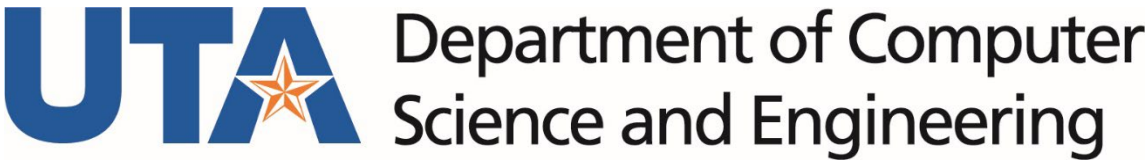


CSE 1325 Project Documentation

SHOP EAZY

Students names, surnames, IDs:
1. Shubh Gupta, 1002055668
2. Manuel Hernandez, 1001911697

Mentor: Marika Apostolova
TA:
Aziz Abdul
Ntokos Nikolaos
Roy Ayon

Object oriented programming CSE 1325

Student declaration:

We declare that:

- *We understand what is meant by plagiarism*
- *The implication of plagiarism has been explained to me by our professor*
- *This assignment is all team own work and we have acknowledged any use of the published and unpublished works of other people.*

1 Shubh Gupta 10012055668 Shubh Gupta
2 Manuel Hernandez 1001911697 Manuel Hernandez

Date: 11/28/2023

	Total number of pages including this cover page	##10
Class Code / Group	CSE 1325	
Lecturer's Name	MARIKA APOSTOLOVA	

Table of Contents

CHAPTER ONE

PROJECT INTRODUCTION

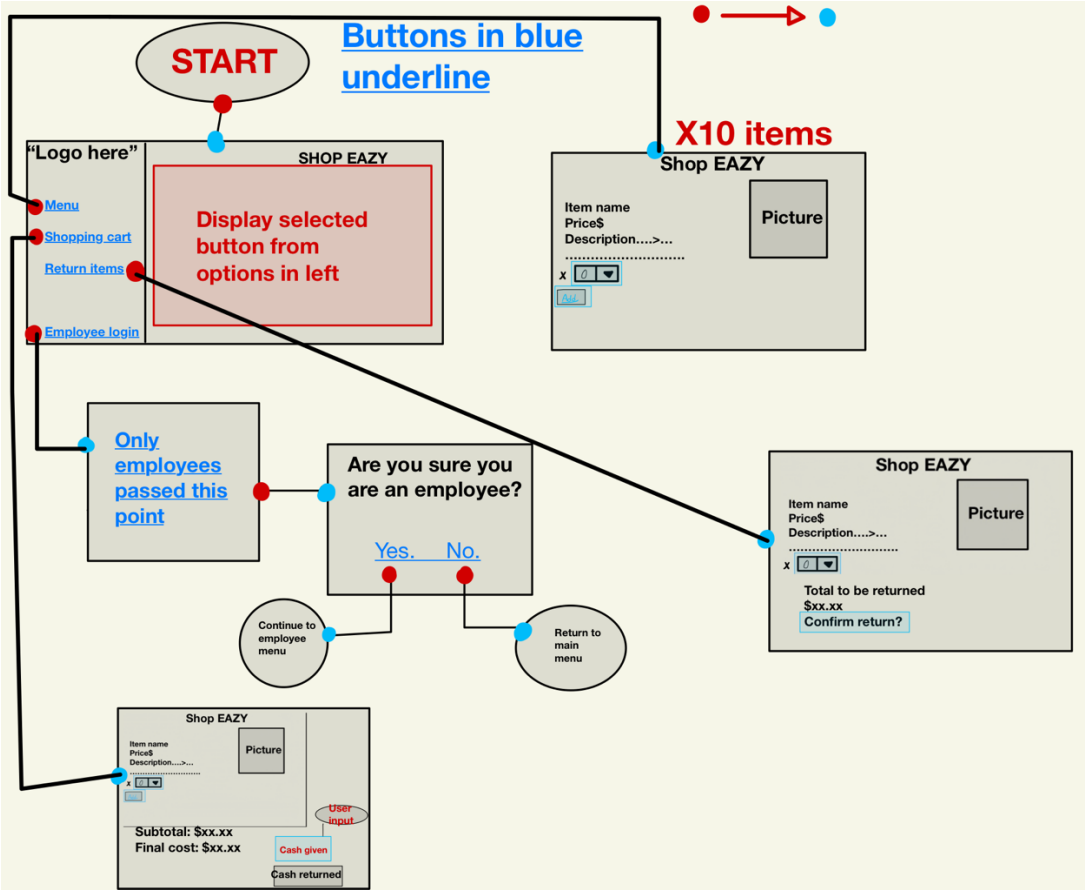
The ShopEazy project is a JAVA-based program designed to simulate an online storefront. This program will revolve around using a graphical user interface GUI5 as the storefront with options for customers to shop store items, check invoice, check out items, and return items for a refund. Additionally, employees can log in with proper credentials to access another menu to perform other employee actions.

