Vanier College

Faculty of Science & Technology

System Development

420-436-VA

Deliverable 03

March 9th, 2023

Purple Team

Nicole Bautista

Mubeen Khan

Deven Patel

Client: Anas & Moustafa

Table of Contents

Statements

Code from our eCommerce class will be used for certain features of this project.

We, the Purple Team, ensure that this document is our own work.

I Deven Patel student ID# #2171750, confirm that I have contributed to this deliverable. D.P

I, Nicole Bautista student ID# 2037770, confirm that I have contributed to this deliverable. N.B

I, Mubeen Khan student ID# 1962558, confirm that I have contributed to this deliverable M.K.

Executive Overview

This deliverable contains documentation related to our client, their problems, and a narrative description of their current inventory management system. The rest of the document contains UML diagrams and use cases.

Our clients, Anas and Moustafa, are the proud owners of Sweemory, a relatively new ice cream shop located in Ahuntsic-Cartierville. Anas brings with him experience from working in a similar field, in his home country, Syria. Moustafa, on the other hand, is a software developer currently studying Internet of Things. You can find their delicious ice cream on UberEats and DoorDash, as well as at their physical location.

As discussed to the narrative description of their current information system in this document, they manually check their products for expiration or low stock. We have determined that their primary business challenge is their inability to track their products efficiently. Their current methods are described as tedious, time-consuming, and requiring significant employee resources. Essentially, their main issue is manual inventory checking.

Our web application seeks to automate the issue mentioned earlier. It offers features such as real-time updates of stock numbers and notifications when products are approaching their expiration date. With all product information easily accessible in just a few clicks, our application is expected to significantly enhance their current system.

The first appendix displays our system, including each component and its interactions. The second appendix features our project's UML diagrams. The third one contains the completed template for use cases, which are based on the first appendix. Finally, the last appendix shows a class diagram that represents the information system.

Summary of the Client

Anas and his twin brother Moustafa have been running their ice cream shop in Ahuntsic-Cartierville, since August of last year. Anas, who previously worked in a bakery shop in Syria, decided to start this business in order to re-experience the feeling of working in a similar environment. Meanwhile, Moustafa, a software developer currently studying Internet of Things, will manage the web application once it is completed. The competition in the area is a Dairy Queen located around the corner.

Despite the competition, the ice cream shop has an impressive 4.9 rating on Google and is on track to becoming a well-known brand. However, there are still some internal issues that they can improve upon, which will be discussed in the next section. Additionally, the business has expanded its reach by partnering with various food delivery platforms like UberEats and DoorDash.

Business Problem

After completing the Flowchart, Class and UML diagrams, and Use Cases, we now have a better understanding on how to approach to our client's business problems. These problems will be discussed in the following.

The first and main problem we can tackle is the lack of efficiency in tracking products. Our client struggles to keep track of product expiration dates. Manually checking each product is time-consuming and tedious, requiring significant employee effort. However, it's important to avoid financial losses and negative impact on reputation from unsellable expired products. Implementing a system, such as software that automatically tracks dates and alerts employees when a product is nearing expiry, would simplify the monitoring process. This would save time, effort, and improve monitoring accuracy and efficiency, ultimately benefiting both the company and its customers.

Additionally, the website will help store employees keep track of products in stock. The website updates in real-time, showing employees the amount of stock available in the store. This will reduce unnecessary stock purchases and prevent overstocking unpopular products. As a result, the store can allocate resources more efficiently. Also, the website will provide a centralized platform for employees to access important product information, including prices, descriptions, and locations within the store. This will make the workforce more knowledgeable and efficient, improving the store's operations and profitability.

On top of that, our client is committed to sustainability, and one of their main goals is to reduce food waste. They believe that a key part of achieving this goal is to help their customers make use of or get rid of their products without waste. To this end, our client is exploring a variety of strategies, including adding promotions when a product is reaching its expiration date. These promotions could take a form of discounts. The website will be able to have display items that may be on the verge of expiring which can include a discounted price if given.

