track inventory[[1]](#footnote-1), and create important reports that help run the business. Our project will create a simple, inexpensive solution to help these new businesses flourish.

## User Profile

Sheri McCoy, the CEO of AVON products, explained “[Our Founder David H. McConnell] understood that women were natural salespeople who could easily relate to other women and passionately market beauty products.”[[2]](#footnote-2) Likewise, and based on our research, the largest demographic for creating new direct sales businesses are women aged 22-40. They are typically suburban Generation X or Millennials. Additionally, our target users are from affluent households with enough disposable income to purchase sufficient inventory to launch the business. However, we see the lure of opening a business decline the wealthier that family is. As a result, the average income for the target audience has incomes roughly in the range of $50k - $150k per year. These are important considerations when considering several aspects of the software solution. First, the application’s look and feel should appeal to young women. Second, the complexity should be very low, as this would drive up the cost of the software and become ultimately unaffordable to the target group. Ideally, the software will cost around $10 per month. Lastly, this segment is not typically formally trained with IT products, so the software should be very easy to use and navigate.

## Assumptions and Constraints

For the development of the software, it is assumed that an information technology (IT) deployment exists in order for the software to be installed. Additionally, the development of the software is designed around portability. By focusing on the use of Java language and minimizing the user interface requirements, this software can be portable to almost all operating system platforms.

It is assumed the end users will not be familiar with information security best practices and will use the software to store private customer information and business sales information. The development team will evaluate these functional requirements to establish appropriate system security constraints in order to ensure confidentiality, integrity and availability of the information.

## Project Deliverables

Project deliverables for the software package are focused on the following:

1. Software package consisting of:
   1. Main GUI
   2. Item Creator
   3. Supplier Creator
   4. Shipment Creator
   5. Inventory Display
   6. Customer Display
   7. Report Builder
2. User guide and manual

The software delivery will consist of a Java application with executable shell scripts for Mac OS and Microsoft Windows. This software will also contain an example configuration file for database connectivity, with configuration options for MySQL. Considering our target user, our goal is to deliver a final software product that is user friendly. The software will be made available for download electronically for customers wishing to receive digital media. The user guide will be delivered via electronic download as well, in the form of a Microsoft Word or PDF document.

Successful delivery of these features is dependent on testing acceptance after the software has passed key milestones where the software is versioned (alpha, beta, version 1.0).

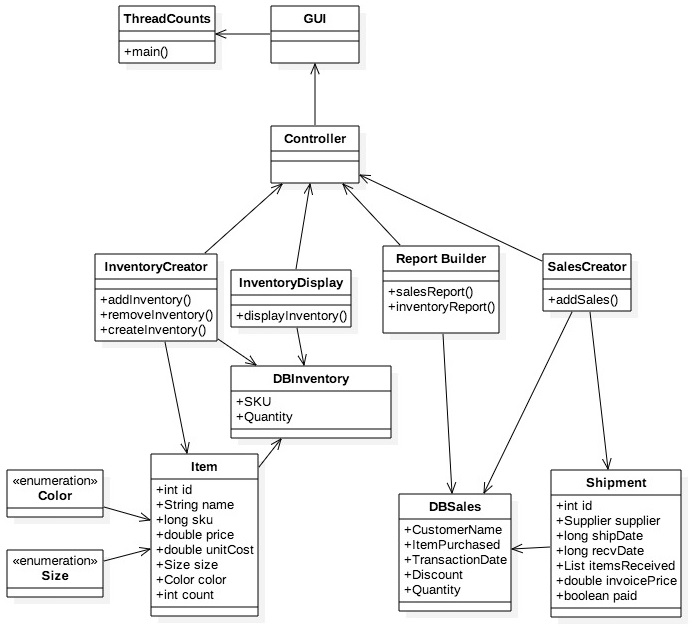
## Project Outline

The basic outline of the ThreadCount application will be as follows:

* GUI Code – Will be generic, calling methods in other classes and passing data.
* Two creator classes; Inventory and Sales
* Basic setup of database(s):
  + Inventory Table columns: include SKU, Style, Size, Wholesale Price, Retail Price, Quantity
  + Sales Table columns: Table key/transaction number, Date, Customer Name, Item purchased, Sales Balance, Address, Email

### 1.6.1 UML Diagram

The UML diagram below depicts the relationships and dependencies among classes for the project’s source code.



***Figure 1. UML diagram***

### 1.6.2 User Scenarios

The software will support 5 major user scenarios. It will allow the user to obtain and manipulate information regarding sales, customers, inventory, catalog, and various reports.

#### 1.6.2.1 Customers

The application will maintain a list of customers and a history for each customer. From the Customer screen, the user will be able to search for a customer by name, address, email, etc. Further, the user will need to get past sales history about customers. For instance, the user will want to be able to identify customers that buy frequently.

#### 1.6.2.2 Catalog

A complete list of items that are possible to be sold through the business will be maintained by the application. These items may or may not be in stock. The user will need to be able to search and find items by searching characteristics of the products.

#### 1.6.2.3 Sales

The user will be able to assemble a list of items for a customer’s purchase. Items can be selected from the pre-existing catalog and dropped into a ‘shopping cart’. Once the item is in the cart, quantities and pricing can be adjusted if it needs to be different than the default.

#### 1.6.2.4 Reports

A report screen will offer the user a set of commonly used reports. These will be preconfigured for the user. Reports such as sales history, customer history, and current inventory will be available.

#### 1.6.2.5 Inventory

The program will keep a list of products that are currently in inventory. From the inventory screen the user will be able to tell what products that they can currently offer to customers. They will need to be able to search by characteristics like clothing type (shirt, pants), size, color, etc. so that they can find suitable offerings for a particular customer’s needs.

# 2. System Specification

This section will list the required hardware and software that will be used to complete the ThreadCount application.

## 2.1 Hardware

* Hardware consisting of:
  + Windows PC
  + MacBook Air using Mac OS

## 2.2 Software

* Software consisting of:
  + Language – Java SE Development Kit (JDK) 8 and associated libraries
  + Database Implementations – MySQL
  + Integrated Development Environment (IDE) – NetBeans 8.1
  + Source Code Repository – GitHub
  + Documentation – Microsoft Word and Adobe Acrobat

# 3. Project Plan

The project plan provides an outline of key milestones, key roles, upcoming tasks, schedule, test plan, and error handling.