The program will implement classes to encapsulate all the options, and methods to perform operations based on user input, and use a defined list of items with prices using individual classes to define those items.

CURRENT SYSTEMS

With the rise of online shopping, now more than ever is there a need for online grocery shopping made easy. ShopEazy is a JAVA-based simple program designed to emulate an online storefront. To run the program, go to the folder labeled "dist" and execute "ShopEazy.jar" through the OpenJDK Platform binary. Currently, ten items can be viewed in the item menu section of the program. Each of the items has a subsequent item name, price, and product description along with an item image. The next tab, the shopping cart allows the user to create their invoice, by writing the name of the item, the quantity, and the price of the item. The program will return the subtotal, then return the final cost including taxes (6.25% according to Texas Goods and Services Tax). You can then input the amount of cash given and it will return the balance, and if the cash is short, return the amount required if need be. The last panel is the Employee Login panel which currently shows a dialog box when clicked on. It can later be implemented with more features in future versions. Currently, the program is in its final version needed for the project.

PROPOSED MODEL DIAGRAM



PROJECT SPECIFICATION/ FUNCTION MODULES

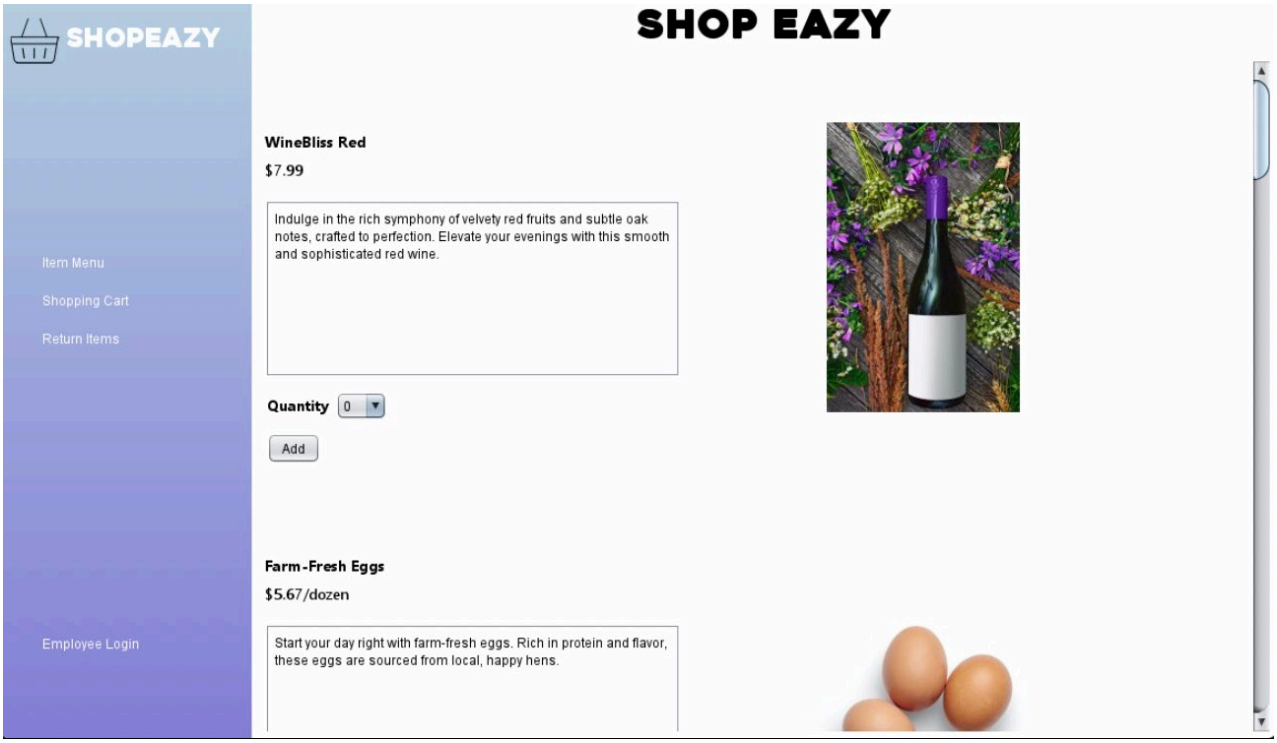
All functions of this project have been intended to be used interactively through a GUI with one another as the design for this program consists of a main menu with all the following functions/classes (itemMenu, invoiceForm, returnItem, and employeeLogin) appearing as sub-menus to the main one. All specific functions have been given a way to interact through JButtons or other user input.

PROGRAM (INPUT/ OUTPUT) SPECIFICATION

