CSC 289 Capstone Project

# S. Apulu

# B. Dalton

# S. Mccartney

# W. MerritT

# Introduction

Using AGILE procedures, our team came together and developed an application that fulfills a business need. The need for a business owner to have a way of tracking inventory and managing employees is important for running a business. Also, employees must be able to search for products and get products details. Using C# along with other technologies this became our focus for developing our application.

Small business owners looking to expand their business may need to use other means of tracking inventory. As a business grows inventory may grow as well. The job of keeping track of that growing inventory can be overwhelming. A database to keep track of that inventory would help with the job of inventory tracking.

As a small business grows, the number of its employees may grow too. Not only that, but employees come and go for whatever reasons. Dealing with employee in/out processing is simplified with our application.

Our team used C# along with other technologies to develop this desktop app. These technologies include WPF (Windows presentation foundation), XAML (Extensible application markup language), SQL (Structured query language) and Dapper (Object-relational mapping).

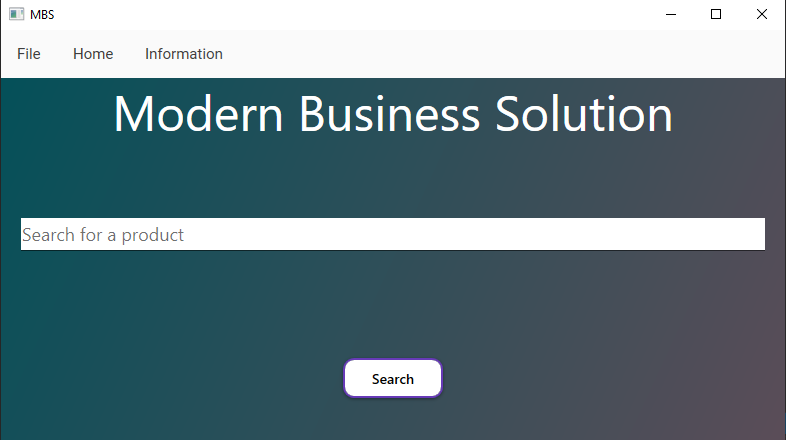
We used Microsoft Visual Studios 2019 as our IDE. Visual Studios allows us to add a local database to our project to store relevant information for our app. We used Dapper, which is a helpful tool that assists with object-relational mapping. In more simpler terms, it helps us retrieve and send data to and from our database.

WPF uses XAML, which is similar to HTML. We used WPF to design how our user-interface looks. There are many controls in WPF that offer a wide range in flexibility to a developers’ UI. WPF also used in ASP.NET projects, which are used to create web-based applications.

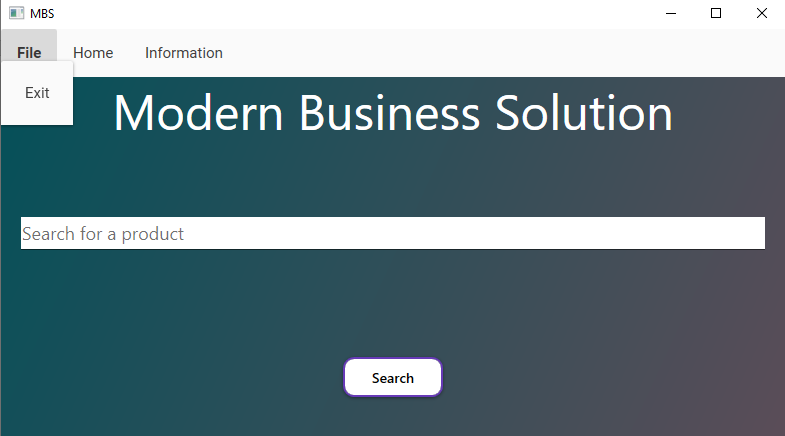
The Model-View-ViewModel structure was the architecture we decided to implement in our application. The MVVM model separates the UI (View), the business logic (ViewModel) and the data (Model). This is a newer software architecture that we had to research on our own.

When I think back, we could have easily made a WinForm App and saved ourselves a lot of researching and headache. Instead, our team decided to go down less familiar roads. So, a lot of experimentation took place as we delve into the world of WPF and MVVM (Model-View-ViewModel) Structure. A world we were, for the most part, strangers to. At times, the task seemed daunting, but I know that going through this unfamiliar territory only made us better at our craft.