## 3.1 Key Milestones

Milestones will be structured so that they coincide with the weekly required documents due.

| **Weekly Document Requirements** | **Due Date** | **Assigned to** |
| --- | --- | --- |
| Project Plan | November 6, 2016 | Edward/Jennifer |
| Test Plan (Front End) | November 13, 2016 | Edward |
| Test Plan (Back End) | November 13, 2016 | Colin |
| User Guide | November 13, 2016 | Shawn |
| Design | November 20, 2016 | Edward/Jennifer (group input) |
| Phase I Source Documentation | November 27, 2016 | Group |
| Phase II Source Documentation | December 4, 2016 | Group |
| Phase III Source Documentation | December 11, 2016 | Group |
| Final Documentation | December 18, 2016 | Edward/Jennifer |

***Figure 2. Document Milestones***

| **Project Milestones** | **Due Date** | **Assigned to** | **Complete Y / N** |
| --- | --- | --- | --- |
| Create User Profile | November 4, 2016 | Shawn | Y |
| Create Product Background Info | November 4, 2016 | Shawn | Y |
| Create UML Diagram | November 5, 2016 | Ian | Y |
| Create GUI Sketch | November 6, 2016 | Shawn | Y |
| Create database with test data | November 13, 2016 | Justin | Y |
| Display inventory from test database | November 15, 2016 | Ian | N |
| Display different report types from database | November 15, 2016 | Ian/Justin | N |
| Enter information into database | November 17, 2016 | Justin | N |
| Modify information in database | November 25, 2016 | Ian | N |
| Display inventory from real database | December 1, 2016 | Justin | N |
| Display different types of reports from real database | December 1, 2016 | Ian | N |
| Backup and encrypt database information | December 8, 2016 | Colin | N |

***Figure 3. Project Milestones***

## 3.2 Key Roles

There are six members on this project: Jennifer Prizeman, Ian Chrisman, Justin Jennings, Edward Caro, Shawn Thompson, and Colin Crowley. Each member of the team is assigned a specific lead role along with assisting others on various aspects during the development cycle of the project. Each team member's specific lead role is given in *Figure 4*, below, along with additional duties which are italicized.

| **Roles** | **Name** | **Contact Info** |
| --- | --- | --- |
| Project Manager / *Documentation* | Jennifer Prizeman | jennifer.prizeman@gmail.com |
| Lead Front End Coding | Ian Chrisman | ichrisman@student.umuc.edu |
| Lead Back End Coding | Justin Jennings | justin.robert.jennings@gmail.com |
| Documentation / *Front End Testing* | Edward Caro | edward.m.caro@gmail.com |
| User Guide / *Front End Coding* | Shawn Thompson | stt9000@gmail.com |
| Security / *Back End Coding* / *Back End Testing* | Colin Crowley | colin.t.crowley@gmail.com |

***Figure 4. Team Roles***

## 3.3 Upcoming Tasks and Schedule

Listed below are the upcoming tasks for the week.

* November 15: Create Project Design Outline
* November 10-17: Code front end GUI and back end Database
* November 17: Class (1st half) – review and finalize Project Design; (2nd half) – discuss assignment due Sunday 11/27 and responsibilities relating thereto.
* November 18-19: Discuss and submit any modifications to Project Design
* November 20: Submit final Project Design to UMUC Assignments Folder
* November 21-24: Draft Phase 1 Source
* November 25: Post current Phase 1 Source document and source code for Peer Review 2
* November 26-27: Discuss and submit any modifications to Phase 1 Source
* November 27: Submit final Phase 1 Source to UMUC Assignments Folder

## 3.4 Test Plan and Error Handling

Below are possible situations for upcoming Test Plan and User’s Guide assignment.

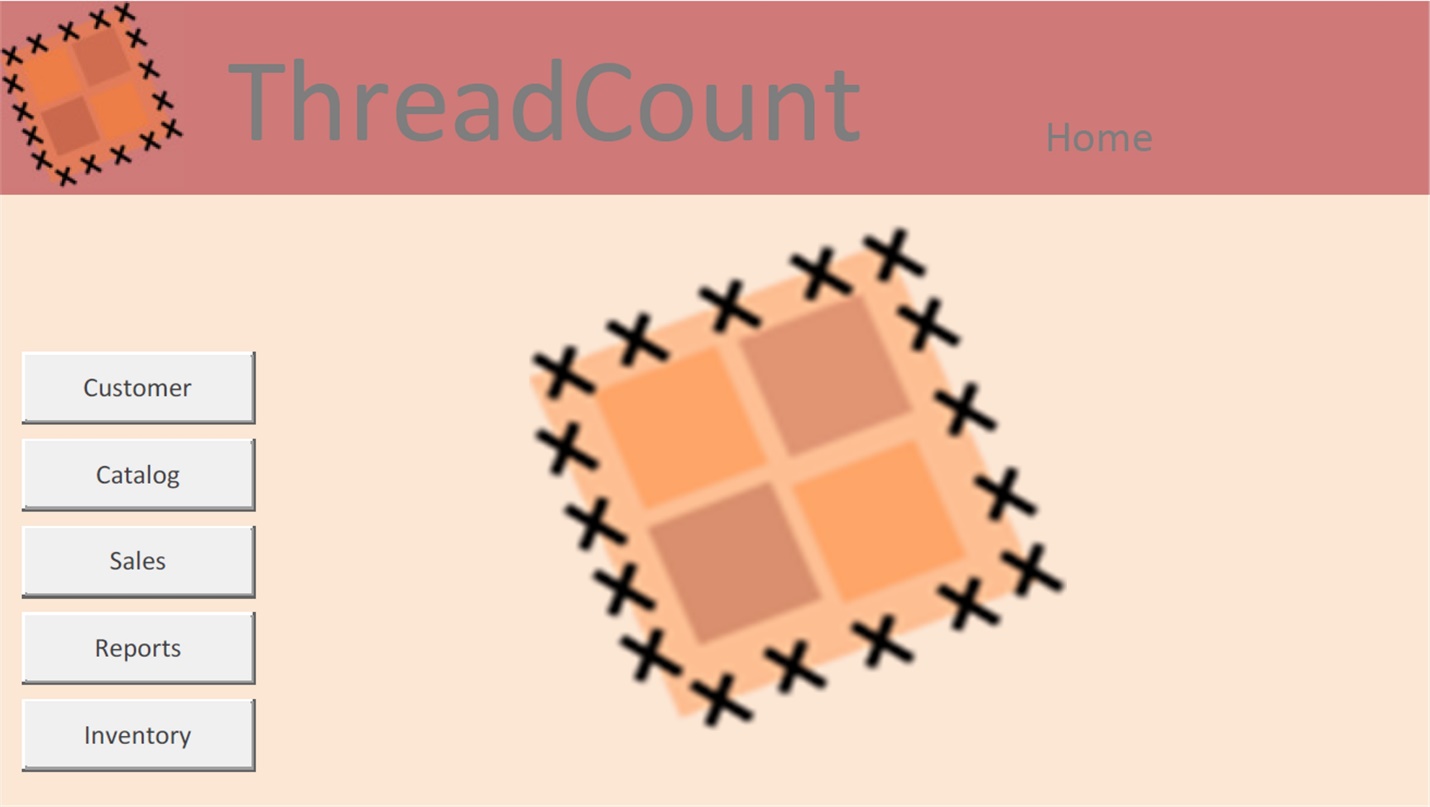
* Selling more inventory than is in inventory (overselling)
* Changing price / sales / discounts
* Modifying entry in database
* Handling incorrect entries. For example, anything other than numerical input for price, numerical input for customer name, etc.
* Backing up customer and inventory databases
* Restoring customer and inventory databases from backup
* Encrypting/decrypting customer database information

# 4. User Guide

This software will assist home based business to track inventory, customers, and record sales. It will provide insights into the business that will aid in decision making such as when and how much inventory should be purchased. The software has an easy to use interface and can be run on standard desktop or laptop computers. An internet connection is required to use the software. All data is kept in the cloud so there is no need to worry about data loss if something happens to the computer.

