Point of Sale
Project Documentation

Ricky Hempel, Nisha Patel, and Josh Cannon

Abstract

A point sale system is an application for small private franchises (retail). It has a functionality of purchasing various products from different departments by adding them into the cart and checks out the cart by paying the total amount. The target audience for this project is anyone who uses a checkout.

1. Introduction

We are creating a point of sale application that can allow customers to purchase clothes, grocery, electronics, and books from different departments. Our point of sale will have a menu-driven interface where a user can view or select items from a drop-down menu. Every department has their own drop-down menu with the list of items and prices. As the user goes though the different departments, he or she can add the items in the cart. Once they have added all the items in the cart, they can go the check-out interface where they can pay the total amount of the purchase with either cash or a card.

We have designed our system for small-emerging private franchises such as specifically targeting retail marketing where their targeted audience will be a general audience such as college students, working employees, and/or retired people. Our point of sale is easily operable by any retail store employee, and is specifically for small departmental stores.

1.1. Planning

It needs a menu-driven interface for all departments and their sub departments with the list of items and prices where a user can add or remove from cart . Furthermore, in our POS keep tracks of added or removed items into the cart. Next, we need a functionality to manage a running display of total with tax calculation. Last, a user have a payment option (cash or credit) when they checkout.

1.2. Background

Point of sale systems are used by the vast majority of retail and restaurant chains today. They simplify the lives of both the employees and customers by making checkout time faster. The also help company operations by automating inventory and accounting. Our point of sale systems needs be easy enough for anyone to learn to use. It should be intuitive enough to use with little or no training and allow employees or customer to place an order and check out in efficient manner. We plan to do this an easy to understand GUI. Just click to add items to cart and check out.

A point of sale is a user interface to purchase objects. We decided to do this project because it is something that everyone uses all the time without thinking. We wanted to learn more about what actually goes on as a user is checking out a product. Josh works at the Pizza Hut and he is using similar system for their store, so he knows about the POS functionality.

1.3. Challenges

The biggest challenge was to the communication between main menu interface and their sub-menus. For instance, one of department called Books is going to be one of the tab in the main menu. Once use clicked on the Book, it will redirect them into the sub menu for the Books where it has list of books with name, author's name, and price. The challenge here would be how are we going to link the main and sub menu together.

Once we get the basics done, the next challenge would be to keep updating the shopping cart after a user adds an item. There will be more once we start working on it. For example, what design pattern are we going to use for the software that is easy to implement, straight forward user interface, and use of local database that can track our inventories. While picking a specific design pattern, we have make sure if it serves our purpose or not before we start scripting.

2. Scope

The final goal of our project is the following: first, it will provide a menu-driven interface of a departmental store. Second, it will allow users to view, select the available products, and add items into the cart. Third, it will allow user to go to the final step where they can check out their cart by paying the total amount in cash or by card.

Currently, we are assuming that we are creating this application to sell it in the small-emerging private retail market. We will be testing this application in our departmental store first, then sell it as a point of sale software for a fixed price. Once the base is done, we can update our system according to the company's requirements.Our two "stretch goals" would be first adding more products that are easily customizable as needed and provide an accurate final result. Moreover, it should allow the user to add any number of items in the cart and check out by paying the right total amount.

2.1. Estimate Time

We would say that our project will be done roughly within a month and a half. If we finish it early, we can add other features.

2.2. Requirements

The requirements we gathered for our point of sales system by brainstorming and doing online research. We created flowcharts of what tools and functionality we needed and how to tie them together. We also watch POS tutorials online to familiarize with the system.

2.2.1. Functional.

- User needs to have a private shopping cart – this cannot be shared between users, and needs to maintain state across subsequent visits to the site
- User can add more than one items – cart should automatically update the total number of items as soon as user adds a new item.
- Users need to have website accounts or store accounts – this will help track recent purchases, keep shopping cart records, etc.
- User can pay with cash or cards – card has to be valid.
- User pays verified total amount – this will verifies and calculate accurate total checkout amount.