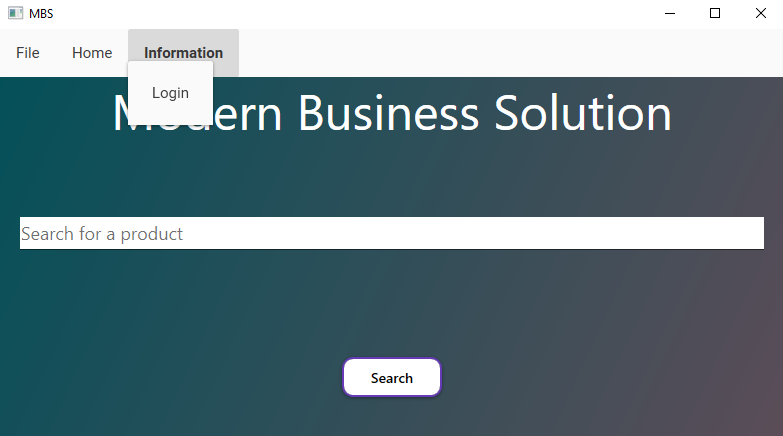
# USER MANUAL



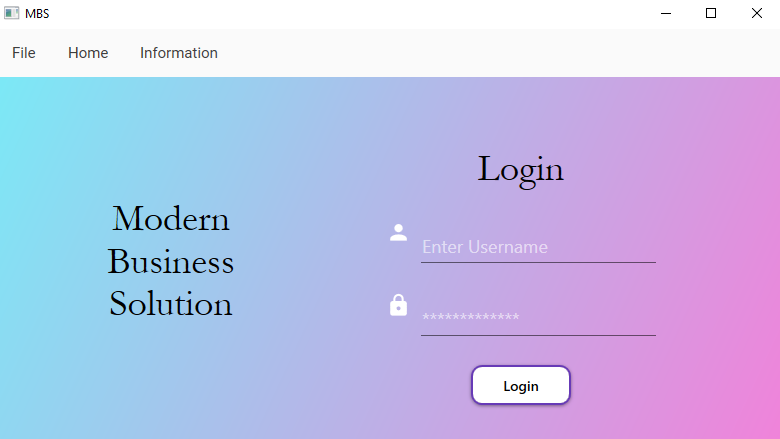
* This is the main page, or the page that shows up when the program is first started. The main page is also treated as the home screen, and it can be accessed clicking the home tab on the menu bar.
* The program may also be closed with the use of the “X” button on the top right of the program.



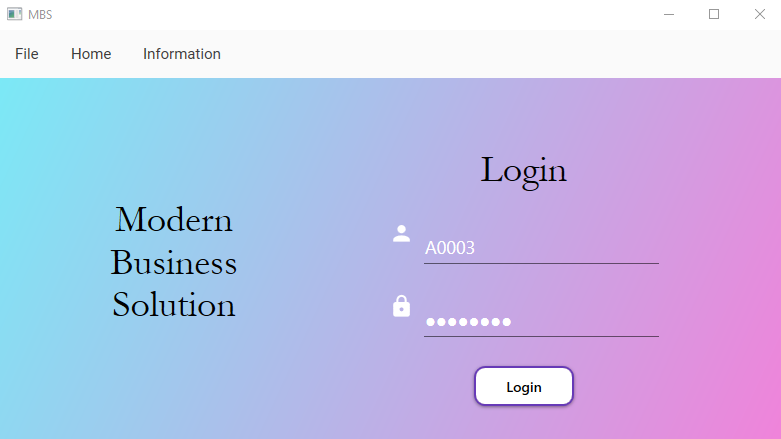
* Starting with the menu bar the “File” tab allows the user to close the program when they are done using it.