After clicking on the ThreadCount icon, the software will open and take you to the main navigation screen. From this screen, you can navigate to any of the 5 modules – Customer, Catalog, Sales, Reports, and Inventory.

## 4.1 Main Screen



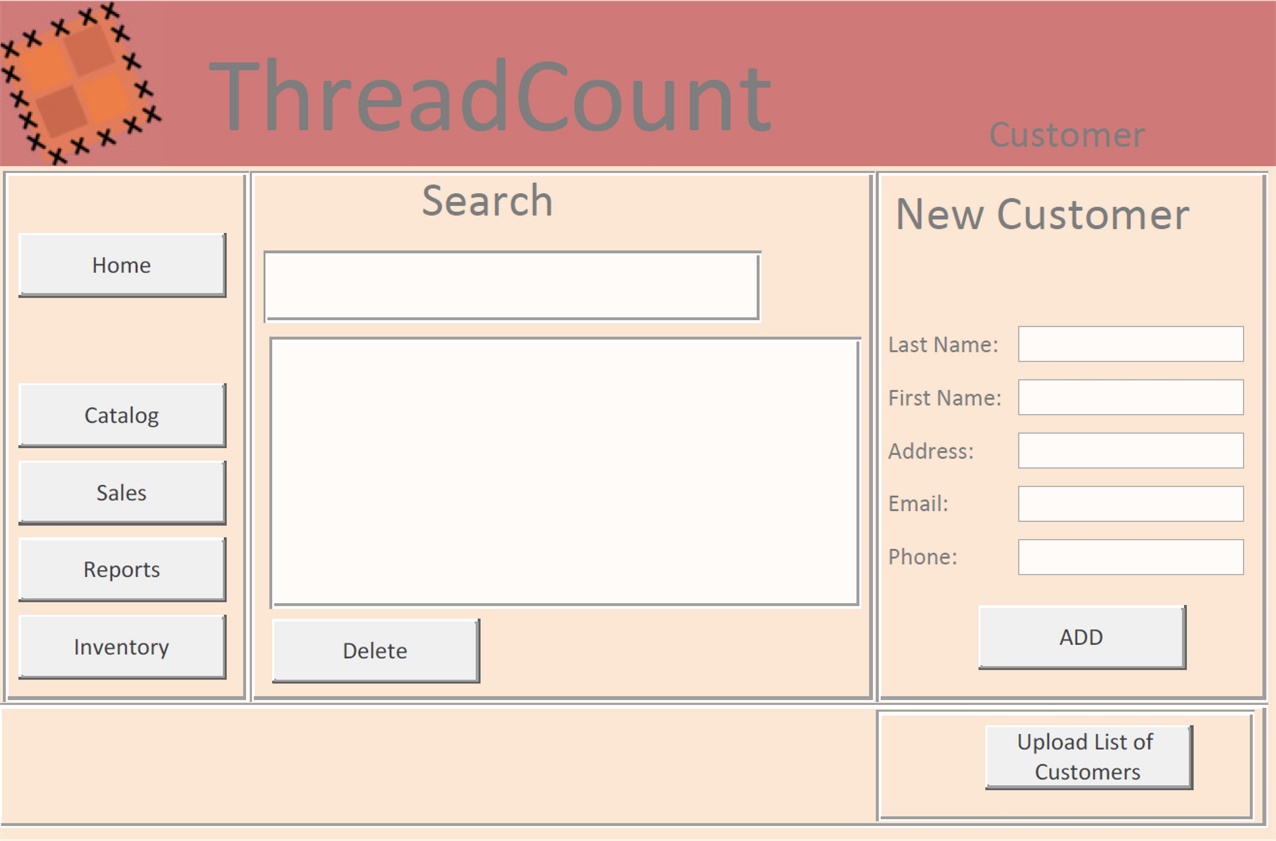
***Figure 5. ThreadCount Home screen***

* Customer – The Customer module provides information about your prospective clients. Here, you will be able to find and store information about your customer, like names, addresses, emails, etc. You can also add, edit, and delete customers.
* Catalog – The Catalog module provides information about any of the products that your business sells. This module will provide information like type, size, color, etc., of the clothing items that you offer for sale. You can add, edit, or delete items from your portfolio.
* Sales – Use this module to enter new sales.
* Reports – Several reports are at your fingertips. Use this module to find out how much in sales your business did last month, last year, and other useful information that helps you grow your business.
* Inventory – This module allows you to see what you currently have in your inventory. You can search to find items that meet your customers’ needs.

Choose one of the modules that are displayed on the left side of the screen. This will take you to a new screen where you can find and edit items related to that category. Go ahead and click the Customer button!

## 4.2 Customer

The Customer module provides information about your prospective clients. Here, you will be able to find and store information about your customer, like names, addresses, emails, etc. You can also add, edit, and delete customers.



***Figure 6. ThreadCount Customer screen***

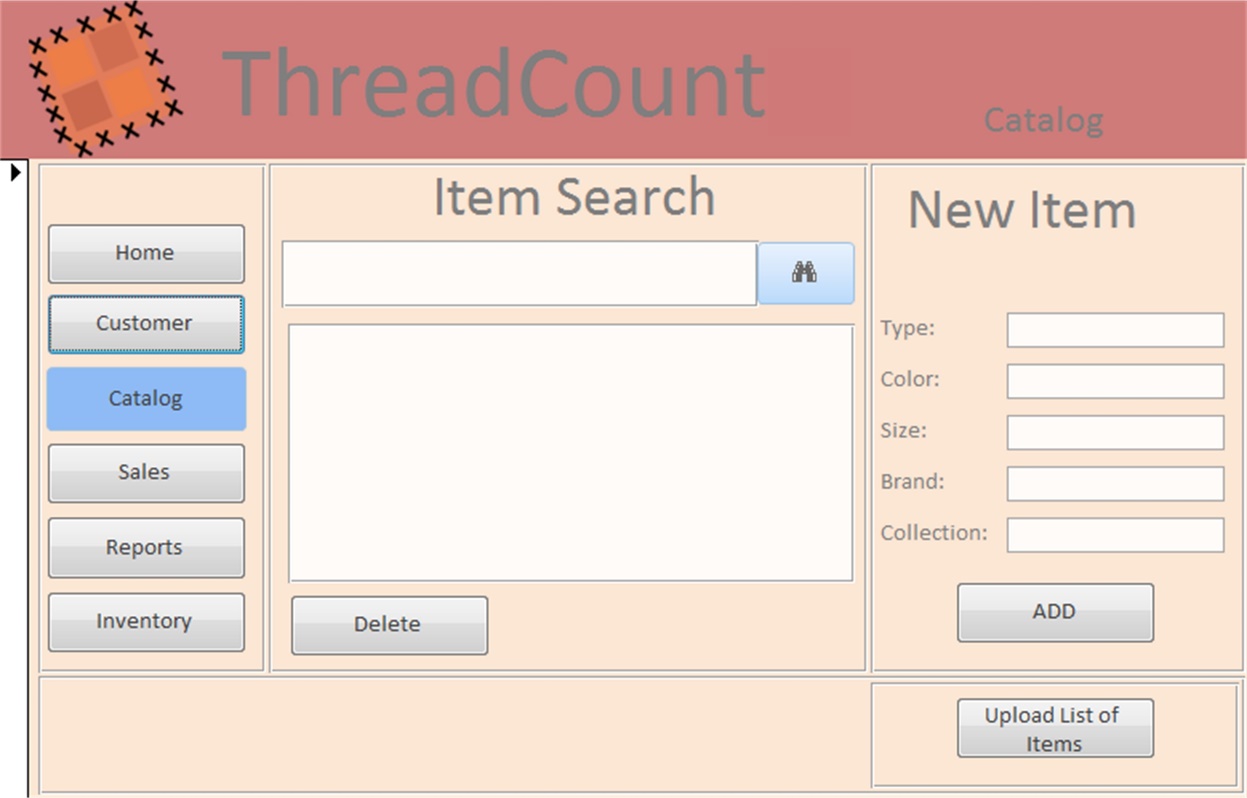
There are 3 main tasks that can be performed in the Customer module:

