Logo, company name

Description automatically generated

Iteration 4 Document for:

*The Department of Information Systems*

INF4027W 2021

Prepared by Team Waffle (Team 11)

# Plagiarism Declaration

I know that plagiarism is wrong. Plagiarism is to use another’s work and pretend that it is one’s own. I have used the APA 7th convention for citation and referencing. Each contribution to, and quotation in, this proposal from the work(s) of other people has been attributed and has been cited and referenced. This document is my own work. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as his or her own work. I acknowledge that copying someone else’s assignment or essay, or part of it, is wrong, and declare that this is my own work. I have not falsified or manufactured any data and declare that all data was ethically collected.

|  |  |  |
| --- | --- | --- |
|  |  | Shape  Description automatically generated with medium confidence |
| Matete Kopano Lebakeng | Kimberley Mugadza | Khanya Ngxabani |

Table of Contents

[Plagiarism Declaration 4](#_Toc86322565)

[List of Tables 7](#_Toc86322566)

[Table of Figures 7](#_Toc86322567)

[1. Project Background 8](#_Toc86322568)

[2. Business Problems 8](#_Toc86322569)

[3. Business Objectives 9](#_Toc86322570)

[4. System and System Objectives 9](#_Toc86322571)

[4.1. Critical Assumptions, Constraints and Exclusions 10](#_Toc86322572)

[4.2. Business Value 11](#_Toc86322573)

[4.2.1. For store owners: 11](#_Toc86322574)

[4.2.2. For Cog3nt: 12](#_Toc86322575)

[5. High Level Package Diagram 13](#_Toc86322576)

[6. Use Case Diagram 14](#_Toc86322577)

[7. Use Case Narratives 14](#_Toc86322578)

[8. Class Diagram 22](#_Toc86322579)

[9. Database Schema 24](#_Toc86322580)

[10. State Machine Diagram 25](#_Toc86322581)

[11. Security 26](#_Toc86322582)

[12. Revised System Architecture 26](#_Toc86322583)

[13. User Interface/ Wireframes 28](#_Toc86322584)

[14. Risk Assessment 36](#_Toc86322585)

[15. Development Approach 45](#_Toc86322586)

[Product Backlog 46](#_Toc86322587)

[15.1 Project and Iteration Plan 47](#_Toc86322588)

[Sprint 1: 03/10/2021 – 10/10/2021 47](#_Toc86322589)

[Sprint 2: 10/10/2021 – 17/10/2021 48](#_Toc86322590)

[Sprint 3: 17/10/2021 – 24/10/2021 49](#_Toc86322591)

[15.2 Velocity: 50](#_Toc86322592)

[16. Test Plan 50](#_Toc86322593)

[Testing Strategies 50](#_Toc86322594)

[Test Cases – MSP System 51](#_Toc86322595)

[18. Conclusion 53](#_Toc86322596)

[19. References 54](#_Toc86322597)

# List of Tables

[Table 1: MSP Regression Algorithm Variables 16](#_Toc83190485)

[Table 2: MSP Security Functionalities 20](#_Toc83190486)

[Table 3: Risk Status Key 31](#_Toc83190487)

[Table 4: Risk Assessment 32](#_Toc83190488)

[Table 5: Sprint backlog Tasks (sprint 1) 40](#_Toc83190489)

[Table 6: Sprint Backlog tasks (sprint 2) 41](#_Toc83190490)

[Table 7: Sprint backlog Tasks (sprint 3) 42](#_Toc83190491)

[Table 8: Sprint Backlog Tasks (sprint 4) 43](#_Toc83190492)

[Table 9: Test Cases 44](#_Toc83190493)

# Table of Figures

[Figure 1: High Level Package Diagram 10](#_Toc86322686)

[Figure 2: MSP Use-Case Diagram 11](https://uctcloud.sharepoint.com/sites/SDTeam11/Shared%20Documents/General/Document%20submissions/Iteration%204/Iteration%204_es%20draft.docx#_Toc86322687)

[Figure 3: MSP Class Diagram 21](#_Toc86322688)

[Figure 4: MSP Database Schema 22](#_Toc86322689)

[Figure 5: MSP Prediction Engine State Machine Diagram 22](#_Toc86322690)

[Figure 6: Revised System Architecture 25](#_Toc86322691)

[Figure 7: MSP Home Page 26](#_Toc86322692)

[Figure 8: User Registration Form 27](#_Toc86322693)

[Figure 9: Login Form 28](#_Toc86322694)

[Figure 10: Product Maintenance Page 29](#_Toc86322695)

[Figure 11: Add New Product Pop-up Form 30](#_Toc86322696)

[Figure 12: Edit Product Pop-up Form 31](#_Toc86322697)

[Figure 13: Delete Product Warning Message 32](#_Toc86322698)

[Figure 14: Reports Page 33](#_Toc86322699)

# 1. Project Background

Research conducted by Cog3nt found that store owners ranging from spaza shops, quick shops, tuck shops, and local stores have limited storage and floor space to keep their inventory. Research conducted by Goyal (2020) also found that most of these businesses have their processes like stock taking and managing sales records done manually. The MSP will fall under one of Cog3nt’s product categories called Retialyx. The MSP aims to help retail businesses improve their processes through intelligent automation by integrating artificial intelligence into their solutions with this product line. The type of artificial intelligence that will be used for this project is Machine Learning which will predict sales quantities based on previous trends for a specified store.

# 2. Business Problems

Inventory management is essential in making all inventory management decisions in an organization. This includes the activities like inventory management policies and inventory management procedures. Performing these activities ensures there is always enough stock available for sales (Shiau Wei Chan et al., 2017). One of the most significant challenges faced by South African SMEs, (i.e. Business problems) relates to ineffective management of inventory which negatively impacts their performance. Though there are many reasons that can affect a business’s performance, one of the biggest factors is the use of a manual inventory tracking method.

1. The issue with most SMEs in the inventory management business is that they record data manually. Their offices have a file department, where employees must go before beginning work to get the company's records on a particular customer or supplier. The disadvantage of this approach is that physical filing takes up a lot of space, and managing paper-based data, particularly historical data, is expensive. And it is not a dependable method of conducting business since a piece of paper may easily be misplaced and hunting for a misplaced file among a pile of papers is time-consuming and annoying.
2. SME's with at least some computer skills also utilize spreadsheet templates to track their inventories. Understandably, adopting spreadsheet templates is a cost-effective alternative that comes pre-installed with your Microsoft Office suite. As a result, spreadsheets are a cost-effective solution for businesses that are just getting started. However, when your business grows, it will encounter obstacles and spreadsheets will no longer suit its requirements (Quickbooks Commerce, 2018).
3. The use of spreadsheet templates to track inventory for store owners is a manual process, as the spreadsheet is not automatically updated when a product is sold or when a customer makes a purchase. When utilizing an inventory spreadsheet template, administration and monitoring of inventory tasks must be done in your brain. Additionally, utilizing a spreadsheet to manage inventories makes collaboration and updating the template difficult for the entire team. As a result, several unique edits and versions of a single inventory spreadsheet file may be generated, resulting in confusion and additional time spent sorting through and creating the 'final' and 'most updated' version (Quickbooks, Commerce, 2018). The result of this manual method of inventory management is an increase in labour and time wastage, an increase in organizational stress, and human errors. As your business grows, it can no longer rely on outdated processes such as those stated above.

As a result of the above problems Cog3nt has found an opportunity to automate these time-consuming data maintenance operations using CRUD. The recorded data will be centered on the customer, supplier, and products, and will be stored on an electronic form than on piece of paper or spreadsheet template. The computer can readily save this data for us in a database of our choice, allowing us to alter the existing data. By automating operations in this manner, the MSP application benefits organizations by reducing paper usage, lowering labour expenses, saving time, and offering additional features such as the ability to search for data.

# 3. Business Objectives

Cog3nt’s objective is to help businesses improve their operative efficiency by automating their current processes with the use of Artificial Intelligence (AI).

The business aims to assist small-medium enterprises to better themselves with the help of their automated solutions.

# 4. System and System Objectives

Store owners need a web-based application that processes historical inventory levels and transactions to perform a predictive analysis. Based on the problem statement, the proposed solution should achieve the following objectives:

1. The system should be able to identify the closet quantity that the user of this system should order for each identified product. The algorithm will be able to predict the quantity that will be sold at an accuracy rate of 98%.

2. The algorithm will also be able to predict the minimum and maximum sales quantities for each individual product at a accuracy rate of 98% which was calculated from the R2 which is the regression score.

3. The algorithm should be able to predict future sales quantities for every month of the year.

4. The system must be safe and secure and allow only authorised users to perform administrative tasks and forecasting of sales.

5. The system should have a functionality that allows users to manage and store information about their products on an online platform. The user should be able to add, update and delete a product and save this information on a central database. The user should also be able to view a record of all products in the database on an interface.

6. The system, with help of artificial intelligence, should be able to provide retail stores with a reasonable count of predicted stock quantities. The system will provide the user with predicted values, which the user can use to test ongoing accuracy by comparing the real value sold to the forecasted quantity.

7. The system must be able to maintain a systemic record of inventory by allowing users to import their excel spreadsheets onto the MSP.

## 4.1. Critical Assumptions, Constraints and Exclusions

In the problem statement, we mentioned that the methods used by retail stores to manage their stock vary depending on the resources to which certain stores have access. Some stores within this category could be using stock-planning technologies in their processes while others not. Based on this, the following can be assumed about our project:

1. The stores currently do not have any extensive ERP system installed to manage their stock.

2. The stores are small with examples being local grocery stores that are owned by sole traders, quick stops at fuel stations and tuck shops.

3. The current stock prediction and product maintenance processes used by these stores is mostly manual. In other words, most of the tasks in these processes are completed by people entering or recording data into a device. We also assume that stock prediction quantities are based on generic estimates that are calculated manually.

The following constraints apply to this project:

1. Time: the project is due on Monday the 25th of October. This gives us about 6 months to complete this project.
2. Data: we will be using one dataset for this project which has been provided by Cog3nt. The dataset is from a small convenience store. The name of the retail/ grocery chain that owns this convenience store cannot be disclosed because of privacy concerns. Ideally, the system should be tested out using a variety of datasets to test its adaptability but for this project, we will be using one dataset and one scenario.

Lastly, the following will be excluded from this project:

1. Robotic Process Automation (RPA). In the background, we mentioned that Intelligent Automation uses a combination of RPA and AI to improve business processes. This project will focus on the first stage of this process which relates to the AI component. This project will, therefore, be the pilot stage in the automation process.

## 4.2. Business Value

### 4.2.1. For store owners:

* **More Accurate Stock Prediction:** Having a manual stock management system means that small retailers must check if they need to update their stock every day or week. According to Jane (2017), manual stock management systems are at a higher risk of inaccuracies. The MSP will reduce, if not eliminate, the risk of human error because its analyses and predictions will be based on figures and statistics.
* **Saves Time**: Manual stock management processes also require more effort from employees. Employees need to keep a consistent track of stock sales and review the system at the end of each business day. The MSP will help to reduce the workload on staff while saving them time because it will perform the prediction part of the process on their behalf. This will also improve business productivity and operative efficiency.
* **Centralised Storage Space:** The MSP will also give store owners a platform to manage their store information relating to their products and where relevant, their suppliers. This information will be stored on a local database.
* **Less Wastage:** Based on the interviews conducted with Rustenburg Pharmacy and Marcels, both participants believe that having wastage or products expiring on shelves is inevitable for any businesses. The use of the MSP can help reduce the number of items that are wasted even if it is inevitable for some items to still be wasted.
* **Optimised Sales:** The use of the MSP will help store owners optimise their sales which will result in less wastage of stock. Store owners have limited storage and floor space to keep their inventory so they must turn over stock as often as possible to keep up with consumer demand. The MSP’s algorithm will be trained to predict the required quantities of stock to optimum levels. This means that if the MSP predicts that 20 loaves should kept in storage, at least 16 or more loaves should be sold. This will reduce the number of units going back into storage and expiring on shelves. The MSP algorithm will also be trained to consider seasonal patterns and consumer preferences. This means that certain items like chocolates will have higher predicted quantities around times of the year like Mothers’ Day or Valentine’s Day.
* **Optimise Available Resources:** Store owners have limited storage and floor space to keep their inventory. Cog3nt’s aim for the MSP is to have it become a tool that helps store owners maximise the resources they have. If sales are optimised and wastage of stock is reduced, business profits/ revenues will be higher.