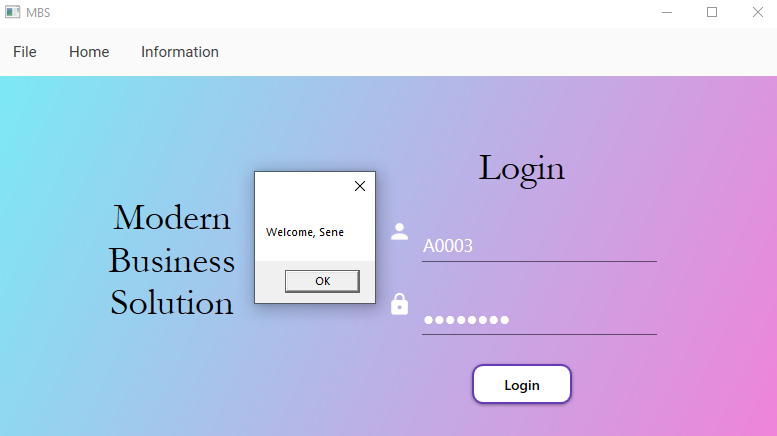
* The “Information” tab holds the menu items that allow the user to login to their account and access new information depending on their access levels.



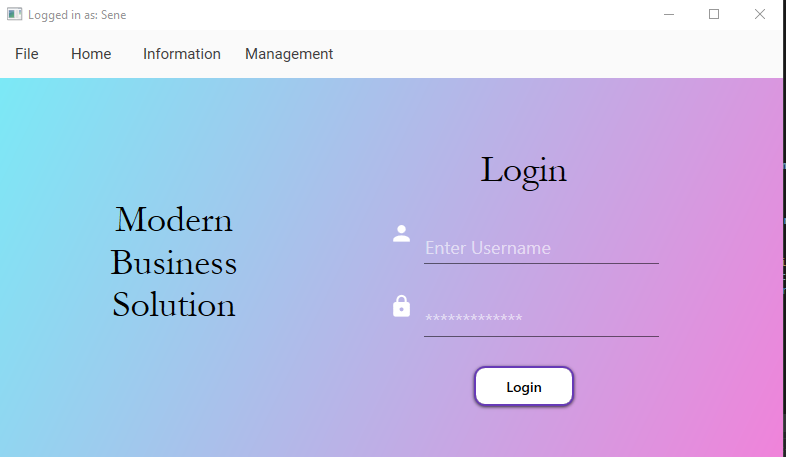
* The user will be brought to the login page after selecting the “Login” button from the menu bar. The login page allows an employee or manager to login to view account information.



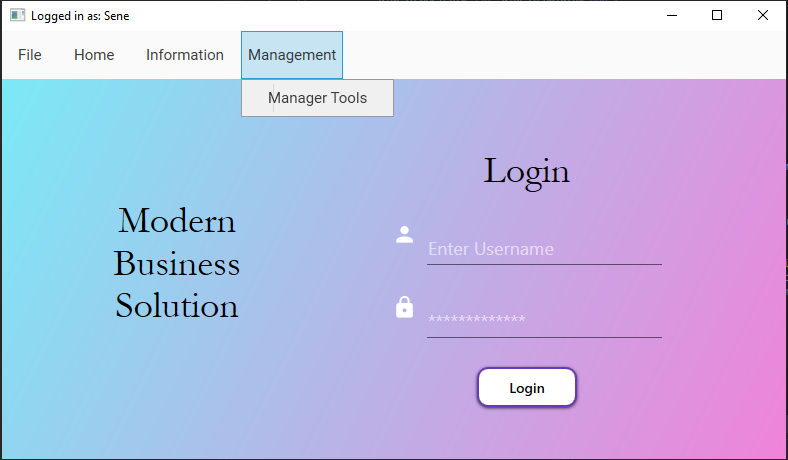
* After typing in the proper login information, the user will be greeted by a “welcome message” or a “login unsuccessful” message.



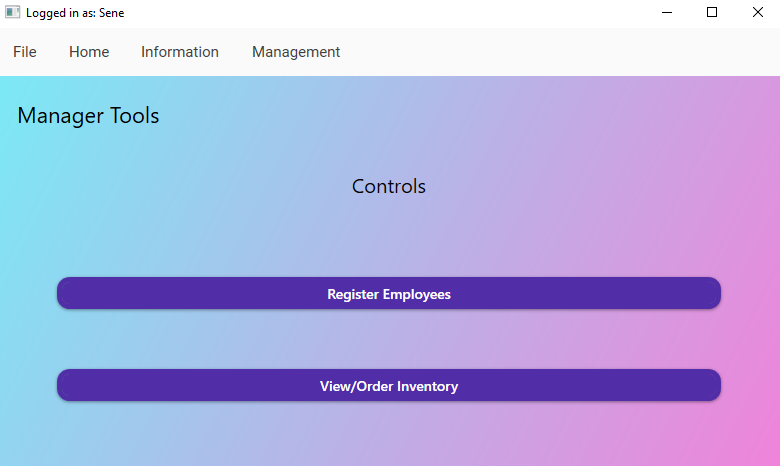
* In this instance the user has login in successfully and must now hit “OK” to continue using the program.