* Search – Enter any part of a name or email address, and any entries that match the search will be displayed in the results area.
* Print shipping label – After searching and finding the correct customer, click on the customer in the results area. Now click the Print Shipping Label button and a shipping label will be printed for you.
* Add new customer – Enter new customer information to add a new customer to your database!
* To add a list of customers, click the Upload List of Customers Button, which specifies acceptable file formats.

Go ahead and add a new customer. Don’t worry, if it is a fictitious for now, you can delete it later. After adding the customer, try to search for it using the search tool.

## 4.3 Catalog

The Catalog module provides information about any of the products that your business sells. This module will provide information like type, size, color, etc. of the clothing items that you offer for sale. You can add, edit, or delete items from your portfolio.



***Figure 7. ThreadCount Catalog screen***

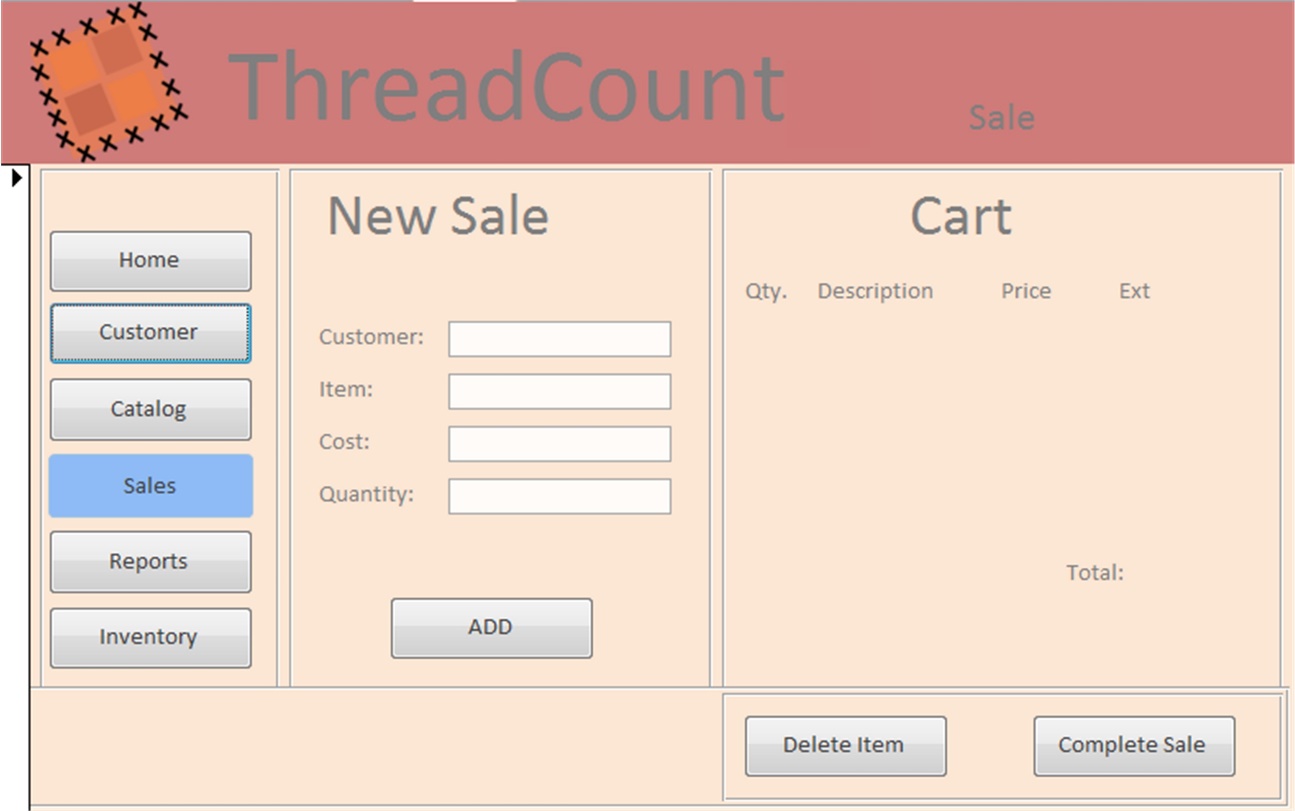
There are 3 main tasks that can be performed in the Catalog module:

* Search – Enter any part of a name or type of clothing, and any matching items will appear in the Results area.
* Delete – After searching and finding the item that you want to delete, select the item by clicking on it in the results area. Now click the Delete button and the item will be removed from your catalog.
* Add new item – Enter new item information to add a new product to your sales offerings!
* To add a list of items, click the Upload List of Items Button, which specifies acceptable file formats.

Go ahead and add an item to your product offering. Don’t worry if its fictitious for now, you can delete it later. After adding the item, try searching and deleting it!

## 4.4 Sale

The Sales module is used to enter new sales.