### 4.2.2. For Cog3nt:

* Cog3nt will expand their product line which will bring the company closer to achieving its goal of providing businesses with automotive solutions to their problems.
* Cog3nt will also benefit from making a positive contribution to its industry by helping its clients make certain aspects of their businesses better.

# 5. High Level Package Diagram

This package diagram outlines the revised scope of the system. Each component shown in this diagram is discussed in more detail in sections 8 – 11. With regards to the “Dashboard Plugins” shown in the diagram, these include libraries like Chart.js and High Charts which were used to display the reports and graphs for the product maintenance and stock prediction pages on the system.

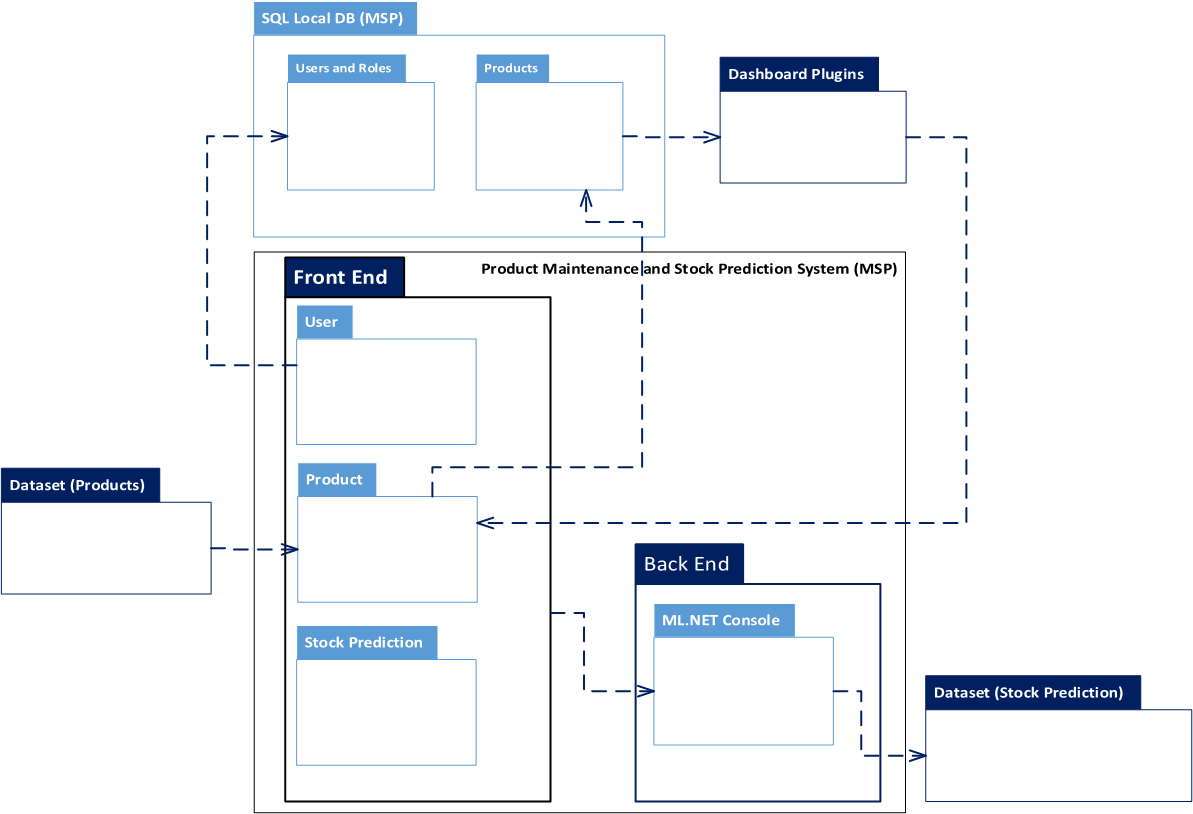
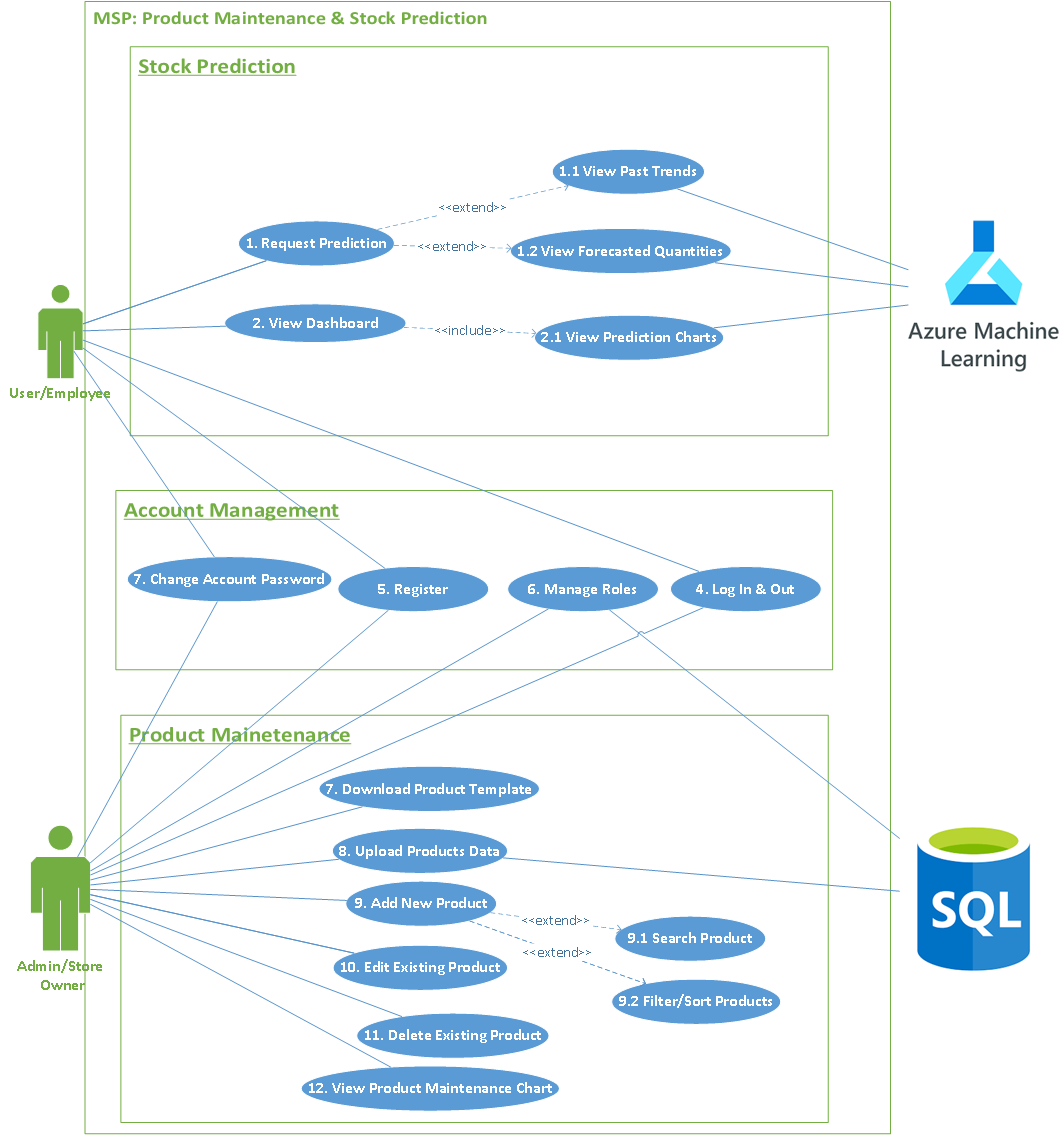


Figure 1: High Level Package Diagram

# 6. Use Case Diagram



*Figure 2: MSP Use-Case Diagram*

# 7. Use Case Narratives

The following Use Case Narratives model the use cases the team developed in this iteration, as well as what was developed in the last one.

|  |  |  |
| --- | --- | --- |
| Name: | | Request Prediction |
| ID: | 1 | |
| Actor: | User/Employee and Store Owner | |
| Goal: | The user and storeowner should be able to request prediction | |
| Pre-conditions: | The user/store owner must be logged in to the system. | |
| Post-conditions: | The user/store owner must be able to retrieve the predictions for each product | |
| Main flow of events: | 1. The user enters login credentials in the login form. 2. The user clicks on the Stock Prediction page. 3. The user enters the product ID or product name in the search bar to request for a prediction. | |
| Alternative Course of events: | A2: Incorrect details entered   * 1. The user enters the incorrect product ID in the search bar   2. The user enters the incorrect product names in the search bar   3. Return to step 3. | |

|  |  |  |
| --- | --- | --- |
| Name: | | View Past Trends |
| ID: | 1.1 | |
| Actor: | User/Employee and Store Owner | |
| Goal: | The user and storeowner should be able to view past trends. | |
| Pre-conditions: | The user/store owner must be logged in to the system. | |
| Post-conditions: | The user/store owner must be able to retrieve the past trends data for each product. | |
| Main flow of events: | 1. The user enters login credentials in the login form. 2. The user clicks on the Stock Prediction page. 3. The user enters the product ID or product name in the search bar to request for a prediction. 4. The system displays the graph for the requested product. 5. The user views the past trends in previous months for the forecasted product. | |
| Alternative Course of events: | A3: Incorrect details entered  1.1. The user enters the incorrect product ID in the search bar  1.2 The user enters the incorrect product names in the search bar  1.3 Return to step 3 | |

|  |  |  |
| --- | --- | --- |
| Name: | | View Forecasted Quantities |
| ID: | 1.2 | |
| Actor: | User/Employee and Store Owner | |
| Goal: | The user and storeowner should be able to view forecasted quantities. | |
| Pre-conditions: | The user/store owner must be logged in to the system. | |
| Post-conditions: | The user/store owner must be able to retrieve the forecasted quantities for each product for the next month. | |
| Main flow of events: | 1. The user enters login credentials in the login form. 2. The user clicks on the Stock Prediction page. 3. The user enters the product ID or product name in the search bar to request for a prediction. 4. The system displays a chart for the requested product. 5. The user views the forecasted quantities for the next month. | |
| Alternative Course of events: | A4: Incorrect details entered  1.1. The user enters the incorrect product ID in the search bar  1.2. The user enters the incorrect product names in the search bar  1.3. Return to step 3. | |

|  |  |  |
| --- | --- | --- |
| Name: | | View Dashboard |
| ID: | 2 | |
| Actor: | User/Employee and Store Owner | |
| Goal: | The user and storeowner should be able to view the dashboard. | |
| Pre-conditions: | The user/store owner must be logged in to the system. | |
| Post-conditions: | The user/store owner must be able to view the dashboard and make well-informed decisions. | |
| Main flow of events: | 1. The user enters login credentials in the login form. 2. The user clicks on the Stock Prediction page. 3. The user enters the product ID or product name in the search bar to request for a prediction. 4. View the various charts | |
| Alternative Course of events: | A2: Incorrect details entered   * 1. The user enters the incorrect product ID in the search bar   2. The user enters the incorrect product names in the search bar   3. Return to step 3. | |

|  |  |  |
| --- | --- | --- |
| Name: | | View Prediction Charts |
| ID: | 2.1 | |
| Actor: | User/Employee and Store Owner | |
| Goal: | The user and storeowner should be able to view various prediction charts. | |
| Pre-conditions: | The user/store owner must be logged in to the system. | |
| Post-conditions: | The user/store owner must be able to retrieve the predictions for each product | |
| Main flow of events: | 1. The user enters login credentials in the login form. 2. The user clicks on the Stock Prediction page. 3. The user enters the product ID or product name in the search bar to request for a prediction. | |
| Alternative Course of events: | A2: Incorrect details entered   * 1. The user enters the incorrect product ID in the search bar   2. The user enters the incorrect product names in the search bar   3. Return to step 3. | |