2.2.2. Non-Functional.

- Security – user credentials must be encrypted on disk, users should be able to reset their passwords if forgotten or update their personal information for the local store accounts.
- Accuracy – system must calculate the correct total amount from the user without any errors.

2.3. Use Cases

It collects all required information from the user about the software he/she want. It asked specific detailed questions about there need and outcome. It gives a rough idea to an user of what and how our point pf sale is interating.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Click on different menus to see all products that are being offered in the store	Shopper	Med	1
2	Add item to cart	Shopper	Med	1
3	Checkout	Shopper and Cashier (applicable to both)	Med	2
4	Stocking	Manager	High	3

TABLE 1. POINT OF SALE USE CASE TABLE

Use Case Number: 1
Use Case Name: Click on different menus to see all products that are being offered in the store
Description: A shopper on our site can explore the different departments and there products. They can click on "Department Name i.e., Books" button to view what types of books we are offering with the book name, author name, and price (options varies as per the department).

Use Case Number: 2
Use Case Name: Add item to cart.

Description : A shopper on our site has identified an item they wish to buy. They will click on a “Add to Cart” button. This will kick off a process to add one instance of the item to their cart. They can add more than one item in cart.

Use Case Number: 3
Use Case Name: Checkout
Description : A shopper on our site has added item in cart and now they are ready to checkout. They will click on “Checkout” button. This will kick off a process to calculate the total amount with price and taxes.

Use Case Number: 4
Use Case Name: Stocking
Description : Manager on our site has access to the inventory list where they can click on ”check inventory” button to see how many items are sold and if they have to restock them.

The flow of the point of sale would be like the following:

1) User navigates to department listing.
2) User navigates to page listing desired item in a particular department.
3) User left-clicks on “Add to Cart” button.
4) User cart is updated to reflect the new item, this also updates the current total.
5) User left-clicks on “Checkout” button to pay the total amount.
6) (Termination Outcome) The user now has a single instance of the item in their cart.

User Stories:
User Story 1: As a user,I want to click on different department to see what products are being offered
User Story 2: As a user has explore the menu, now he or she can add items in to the cart by clicking “add to cart” button. Furthermore, the user can see the current running total of the cart on the display.
User Story 3: As a user is ready to checkout, he or she can click“checkout” button on the screen and that will allow them to pay the final price including the taxes. User Story 4: As a manager, I want to see the inventory list with what products we have sold and in what quantity. Once the store restocks the inventory, I should be able to add them into the inventory database.

2.4. Interface Mockups

Point of Sales main menu consists of four options: Grocery, Books, Electronics, and Clothes. Once user clicks on any of the menu, it will redirect them into sub menu of that selected department. It continues navigating the sub sub menu until user finds the item to add in to the cart. For example, user has clicked on Books $->$ Name or Author Name $->$ Clicked Name $->$ List of Books $->$ Clicked on a book $->$ Author’s Name and Price, and an option to Add in the cart.
Below is a sample diagram that represents point of sales functionality.