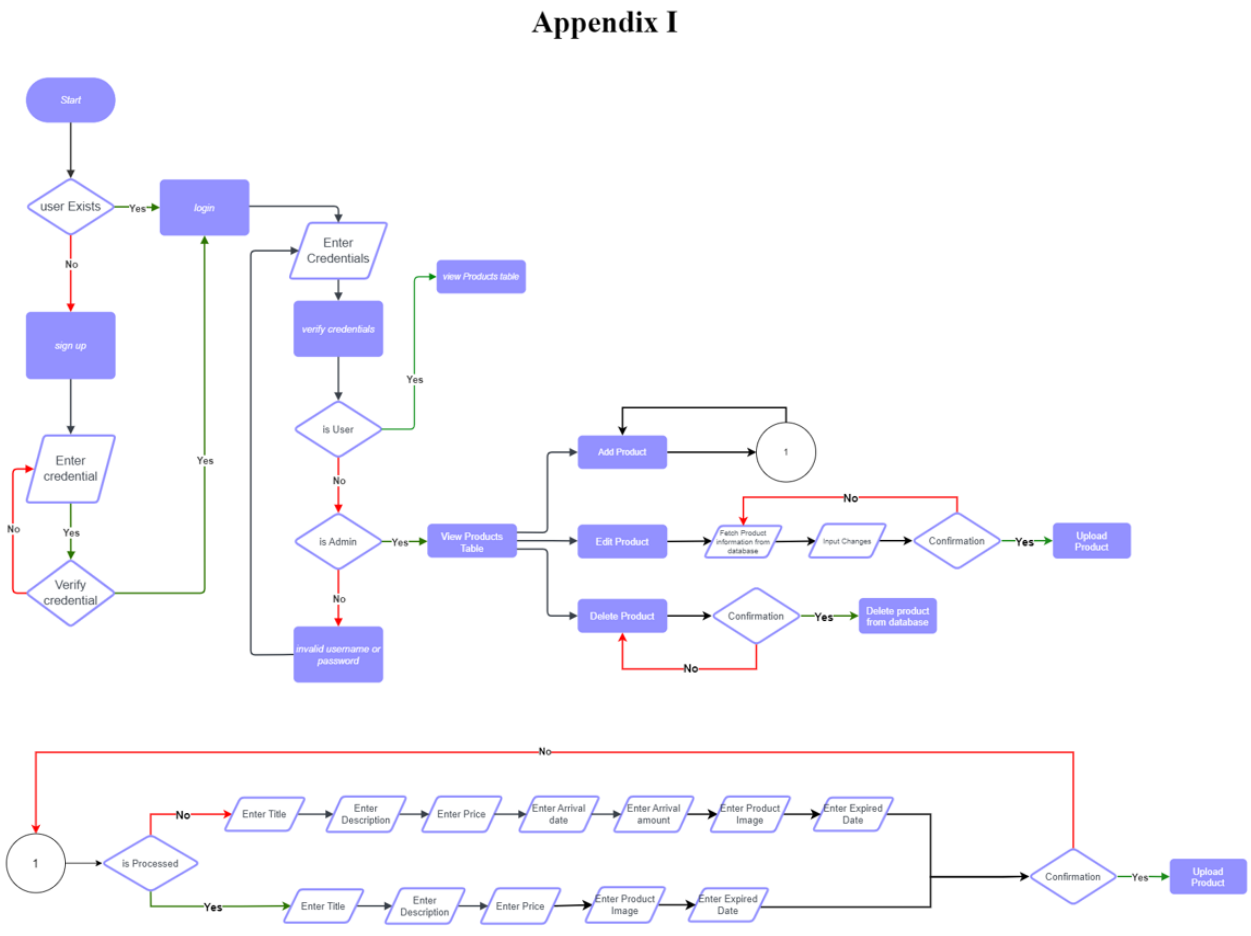
Narrative Description of Present Information System

The inventory is managed when the owners or employees notice that a product is low in stock and take note of it. One of the owners will then go to the store to procure more of the required products that are in their list. The importance of the item will determine when it will be replenished. Products that are used for making ice cream will be noticed quicker and replaced faster, compared to toppings that are unlikely to be requested by the customers.

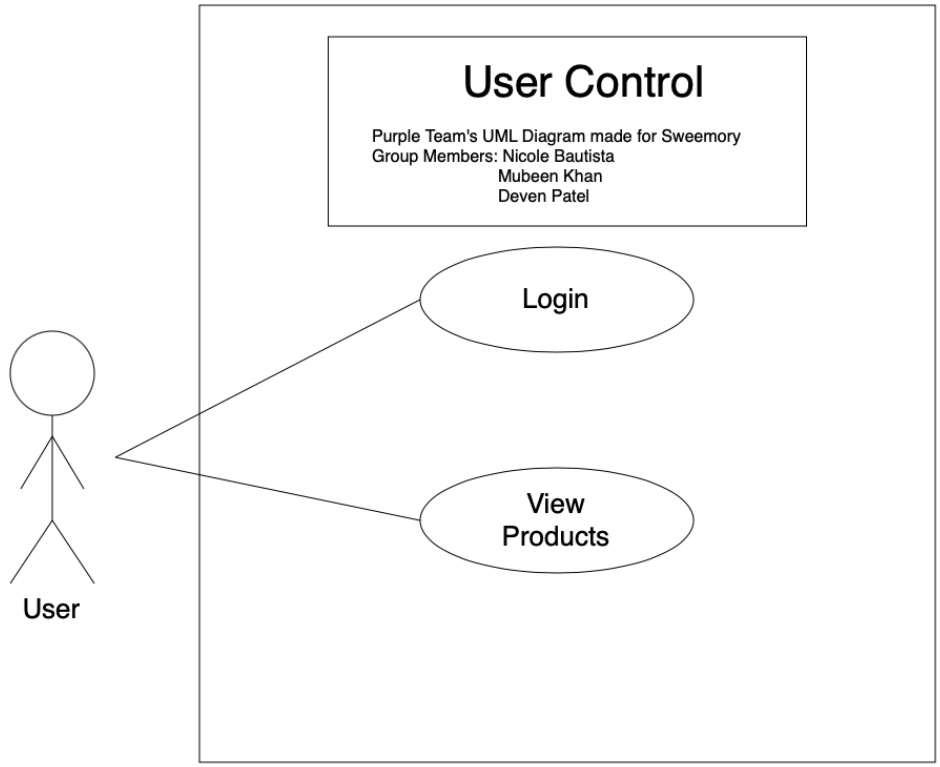
The owners and employees of Sweemory will check the expiration dates of their products every so often. But it also depends on the type of product, some products expire quicker than others. Which means that items such as milk will be checked more often than M&M's. Other than checking the product's expiration date manually, they sometimes look at their Excel sheet. When new products are purchased, one of the owners will make an entry in their Excel sheet about the specific product(s). They take note of the quantity arrived, the expiration date, arrival date, name and if it has been opened or not.

