

Inventory Management Program for UTM CSCI 352 Spring 2018

Brayden Faulkner
Johnathon Sulcer

Abstract

With this program we wish to provide an inventory management system that can be used by a variety of consumers such as commercial retailers and warehouses. The program will provide different features depending on who is using it. For example if the program is being used by a distribution warehouse the user will probably want to track where outgoing products are going, while a retail store would just like to note how much the product was sold for. This

1. Introduction

With this program we want to help the user track the contents of their inventory as efficiently as we can and provide features to aid them. This means we will have a user specific database filled with the user's inventory and information provided by the user, such as what the inventory is for. This program will be designed with the intention of being usable by virtually anyone who has an inventory to manage. While the program will provide specialized features for things such as restaurants, warehouses, and retail stores, there will also be a basic set of features for those cases we do not have the time or imagination to account for. This means that hopefully any user will be able to manage any sort of inventory in a simple and efficient manner.

Should you need to cite anything, use the *cite* keyword, and refer to something from your bibliography. For example, this was put together with the help of a L^AT_EXguide [1].

Make sure that by the end of your introduction the reader knows what your project is and why you are doing it.

1.1. Background

The main reason we decided to do this program is we thought about all the similar programs we had done in the past and wondered if we could adapt the idea behind some of them into a useable end product. We also thought of all the places that still use ineffective methods to keep track of their inventory due to fear of technology or the misguided view that modernizing would be too expensive.

1.2. Challenges

Two of the biggest problems we anticipate are dealing with working with the databases and the login procedure. We'll need the login procedure to store what the client has in their inventory and what kind of inventory they have. The databases are sort of an x factor here as since neither of us have extensive experience with databases, so we're not really sure what to expect.

2. Scope

As a minimum, we would like the program to be able to track the inventory, list its contents, keep data on, and add and take things away from the inventory as well. Then we will add in some features specialized for retail stores, including tracking profits and the dates items were received. There will also be some features specialized for warehouses, like tracking where and when items are being shipped. As a stretch goal we would like to add in features specialized for a library, such as tracking who has what book and when they should return it. As another stretch goal we would like to add a basic setting screen to change simple things such as text size and background color.

2.1. Requirements

As part of fleshing out the scope of your requirements, you'll also need to keep in mind both your functional and non-functional requirements. These should be listed, and explained in detail as necessary. Use this area to explain how you gathered these requirements.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Add item to cart	Shopper	Med	1
2	Checkout	Shopper	Med	1

TABLE 1. SAMPLE USE CASE TABLE

2.1.1. 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
- Users need to have website accounts – this will help track recent purchases, keep shopping cart records, etc.
- You’ll need more than 2 of these...

2.1.2. Non-Functional.

- Security – user credentials must be encrypted on disk, users should be able to reset their passwords if forgotten
- you’ll typically have fewer non-functional than functional requirements

Use Case Number: 1

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.

You will then go on to (minimally) discuss a basic flow for the process:

- 1) User navigates to page listing desired item
- 2) User left-clicks on “Add to Cart” button.
- 3) User cart is updated to reflect the new item, this also updates the current total.

Termination Outcome: The user now has a single instance of the item in their cart.

You may need to also add in any alternative flows:

Alternative: Item already exists in the cart

- 1) User navigates to page listing desired item
- 2) User left-clicks on “Add to Cart” button.
- 3) User cart is updated to reflect the new item, showing that one more instance of the existing item has been added. This also updates the current total.

Termination Outcome: The user now has multiple instances of the item in their cart.

Use Case Number: 2

Use Case Name: Checkout

Description: A shopper on our site has finished shopping. They will click on a “Checkout” button. This will kick off a process to calculate cart total, any taxes, shipping rates, and collect payment from the shopper.

2.2. Interface Mockups

3. Project Timeline

4. Project Structure

4.1. UML Outline

Show the full structure of your program. Make sure to keep on updating this section as your project evolves (you often start out with one plan, but end up modifying things as you move along). As a note, while Dia fails miserably at generating pdfs (probably my fault), I have had much success with png files. Make sure to wrap your images in a `figure` environment, and to reference with the `ref` command. For example, see Figure ??.

4.2. Design Patterns Used

5. Results

5.1. Future Work

References

- [1] H. Kopka and P. W. Daly, *A Guide to L^AT_EX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.