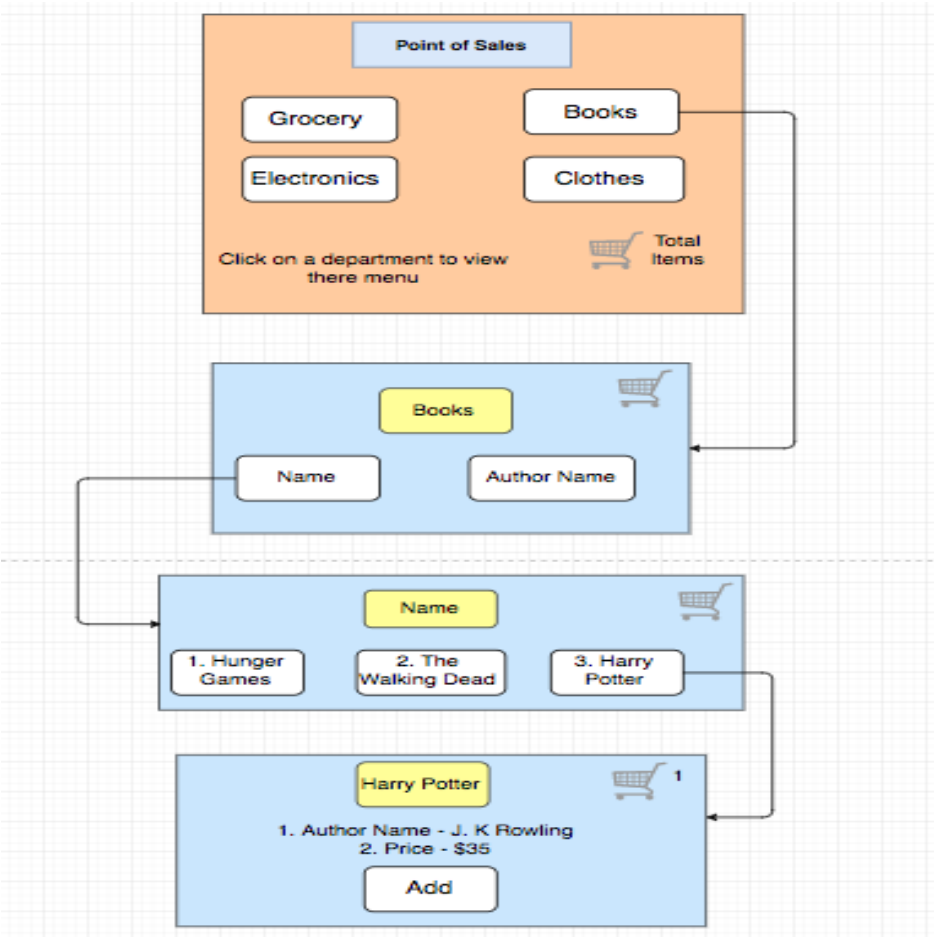

Figure 1. Sample Interface

We have modified the previous interface with this one because navigating from one window to the another until you reach the final sub menu was a bit challenging. We decided to go with a simple option where our design looks simple and easy to implement.