|  |  |  |
| --- | --- | --- |
| Name: | | Register New Account |
| ID: | 5 | |
| Actor: | User/Employee and Store Owner | |
| Goal: | The user and storeowner should be able to login and out of the system. | |
| Pre-conditions: | The user/store owner must have a valid email address to create an account | |
| Post-conditions: | The user/store owner login details must be stored on the database. | |
| Main flow of events: | 1. User requests to register new account 2. System displays the registration interface. 3. User completes required fields 4. System stores details and displays home page. | |
| Alternative Course of events: | A1: Error with details entered   * 1. System notifies the user that they have entered incorrect to check registration details   2. If the user enters incorrect details the textboxes in the registration form will be highlighted in red.   3. The user will have the option to reset the details they have entered by pressing the reset button.   4. Return to step 2. | |

|  |  |  |
| --- | --- | --- |
| Name: | | Login & out |
| ID: | 4 | |
| Actor: | User/Employee and Store Owner | |
| Goal: | The user and storeowner should be able to login and out of the system. | |
| Pre-conditions: | The user/store owner must have a valid email address to create an account | |
| Post-conditions: | The user/store owner login details must be stored on the database. | |
| Main flow of events: | 1. The user enters login credentials in the login form. 2. The user enters information such as valid email, password and selects a role between employee and storeowner. 3. System notifies the user the details are correct details and logs them in. 4. After user has been logged in the system, they can also log out by pressing the log out button. | |
| Alternative Course of events: | A2: Incorrect details entered   * 1. System notifies the user that they have entered incorrect login details   2. If the user enters incorrect details the textboxes in the login form will be highlighted in red.   3. The user will have the option to reset the details they have entered by pressing the reset button.   4. Return to step 2. | |

|  |  |
| --- | --- |
| Name: | Manage Roles |
| ID: | 6 |
| Actor(s): | Store owner |
| Goal: | Store must be able to edit and delete roles |
| Pre-conditions: | Store owner must be logged in. |
| Post-conditions: | Updated user roles or new user roles must be saved and stored in the database. |
| Main flow of events: | 1. Store owner enters login credentials. 2. The system validates credentials and logs the storeowner in. 3. Store owner clicks on “Manage Roles” page. 4. The system displays a list of roles saved in the database. 5. The store owner can either edit each role or assign permissions. 6. The store owner can also remove a user. 7. The storeowner can also add a new user. 8. The system will update these changes and they will be stored in the database. |
| Alternative course of events: | Logs off   1. The storeowner logs out of the system by clicking the log out button 2. Terminate use case. |

|  |  |  |
| --- | --- | --- |
| Name: | | Change Account Password |
| ID: | 4 | |
| Actor: | User/Employee and Store Owner | |
| Goal: | The user and storeowner should be able to enter a new password for their account. | |
| Pre-conditions: | The user/store owner must have a valid email address to create an account | |
| Post-conditions: | The user/store owner’s new account details must be updated in the database. | |
| Main flow of events: | 1. User clicks on Forgot Password link 2. User enters his email ID 3. System verifies an account exists with email id. 4. User hits Reset password button. 5. System confirms password reset instructions were sent to the email. 6. The user enters new password 7. The system verifies new password and updates the users' credentials and stores it in the database. | |
| Alternative Course of events: | 1. The user selects the Forgot Password link. 2. The user provides his email address. 3. The system confirms that an account with the specified email does not address exists 4. The system notifies the user that they do not have an account. 5. Return to step 2. | |

|  |  |  |
| --- | --- | --- |
| Name: | | Download Product Template |
| ID: | 7 | |
| Actor: | Store Owner | |
| Goal: | The user and storeowner should be able to download a product template that they can upload to the system with their own data. | |
| Pre-conditions: | The user/store owner must be logged into the system. | |
| Post-conditions: | The products template must be downloaded and stored in the user’s PC. | |
| Main flow of events: | 1. The user enters their credentials. 2. The system verifies the credentials and logs them in. 3. The user clicks on the “Product Maintenance” page. 4. The user clicks on the “Download Product Template” button. | |
| Alternative Course of events: | A2: Incorrect Details Entered   1. The user enters incorrect credentials. 2. The system notifies the user that the credentials entered are incorrect. 3. Return to step 1. | |

|  |  |
| --- | --- |
| Name: | Upload Products Data |
| ID: | 8 |
| Actor(s): | Storeowner |
| Goal: | To upload an excel spreadsheet of data so that it can be populated on the database. |
| Pre-conditions: | The user must be logged in the system. |
| Post-conditions: | The data must be populated on the database and be displayed on the data table. |
| Main flow of events: | 1. Store owner enters login credentials. 2. The system validates credentials and logs the storeowner in. 3. The store owner clicks on the “Product Maintenance” page in the navigation bar. 4. In the products maintenance page, store owner clicks on the “choose file” button and selects an excel file. 5. After they select the file, the store owner clicks on the “Upload to database” button and the data is then populated and saved on the database. 6. Store owner reloads the products page. 7. The system displays the data on the data table. |
| Alternative course of events: | A5: Store owner selects an incorrect file type   1. The store owner selects a file that isn’t an excel file 2. The system will notify the user that an incorrect format for the file was selected and that they have to choose an excel file. 3. The data table will read as follows “No available data has been entered”. 4. If the store owner selects an excel file return to step 4. |

|  |  |
| --- | --- |
| Name: | Filter/Sort Products |
| ID: | 9.2 |
| Actor(s): | Store owner |
| Goal: | The store owner must be able to filter or sort products they need from the data table. |
| Pre-conditions: | The store owner must be logged in. |
| Post-conditions: | The products must be filtered or sorted according to the storeowners needs. |
| Main flow of events: | 1. The user enters correct login details. 2. System verifies details and logs the user in 3. The user clicks on the “Product Maintenance” page. 4. The user clicks on the heading of the column to filter products according to product type. 5. The system displays the products under each product type. 6. The user clicks on the heading of the column to sort products in ascending or descending order. 7. The system displays products in ascending or descending order. |
| Alternative course of events: | A2: Enters incorrect Login details   1. Store owner enters incorrect login credentials. 2. The system notifies user incorrect details were entered. 3. Return to step 1. |

|  |  |
| --- | --- |
| Name: | Search Product |
| ID: | 9.1 |
| Actor(s): | Store owner |
| Goal: | The store owner must be able to search for the product they need from the data table. |
| Pre-conditions: | The store owner must be logged in. |
| Post-conditions: | The product must be displayed that is being searched using the search bar. |
| Main flow of events: | 1. The user clicks on the search bar. 2. The user types the name of the product. 3. The system returns the product and its details in the data table. |
| Alternative course of events: | A2: Enters incorrect name of product   1. Store owner enters login credentials. 2. The system validates credentials and logs the storeowner in. 3. The store owner clicks on the products page in the navigation bar. 4. The user enters incorrect name of the product in the search bar. 5. System will display no product. 6. If the store owner enters correct name of the product return to step 2.     A3: Product does not exist   1. The user enters product name of a non-existing product in the database. 2. The system will display no product in the database. |

|  |  |
| --- | --- |
| Name: | Add new Product |
| ID: | 9 |
| Actor(s): | Store owner |
| Goal: | The store owner must be able to add a new product in the products page. |
| Pre-conditions: | The store owner must be logged in. |
| Post-conditions: | The product must be saved and stored in the database. |
| Main flow of events: | 1. Store owner enters login credentials. 2. The system validates credentials and logs the storeowner in. 3. The store owner clicks on the products page in the navigation bar. 4. The store owner clicks on the “Add New Product” button. 5. The system displays a pop-up form. 6. The store owner enters correct product information in the form. 7. Store owner presses the submit button 8. The product is then saved in the database. |
| Alternative flow of events: | A5: Store owner enters incorrect product information.   * 1. The system will alert the store owner if they enter incorrect information on the form using input data validations.   2. The textbox will have a red border and a message notifying the store owner of the error.   A5.1: Reset the form   * + 1. The store owner can reset the whole form if they have made errors on it and start afresh.     2. Return to step 6.   A5.2: Store owner logs off   * + 1. The store owner clicks on the log out button.     2. Terminate use case. |

|  |  |
| --- | --- |
| Name: | Edit Existing Product |
| ID: | 10 |
| Actor(s): | Store owner |
| Goal: | The user must be able to edit an entry from the data table. |
| Pre-conditions: | The store owner must be logged in |
| Post-conditions: | The new edited product must be saved in the database. |
| Main flow of events: | 1. Store owner enters login credentials. 2. The system validates credentials and logs the storeowner in. 3. The store owner clicks on the products page in the navigation bar. 4. The store owner clicks the “Edit” button on the product that they want to edit. 5. The system displays a pop-up form. 6. The store owner edits product information for an existing product using the pop-up form. 7. Store owner presses the submit button. 8. The product is then saved in the database. |
| Alternative course of events: | A7: Input Data Validations   * 1. The system will inform the store owner if they make an error on the form. E.g., adding a letter instead of digits in a text box.   2. The textbox will have a red border and a message notifying the store owner of the error.   3. The store owner can reset the whole form if they have made errors on it and start afresh.   A7.1: Reset Form   * + 1. The store owner can reset the whole form if they have made errors on it and start afresh.     2. Return to step 6. |

|  |  |
| --- | --- |
| Name: | Delete Existing Product |
| ID: | 11 |
| Actor(s): | Store Owner |
| Goal: | The store owner must be able to remove a product that is no longer needed from the database. |
| Pre-conditions: | The store owner must be logged in. |
| Post-conditions: | The product must be removed from the database. |
| Main flow of events: | 1. Store owner enters login credentials. 2. The system validates credentials and logs the storeowner in. 3. The store owner clicks on the products page in the navigation bar. 4. The store owner clicks the “Delete” button on the product that they want to remove. 5. The system displays a pop-up form that notifies the store owner if they are certain that they want to remove the product. 6. The store owner clicks on “Yes” and the product is removed from the data table and the database. |
| Alternative course of events: | A7.2: Storeowner clicks “No”   1. The store owner can click “No” if they are not certain that they want the product to be deleted from the data table and database.   Store owner logs off   1. The store owner clicks on the log out button. 2. Terminate use case. |

|  |  |
| --- | --- |
| Name: | View Product Maintenance Chart |
| ID: | 12 |
| Actor(s): | Store owner |
| Goal: | The store owner must be able to view the product maintenance chart and be able to count the number of products in each product type. |
| Pre-conditions: | The store owner must be logged in. |
| Post-conditions: | The store owner views the number products under a specified product type |
| Main flow of events: | 1. Store owner enters login credentials. 2. The system validates credentials and logs the storeowner in. 3. The store owner clicks on the “Product Maintenance” page in the navigation bar. 4. The user clicks on the pie chart until they see a product type of their choice 5. The system filters all products under that specified product type and displays a percentage of products in that specified product type. |
| Alternative course of events: | Store owner Logs off   1. The store owner clicks the log out button 2. Terminate use case. |

# 8. Class Diagram

A regression algorithm was used for the machine learning component of this system. The main class that is central to the backend is the ProductData class that takes the data from the spreadsheet and loads it into the console. The dataset we received was cleaned to obtain the following variables:

Table 1: MSP Regression Algorithm Variables

|  |  |
| --- | --- |
| **MSP Regression Algorithm Variables** | |
| **productId** | The identifiers used to link a forecast to a specific product. |
| **year** | The year for the required forecast period – this will be useful as the algorithm continues to trains, learn and adapt its repository. |
| **month** | Month of the year for the required forecast period. |
| **units** | The maximum number of products that can be sold for a specific product. |
| **avg** | The average quantity sold for a specific product within a month. |
| **next** | The quantity sold for a product in the next forecast period – this is useful when training the data because you can compare the forecast against the real value of sales. |
| **count** | The actual quantity sold of the during the current forecast period. |
| **min** | Lowest number of recorded sales for that product on a specific day. |
| **max** | Highest number of recorded sales for that product on a specific day. |
| **prev** | Units sold for that product in the previous month. |

The class *“SampleProductData”* is a copy of the *“ProductData”* class and it was created to test the algorithm. The *“ProductUnitPrediction”* class stores the evaluation metric for the algorithm which is the regression score.

The class diagram also shows a dependency between the “Algorithim” (console application) and “MSP” (web application) projects to show the merge of the two projects. The console application was integrated into the web app via the project dependencies where the “Algorithim” project was added as a project reference.

Lastly, with regards to the front end, the only class the team coded with regards to the system is the product class which stores data about the records sold in this particular store. The other information relating to product maintenance (like the bulk upload functionality) have been coded in the ProductsController and the respective views.