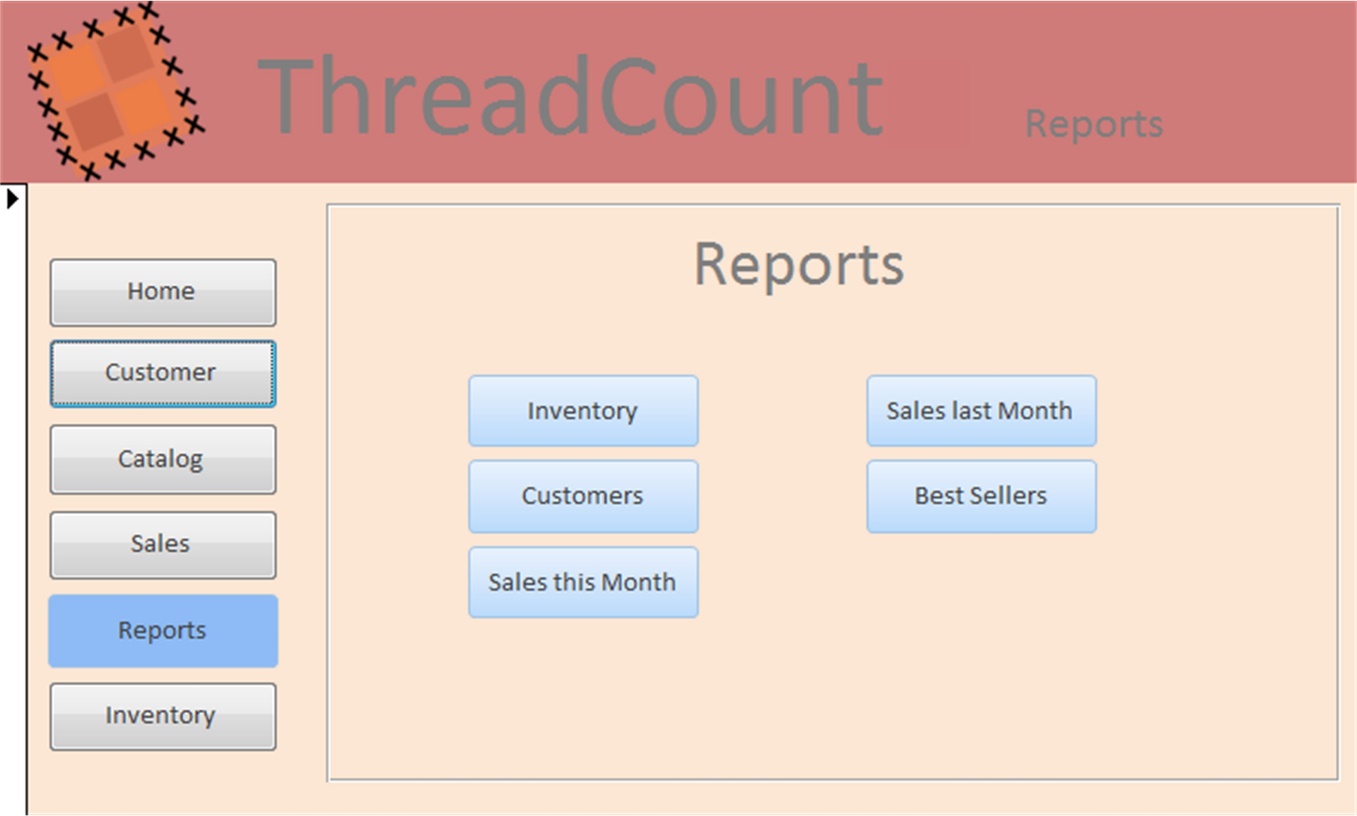
***Figure 8. ThreadCount Sale screen***

The Sale module allows you to create a shopping cart and execute a sale to a customer:

* Item is a drop-down menu that shows the items are currently in your inventory.
* When selected, the Cost is automatically populated with the suggested price, but you can update this to any cost you want.
* Customer is a drop-down menu allowing you to choose the correct customer that you are making the sale to.
* After filling out the customer, item, price and quantity, click the add button to add the listing to the shopping cart.
* After the shopping cart is correct, click on Complete Sale to finalize the transaction.

## 4.5 Reports

Use this module to find out how much in sales your business did last month, last year, and other useful information that helps you grow your business.



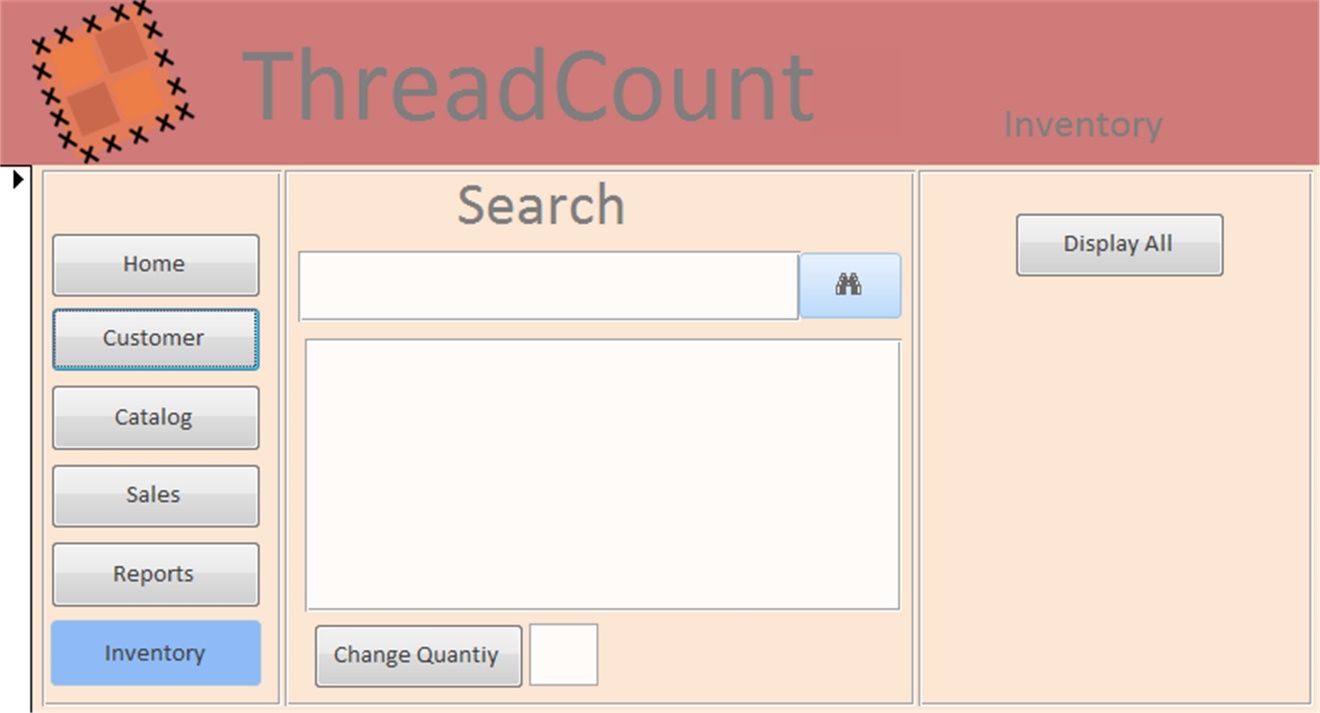
***Figure 9. ThreadCount Reports screen***

The Report module provides information and key insights about your business:

* Inventory – This report will give you a complete listing of your current inventory.
* Customers –This report outputs a complete listing of all your customers in alphabetical order.
* Sales this Month – Outputs a report with month-to-date statistics.
* Sales last Month – Outputs a sales report with lasts month’s statistics.
* Best Sellers – Creates a list of the top ten inventory sellers.
* Time to Depletion – Helps determine when you should place new orders to add to, or replenish, your inventory.

## 4.6 Inventory

The Inventory module allows you to see what you currently have in your inventory. You can search to find items that meet your customers’ needs. After searching, click on the item to manually change the quantity currently in inventory. You can also click the Display All button to see a complete list.