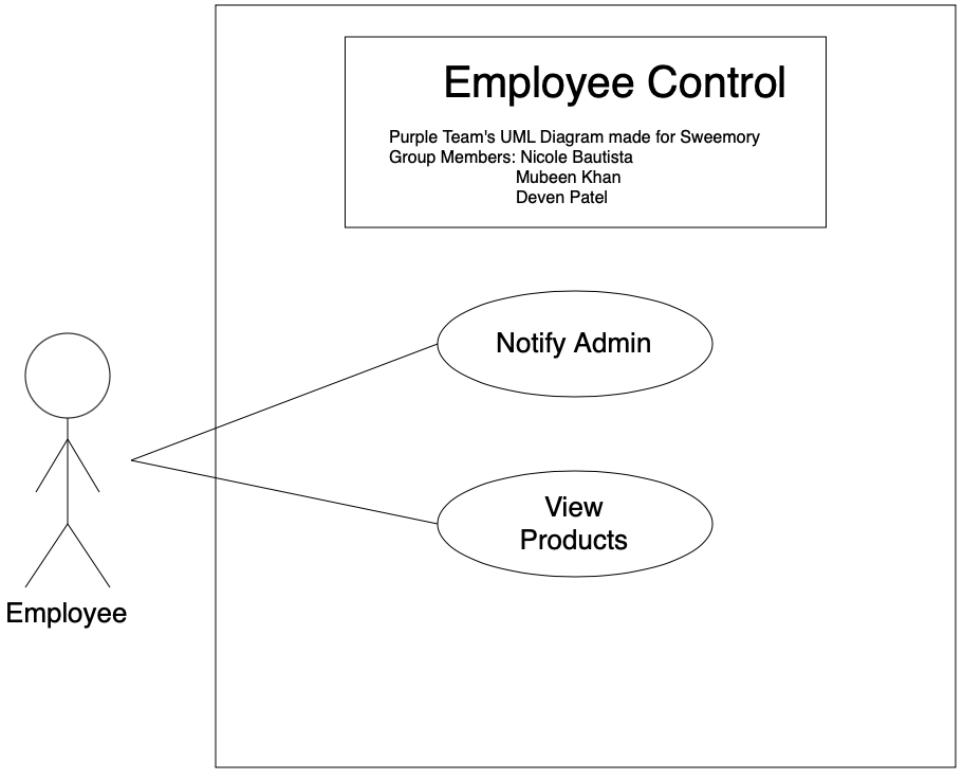
Appendices

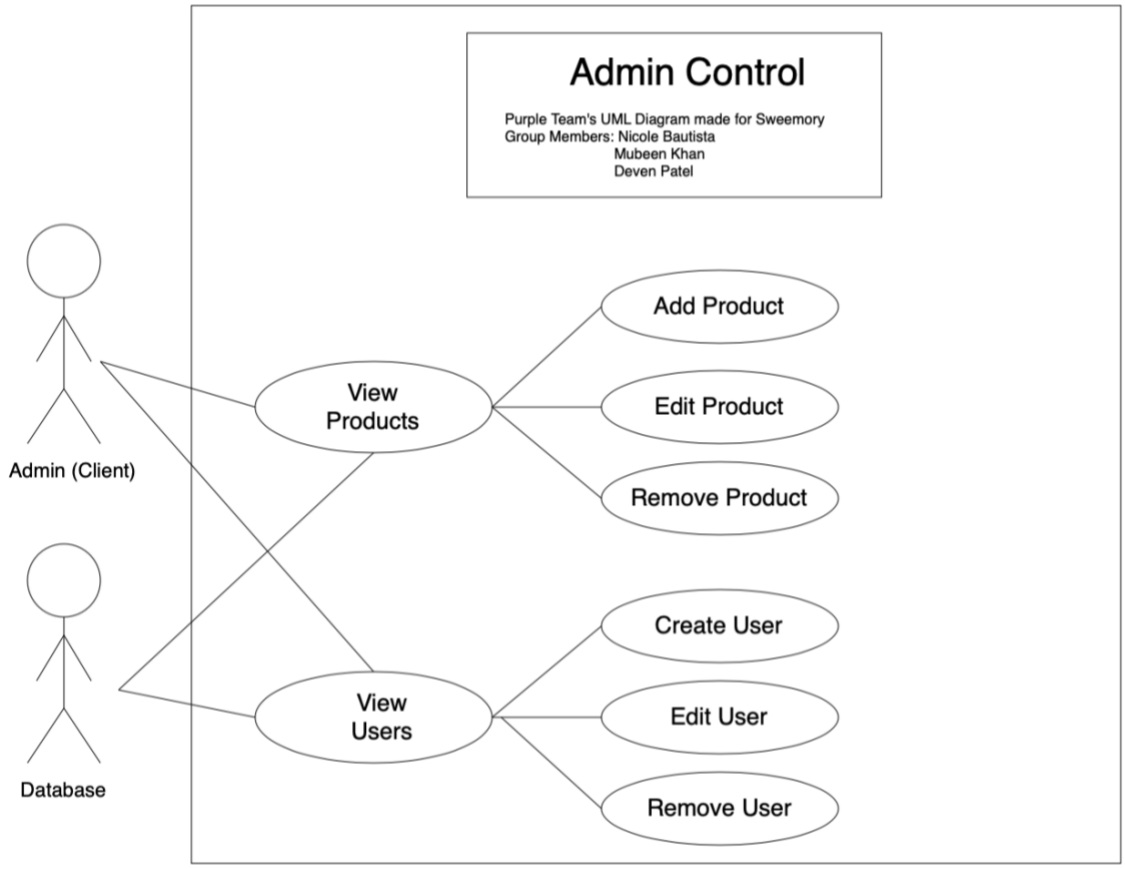
Appendix 1

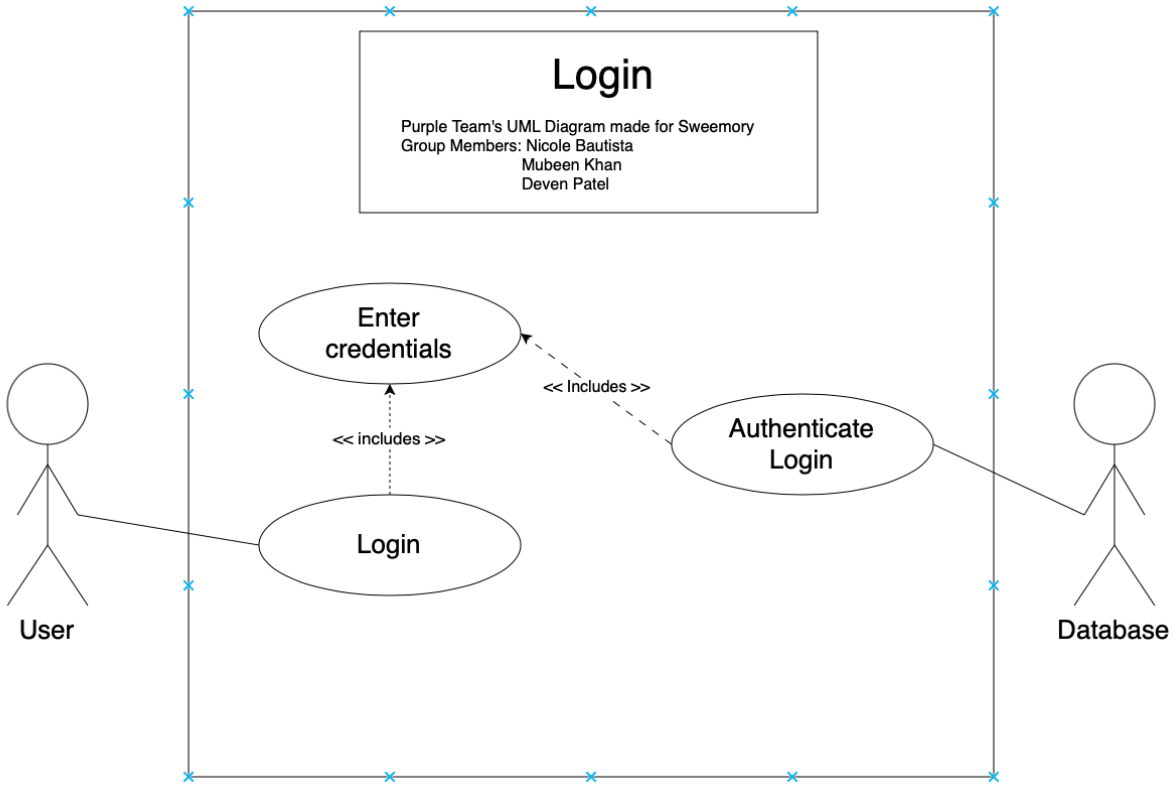