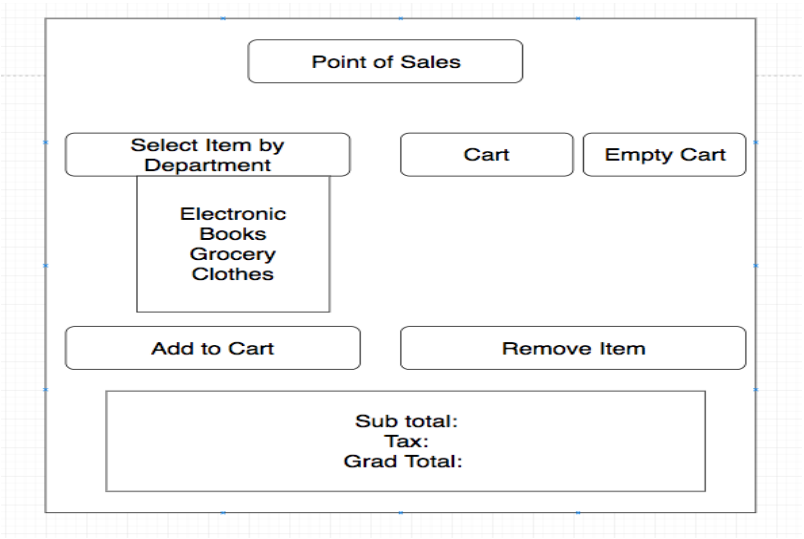

Figure 2. is a representation of new interface

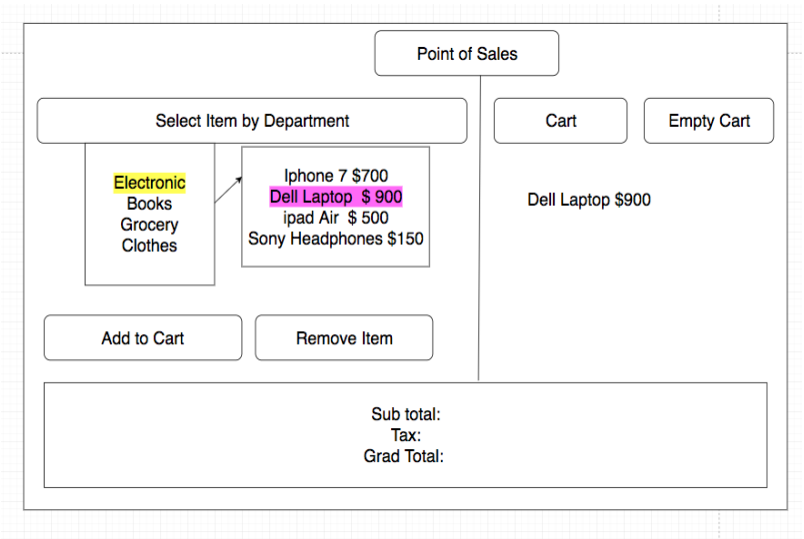

Figure 3. is a representation of how the point of sales system is working internally.

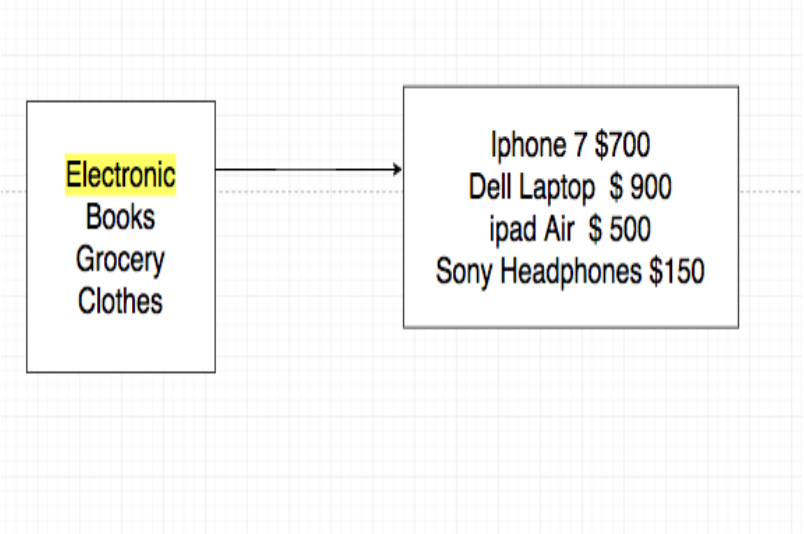

Figure 4. is a representation of submenu

3. Project Timeline

Below is the timeline planned out about how we are following project development life cycle for the Waterfall approach.

Task Name	Start	Finish	Duration	% Complete	Status
CSCI 352: Point of Sales system					
Project: "Create a timeline with assigned tasks for the project with a timeline planned out"					
Teams - Selecting team members	08/29/17	09/517	1d	100%	Complete
Project Proposal Draft - First Draft - Proposing our ideas and a rough plan	08/29/17	09/21/17	18d	100%	Complete
Project Proposal Update - Second Draft - Updated previous draft	09/27/17	10/12/17	12d	100%	Complete
Project Proposal Presentation - interface, use cases and functional and non functional requirements	09/27/17	10/12/17	12d	100%	Complete
Demo - Rough demo of the working wpf application wth minimum functionality of selecting an item(s), adding or removing in or from the cart	11/14/17	11/21/17	6d	100%	Complete
Penultimate Writeup - Updated report with user stories, timeline planned out, structure notes and UML outline	11/16/17	11/28/17	9d	100%	Complete
Final Writeup - Finish up remaining sections - design decisions, results, and future works	11/16/17	11/28/17	9d		Complete
Final Presentation - Final presentation of point of sales system	11/16/17	12/06/17	15d		Complete

Figure 5. is a detailed representation of project development with the tasks, start and end dates, duration, percentage complete, and status