***Figure 10. ThreadCount Inventory screen***

# 5. Test Plan

This test plan will illustrate the testing plan, approach, and overall framework that will drive the testing of ThreadCount. The Test Plan includes:

* Test Team
* Test Environment
* Test Acceptance Criteria
* Test Deliverables
* Graphical User Interface (GUI) Test Plan and Test Cases
* Database Test Plan and Test Types

## 5.1 Test Team

The table below shows the members of the Testing Team. Coding Leads will collaborate with their respective Testing Leads in order to get the project fully functional.

| **Team Member** | **Lead Area** | **Test Responsibilities** |
| --- | --- | --- |
| Ian Chrisman | Front End Coding | Create and manage front end coding of project |
| Justin Jennings | Back End Coding | Create and manage back end coding of project |
| Edward Caro | GUI Testing | Test front end of project |
| Colin Crowley | Database Testing | Test back end of project |

***Figure 11. Testing Responsibilities***

### 5.1.1 Lead Tester Responsibilities (Expanded)

Creates the test plan, creates and reviews test data and queries, analyzes and complies test results for final deliverable, conducts test queries, documents individual test results

## 5.2 Test Environment

In order to fully test the application, the following will be required for the testing environment. Each team member will take part in testing to some degree. This is done to test a variety of operating systems and other variables that might not appear one just one system.

| **Requirements** | **Client Side** |
| --- | --- |
| Hardware | Intel/AMD processor 1GHz+ |
| Operating System | Windows 7, 10, Mac OS X |
| Software | JDK 8.0, NetBeans IDE |

***Figure12. Test Environment***

## 5.3 Test Acceptance Criteria

Testing will use Pass/Fail criteria for all test cases. A test case will only be considered passed if the result matched the expected result. Otherwise it will be considered failed.

## 5.4 Test Deliverables

The following deliverables will be updated and added to throughout the lifecycle of the project.

| **Deliverable** | **Description** | **Due Date** |
| --- | --- | --- |
| Test Plan | Document that provides an overall guidance of testing efforts throughout the project’s development cycle | November 13, 2016 |
| Phase 1 Test Report | Testing to ensure coding of Phase 1 is working correctly | November 18, 2016 |
| Phase 2 Test Report | Continued testing of Phase 1 along with the testing of Phase 2 coding | December 1, 2016 |
| Phase 3 Test Report | Final Testing to ensure all coding in the project is working as designed | December 8, 2016 |

***Figure 13. Testing Deliverables***

### 5.4.1 Phase 1 Testing

| **Phase 1** | | | |
| --- | --- | --- | --- |
| **Task** | **Description** | **Assigned to** | **Due Date** |
| GUI Testing | * Test the five module buttons and Home button work * Test the functionality of the five module buttons thoroughly | Edward | November 18, 2016 |
| Database Testing | * Input tests * Retrieval tests * Report tests | Colin | November 18, 2016 |
| Execute Testing Plan | Run tests and document any problems | Colin/Edward | November 20, 2016 |
| Phase 1 Testing Report | Compile a report of the documented problems | Colin/Edward | November 25, 2016 |
| Review of Report | Review the report and make any necessary changes to the code | Group | November 26, 2016 |

***Figure 14. Phase 1 Testing***

## 5.5 GUI Test Plan

The objective of the GUI test plan is to assure that the ThreadCount software application is working correctly according to the User’s guide in section 4. This section is subject to change. Additional functionality may be added to the project or functions may be removed as needed. There is a back end portion for this application, but it is out of scope for the GUI test and accounted for in the Database test plan.

### 5.5.1 Scope

The purpose of this testing is to make sure the GUI is working and interfacing correctly with other modules. This testing is carried out by clicking the various module buttons to ensure the output is as expected.

### 5.5.2 GUI Test Cases

Below are the current test cases for the GUI portion of the project. The list will change as the project progresses.

| **GUI Test Cases** | | | | | |
| --- | --- | --- | --- | --- | --- |
| **Test #** | **Test Objective** | **Test Steps** | **Expected Result** | **P/F** | **Comments** |
| 1 | Start Application | Double Click on the ThreadCount icon, or right click and select open | The ThreadCount application loads on system |  |  |
| 2 | Check that the Customer button opens | Click the Customer button | A new screen will open where customer information can be entered |  |  |
| 3 | Check that the Home button works | From the Customer screen, click the Home button on the left side of the screen | The screen should return to the Home screen |  |  |
| 4 | Check that the Catalog button opens | Click the Catalog button | A new screen will open where Catalog information can be entered |  |  |
| 5 | Check that the Sales button opens | From the home screen, click the Sales button | A new screen will open where Sales information can be entered |  |  |
| 6 | Check that the Reports button opens | From the home screen, click the Reports button | A new screen will open where Reports information can be entered |  |  |
| 7 | Check that the Inventory button opens | From the home screen, click the Inventory button | A new screen will open where Inventory information can be entered |  |  |

***Figure 15. GUI Test Cases***

## 5.6 Database Test Plan

This is the test plan for the back end Database (“DB”) of the ThreadCount inventory management software application. ThreadCount is used to enable analysis, storage and retrieval of customer and inventory information for a small-to-mid sized clothing retailer. There is a front end GUI interface for this application, but it is out of scope for the DB test and accounted for in the GUI test plan.

### 5.6.1 Scope

The following information is critical to the performance of the inventory management system database:

* The type fidelity of DB information
* That the DB schema accurately captures the required inventory and customer information

### 5.6.2 Database Test Types

* Input tests
  + Input tests will use MySQL commands to determine if customer and inventory data is correctly formatted and saved in the DB according to the design requirements
* Retrieval tests
  + Retrieval tests will use MySQL queries to determine if customer and inventory data is:
    - Retrieved and not altered during queries
    - Correctly pulls the needed data
* Report tests
  + Report tests will use MySQL queries to determine if customer and inventory data is:
    - Displayed in the correct format
    - Displays the correct DB fields and attributes
    - Completes in a reasonable time frame

### 5.6.3 Database Test Approach

* Customer Information tests
  + SQL command to add Last Name, First Name, Address, Email, Phone info
  + SQL command to get Last Name, First Name, Address, Email, Phone info
* Catalog Information tests
  + SQL command to add Type, Color, Size, Brand, Collection info
  + SQL command to get Type, Color, Size, Brand, Collection info
* Sales Information tests
  + SQL command to add Customer, Item, Cost, Quantity info
* Reports Information tests
  + SQL command to generate inventory report
  + SQL command to generate customer report
  + SQL command to generate sales this month report
  + SQL command to generate sales last month report
  + SQL command to generate best sellers report
* Inventory Information tests
  + SQL command to generate inventory report

### 5.6.4 Database Test Activities and Schedule

* Analyze and review database requirements (required input)
* Prioritize risks and identify most likely scenarios (required input)
* Create test queries based on scenarios
* Create sample test data
* Finalize test queries
* Finalize sample test data
* Conduct test approach for all five test scenarios
* Compile and analyze test results
* Prepare final deliverables

### 5.6.5 Risks

The following items and events present risk to successful test implementation and completion:

* Insufficient test DB size. Ex., small sample DB vs. one of similar size and complexity to real DB
  + Mitigated by random data generation
* Insufficient number of test inputs to validate schema
  + Conduct analysis to identify most likely input scenarios

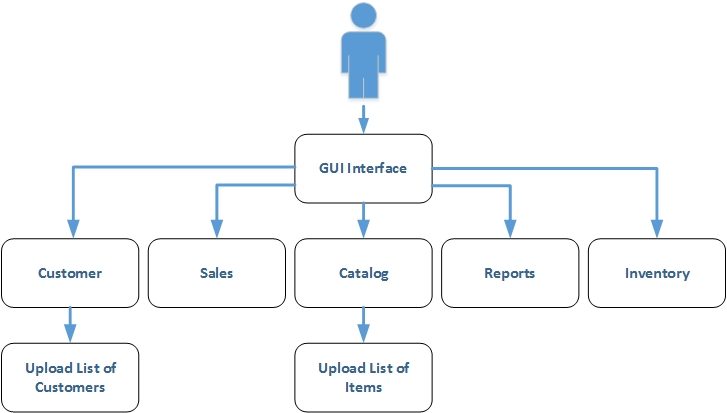
# 6. Design

The ThreadCount software application utilizes a GUI interface to allow the user to manage inventory, sales, and customer information for a small clothing boutique. The user interface will enable the application user to display various reports about sales, customer, and inventory data.

