Design Document

Introduction- This project will be an inventory management system. It will be designed to track customers, orders and stock. We will be creating it in C# using Visual Studio and MySQL. The repository will be stored in github and we want to host the database on an AWS test server. It will be designed to streamline checking in stock and removing stock when sold. It also will create an easier interface for finding customers and what they’ve ordered. It is being designed for an air cleaner reseller that Paul knows out of Fort Wayne.

Architecture- I believe that we will be using a layered architecture. We will be separating the program into several. I would like to use the MVC model. The business logic and database logic will be broken into folders and kept separate from any of the other code.  There will be a folder BAL for the business access logic and a DAL folder for the data access logic. The UI will be made and configured in Visual Studio designer and the Views will be updated separately. The actual functionality will be kept separate and events will be tied to classes and functionality built. Any changes will go be stored and the UI will update from that data. This will be illustrated with a table.

|  |
| --- |
| User Interface(windows and their controls. Any updates will be controlled by UI management and updates will be passed to the UI to change or fill data. This is where the data from UI management will be placed.) |
| UI management(This is for checking all data entered for validity. It will also include any authorization and validation required for use. The database connection on the first layer will provide information to be management and placed in controls) |
| Application Functionality(Core functionality for access and updating program. Sends control information for UI Management. Includes all API’s, classes, and methods) |
| System Support(DAL, BAL and functionality for windows) |

Use case-

* View Inventory
  + A user logs in and the home screen is shown. They click the View Inventory button and are taken to a query page and input parameters for searching inventory.This can include names, serial numbers, sold, or under warranty. When search is pressed a new screen is created displaying the stock of inventory the user chose and any other criteria requested. A button for a new query or to exit will be provided.
* View Customers
  + A user logs in and the home screen is shown. They click the View Customer button and are taken to a query page and input parameters for searching customers. This can include the name, address, phone number or email.  When search is pressed a new screen is created displaying customers meeting the criteria provided. A button for a new query or to exit will be provided.
* Add Customer
  + A user logs in and the home screen is shown. The Add Customer button is pressed. A form for entering new customers shown. The user enters the customer information and clicks the add button. The customer is added to the database.
* Add Existing Inventory
  + A user logs in and the home screen is shown. The Add Inventory button is pressed. A form for entering the UPC is shown. The user can either enter the UPC manually or scan it. It is an existing product. The fields are populated and after everything has been scanned the user clicks the add button and the quantity is updated.
* Generate UPC
  + A user logs in and the home screen is shown. The Generate UPC button is pressed. A form for entering the data for a new product is shown. The users fill out the information for the new product. If a serial number is required a text box is shown to enter it. The user clicks the add button and an upc is generated for the product. A print screen is shown and the user is given an option to print the barcode for future use.

UML- See UML file in documentation

Test Cases-

User updates the amount of stock in inventory-  This can include adding and subtracting the user should be able to scan the products or enter the data if adding existing stock and the program should update the database. The user should be able to create an upc and enter data for a new product.

View Customer- User should be able to click the view inventory button and enter the last name of a customer and find all the information available on that customer. A new query can be sent with another click of the search button or the back button can be clicked to return to the home screen.

View inventory- same as view customer, but it will show product information based on name entered or upc scanned or entered.

Generate UPC-  Users should be able to click the generate upc button and be brought to a screen that lets them enter a new number for an upc. It should display the upc and give a print option as well as add the product to the database after the print option is selected.

Print- User should be able to click a button to print a upc or the inventory or customer query.

Summary- This project will be implemented and deployed by the end of the semester. We will be hosting the MySQL database on AWS. There will be user authentication for the database through logging into the application. The technologies used for this are new to the developers and though we would like to deploy this in the actual business we are building this to learn the systems and software development process in more detail. It will also create a portfolio item for potential employers to view. We will continue to add to the documentation as we discover and learn more about the project.