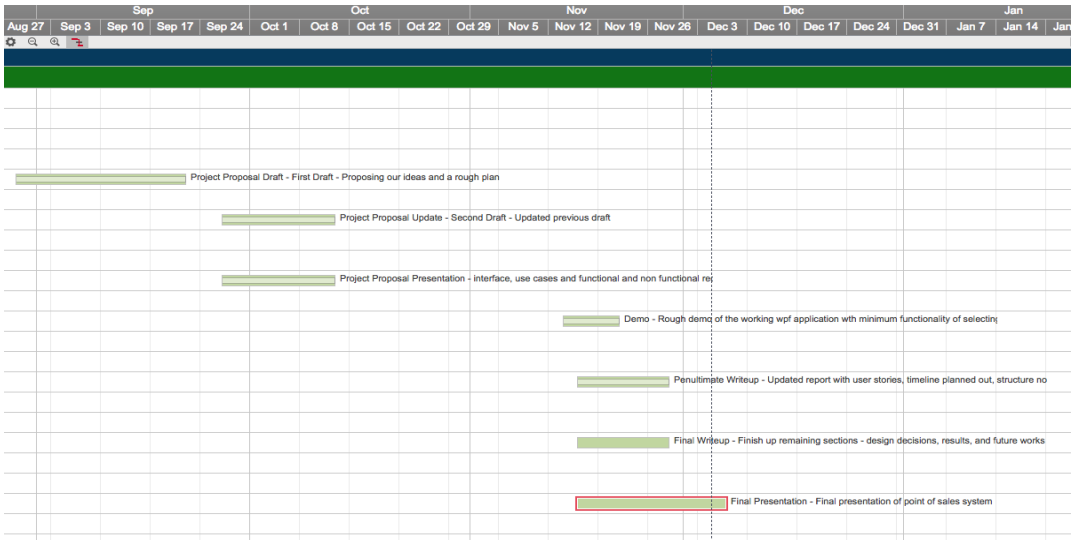

Figure 6. is a representation of proposed timeline

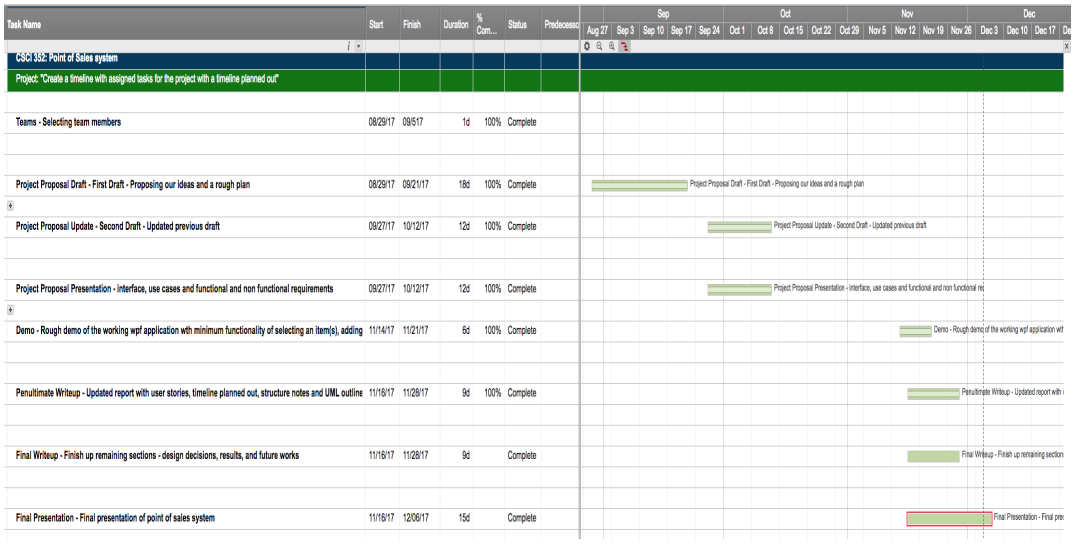

Figure 7. is a representation full timeline showing both details plotted timeline for rest of the semester

4. Project Structure

We are planning on using a built-in observer pattern that can maintain a list of object's dependents. We decided to use this pattern because the observer pattern will notify its dependent objects whenever a subject is modified. For instance, if a user clicks add to cart, it will add that item into the cart and likewise if the user removes an item from the cart, it will update the cart by changing the object's state. We are planning to use Abstract Factory Pattern to create different themes for different stores. However, now we have decided to use builder design pattern.