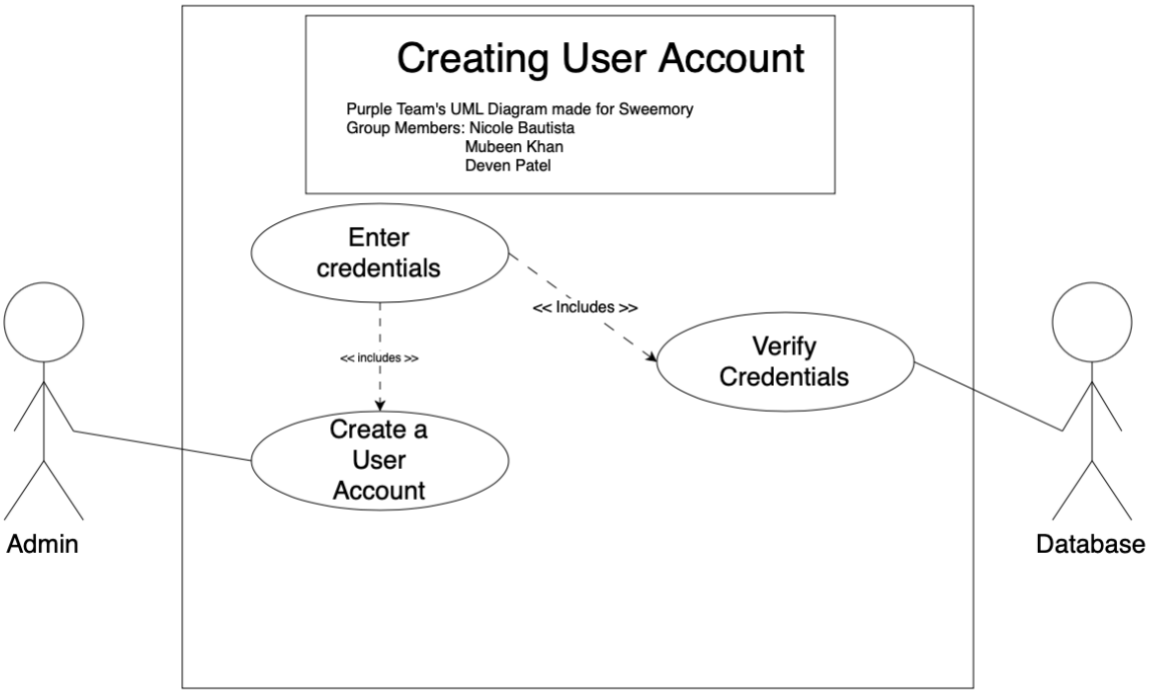
Appendix 2

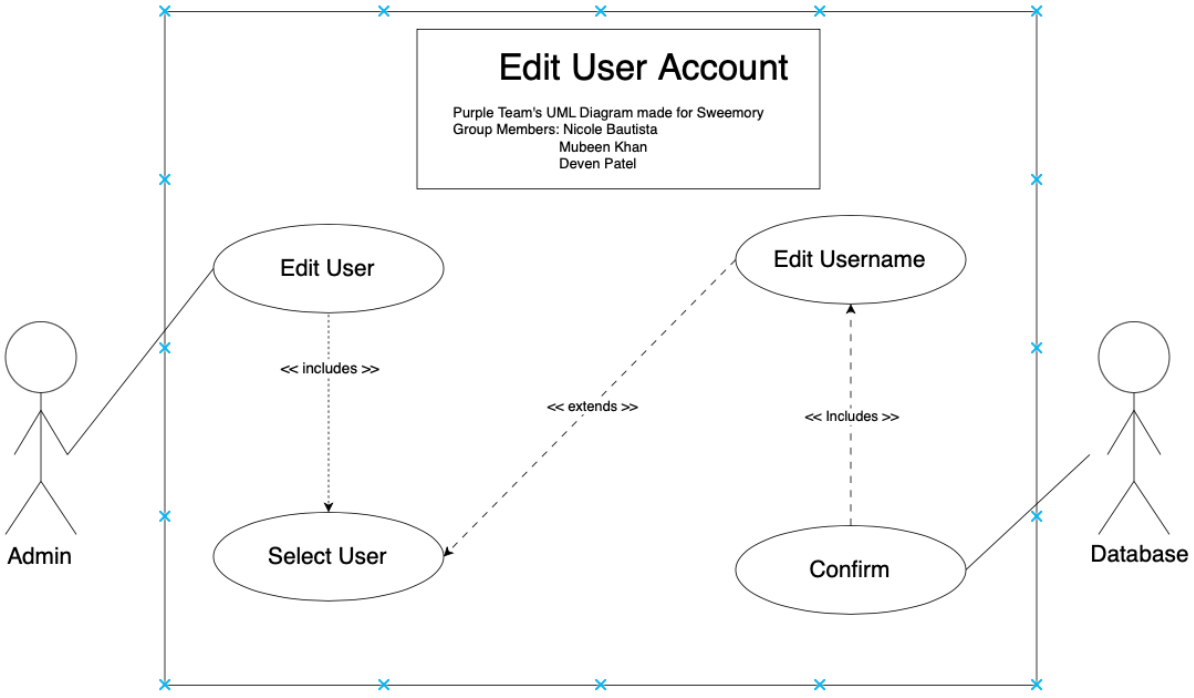

User Control
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Login
View Products