This section will provide specifications for the system architecture in a modular format along with detailing the class and structure design of the application. Diagrams and descriptions are provided to illustrate the program’s functionality.

## 6.1 Overall Approach

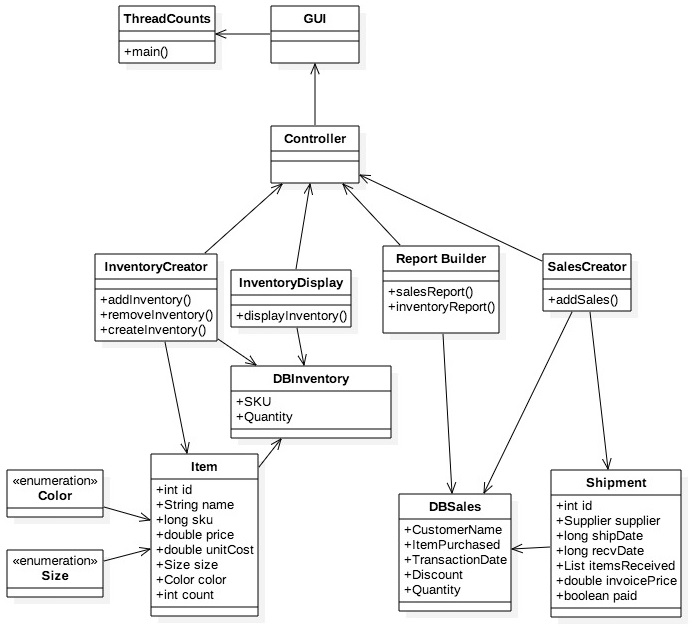
### 6.1.1 Workflow



***Figure 16. Workflow diagram***

## 6.2 Program Structure

### 6.2.1 UML Diagram



***Figure 17. UML diagram***

### 6.2.2 Classes, Methods, Fields, & Interfaces

* Class ThreadCount
* Class ThreadGUI
* Class Item
* Class Customer
* Class ReportBuilder
* Class Controller
* Class Config

### 6.2.3 Class Design

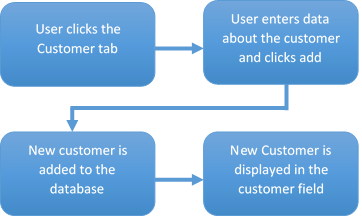
| **Class** | **Description** |
| --- | --- |
| ThreadCount | This class is the starting point for the application. It instantiates the GUI. |
| ThreadGUI | This class defines the GUI for the application, and sets up ActionListeners, etc. |
| Item | This class contains the data structure for a catalog item, as well as related methods. |
| Customer | This class contains the data structure for Customer information, as well as related methods. |
| ReportBuilder | This class will be used to generate various reports on data points of interest to the user (for example monthly sales, profits, etc.). |
| Controller | This class does all the SQL work, object instantiation, etc., and exposes useful methods to the GUI class. |

***Figure 18. Class Design table***

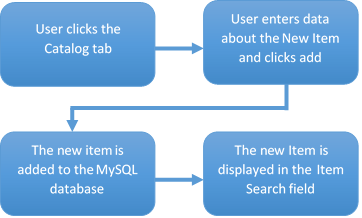
## 6.3 User Interface

The user interface will be a single window that dynamically changes depending upon which module is selected.

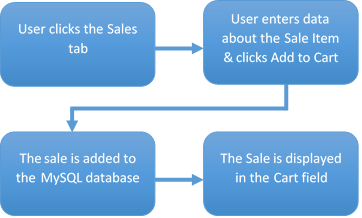
### 6.3.1 Customer module flow chart



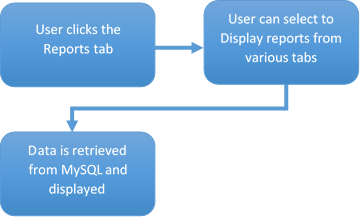
### 6.3.2 Catalog module flow chart



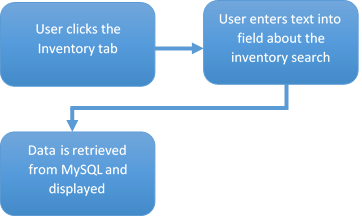
### 6.3.3 Sales module flow chart



### 6.3.4 Reports module flow chart

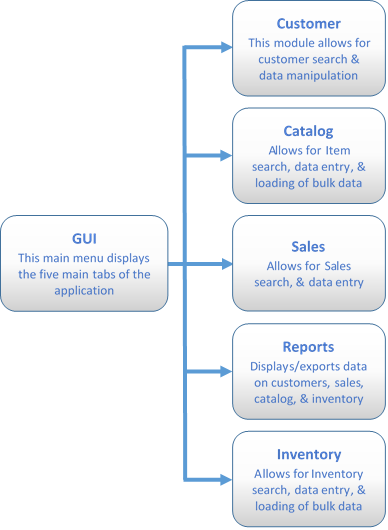


### 6.3.5 Inventory module flow chart



## 6.4 GUI

### 6.4.1 GUI Workflow



### 6.4.2 GUI Design

The design of the GUI provides visual and conceptual elements for the user in the form of Java Buttons, Text Fields, and Labels. The GUI is designed to take value inputs from the user to manipulate data within the MySQL database.

### 6.4.3 GUI Components

| **Component** | **Description** |
| --- | --- |
| JFrame | Creates a top-level window with a border and title, in which other GUI components are placed. |
| JTabbedPane | Allows several panels to share the same space. The ThreadCount GUI will have a tabbed pane that includes the Customer, Catalog, Sales, Reports, and Inventory tabs. |
| JPanel | Multiple JPanels will display GUI components like JLabels, JTextfields, and JButtons. JPanels include customerPanel, catalogPanel, salesPanel, reportPanel, and inventoryPanel. |
| JComponent | Adds components to subpanels, such as centerSalesComponent and rightCartComponent. |
| JLabel | Displays unselectable text labels that identify corresponding fields and buttons. Labels used in the ThreadCount GUI include Last Name:, First Name:, Address:, Style:, Color:, Size:, Cost:, as well as Display Reports, Export Reports, etc. |
| JTextfield | JTextfields accompany the labels, and allow users to edit a line of text in order to input information. The textfields are used for entering customer and clothing item information, as well as search functions. They include firstNameText, emailText, searchCustomerText, styleText, colorNameText, searchClothingText, etc. |
| JButton | JButtons have labels and can be clicked in order to generate an event. JButtons include addCustomerButton, customerSearchButton, addSaleButton, inventoryButton, salesReportButton, displayAllButton, loadButton, etc. |
| JScrollPane | Scroll panes create a scrollable view of a component, and the ThreadCount GUI has shoppingCart, reportPane, and inventoryPane scroll panes. |
| JComboBox | The combobox allows the user to choose one of several options. Our GUI uses comboboxes for selecting size and item. |