4.1. UML Outline

Below is the rough UML of the point of sales system. It is not accurate because we are still in the design process. However, we will have an up-to-date UML on the day of our presentation.

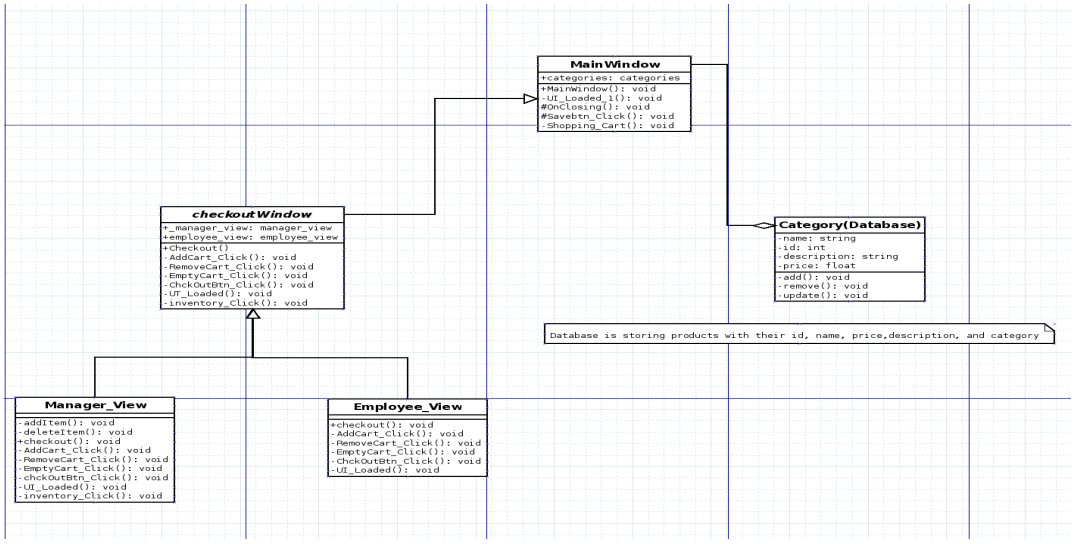

Figure 8. is a representation of a rough UML of POS

4.2. Design Patterns Used

We used built-in observable pattern that can maintain a list of object's dependents, and we decided to use this pattern because observer pattern will notify its dependent objects whenever a subject is modified. Furthermore, we are using Builder

design pattern because it separates the construction of complex objects from its representation so the same construction process can create different representations. In our POS, we are using this method to separate ”manager” and ”employee” view. For example, on the manager window’s, they should be able to view the inventory list where they can add/remove items from the database on the fly. However, on the employee window, they can only have limited access. For instance, the employee cannot view the inventory list, and they do not have access to add/remove items from the database. We used Microsoft SQL database to store our products according to there category. We also used entity framework that offers an automated mechanism to developers for storing and accessing data in the database.

5. Results

Our point of sales can do the following tasks: First, a user can view the menu-driven interface for all departments with the list of items, prices, and products description. Second, a user can select a department and view the offered items. Third, a user can select the item(s) and add to the cart by clicking ”Add to Cart” button. Fourth, it can keep track of added or removed items from the cart (constantly updating the cart whenever a user makes any changes). Fifth, when a user clicks on ”checkout” button, it will help them the current running total of the taxes.