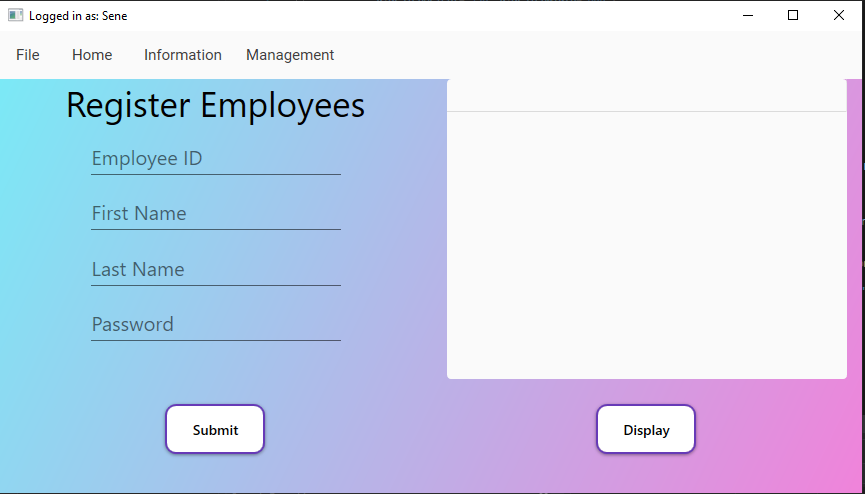
* After logging in the user will be greeted by a new tab depending on their access level given by the store owner.



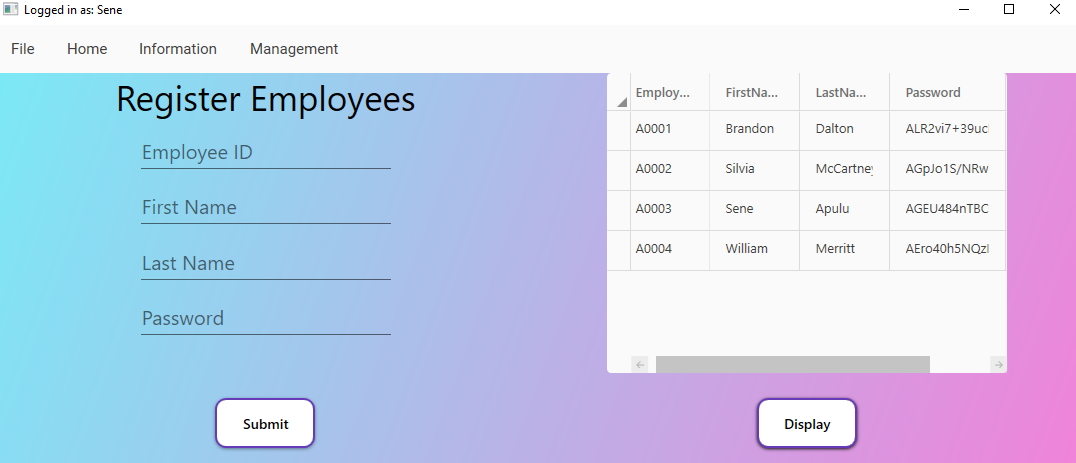
* The managers of the store have access to the “Manager Tools” tab/page. The Allows the manager to register new employees or check inventory that needs to be ordered.