User

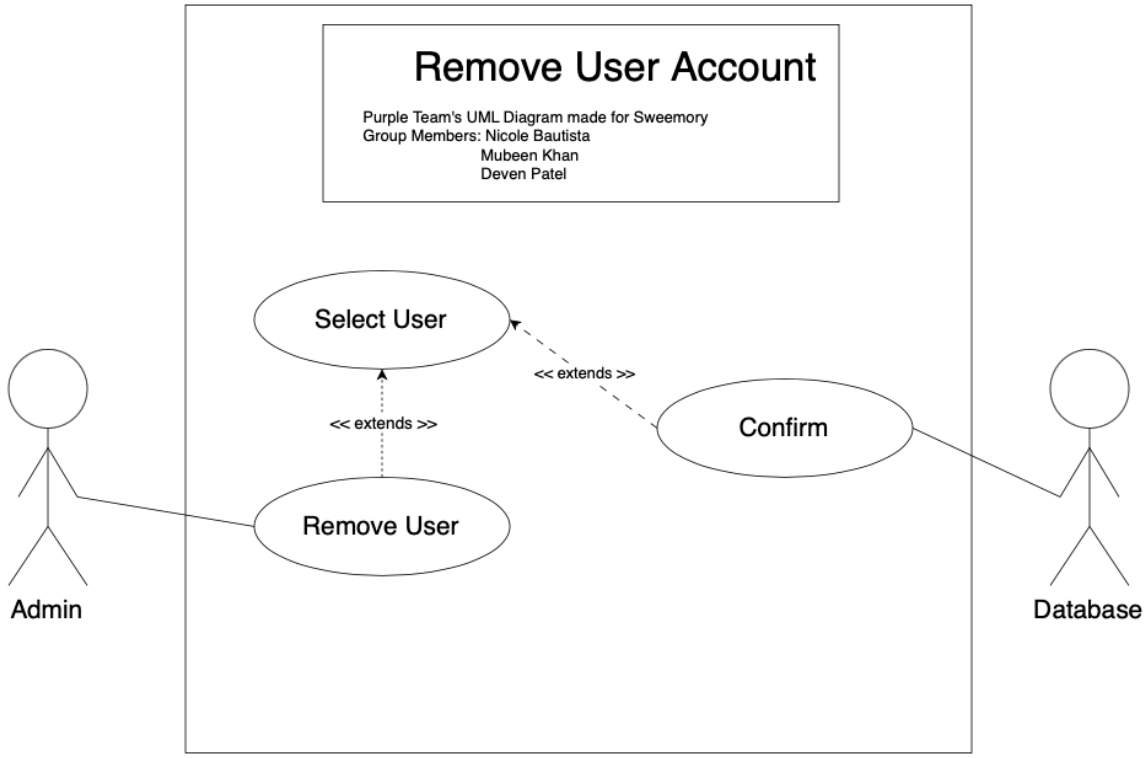

Employee Control
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Notify Admin
View Products
Employee

