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 some basic different functionality depending on who is using it. For example, some information needed to be stored may about the inventory may change from different consumers, so the consumer will be able to store specific information on their inventory through some methods that will eventually be provided by the program.

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 provided specialized features (which may be things easily set up by the consumer) for things such as restaurants, warehouses, and retail stores; 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.

1.1. Background

The main reason we decided to do this program is we thought but on all the similar program we had done in the past and wondered if we could adapt the idea behind some of them into a usable 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. Since we plan to use a login, 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, be able to manage some sort of data on the inventory, and add and take things away from the inventory as well. As a stretch goal, 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. Finally, 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

2.1.1. Functional.

- User needs to have a private inventory – 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 keep an online database and keep track of inventory with details.
- Needs to have a place with sufficient capacity to record all required information, and reliably store it in a database.

Use Case ID	Use Case Name	Primary Actor	Complexity	Priority
1	Basic Inventory	Admin	Hard	1
2	Retail Inventory	Admin	Med	2
3	Warehouse Inventory	Admin	Med	2
4	Library Inventory	Admin	Med	2
5	Visual Settings/Appeal	Normal User	Easy	3

TABLE 1. INVENTORY PROGRAM USE CASES

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

2.2. Use Cases

Use Case Number: 1
Use Case Name: Basic inventory
Description: A user wants to set up an inventory that requires a login to access and contains basic information about the inventory, minimally including quantity and a way of giving a brief description of an item.

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

1) User downloads/runs program and selects options to set up his account/inventory.
2) User adds inventory through a method(?).
3) User inventory is updated to reflect new items, this also updates the current total.

Termination Outcome: The user now has a inventory of items updated and modified through the program.

You may need to also add in any alternative flows:
Alternative: Item already exists in the inventory

1) User navigates to page detailing the item already exists in inventory
2) User updates information if necessary on item in inventory.
3) User inventory is updated to reflect the new item's details, displaying any worthy information about that item.

Termination Outcome: The user now has a inventory of items updated and modified through the program.

Use Case Number: 2
Use Case Name: Retail Inventory
Description: An inventory database for a retail store. For example, it will have specialized features such as the ability to have a shopping cart and checkout methods that can automatically add and take away from the inventory database.

Use Case Number: 5
Use Case Name: Visual Settings/Appeal
Description: The user may change visual settings to make the program more usable or appealing.

1) User navigates to a settings page.
2) User changes to desired visual appeals.
3) Program visuals are updated to reflect the desired visual changes for the program.

2.3. Interface Mockups

3. Project Timeline

Design: We have already decided on the overall idea and plan of what is to be done as read in the overview of the project, but more specifically the project UML, chosen design patterns, project structure, and interface mockups will also be mostly fleshed out by March 15.

Implementation: We hope to at a minimum have a working inventory on a at least a very basic level with a simple interface that has a login connected to a user database. Our deadline for this is by April 5.

Verification: TBD

Maintenance: TBD

4. Project Structure

4.1. UML Outline

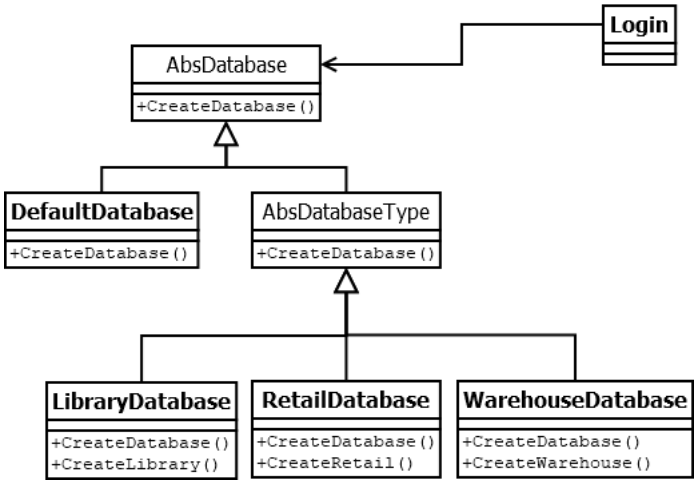

Figure 1. The idea of how modified databases will be implemented, such as specific databases for a library, retail store, or warehouse, using the decorator design pattern.

4.2. Design Patterns Used

1 Factory method used for creation of a new user.
2 Decorator pattern used to create some different types of databases with added functionality.

5. Results

5.1. Future Work

References

[1]