The program's users interact with their own input through different GUI menus. Access to all the displayed menu options is granted by user input by clicking on each respective button, which in return will open said class and GUI for further use. In class invoiceForm, the user has access to input the desired amount of item x through keyboard input or the drop-down menu with quantities 0-9. The overall goal of the design implemented was to be extremely user-friendly and intuitive to use for any person.

Screen Design

Shown is the shopping cart opened in the main menu.



CHAPTER TWO

PROGRAM DESIGN AND CODES

As stated, the program has been designed to be intuitive and user friendly. Here is a breakdown of all the main components of the program.

MAIN

Main is the main class for this Java program. This class imports everything in and extends javax.swing.JFrame. Here is main itemMenu, invoiceForm, and employeeLogin are private values. This class also had the implementation to change panels in the main menu by setting and changing specific panels chosen.

MENU

Menu is the main menu page for this program. This class incorporates all the menu panel options for use in the program. Menu also has a method for moving the program by clicking and holding on the logo in the menu selection.

ITEMMENU

This class creates one of the forms needed for this program. In this class, all item cards are loaded into this menu as a scrollable page ready to use.

INVOICEFORM

This class imports java.util.ArrayList to store itemName, quantity, itemPrice, and subtotal. This class is implemented as a GUI with methods for the “add” button in the invoice as well as for clearing everything, dispensing change, and taking in user input. This GUI is designed to look like an invoice and to be easy to navigate a transaction.

ITEMCARDS

itemCards is the package for all the items in this program available for purchase. The individual item classes give us all the information. Needed to display and be able to conduct transactions with said items.

RESOURCES

There exists a package called resources which simply holds all the .jpg, .jpeg, and .png files for our logo and item pictures.