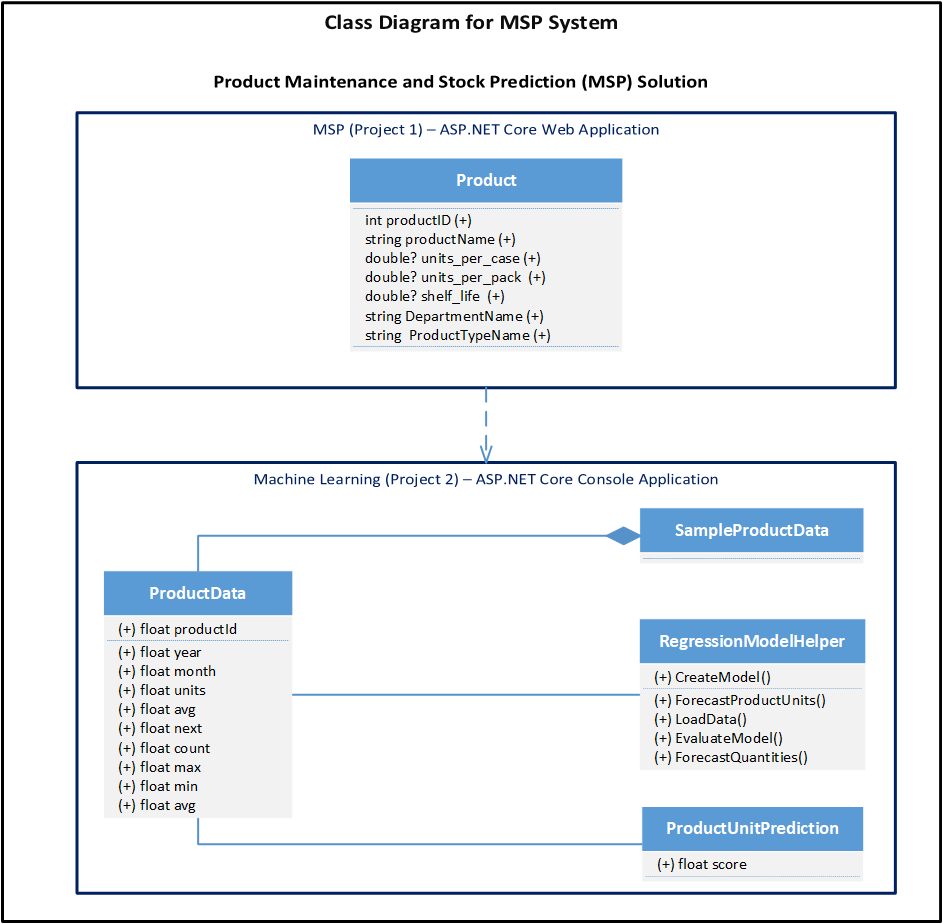


Figure 3: MSP Class Diagram

# 9. Database Schema

The system stores its data on a local SQL database called the MSP. This database stores and manages the “Products” table which stores the 2057 product records for this store. The database also stores and manages the tables relating to the users and role allocations for the system. These two tables were auto generated by Visual Studio. Lastly, the machine learning component of the system has been configured to store its data to a zipped folder named *“product\_month\_fastTreeTweedie”*. When the console app runs and trains the algorithm for the first time, the “*product\_month\_fastTreeTweedie”* zipped folder is created. Any further data fed into the algorithm will then be stored there.

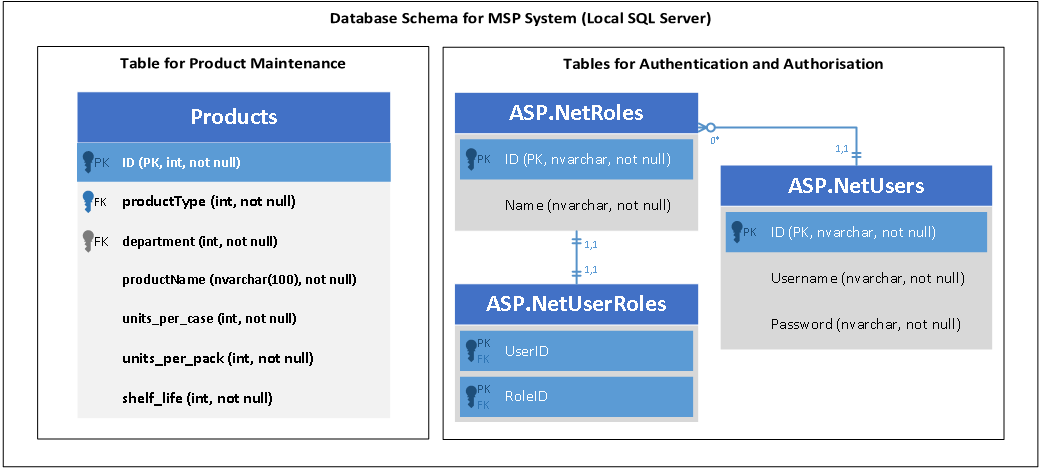


Figure 4: MSP Database Schema

# 10. State Machine Diagram

The main dynamic function on this system that goes through various states is the prediction engine in the back end which is required to display the forecasts for the viewer. Seeing that the physical management, planning and ordering of stock is out of scope, the following state machine diagram is the only one that is applicable to this system. Since the system is now complete, the algorithm no longer needs to go through the states of loading, training or evaluation. The prediction engine is now in use and will remain idle unless initiated as shown by the State-Machine Diagram below.

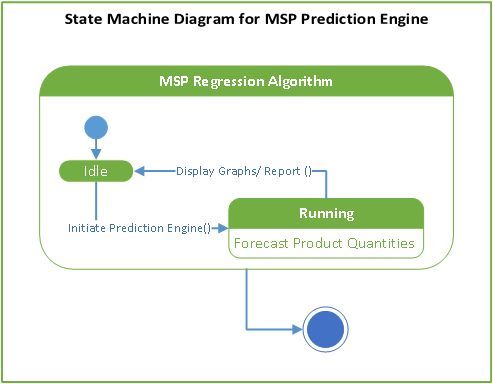


Figure 5: MSP Prediction Engine State Machine Diagram

# 11. Security

The MSP deals with sensitive information about a store’s transaction records and product information thus the system must ensure that the user’s information is kept safe. This will be achieved by adding the following functionalities:

Table 2: MSP Security Functionalities

|  |  |
| --- | --- |
| **MSP Security Functionalities** | |
| User Authentication: | User logins and profiles. Users will only have access to data relating to their store and will not have access to the information from another store. |
| User Authorisation: | Users Roles and access privileges. Only users with access rights will be able to perform certain functionalities on the system. For example, only the administrator will be able to create and assign user roles and the manager will have read and write privileges for product maintenance while an employee will only be able to view the data. This helps the store maintain control over its products because only the admin will have access to change the store’s data records. |

# 12. Revised System Architecture

For the design of our proposed system, we used a three-tier software architectural framework which includes components such as the presentation, business logic and data layers this is depicted in the figure below.

**Presentation Layer**

The highest level of the application is the user interface. The basic goal of the interface is to transform actions and outcomes to something that the user can comprehend. Typically, it operates on a desktop, PC, or workstation, employs a standard graphical user interface (GUI), and shows information linked to services. The presentation tier connects with other layers. To build the user interface of MSP we are going to employ web-based application languages such as HTML, CSS, jQuery, Ajax and JavaScript.

**Application Layer**

This layer is responsible for application coordination, command processing, logical decision-making, evaluation, and calculation. Additionally, it transmits and analyses data between the two adjacent layers. Visual Studio, Azure Devops for collaboration, and the Azure Machine Learning studio services will be used. The application's functionality will be developed using programming languages such as C#, ASP.NET Core, and ML.NET.

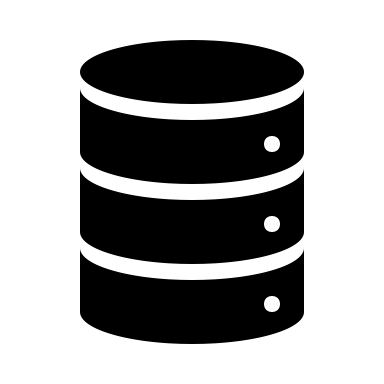
**Data Layer**

The data is stored and retrieved from a database or file system in this area. The data is subsequently returned to the logic tier for processing before being sent to the user. Our application's data will be stored and accessed via an Azure SQL Database and Azure storage.

Diagram

Description automatically generated

Data Layer:



Software: SQL server embedded in Visual studio

Type: local

Figure 6: Revised System Architecture

# 13. User Interface/ Wireframes

The following are low-fidelity wireframes for the MSP Web Application; prior to beginning the development process, the team determined that prototyping each page and how it would appear in an ideal state would be important. These wireframes are mostly concerned with product maintenance, the user interface, login/user registration pages, and reports/charts. Prototyping assisted the team by offering a sense of how the application will seem and by allowing team members who were not involved in the development process to visualize the application and its appearance. Additionally, it enabled the team to respond quickly to team feedback.

The home page for MSP is illustrated below. This is a stand-alone web page with the primary objective of informing the visitor about MSP. The home page will provide information about MSP's services, which include product maintenance and stock prediction.

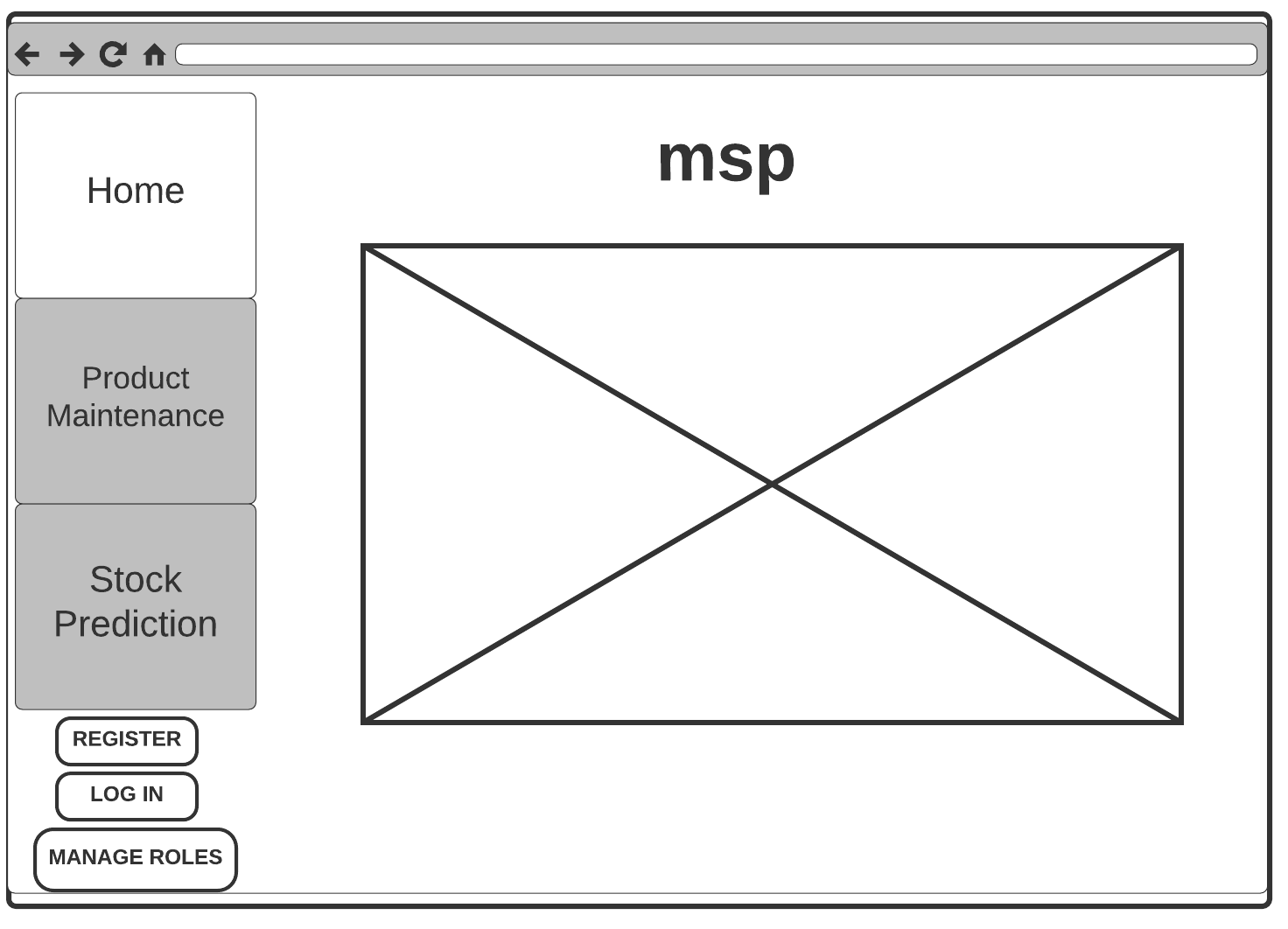


Figure 7: MSP Home Page

From the MSP's landing page, the visitor has the option of registering as a new user or logging in as an existing user. The following illustrates the application's user and login pages.