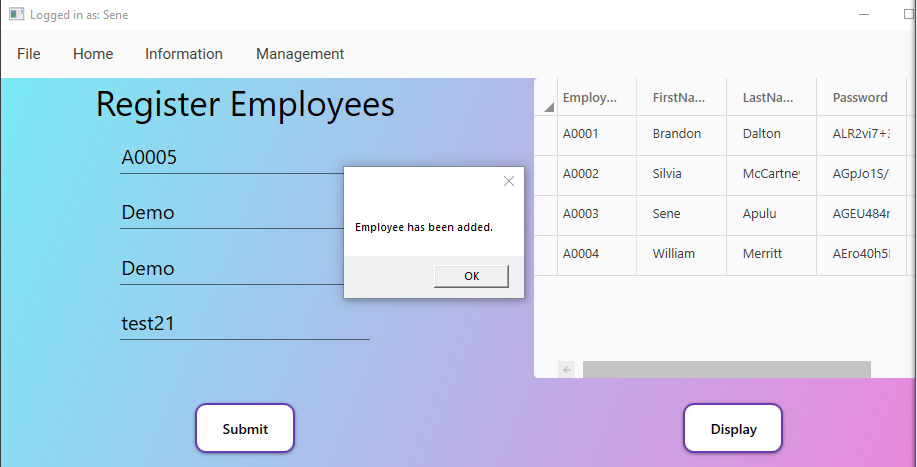
* This view holds the buttons the managers can use to view the current employees or the inventory that the company is currently is low on and may need to order.



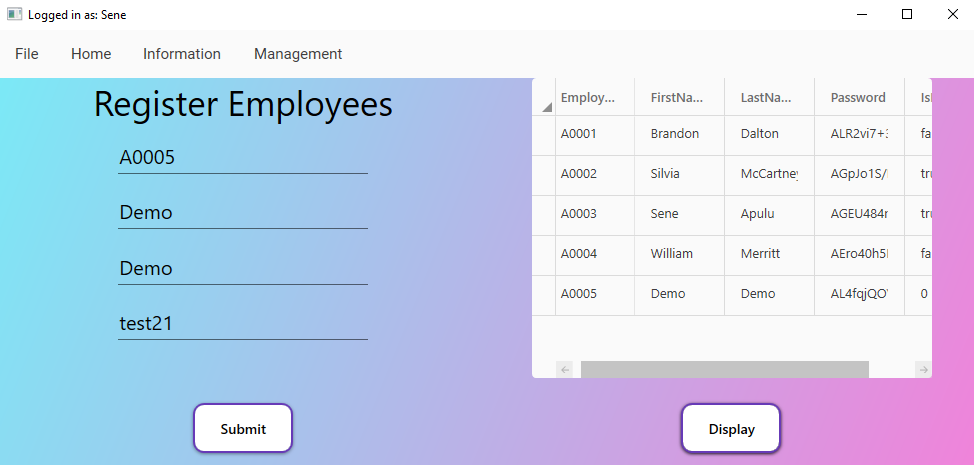
* This is the view the user is taken to after selecting the “register employee” button.



* This is the updated view after selecting the “Display” Button. The manager is shown all the employee currently working for the company.



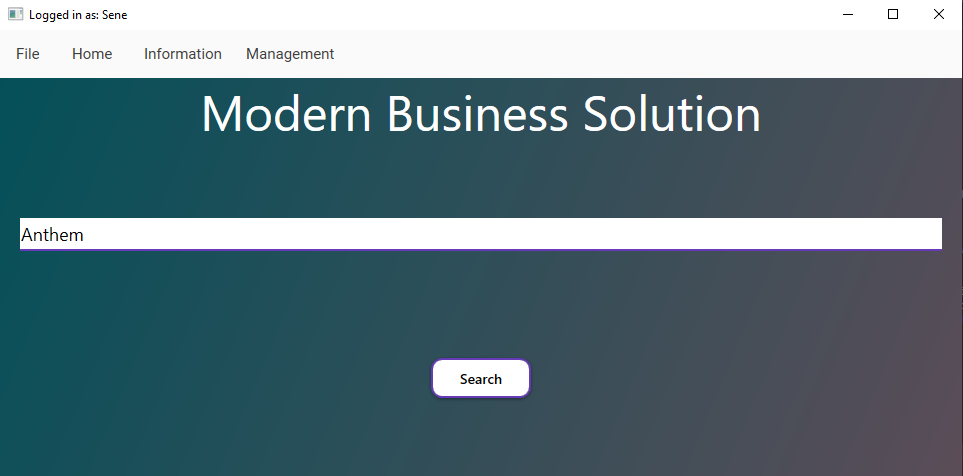
* These slides show the process of adding a new employee and showing them newly added into the database.