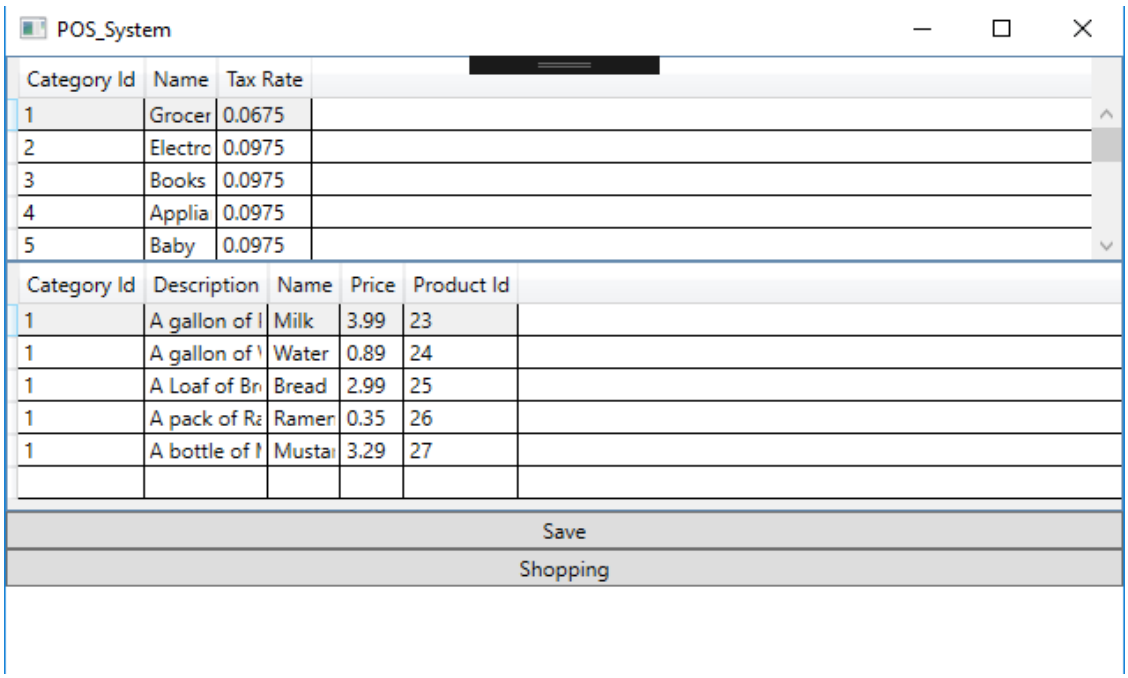

Figure 9. is a representation of inventory window where a manager can add products on the fly

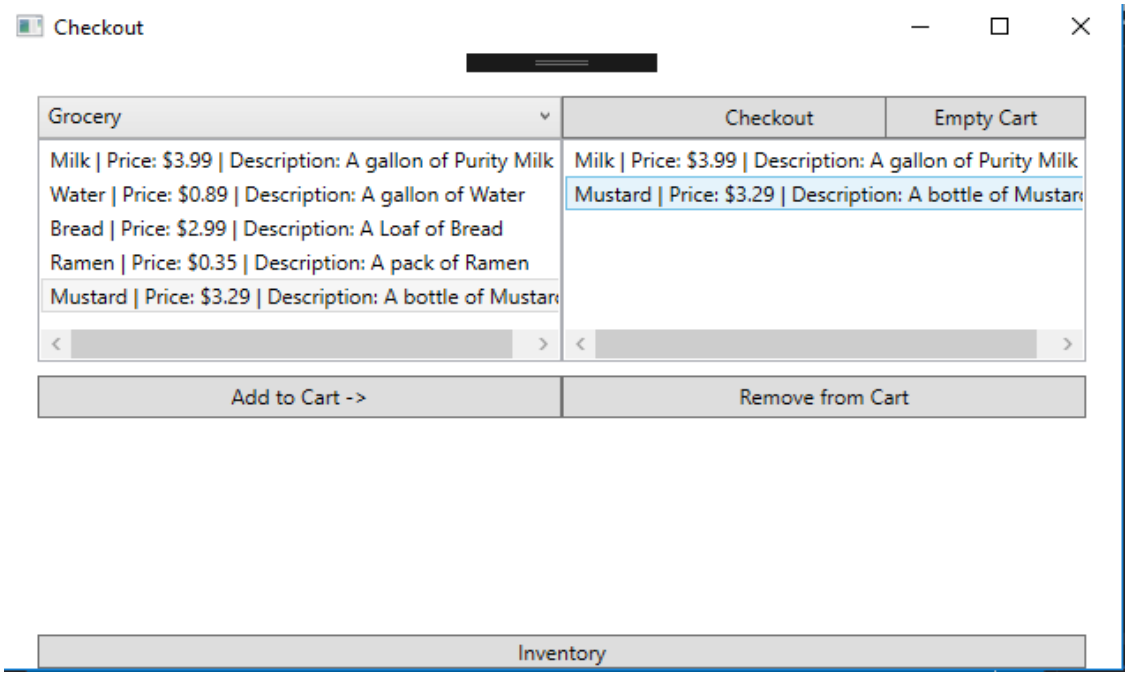

Figure 10. is a representation of basic functionality where a user can view, add, and delete items in the cart