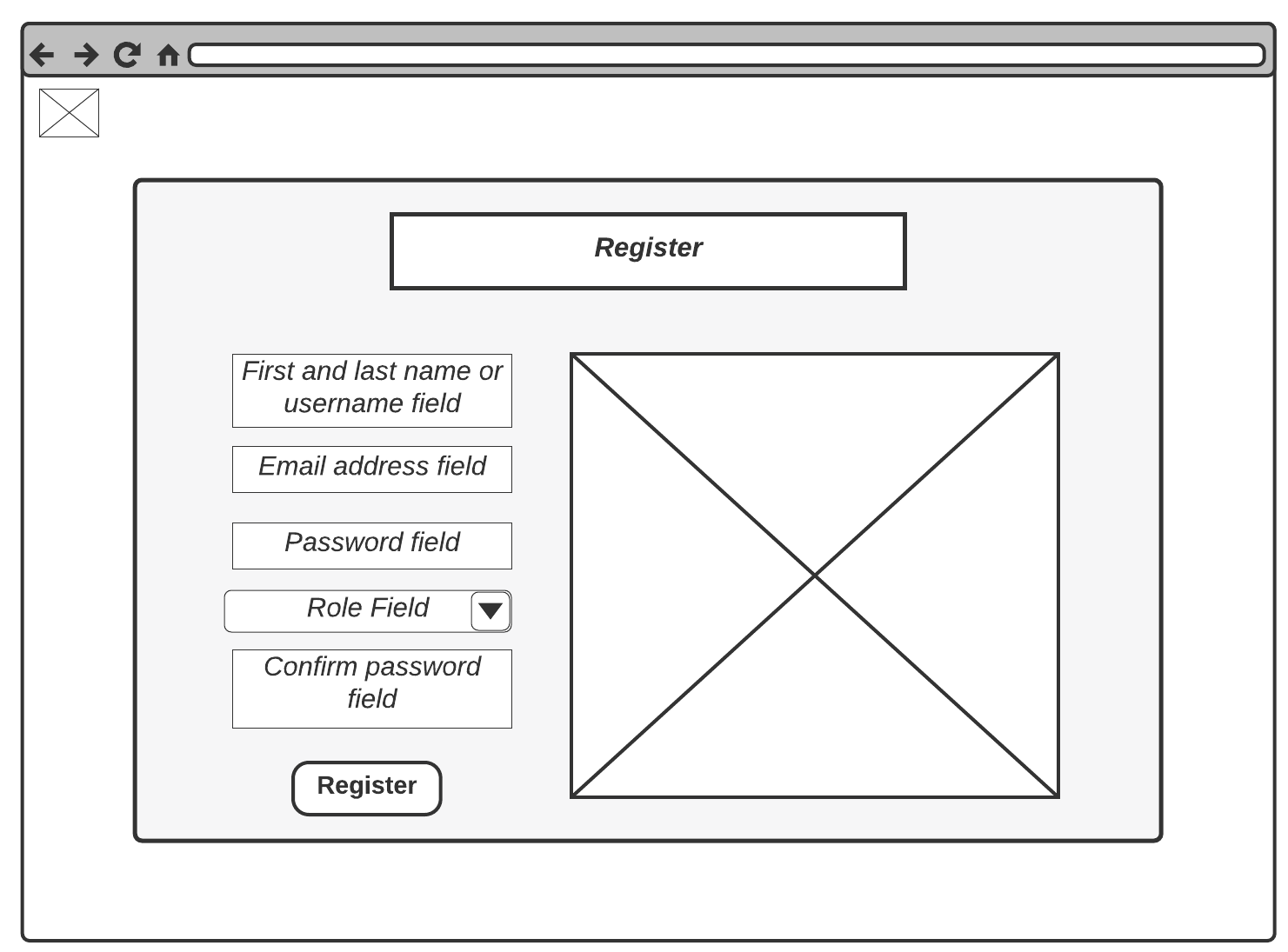


Figure 8: User Registration Form

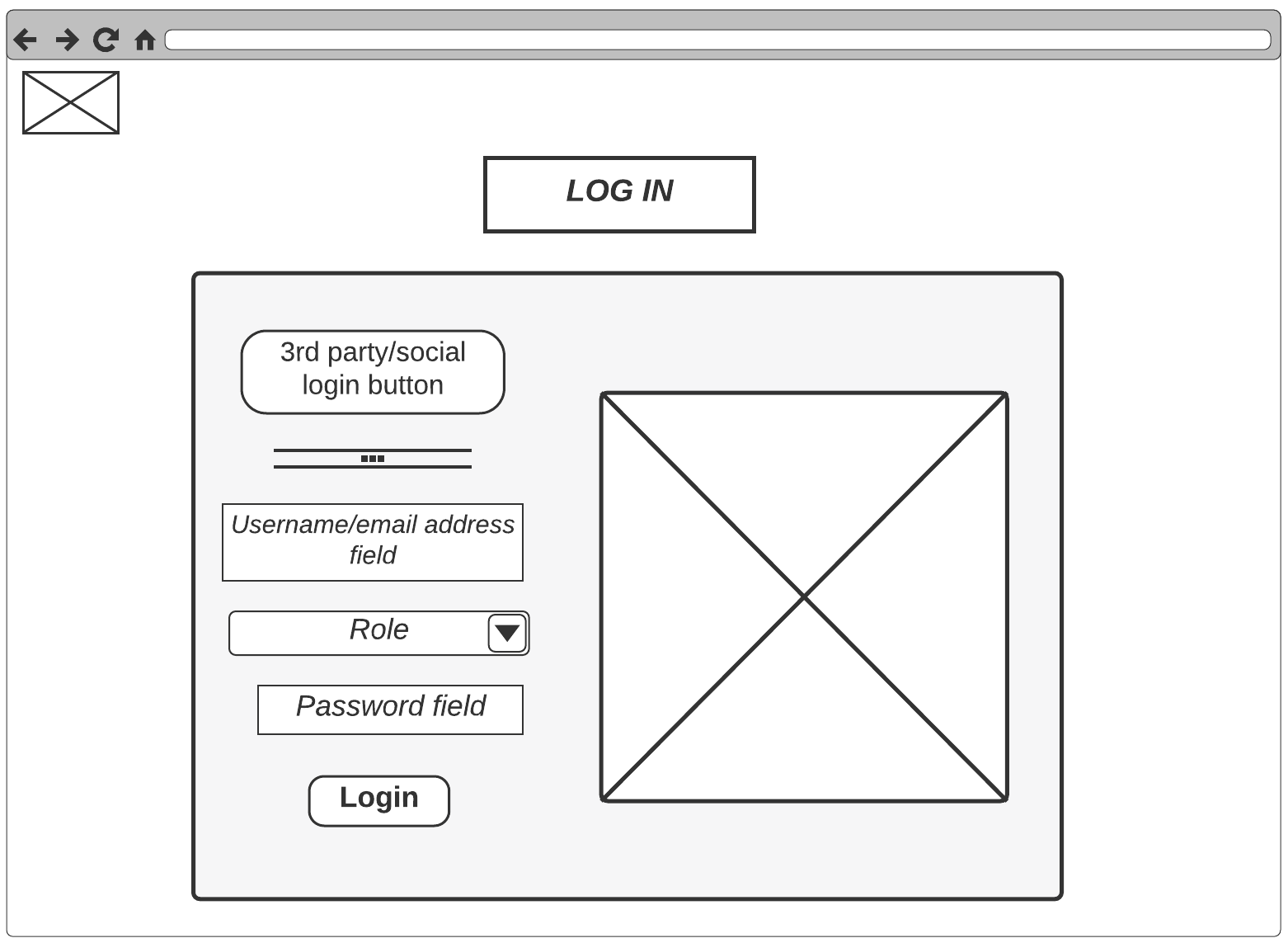


Figure 9: Login Form

The Products page is seen below. The user may perform CRUD actions on this page, such as updating and deleting the information of an existing product in the data table. Additionally, by selecting the "Add New Product" button, the user may create a new product. Furthermore, the user may upload a spreadsheet of their products by clicking the "Upload Data" button.

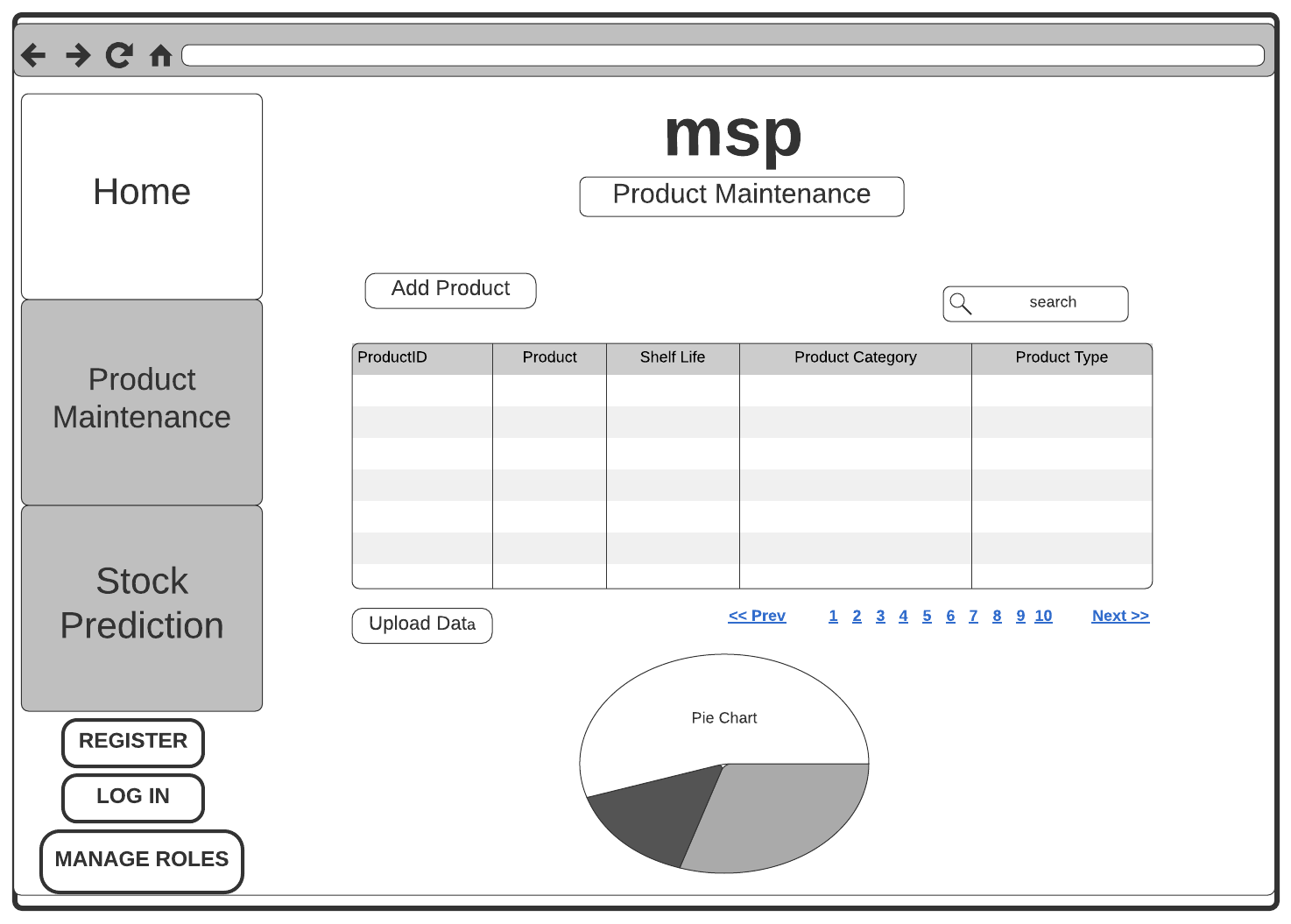


Figure 10: Product Maintenance Page

A form for adding a product to the database is depicted in the following figure. The user may add product details in this form, and upon submission, the product will show in the data table among the other products currently in the database. Additionally, the user can reset the product information they typed onto the form if they made an error and re-enter the product information.