***Figure 19. GUI Components***

### 6.4.4 GUI Tabs

ThreadCount uses the following tabs to allow the user to manipulate customer, catalog, sales, reports, and inventory data.

| **Component** | **Description** |
| --- | --- |
| Customer | The Customer tab provides information about prospective clients. The left side of the panel allows the user to add a new customer. The right side of the panel allows the user to search for a customer. |
| Catalog | The Catalog tab provides information about the products that the user’s business sells. The left side of the panel allows the user to add a new item. The right side of the panel allows the user to search for an item. |
| Sales | The Sales tab is used to enter new sales. The left side of the panel allows the user to add a new sale to cart. The right side of the panel allows the user to see all the sales in the cart. |
| Reports | The Reports tab is used to find out how much in sales your business did last month, last year, and other useful information. The top half of the panel allows the user to display reports for inventory, customers, sales/month, and all sales. The bottom half of the panel allows the user to export reports to a file. |
| Inventory | The Inventory tab allows the user to see what items are currently in the inventory. The top half of the panel allows you to search for an item by SKU and the ability to display all items within the inventory. The bottom half of the panel allows the user to enter a new quantity for an item or change the quantity, along with loading items from a file. |

***Figure 19. GUI Tabs***

## 6.5 Database

The ThreadCount application will use a MySQL as the database to store its data.

### 6.5.1 Sample SQL Schemas

Below are the schemas for the ‘catalog’, ‘customer’, and ‘sales’ tables.

Catalog:

+-----------+---------------+------+-----+-------------------+-----------------------------+

| Field | Type       | Null | Key | Default        | Extra                    |

+-----------+---------------+------+-----+-------------------+-----------------------------+

| id     | int(11)    | NO   | PRI | NULL           | auto\_increment           |

| style | varchar(30)   | NO   | | NULL           |                          |

| color | varchar(15)   | NO   | | NULL           |                          |

| size   | varchar(15)   | NO   | | NULL           |                          |

| quantity  | int(11)    | NO   | | NULL           |                          |

| unit\_cost | decimal(10,2) | NO   | | NULL           |                          |

| price | decimal(10,2) | NO   | | NULL           |                          |

| img\_uri   | varchar(50)   | YES  | | NULL           |                          |

| sku    | bigint(20) | NO   | | NULL           |                          |

| created   | timestamp | NO   | | CURRENT\_TIMESTAMP | on update CURRENT\_TIMESTAMP |

+-----------+---------------+------+-----+-------------------+-----------------------------+

Customer:

+--------------+--------------+------+-----+-------------------+-----------------------------+

| Field     | Type      | Null | Key | Default        | Extra                    |

+--------------+--------------+------+-----+-------------------+-----------------------------+

| id        | int(11)   | NO   | PRI | NULL           | auto\_increment           |

| last\_name | varchar(30)  | NO   | | NULL           |                          |

| first\_name   | varchar(30)  | NO   | | NULL           |                          |

| address   | varchar(100) | NO   | | NULL           |                          |

| email     | varchar(30)  | NO   | | NULL           |                          |

| phone\_number | varchar(15)  | NO   | | NULL           |                          |

| created   | timestamp | NO   | | CURRENT\_TIMESTAMP | on update CURRENT\_TIMESTAMP |

+--------------+--------------+------+-----+-------------------+-----------------------------+

Sales:

+-------------+---------------+------+-----+-------------------+-----------------------------+

| Field    | Type       | Null | Key | Default        | Extra                    |

+-------------+---------------+------+-----+-------------------+-----------------------------+

| id       | int(11)    | NO   | PRI | NULL           | auto\_increment           |

| customer\_id | int(11)    | NO   | | NULL           |                          |

| item\_id | int(11)    | NO   | | NULL           |                          |

| quantity | int(11)    | NO   | | NULL           |                          |

| sale\_price  | decimal(10,2) | YES  | | NULL           |                          |

| created | timestamp | NO   | | CURRENT\_TIMESTAMP | on update CURRENT\_TIMESTAMP |

+-------------+---------------+------+-----+-------------------+-----------------------------+

### 6.5.2 SQL Queries

Below are some examples of the kinds of SQL queries that will be run on these tables in order to get meaningful information to the user.

Search for a customer by first or last name:

PreparedStatement ps = c.prepareStatement("SELECT id, last\_name, first\_name, address, email, phone\_number FROM customer where last\_name like ? or first\_name like ?");

ps.setString(1,  "%" + searchTerm + "%");

   ps.setString(2,  "%" + searchTerm + "%");

Add an item to the catalog:

PreparedStatement ps = c.prepareStatement("INSERT INTO catalog (style, color, size, quantity, unit\_cost, price, sku) VALUES (?, ?, ?, ?, ?, ?, ?)");

   ps.setString(1, i.style);

   ps.setString(2, i.color);

   ps.setString(3, i.size);

   ps.setInt(4, i.quantity);

   ps.setDouble(5, i.unitCost);

   ps.setDouble(6, i.price);

   ps.setLong(7, i.sku);

### 6.5.3 Connectivity

There is a Config class, which handles the database connection information (database host, database name, username and password. The requirement is a MySQL database. The program connects to it using JDBC and J/Connector. The J/Connector driver is required to be in the classpath.

### 6.5.4 Database Components

The database for this application consists of a MySQL database to which the application connects, and stores its data. This database consists of three tables- ‘customer’, ‘catalog’, and ‘sales’.

### 6.5.5 File Import/Export Mechanism

The user will be able to import, in CSV format, a file specifying new items or quantities of items to be added to inventory. The ability to save generated reports to file is also provided.

## 6.8 Design Phases

### 6.8.1 Phase 1 (Due Nov 27th)

**Expectations:**

* Basic GUI functionality, with five modules working correctly
* Sample Database created
* GUI and MySQL databases connected successfully
* Sample queries of database can be performed

### 6.8.2 Phase 2 (Due Dec 4th)

**Expectations:**

* Several queries of sample database can be performed, with expected results
* GUI allows user to retrieve and display a variety of data from database

### 6.8.3 Phase 3 (Due Dec 11th)

**Expectations:**

* Database with realistic amount of inventory, sales, and customer data created
* Several queries of database are performed successfully
* GUI allows user to retrieve and display data from database
* GUI allows user to modify database, including adding and deleting information

1. Hughes, B. (2015, Dec 30). *How to Improve Inventory Tracking: 5 Solutions for Small Businesses*. Retrieved from https://smallbiztrends.com/2015/12/improve-inventory-tracking-systems-top-5-solutions-for-small-businesses.html. See also, Lovering, J. (2016, Oct 16). *Small businesses consider Facebook critical to business success, survey says*. Mississauga News. Retrieved from http://www.mississauga.com/news-story/6913475-small-businesses-consider-facebook-critical-to-business-success-survey-says/ [↑](#footnote-ref-1)
2. Emmert, J.M. (2014, May 29). *The Big History of Direct Selling*. Retrieved from http://directsellingnews.com/index.php/view/the\_big\_history\_of\_direct\_selling#.WB545\_krLb0 [↑](#footnote-ref-2)