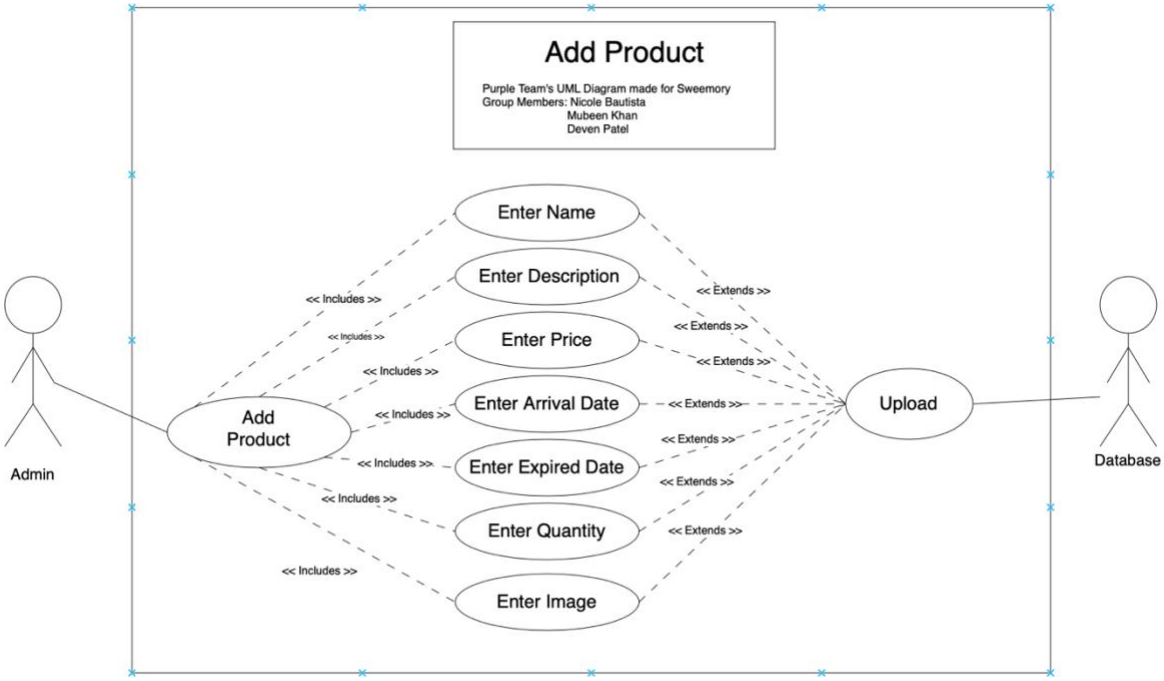

Admin Control
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Admin (Client)
Database
View Products
View Users
Add Product
Edit Product
Remove Product
Create User
Edit User
Remove User

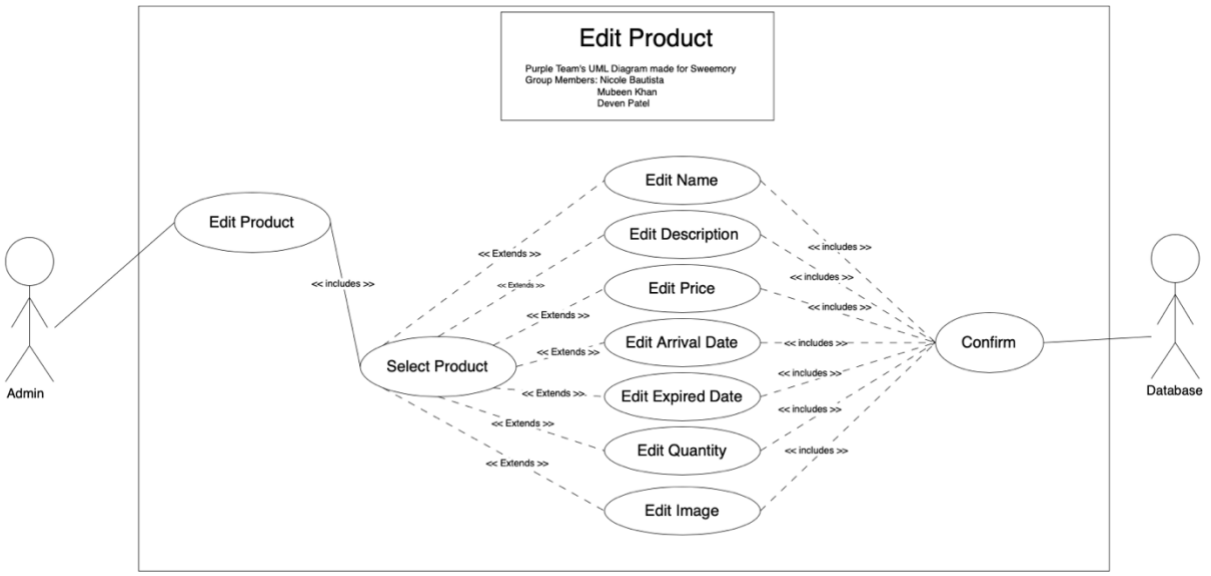

Login
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Enter credentials
<< Includes >>
<< includes >>
Authenticate Login
Login
User
Database