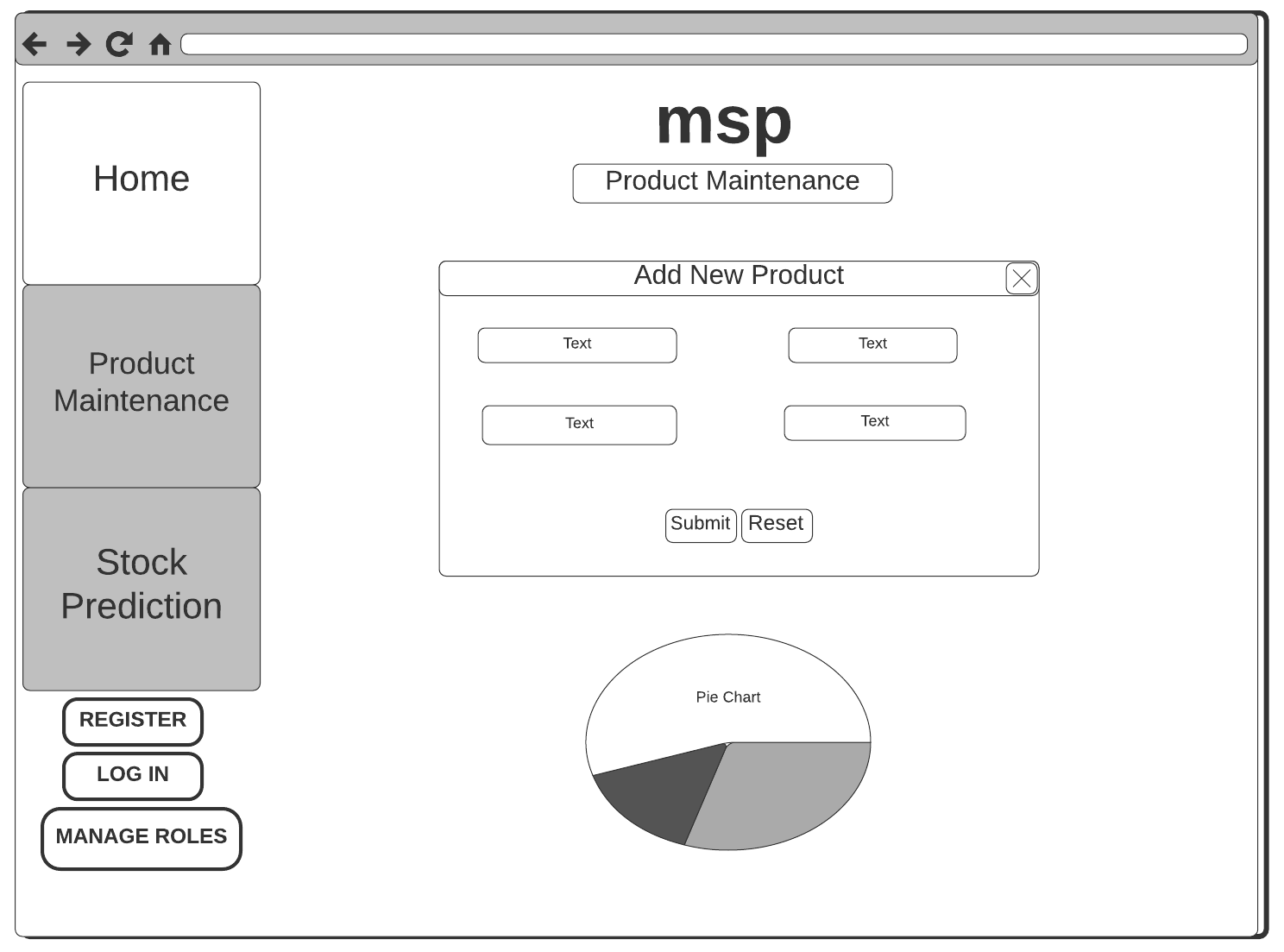


Figure 11: Add New Product Pop-up Form

The following illustration depicts a pop-up form that will be displayed when the user clicks the table's "Edit" button. By clicking on the "Edit button" next to a product in the table, the user can edit it details of an existing product in the data table.

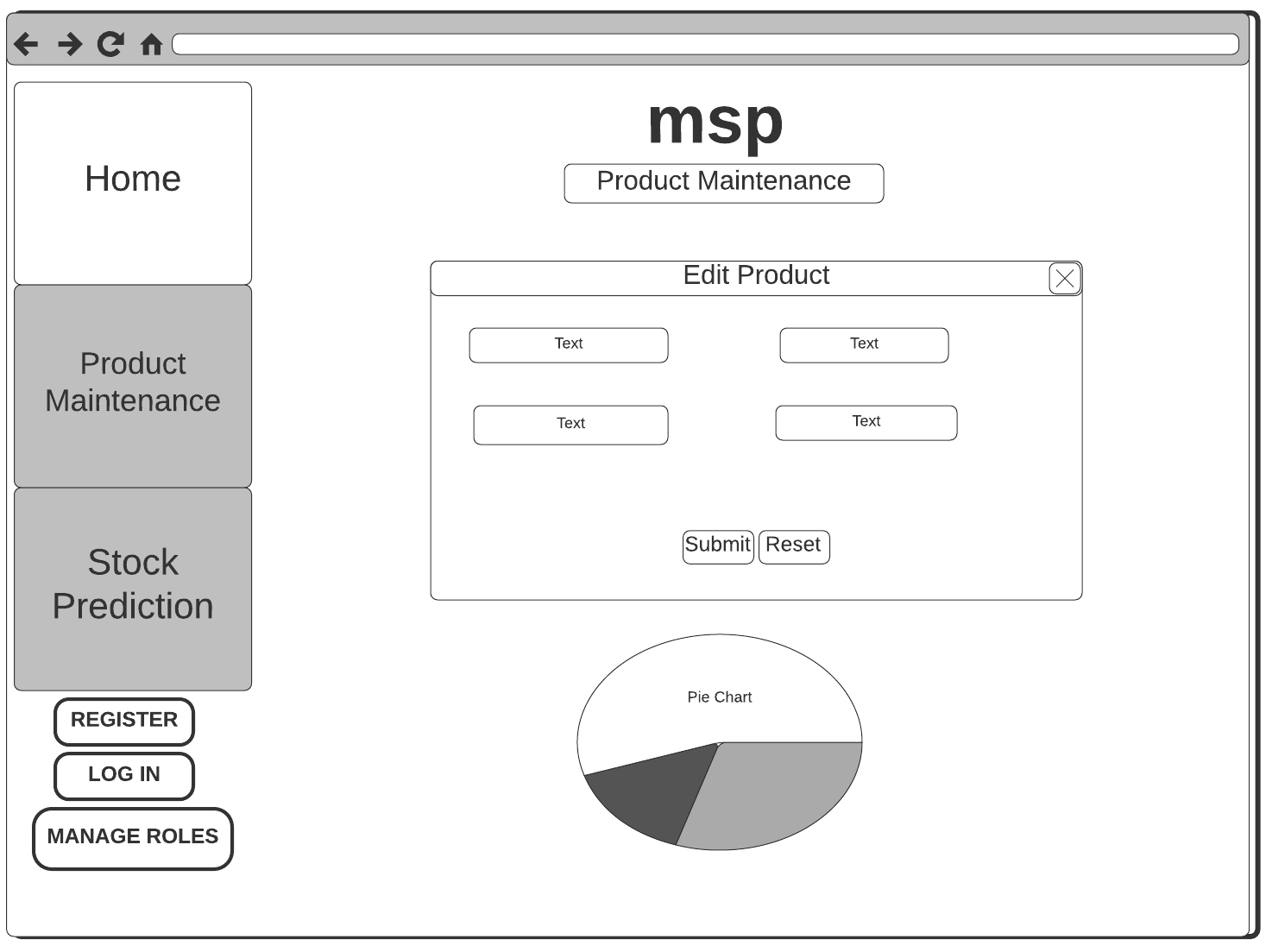


Figure 12: Edit Product Pop-up Form

The following figure illustrates the warning message displayed when a user attempts to delete an existing product entry from the data table. The user may edit, view, and remove existing products from the data table. When the user chooses to delete the product data from the data table, a form pops up prompting the user to confirm their decision. The user is then presented with the option of selecting "Yes" or "No."

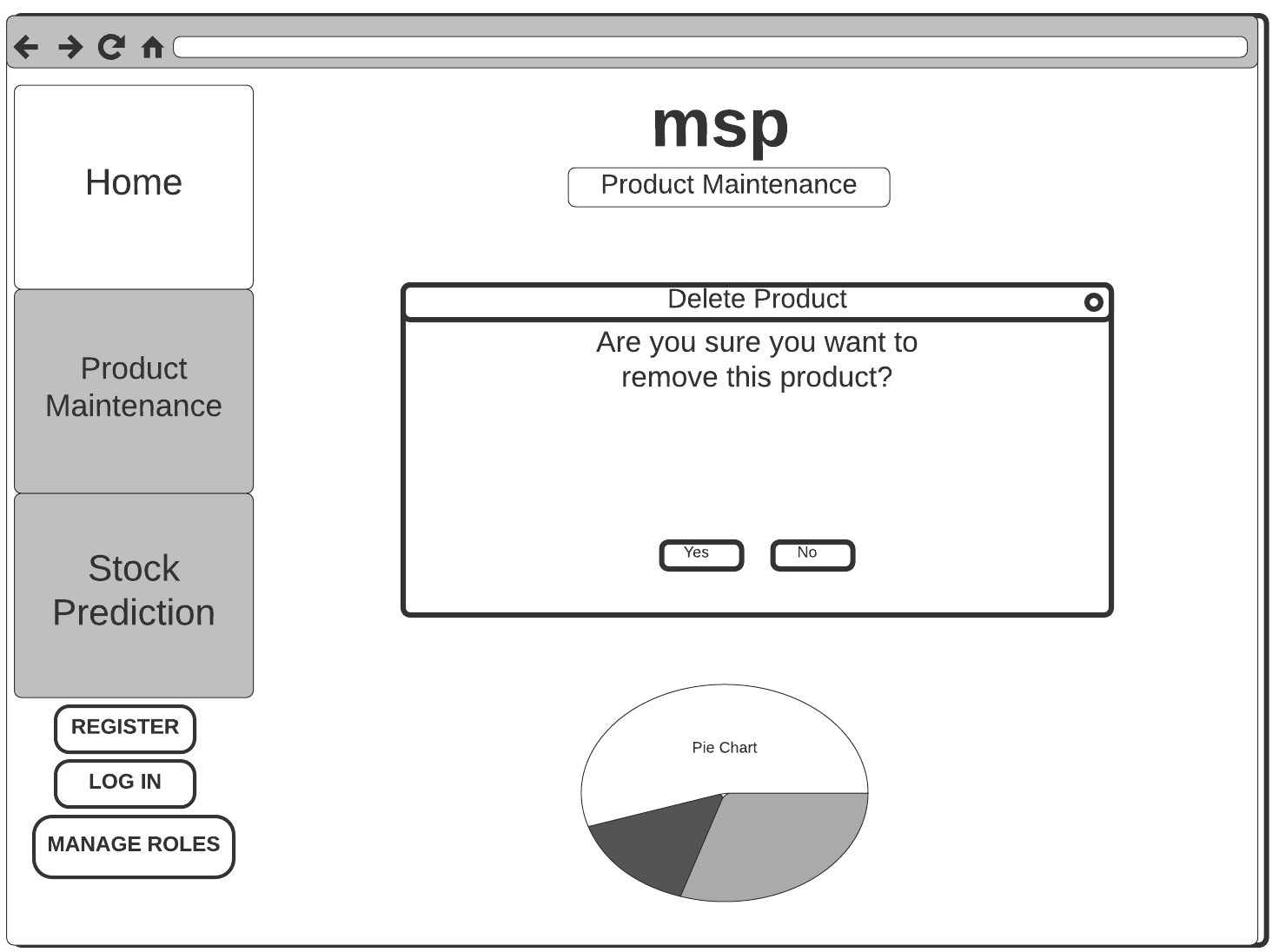


Figure 13: Delete Product Warning Message

The Stock Prediction page is seen below; this page will provide charts and statistics pertaining to stock forecasting, the dashboard for the stock prediction is also nested inside of this page and includes highest selling products, lowest selling products, sales over time etc.