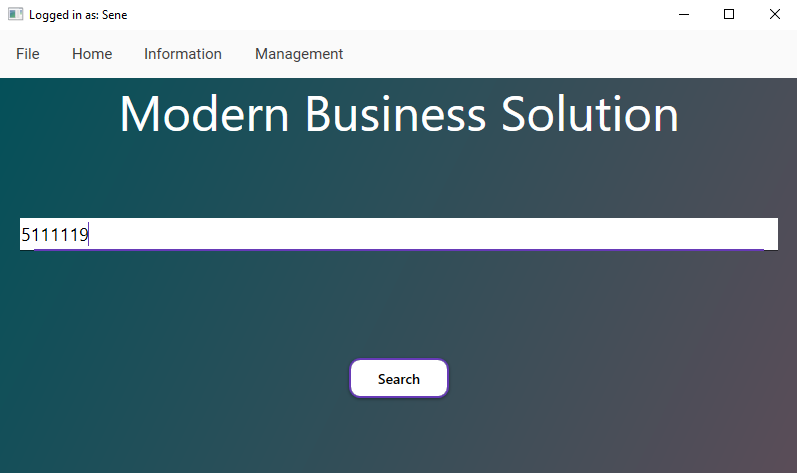
* The new user is displayed in the database with a hashed version of their password as well.



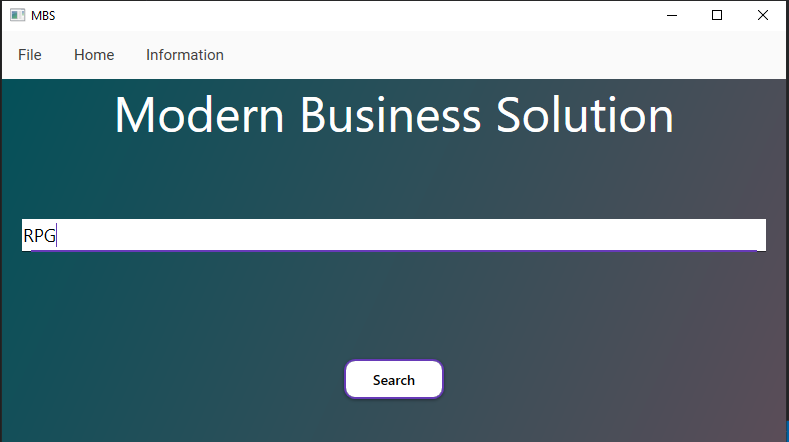
* The other option in the managers tools is the “Check Inventory” button that shows the manager a detailed report of what items are low and may need to be ordered.



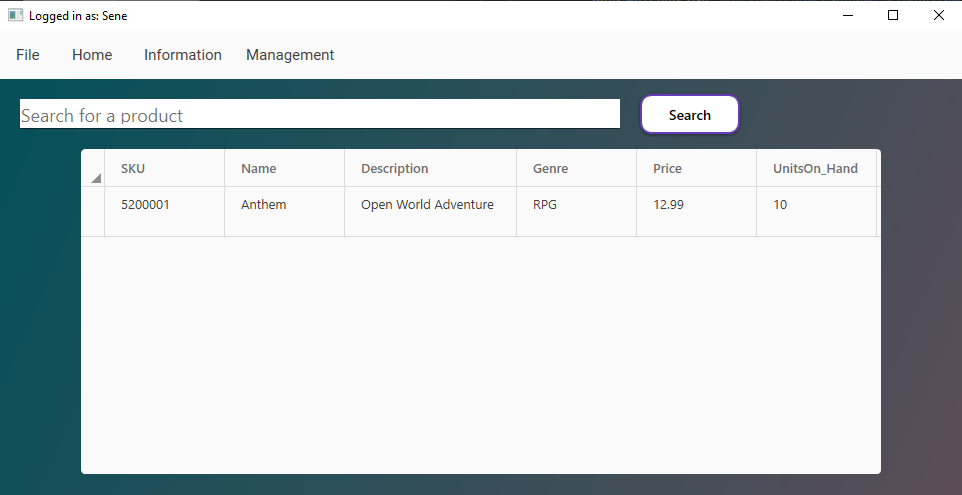
* The home screen acts as a search page as well. The main page allows the user to search for inventory and see if we have it in stock.