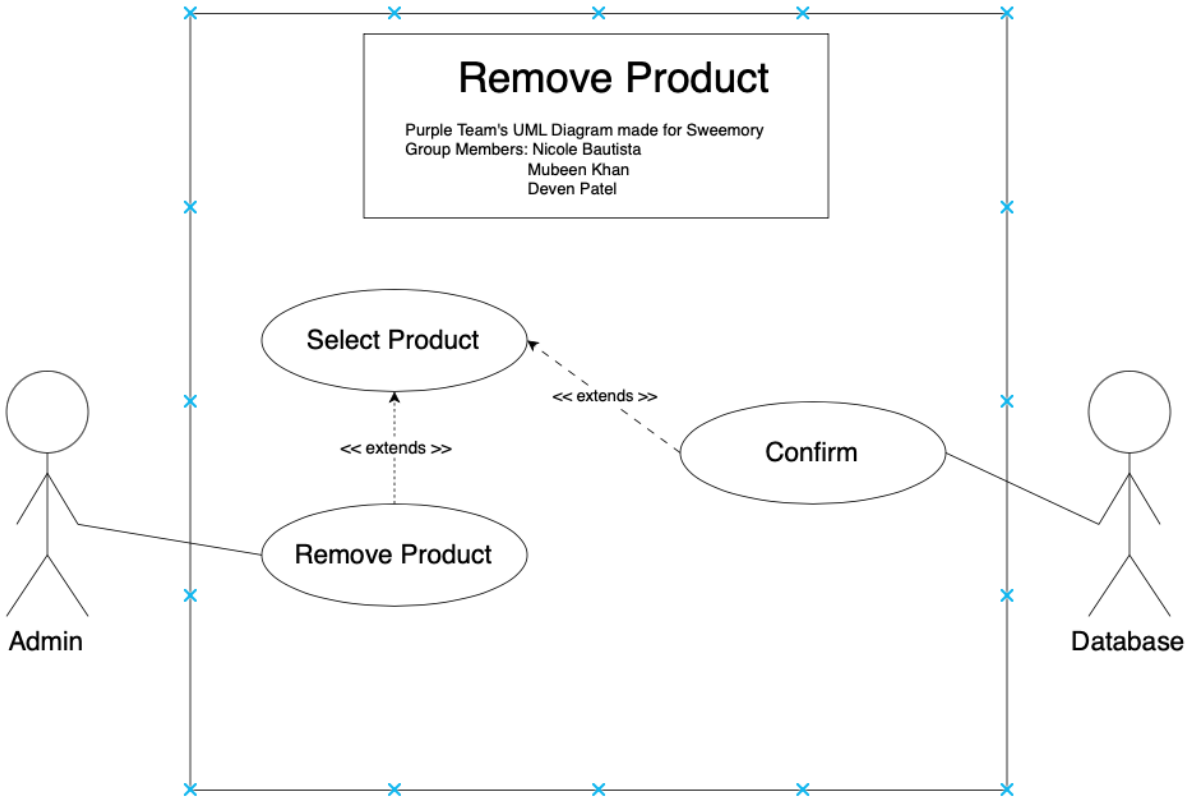

Creating User Account
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Enter credentials
<< Includes >>
<< includes >>
Verify Credentials
Create a User Account
Admin
Database


Edit User Account
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Edit User
Edit Username
<< includes >>
<< extends >>
<< Includes >>
Admin
Select User
Confirm
Database


Remove User Account
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Select User
<< extends >>
<< extends >>
Confirm
Remove User
Admin
Database


Add Product
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Enter Name
Enter Description
Enter Price
Enter Arrival Date
Enter Expired Date
Enter Quantity
Enter Image
<< Includes >>
<< Includes >>
<< Includes >>
<< Includes >>
<< Includes >>
<< Includes >>
<< Includes >>
Add Product
Admin
<< Extends >>
<< Extends >>
<< Extends >>
<< Extends >>
<< Extends >>
<< Extends >>
<< Extends >>
Upload
Database


Edit Product
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Edit Product
Admin
<< includes >>
Select Product
<< Extends >>
Edit Name
Edit Description
Edit Price
Edit Arrival Date
Edit Expired Date
Edit Quantity
Edit Image
<< includes >>
Confirm
Database


Remove Product
Purple Team's UML Diagram made for Sweemory
Group Members: Nicole Bautista
Mubeen Khan
Deven Patel
Select Product
<< extends >>
Confirm
<< extends >>
Remove Product
Admin
Database

Appendix 3

Use Case ID	SM-CreateUser		
Use Case Name	Creating Users		
Created By:	Deven Patel	**Last Updated By:**	N/A
Date Created:	2023-03-02	**Last Revision By:**	N/A
Actors:	Owner of Sweemory (Primary)		
Description:	This use case adds a new employee to the database and allows employees to login.		
Trigger:	Owner fills out form with employee's information.		
Pre-Conditions:	Owner has the employee's information and decides to give them access to the application.		
Post-Conditions:	1- Employees information is saved in the system 2- Employees can login and have access to the application		
Normal Flow	1. Owner fills out the form with specific employee's personal information 2. System checks if password verifications and uncomplete form exceptions 4. System adds the employee' information to the database		
Alternative Flows	N/A		
Exceptions:	A) if password and re-enter password do not match 1. System displays error message. 2. Makes owner retype password. B) if form is not fully complete 1. System displays error message. 2. Makes owner fully fill the form.		
Includes:	N/A		
Frequency of Use:	When needed		
Special Requirements:	N/A		
Assumptions:	Owner has the necessary information to fill out the form		
Notes and Issues:	N/A		