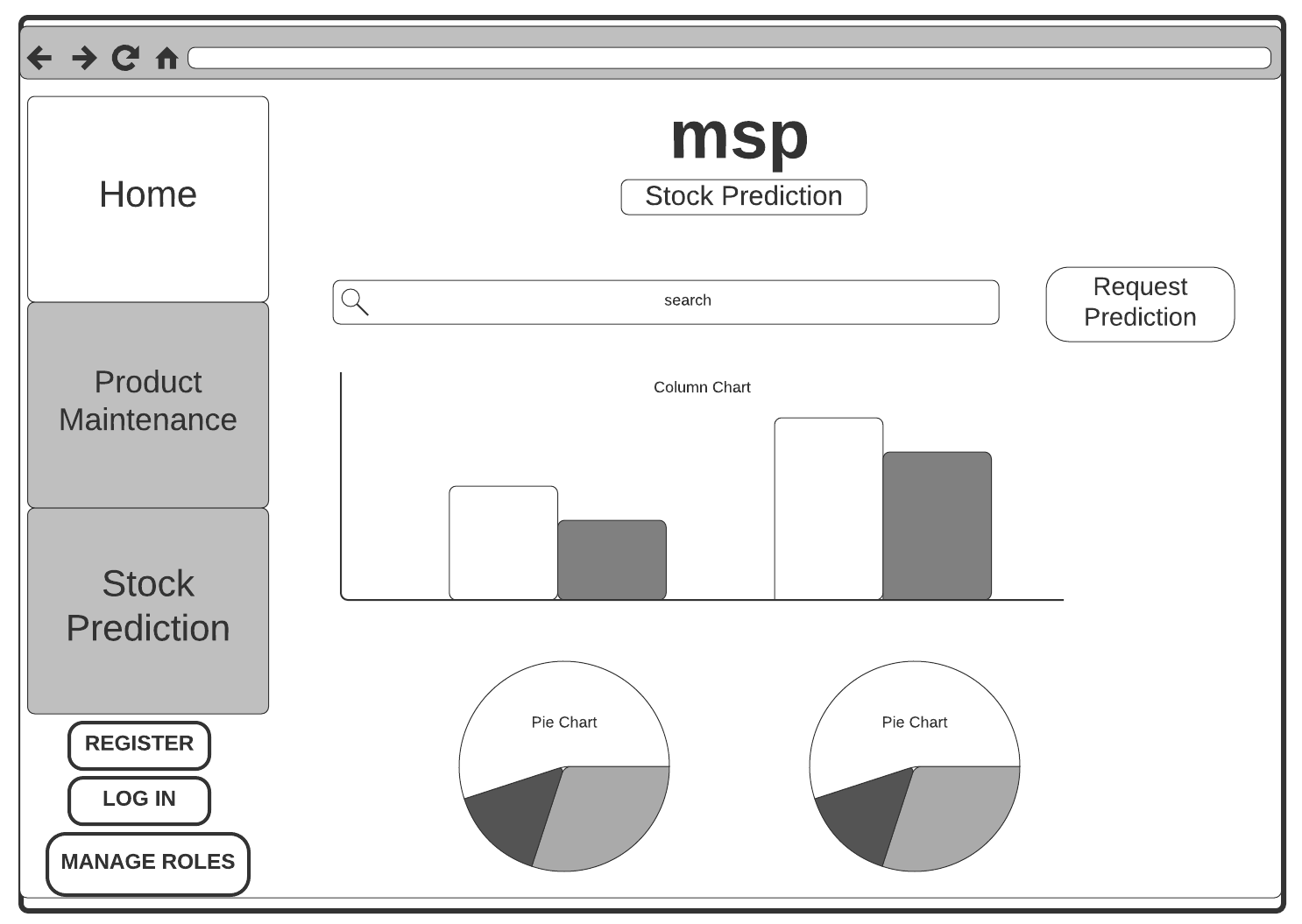


Figure 14: Reports Page

# 14. Risk Assessment

The following is a risk response plan that includes a list of all potential risks. The risks are ranked according to their likelihood of occurrence, starting with the most likely. This table summarizes the risks, the factors that might contribute to them, and how to reduce them. Additionally, it emphasizes the contingency plan in case that risk occurs, ensuring that the project is not negatively impacted if any of the following risk were to occur.

Contrary to popular perceptions of "risk," a project risk might have either a negative or a positive influence on the project's progress toward its objectives. Project risk management entails identifying possible issues that might arise over the course of the project and determining how they can jeopardize the project's success. The following is a risk response plan that takes into account both general project risks and those unique to the MSP application. The tables below summarize the risks that have been found, their possible fundamental causes, and ways to mitigate the risk. Additionally, they outline the contingency plan in case the risk occurs, ensuring that the project is not severely impacted.

**Risk Status Key:**

Table 3: Risk Status Key

|  |  |
| --- | --- |
| **Colour** | **Status Definition** |
| Red | There are significant issues with the project. The issue cannot be handled solely by the project manager or project team. |
| Amber | A problem has a negative effect on project performance but can be dealt with by the project manager or project delivery team. Action is taken to resolve the problem or a decision is made to monitor the situation. |
| Green | The project is performing to plan. All aspects of project viability are within tolerance. |

Table 4: Risk Assessment

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Description** | **Impact** | **Likelihood** | **Root Cause** | **Mitigation Strategy** | **Contingency Plan** | **Status** |
| **1** | Inadequate technical skills to successfully meet project requirements. | Significant | Possible | The team has never developed any applications that embody machine learning. | Leverage individual strong technical/ research skills.    Regularly consult the Microsoft ASP.Net Core Workshops to solidify knowledge on crucial functionalities. | Discuss possible solutions about technical difficulties with the project sponsor.    Seek additional resource help from project sponsor.    **Daily check-in on WhatsApp?** | Green |
| **2** | Team member unavailability due to physical/mental health challenges. | Significant | Likely | Individual proneness to illness/ poor individual health practices (e.g., not adhering to Covid-19 regulations).    Inadequate planning and workload management leading to work-overload and/or burnout. | Adequate workload management for each team member during sprint planning to avoid overworking.    Team members to individually adopt healthy practices such as adequate rest and taking personal time.    Adhere to recommended Covid-19 regulations. | Adjust sprint schedule and consider the availability of team members and how much work can realistically be completed.    Create a safe space and community culture in the team to support each other.    Communicate with course convenor to make any adjustments to deadlines if necessary. | Green |
| **3** | Sponsor (Nkosana) and lecturing team increases or propose changes to the scope of the project. | Minor | Possible | Sponsor or lecturing teams request additional features not previously agreed on. | The team signs a contract stipulating the requirements of the project and the scope. | Ensure that the main functionality of the system and the agreed upon requirements are met.    Brainstorm possible ways to incorporate additional features based on availability of time and resources. | Green |
| **4** | Conflict amongst team members impacting overall team productivity and quality of work. | Significant | Likely | Individual character clashes and working in isolation.    Poor communication and lack of harmony in understanding the project requirements and execution plans. | Stick to stipulated scrum values created at the undertaking of the project.    Openly communicate any personal/technical conflicts when they occur to maintain harmony. | Consult with course Lecturers to help us solve our differences and find a way forward and common ground. | Green |
| **5** | Scope Creep | Significant | Possible | Complexity of project overwhelming team member the more they uncover and understand project requirements. | Work consistently and manage workload to avoid overworking and underworking.    Ensure open communication lines with project sponsor and lecturing team to maintain a reasonable scope according to time and skills of the team. | Focus on the functional requirements then do the non-functional requirements after. | Green |
| **6** | Team members not completing work planned for each of the sprints. | Significant | Possible | Poor communication on sprint requirements and expectation.    Pressure and deadlines from other courses. | Conduct regular documentation and code review in the form of status updates. | Adjust sprint schedule and consider the availability of team members and how much work can realistically be completed.    Communicate with team on possible issues an individual may be facing and if the team can be of any help.    Consult with course Lecturers to help us solve our differences and find a way forward and common ground. | Green |
| **7** | Miscommunication between Nkosana and the team about specific requirements leading to team not doing what the sponsor requires. | Significant | Possible | Different communication styles and time constraints | Employ multiple communication strategies (WhatsApp, Email, and Microsoft Teams).    Update the sponsor on progress after every sprint and what the next components that will be worked on. | Adjust the plans for the next sprints and work accordingly. | Green |
| **8** | Nkosana states that the project should be terminated. | Significant | Unlikely | Nkosana can be influenced by other stakeholders at Cog3nt.    The CSR initiative to better small medium enterprises may be terminated. | Ensure an open communication line between all project stakeholders to identify any potential issue that may fatally affect the project. | Identify new project for waffle? | Green |
| **9** | Loss of application project file | Significant | Unlikely | Corrupt files and/ poor and unsecure storage. | Use GitHub as a version control tool.    Use university provided secure cloud storage for documents. | Revise project plan and possibly reduce scope. | Green |
| **10** | Inconsistencies in project documentation because of delegation | Significant | **Possible** | Sections being divided and delegated amongst team members. | Conduct scheduled documentation review. | Have a meeting to discuss where the discrepancies are and consult Nkosana and the lecturing team where agreements cannot be made. | Green |

Below we have additional risks that pose a threat to the completion of our application. We decided to divide the rest of the table for better viewing pleasure.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Risk ID** | **Description** | **Impact** | **Likelihood** | **Root Cause** | **Mitigation Strategy** | **Contingency Plan** | **Status** |
| **1** | Store owners not understanding how to use the application. | Significant | Unlikely | Technical challenges with the use of digital technology.    Inexperience with using an application of its kind. | Ensure there are help functions on the application.    Make the MSP as intuitive as possible for the user and test if others can understand it. | Provide an in-depth training manual.    Provide an auditory/ visual guide explaining how to use the functionalities of the MSP. | Green |
| **2** | MSP prediction algorithm could provide wrong predictions. | Significant | Possible | Not enough data training in machine learning code. | Ensure the use of enough training data that the machine can make more accurate predictions.  . | Continue training the application for longer. | Green |
| **3** | Dataset file could not be compatible with ML.NET. | Significant | Possible | Data is qualitative instead of quantitative. | Scrum Team should ensure that the dataset is in the appropriate format.    Provide supporting documentation for the application listing the filetype and dataset type the application is compatible with. | Request guidance from project sponsor, Nkosana.    Convert the dataset to  .csv file (quantitative data) | Green |
| **4** | Lack of adoption of the MSP | Significant | Possible | High data and Wi-Fi costs and Poor network infrastructure in South Africa.    Store owners not seeing the value in the application.    Reluctance to change due to familiarity with current manual method    Issues of privacy and safety of financial data | Request feedback after every iteration to ensure that we a building a solution that is indeed adding value. | Give them consultancy with Cog3nt who can explain the value gained from the MSP.    Possible offline capability? | Green |
| **5** | Store owners may fail to regularly update the MSP with new training data. | Significant | Possible | Technical difficulties with the use of technology.    Trouble adjusting to new operating conditions and forgetting to complete task. | MSP must provide regular notifications when an update of data is due. | Make the notifications pop up on the application. | Green |

# 15. Development Approach

The team still follows the Agile Project Management Framework, Scrum Methodology. And due to the ongoing Covid-19 pandemic, the need for Remote Learning and interaction is still necessary. The team still employs a host of digital collaboration technologies to aid us in usefully delivering this project. What follows is a brief overview of the platforms we use to engage with for our team collaboration and communication, namely, *WhatsApp, Miro, Microsoft Teams, and Microsoft Outlook.*

#### WhatsApp

Remote learning necessitates the need for virtual interaction. This will require the use of digital communication and collaboration tools. The team still uses *WhatsApp* as a communication medium to keep all team members updated on what we are doing and any quick updates as they arise. WhatsApp is also useful for conducting daily scrum meetings as well as keeping reminders about team meetings and deadlines. Clarification on misunderstandings and team members approving each other’s work can be communicated on this platform which also ensures that all team members are kept in the loop without necessarily having to meet them at a specific time, as they can read the message whenever they can.