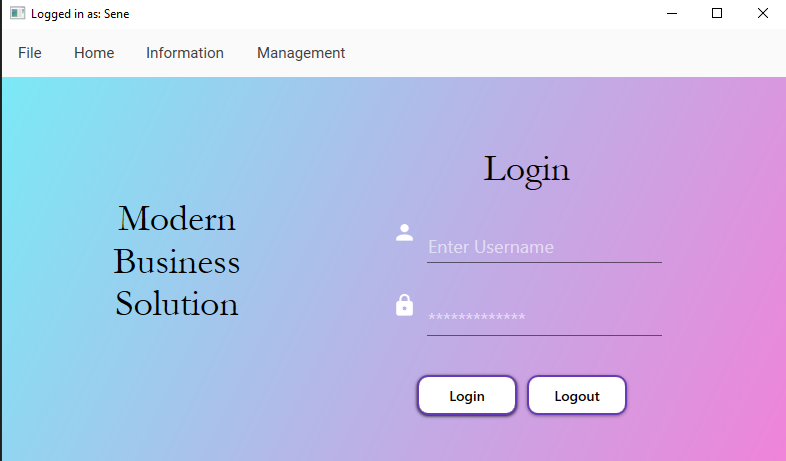
* Showing that the customer can search by “SKU” number. Allowing them another way to find products.



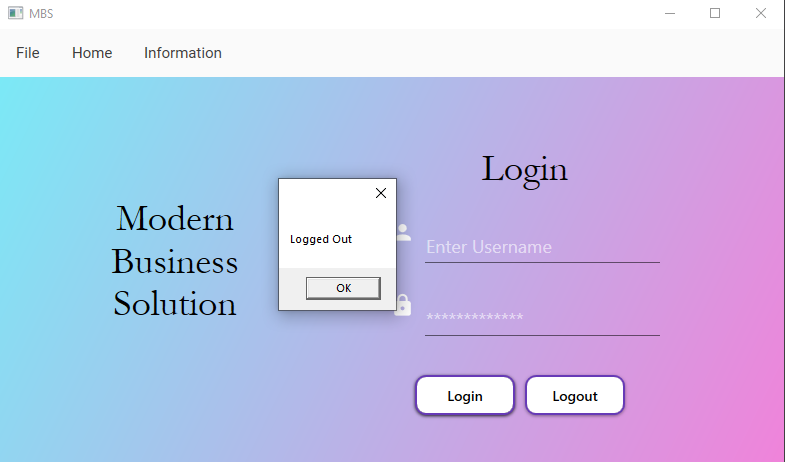
* Showing the user that they can search by genre as well.



* After clicking the search bar, the user is brought to the results page. The results page allows the user to search for more products if they choose to do so.

