

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Team Name: Apple Addicts |  | Simon  Tyler |  | arrudat@sheridancollege.ca  [Web address] |

The Register

Enter Products, Calculate Change and Retain a store copy

Table of Contents

1. [Project Overview 1](#_Toc340506951)

Purpose of the Application

1. [User Interface Prototype 1](#_Toc340506956)
2. [Data Design 1](#_Toc340506956)

# Project Overview

## Purpose of the Application

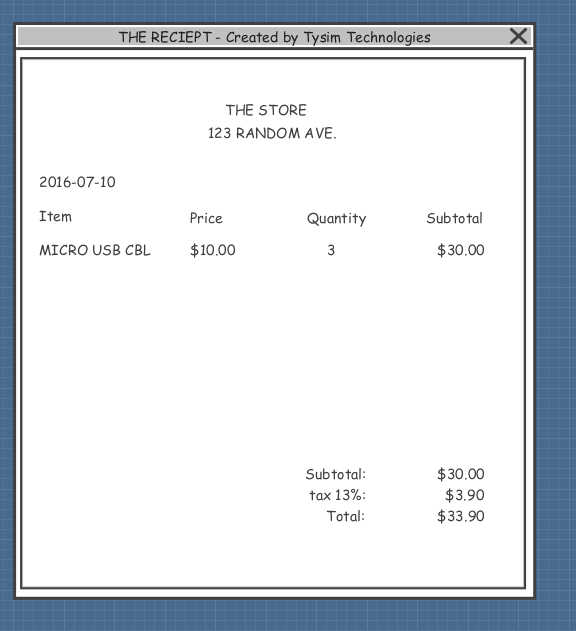
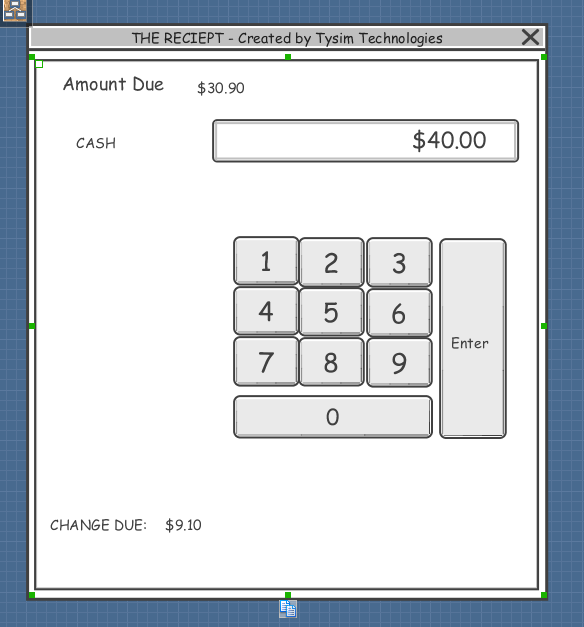
This application is designed to complete the functions required by a simple cash register. These functions include entering products to the total purchase list, apply taxes and total the list of products, accept money from the customer, calculate their change and print out a receipt for the customer along with the store copy to be saved in records.

## Development Environment

The development environment being used for this project is Visual Studio 2015 Enterprise. All the work is being done using the Universal Windows Platform meaning that the code will be written mainly in C# other than the design elements that are handled with XAML.

# User Interface Prototype

This is a Wireframe project made using Visual Paradigm. This will be the main screen of the register

This portion will be displayed upon cash out This portion will be displayed after the customer has paid, it will also include the amount paid and the change received (this is not shown in this version of the reciept)

# 

# Data Design

This application will work with predetermined products that will be activated or added to the total with buttons specific to the product. The Register will add items to the total and remove inventory from the store. The cash register will then generate the receipts for the user and for the store. These receipts will be generated separately as the store copy generally differs from the customer copy. The customer copy will be displayed as a message box (as we are not going to physically print any receipts) and the store copy will not be displayed as it will just be saved to a text file.

# The Outcome

There is quite a big difference between our proposal and the final product. We decided to take a more modern approach with simplicity in mind. Instead of the traditional POS system with a bunch of clustered buttons and outdated display, we went with a natural layout that would work well with a touchscreen. There are two list view items placed on the screen, one to hold preset products and the other to show the items that have been added to the cart. The list view with the items has the ability to scroll up and down considering the number of products do not fit into the list view. The idea here is still the same as the application is intended for a small vendor with limited stock options. The project consists of the main page where the user interface lives, and then five other classes that help form the application. These are the classes; CheckOutController, Items, Order, OrderItem and Product. We also have a UML diagram as part of our visual studio project. The final product of our application still does what we wanted to achieve from the start. The user can add items and have a visible list view of what they have selected. After that, once all the items they would like to have are added to the cart, they can checkout and this is where the backend process begins. The application will then prompt the user with a message saying that their file has been created. What happens in the background is that the application will then generate a receipt and store that as a text file so that the products and the total are available in a printable txt file. The same concept has been achieved with a cleaner look and that’s why we chose to go with the list view with product info opposed to having the buttons that did not say anything about the products. We also added a currency converter that allows the user to pick what currency they want to use in the store. The currency is accurate with commas and decimal places and uses the IValueConverter interface.

Overall, we are happy with our application and the final outcome was good. The application does what it is intended to do and keeps the modern look of the UWP environment. To finish off, we would like to thank you for the knowledge and the help you have given us in the last half of the semester and we wish you all the best in the future.