#### Microsoft Teams & Miro

Remote learning also has the consequence of not physically meeting the collaborators you are working with, therefore the ability to host meetings on *Microsoft Teams* is paramount because of its video and audio call functionalities. It also enables us to simultaneously collaborate on documentation and safely store it within the platform. Microsoft Teams also integrates third party applications into its platform, allowing us to integrate our collaboration tool Miro for ease of access to features such as our task board for team member’s perusal. We will also be hosting our meetings with the sponsor and lectures through Microsoft Team.

#### Microsoft Outlook

Microsoft Outlook is also a critical technology we are using in this remote environment as it allows us to integrate email communication, organize meeting appointments and automate invitations or reminders for Microsoft Teams meetings, which for us is the most useful feature.

### Product Backlog

The table below elaborates on the user stories that were used for this iteration. These user stories are linked to the tasks outline for this iteration in the Project and Iteration Plan section.

Table 21: Product Backlog

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Story ID | User Stories | Reasoning | Priority | Sprint |
| S001 | As a store owner, I want to be able to upload my product list (spreadsheet) onto the system in bulk. | To avoid manual entries consisting of thousands of products. | High | 1 |
| S002 | As a store owner, I want to be able to create new products. | If the owner has new products that need to be maintained and predicted. | High | 1 |
| S003 | As a store owner I want to be able to be able to edit or remove existing products | Existing product details may change or no longer be required. | High | 1 |
| S004 | As a user I want to be able to search for products on the system | A user should be able search for the desired product instead of scrolling through the whole product list. | Medium | 2 |
| S005 | As a store owner I want to be able to page through a list of products | This is to avoid having one long page that scrolls continuously. | Medium | 2 |
| S006 | As a store owner I want a forecast of the number of products I need to stock for the next 7 days. | The user should know the optimal levels of inventory to keep on hand/ stock up on. | High | 2 |

## 15.1 Project and Iteration Plan

For this iteration, the average daily hours burned per user story increased significantly. This is a depiction of the consistency of work put in by the development team since the beginning of the iteration, owing to the project was running behind schedule and the team allocating more of their time and effort to make considerable coding progress for this iteration.

The next subsections will focus on each sprint. Each section details the tasks completed in the sprint backlog, the velocity, and burndown charts.

### Sprint 1: 03/10/2021 – 10/10/2021

**Tasks in Sprint Backlog that were completed**

|  |  |
| --- | --- |
| Tasks | Hours |
| Create Login pages | 3 |
| Create Registration Page | 2 |
| File Upload Validations | 5 |
| Implementing ML graph (responsive area + line graph for stock prediction dashboard) | 6 |
| Total | 16 |

**Burndown Chart:**

### Sprint 2: 10/10/2021 – 17/10/2021

**Tasks in Sprint Backlog that were completed**

|  |  |
| --- | --- |
| Tasks | Hours |
| Create User roles | 3 |
| Input validations for Login in and Registration Page | 4 |
| Upload template for product Maintenance | 4 |
| Input Validations for “Add new product Form” | 1 |
| Redesign Application | 4 |
| Fix Nav bar (minimizing buttons and spacing) | 1 |
| Push ML graphs onto Project Solution | 4 |
| Total | 21 |

**Burndown Chart:**

### Sprint 3: 17/10/2021 – 24/10/2021

**Tasks in Sprint Backlog that were completed**

|  |  |
| --- | --- |
| Tasks | Hours |
| Assign User roles and views | 9 |
| Forgot password | 0 |
| “Remember Me” Function | 1 |
| Re-design Homepage | 5 |
| Research on Tableau connecting to VS database for front end | 2 |
| Complete ML Dashboard | 12 |
| Product maintenance pie chart | 4 |
| Total | 33 |

This sprint had the most hours worked as a result of the final iteration being due on the 25th of October and there was quite a bit of pressure.

**Burndown Chart:**

# 

# 15.2 Velocity:

= Total story points completed / number of sprints

= 70/ 3

= 23,33 hours

This would equate to the average of 23,2 hours worked per sprint.

# 16. Test Plan

## Testing Strategies

The following testing strategies will be adopted by the team throughout the development lifecycle:

**Integration Test:** Combining the console application with the web application. The system should run the console application and the web application simultaneously – this signifies a successful merge. However, to identify a successful integration, only the web application should run at run time unless the console application was selected as the start-up project. In a successful integration, the console app with the prediction engine should be idle unless called upon on the stock prediction page (as explained in the [State-Machine Diagram).](#_10._State_Machine)

**System Test:** The system code was tested using breakpoints to ensure that variables that were assigned data were being populated properly when certain tasks were performed. Another example involves checking to see if the system will request the user to login/register before they can access any of the system’s features.

**Functional Acceptance Test:** System validations to ensure that the system will catch user exceptions instead of crashing. Examples of this include displaying an error message when a user requests a prediction for a product/ productID that does not exist. Another example of this includes displaying a notification when a user has added/updated or deleted a product successfully on the products page.

## Test Cases – MSP System

Table 9: Test Cases

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | TC. V101 | | | | |
| Description: | System Authentication – Registering and Logging in | | | |
| Pre-Conditions: | 1. MSP Application should be loaded.  2. Login/ Register form should be loaded | | | |
| Tester: | Matete Kopano Lebakeng | | Date:23/10/2021 | |
| Test Scenario | | **Test Data** | **Expected Results** | **Pass/Fail** |
| 1. Run the MSP Application.  2. The system should request for users to login before they can see the homepage.  3. Only once a user has registered or logged in successfully should they be able to see the MSP homepage.  4. When users have logged out they should not have access to any MSP pages. | | **Email address:** [**kim@gmail.com**](mailto:kim@gmail.com)  **Password: @Kimberley123** | - Only once Kim has registered and logged in successfully should she be able to see the MSP homepage.  - When Kim has logged out, she should not have access to any MSP pages. | **Pass** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | TC. V102 | | | | |
| Description: | System Authentication – Assign user Kimberley to role of manager. | | | |
| Pre-Conditions: | 1. User Kimberley must be logged in to the MSP  2. User Kim should not be assigned to any pre-existing roles | | | |
| Tester: | Matete Kopano Lebakeng | | Date:23/10/2021 | |
| Test Scenario | | **Test Data** | **Expected Results** | **Pass/Fail** |
| 1. User Kimberley will be assigned to the role of manager so she can access and view the dashboard on the Stock Prediction Page. | | **userId: kim@gmail.com** | - User Kimberley should be assigned the role of manager.  - User Kimberley should be able to view the “Product Analytics” part of the Stock Prediction page.  - User Kimberley should be not be allowed to access the “Administrator” page where roles are assigned. | **Pass** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | TC. V103 | | | | |
| Description: | Bulk Upload – Excel Spreadsheet load into database | | | |
| Pre-Conditions: | 1. MSP Products Page Interface should be loaded  2. User is registered and is logged in | | | |
| Tester: | Khanya Ngxabani | | Date:29/08/2021 | |
| Test Scenario | | **Test Data** | **Expected Results** | **Pass/Fail** |
| 1. Run the MSP web application and access the Product Maintenance page vis the nav bar.  2. Users should download the template and populate the allocated columns with the required data for their store.  3. Click “Upload” to perform bulk upload and pass entries into system database | | **Products.xlsx** | The system should read the entries in the excel file and should populate the table on the “Product Maintenace” page with the relevant data. | **Pass** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | TC. V104 | | | | |
| Description: | CRUD Functionalities – Product Maintenance | | | |
| Pre-Conditions: | 1. MSP Products Page Interface should be loaded  2. User should be registered and logged in | | | |
| Tester: | Khanya Ngxabani | | Date:29/08/2021 | |
| Test Scenario | | **Test Data** | **Expected Results** | **Pass/Fail** |
| 1. Run the Time Series Console Application | | **mspdataset.xlsx** | The system should take the user’s input from the pop up form and add/edit the entry as intended by the user to the database. If a user wants to delete and item, the selected item should no longer be visible on the database. | **Pass** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | TC. V105 | | | | |
| Description: | Test Regression Algorithm | | | |
| Pre-Conditions: | 1. MSP Products Page Interface should be loaded  2. ML Console Application without Errors | | | |
| Tester: | Kimberley Mugadza | | Date:29/08/2021 | |
| Test Scenario | | **Test Data** | **Expected Results** | **Pass/Fail** |
| 1. Run the Regression Console Application | | **mspdataset.xlsx** | The program should run the console app and display a forecast for the selected product. Accuracy rate should be at least 95% (regression score above 0.45 out of 0.5). | **Pass** |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | TC. V106 | | | | |
| Description: | Test Prediction Accuracy | | | |
| Pre-Conditions: | 1. MSP Products Page Interface should be loaded  2. ML Console Application without Errors | | | |
| Tester: | Kimberley Mugadza | | Date:25/09/2021 | |
| Test Scenario | | **Test Data** | **Expected Results** | **Pass/Fail** |
| 1. The Forecasted Value close to the “Real Value” column should be close by +/- 5 units. | | **prosductsstats.xlsx** | The program should run the console app and display a forecast for the selected product. Accuracy rate should be at least 95% and the system should save the model to an auto-generated zipped folder called “*product\_month\_fastTreeTweedie”.* | **Pass** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test ID | TC. V107 | | | | | |
| Description: | Display Forecast using Chart.js | | | | |
| Pre-Conditions: | 1. MSP Products Page Interface should be loaded  2. Access to SQL Management Studio | | | | |
| Tester: | Kimberley Mugadza | | Date: 24/10/2021 | | |
| Test Scenario | | **Test Data** | | **Expected Results** | **Pass/Fail** |
| 1. Select/ search for forecast for “Soft Eating Gum 125g”. | | **Forecast for ProductId 44** | | The interface should display the forecast for the selected product for the next month on a graph. | **Pass** |

# 18. Conclusion

In conclusion, this project can be described as an initial development with regards to the applications that AI and machine learning can be used in business. This project can be combined with the other product clustering project from Cog3nt to make the MSP system more robust. If the two projects are combines, then users can understand any correlations between their best selling products while planning for their stock, adding more business value. Similarly, the system can also be configured for future integration with barcode system so that the system can also alert users when they are about to run out of stock (predict the date they are likely to run out of stock) which will add more business value. There are several ways in which the MSP can be expanded to generate more business value and the team has enjoyed being a part of this project.

# 19. References

Webb, J. (2018). *How Many Suppliers Should Businesses Have? How Many Should They Have?* Forbes. <https://www.forbes.com/sites/jwebb/2018/02/28/how-many-suppliers-do-businesses-have-how-many-should-they-have/?sh=1afc62749bb7>

Chan, S. W., Tasmin, R., Aziati, A. N., Rasi, R. Z., Ismail, F. B., & Yaw, L. P. (2017, August). *Factors influencing the effectiveness of inventory management in manufacturing SMEs.* In *IOP Conference Series: Materials Science and Engineering* (Vol. 226, No. 1, p. 012024). IOP Publishing.

Cooper, I. (2015). *Why CRUD might be what they want but may not be what they need*. <http://codebetter.com/iancooper/2011/07/15/why-crud-might-be-what-they-want-but-may-not-be-what-they-need/>

Hayes, A. (2021). *What Is a Time Series?* <https://www.investopedia.com/terms/t/timeseries.asp>. Investopedia.com

Herlings, F. (2020). *What is CRUD and how does it help me with my application?*. <https://florianherlings.de/posts/2020-04-28-crud-create-read-update-delete/>

Mary Jane. (2017). *Difficulties in Using a Manual Inventory System*. <https://bizfluent.com/info-8007357-difficulties-using-manual-inventory-system.html>

Quickbooks Commerce. (2018). T*he dangers of Using an Inventory Spreadsheet Template for your business*. <https://www.tradegecko.com/blog/inventory-management/dangers-of-using-inventory-spreadsheet-template>