Use Case ID	SM- Login		
Use Case Name	Login		
Created By:	Nicole Bautista	**Last Updated By:**	N/A
Date Created:	2023-03-02	**Last Revision By:**	N/A
Actors:	Employees and Owner of Sweemory (Primary) Database (Secondary)		
Description:	This use case allows the employees and owner of Sweemory to login to their accounts in order to view the store's products.		
Trigger:	Employee or Owner completes the login form.		
Preconditions:	1- The employees and the owners have a valid account created. 2- The employees and the owner are not logged in.		
Postconditions:	1- The user is signed into their account. 2- Regular users (employees) can view the store's product, while Admin users (owner) are able to view, add, edit and delete the products.		
Normal Flow	1- The system needs user's id and password. 2- Employees fill the login form. 3- The system will authenticate the user's login information.		
Alternative Flows	3a- If authentication fails, but the user does not have an account: 1- System will ask the user to ask their higher-up to create an account. ->Creating User (Extend SM-CreateUser)		
Exceptions:	3a. If authentication fails and user has an account: 1- System displays an error message. 2- User must restart the login form without being logged in.		
Includes:	SM-CreateUser		
Frequency of Use:	On demand		
Special Requirements:	N/A		
Assumptions:	Users has an account created. Users can enter their information into the login form.		
Notes and Issues:	N/A		

Use Case ID	SM-Add Product		
Use Case Name	Adding a Product		
Created By:	Mubeen	**Last Updated By:**	N/A
Date Created:	2023-03-02	**Last Revision By:**	N/A
Actors:	Employees and Owner of Sweemory (Primary) Database (Secondary)		
Description:	This use case will allow the admin to add a new product to the database.		
Trigger:	The admin clicks on “add product”		
Preconditions:	The product is not listed on the system.		
Postconditions:	A product will be added to the database. The database will list the amount added on the date it was added.		
Normal Flow	1. The admin logs in with an account that has administrator privileges 2. The admin views the products table 3. The admin clicks on the “add product” button 4. An empty form will be displayed on the screen 5. The admin is required to enter a title, a description, a price, the arrival date, the arrival amount, the expiration date, and a picture. 6. The admin clicks on the “add product” button		
Alternative Flows	6a. if one or more information was not added to the form: 1. The system will display a “missing information” message. 2. The admin will be required to fill out everything in the form without submitting the product. 3. After filling out the form, the admin may proceed with step 6. 6b. if the admin no longer wishes to add the product: 1. The admin can press the “cancel” button. 2. The system will send the admin back to the “view product table” page.		
Exceptions:	6b		
Includes:	N/A		
Frequency of Use:	On demand		
Special Requirements:	N/A		
Assumptions:	The admin is logged in with administrator privileges. The admin will be able to fill out a form.		
Notes and Issues:	N/A		

Use Case ID	SM-Delete Product		
Use Case Name	Deleting a product		
Created By:	Mubeen	**Last Updated By:**	N/A
Date Created:	2023-03-08	**Last Revision By:**	N/A
Actors:	Employees and Owner of Sweemory (Primary) Database (Secondary)		
Description:	This use case will delete a product from the database and remove it from the system.		
Trigger:	The admin clicks on the button that deletes a product.		
Preconditions:	The product is listed on the system		
Postconditions:	The removed product is no longer listed among the products on the system. The database is updated.		
Normal Flow	1. The admin logs in with an account that has administrator privileges 2. The admin views the products table 3. The admin clicks on the “delete” button near the product they want to remove. 4. A confirmation appears on the screen. 5. The admin clicks on “yes” to delete the product 6. A message will appear on the screen saying that the product has successfully been deleted		
Alternative Flows	4a. if the admin does not want to delete the item anymore: 1. The admin clicks on “No” to cancel. 2. The product will stay on the list of products.		
Exceptions:	4a		
Includes:	N/A		
Frequency of Use:	On demand		
Special Requirements:	N/A		
Assumptions:	The admin is logged in with administrator privileges.		
Notes and Issues:	N/A		

Use Case ID	SM-Notify Admin		
Use Case Name	Notify the Admin		
Created By:	Nicole Bautista	**Last Updated By:**	N/A
Date Created:	2023-03-09	**Last Revision By:**	N/A
Actors:	Employee of Sweemory (Primary)		
Description:	Employees will be able to notify the owners/admins when a product is low on stock or when it is done.		
Trigger:	Employee clicks on notify admin button.		
Pre-Conditions:	The product is out of stock or low on stock. Employee has an account to access these features.		
Post-Conditions:	1- System will notify admin by showing an alert when admin first logs in.		
Normal Flow	1- Employee is logged in. 2- Views Product table. 3- Clicks on “Notify Admin” button on the page. 4- A confirmation notice will appear. 5- Employee presses “yes” to notify. 6- Message will appear that they have successfully notified the admin.		
Alternative Flows	5a – Employee changes their mind. 1. They press “cancel”. 2. They go back to the view products page.		
Exceptions:	5a		
Includes:	N/A		
Frequency of Use:	When needed		
Special Requirements:	N/A		
Assumptions:	The product is either about the expire, low on stock or out of stock.		
Notes and Issues:	N/A		

Appendix 4

Appendix 4 - Class Diagram