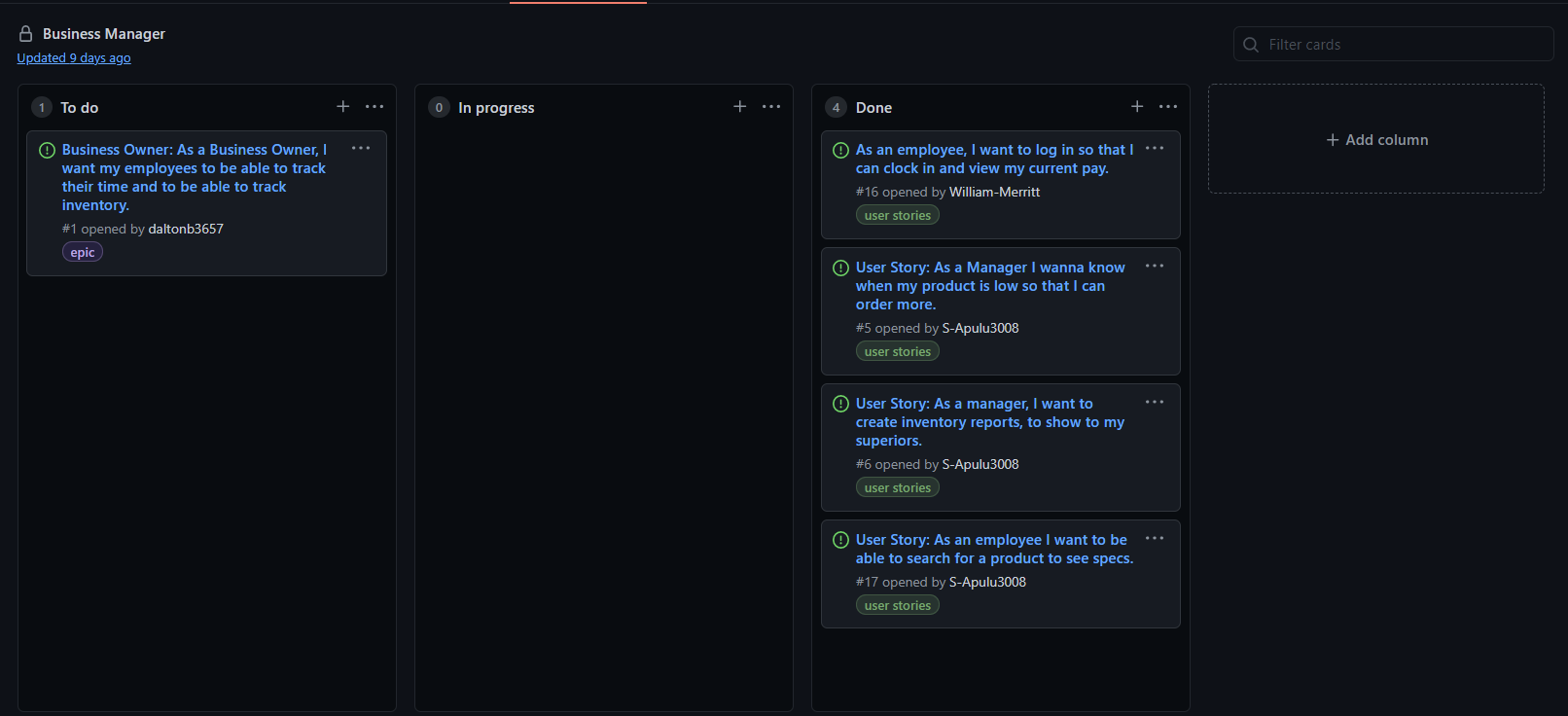


* The logout button is found on the login page. This allows the user to logout and quit their session.



* The user has been logged out, and the management tab has been hidden from public use. The name at the top of the screen has also been taken away. You are now back to the main program.

# User stories



* (EPIC)As a Manager, I want to manage inventory and employees.
* (User Story) As an owner, I want to give employees different levels of access.
* (User Story) As a Manager I want to be able to add, delete, modify employees.
* (User Story) As a Manager, I want to add, delete, modify products.
* (User Story) As an Employee, I want to search for products.
* (User Story) As an Employee, I want to view my search results on another screen.



* We want the employee to have more power with the program. In future sprints would we like to give the employee the power to clock in/clock out and view their time sheets.
* At the start of the project our main idea was to recommend projects to the user. Because of time and difficulty, we ended up having less time to add this feature. Later we would like to completely add this feature to the program.
* A P.O.S. system would allow the employee or manager to sale inventory and order more inventory for the customer. Currently, the program is just used to check inventory and shows off the user of manager options. This would take more time than just one semester to complete.
* Adding/Deleting products was brought up towards the end of the program. Since this is just the start of a program that businesses can use, we wanted to show basic functionality, so we could ask for more funding for later.

# Data Flow Diagrams /Data Dictionary

Context Diagram:

Diagram

Description automatically generated

Data Dictionary:

Employee Table

|  |  |  |
| --- | --- | --- |
| **Field** | **DataType** | **Notes** |
| EmployeeID | NChar(10) | Primary Key; Used to sign in; |
| FirstName | NVarChar(50) |  |
| LastName | NVarChar(50) |  |
| Password | NVarChar(50) | Hashed with ASP.NET PasswordHasher |
| IsManager | NChar(10) | True, if employee is a Manager |

Products Table

|  |  |  |
| --- | --- | --- |
| **Field** | **DataType** | **Notes** |
| SKU | Int | Primary Key |
| Name | NVarChar(50) | Product Name |
| Description | NVarChar(50) | Product Description |
| Genre | NVar(10) | Product Genre |
| Price | Money | Product Price |
| Units\_On\_Hand | Int | Number of Product on hand |

# Source code

<https://github.com/S-Apulu3008/CapstoneProject>