Sprint 4 - User story document

Document History

Author	Version	Date	Description	Comments
Yogesharvind Nedumaran	0.1	04-Mar-2024	Functionality and Technical document update	
Giridharan Sankaran	0.3	10-Mar-2024	Test results document to be updated	
Ashwin Krishna Mani	1.0	13-Mar-2024	Reviewed and baselined	

Background

In the 'Frontend development 2 & Cucumber implementation' sprint, our team is dedicated to advancing the frontend development and incorporating Cucumber for automating tests for the backend code. The primary objective is to achieve significant progress in frontend design, aiming for around 75 percent completion. Additionally, we are implementing Cucumber to automate tests for the backend code, focusing on the Product, Expiry, and Discount tables. Furthermore, a new functionality is being introduced wherein a Python script embedded in the code will automatically send emails to suppliers to restock based on the manager's selection. Another important aspect being addressed is the inclusion of date synchronization into the code to ensure real-time stock inventory updates, enhancing the overall efficiency and reliability of our application.

Proposed Functional Solution

In the 'Frontend development 2 & Cucumber implementation' sprint, we aim to make substantial progress in frontend development, targeting around 75 percent completion. We will focus on refining the user interfaces for key components such as the Supplier, Inventory, Expiry, and Discount pages, ensuring a seamless and intuitive user experience. Concurrently, we will introduce Cucumber for automating tests for the backend code, specifically targeting the Product, Expiry, and Discount tables. This automation will streamline the testing process and improve code reliability. Additionally, a new feature will be implemented to automatically send restocking emails to suppliers based on the manager's selection, enhancing inventory management efficiency. Finally, date synchronization will be incorporated into the code to ensure real-time updates to the stock inventory, providing accurate and up-to-date information to users.

Design Analysis and Solution:

3.1 Technical changes:

3.1.1 Design Analysis:

In this sprint, our focus is on advancing frontend development and incorporating Cucumber for backend code testing. We will refine the user interfaces for various pages and ensure they meet user requirements. Additionally, we will design and implement Cucumber scripts to automate tests for the backend code, specifically targeting the Product, Expiry, and Discount tables. The introduction of a Python script embedded in the code will enable automatic email notifications to suppliers for restocking, based on the manager's selection. Furthermore, date synchronization will be included in the code to ensure real-time updates to the stock inventory, enhancing the application's reliability and usability.

3.1.2 Frontend Development:

Under the 'Frontend Development' phase, we will focus on refining the user interfaces for key components such as the Supplier, Inventory, Expiry, and Discount pages. These interfaces will be designed to provide a seamless and intuitive user experience, with features including record display, filtering, and editing functionalities. Our design analysis will ensure that each component adheres to React Native best practices, optimizing performance and usability. Through careful consideration of user feedback and iterative design improvements, we aim to achieve around 75 percent completion of frontend development by the end of this sprint.

3.1.3 Cucumber Implementation:

Incorporating Cucumber for backend code testing involves the creation of feature files and step definitions to automate tests for the Product, Expiry, and Discount tables. These tests will cover various scenarios to ensure the correctness and reliability of the backend functionalities. By automating these tests, we aim to streamline the testing process and improve overall code quality. Our design analysis will focus on identifying key test scenarios and crafting efficient and maintainable Cucumber scripts. Through rigorous testing and validation, we aim to enhance the robustness and reliability of our backend code.

3.1.4 Automatic Restocking Email Functionality:

The introduction of automatic restocking email functionality involves embedding a Python script in the code to send email notifications to suppliers based on the manager's selection. This functionality will streamline the restocking process, improving inventory management efficiency. Our design analysis will focus on integrating the Python script seamlessly into the codebase and implementing robust error handling mechanisms to ensure reliable email delivery. By automating the restocking process, we aim to reduce manual effort and improve the overall efficiency of our inventory management system.

3.1.5 Date Synchronization:

Date synchronization will be included in the code to ensure real-time updates to the stock inventory. This functionality will enable the application to maintain accurate and up-to-date inventory information, enhancing the overall reliability and usability of the system. Our design analysis will focus on implementing efficient data synchronization mechanisms to minimize latency and ensure consistency across the application. By synchronizing dates across different components of the system, we aim to provide users with reliable and timely information about stock availability.

Use Test Cases:

To be updated based on Cucumber script execution for backend code.