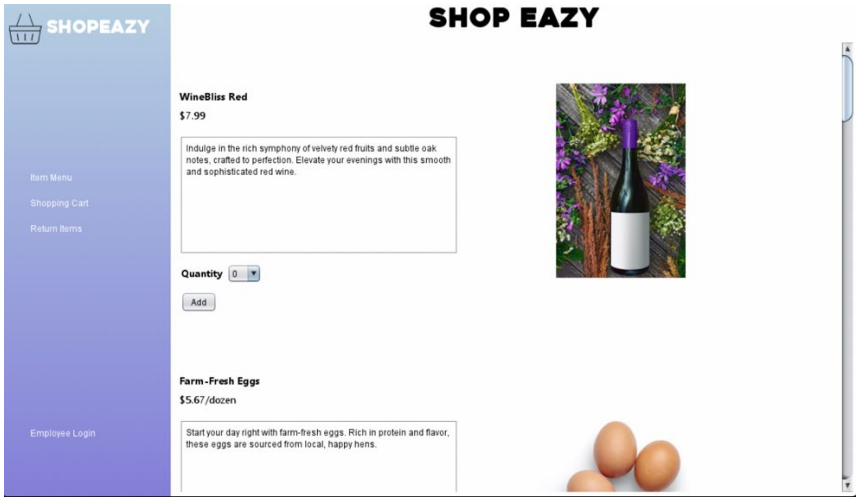
EMPLOYEE

Class employee is here for the future development of this program. EmployeeLogin asks users if they are an Employee, as content opened from there will be for employees only like withdrawing money from the register, putting money in the register, changing item prices, and discontinuing items. (for future development)

MODULE MENU SCREEN

Shown is the main menu

The menu design was made to be appealing and easy to navigate through, different pages are made accessible by clear buttons on the left with respective buttons opening their respective menus on the right (shown here is the shopping cart). Selecting and adding user input is made clear by different buttons such as the quantity dropdown menu or the add button which adds to the invoice.



CHAPTER THREE

PROGRAM TESTING LOG & CASES

To have a final product, there must be test cases to test different possible outcomes from the software to test for bugs or other issues needing to be fixed. Here is a log of manual testing done with different cases in mind.

CASE	METHOD	RESULT
Run the program as stated in the README.TXT file to test for no run time errors or warnings.	go to the folder labeled "dist" and execute "ShopEazy.jar" through the OpenJDK Platform binary.	PASS
Open every main menu panel option to verify proper program running.	Click on every main menu option to see if they will open on the right hand side panel and display their intended page.	PASS
Check every close button works	Go to every page with a close option (x button) and see if it closes the page properly.	PASS
Check for invoice accuracy	Max out every item in the invoice and repeatedly change values to ensure subtotal and final total display proper results	PASS
Check for bugs when changing the program display size	Move around the software tab as much as possible to see if all pages move accordingly or get stuck.	PASS

CHAPTER FOUR

CONCLUSION

The ShopEazy program gives users the ability to easily navigate a virtual storefront with essential groceries through an intuitive and easy-to-use GUI. The software implements all needed aspects of a store such as an item catalog, shopping cart (in the form of an invoice), return page, and an employee login section for employee access to employee actions (to be later implemented).

Program weakness

Although this program is intuitive and user-friendly, it lacks some aspects of a modern storefront. While the invoice is easy to use, a better implementation would be that in both the catalog and invoice you could edit the quantity in your cart, but we encountered issues with saving those values while navigating the store. A better understanding of data structures and databases would allow this program to take on a more developed form.

Program Strength

As stated before, this program is extremely user-friendly and intuitive. The strength of this program is its GUI implementation that follows a familiar design to that of the majority of websites today making it easy to navigate for the average Joe. This program also clearly states all of its functions so that there isn't any confusion for the user.

Program Enhancement

As stated previously, a better understanding of data structures and databases would allow this program to take on a more developed form. With a clear understanding of databases, this program would be enabled to be more cohesive. Instead of every panel doing something different with no relation to the others, actions and input from one panel could be introduced to the rest of the program. This would allow for the implementation of a register where the withdrawals, deposits, and other actions can be crossed between the program parts. This would also allow for the implementation of a saved cart rather than needing to create an invoice at checkout.