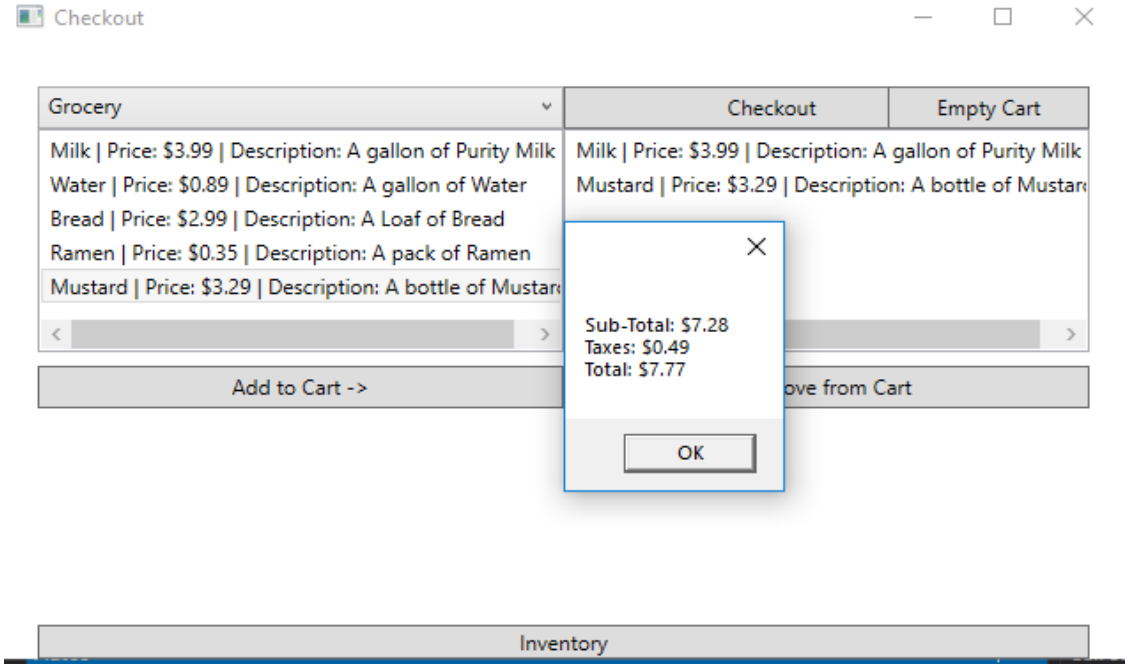

Figure 11. is a representation of checkout process

We have met our two stretch goals: a user can add more products that are easily customizable as needed and provide an accurate final total. It also allows a user to add any number of items with repetition in the cart and check out by paying the right total amount.

In addition, we have implemented all the use cases in out software. First, a user can select different departments and view what products are being offered. Second, user can add or remove items anytime in the cart. Third, a user can see the final price including the taxes when they click on checkout. In short, our project has all the functionality as we planned.

5.1. Future Work

For the future development, we have to work on the security issues by creating a login page for the user, and it will grant access according to the entered credentials. For instance, an admin has full access to the system whereas employees have limited access. Next, we want to create a prettier interface by adding additional functionality such as other images of the products, so the user knows what they are buying. Then, create a finite list of inventory with the item counts, so the manager can see how many items have been sold and restock them as needed. Last, we want to add a functionality where a user can select a payment type (cash or credit) and process the payment. We almost met the expected goals. We decided to shift gears with the initial design because changing a window whenever a user click was bit challenging, so we decided to go with a more straightforward approach by using a list box. We will be selling this software directly to the retailer once it has all the functionality and we will be linking to your resume.