**Inventory Module**

Sarah and Ted buy products from local farmers and crafters. One of their frequent customers has been diagnosed with celiac disease and has approached them to include gluten free products.

Also, Sarah and Ted wonder about food recalls and how would they be able to inform their customers if they have purchased a recalled product.

Sarah and Ted would like a system that helps them organize their purchases by location, instead of having to look through a combination of online orders and paper receipts. They would like the ability to attach a copy or picture of the receipt to the purchase for tax purposes.

Please create a class diagram and sequence diagrams for the User Stories and Systems Use Case Specifications detailed below.

Use Case: Maintain Products

User Story

As the owner of this business, I would like to record products so that I can easily see what I’ve purchased, what I’ve sold and how much inventory I have at each location.

Acceptance Criteria:

1. Must be able to record critical information about each product.
2. Allow deletion of product in case the owner makes a mistake. The owner must be prompted to confirm, and product must not be involved in any purchases or sales.
3. Must be able to easily retrieve product information.
4. Must be able to retrieve a list of products by product type.

Diagram

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | **Create Product Information** | | |
| Triggering Event | A new product of interest to the business | | |
| Brief Description | Allows the Owner to record a new product. | | |
| Actors | Owner | | |
| Related Use Cases |  | | |
| Preconditions | Owner has opened the Main Menu. | | |
| Post Conditions | Product is saved to the database and now can be purchased. | | |
| Flow of activities | Actor | | System |
|  |  | Requests to add a new product | Displays a list of product types and prompts for selection.  Prompts for product name, ingredients, description, price |
|  |  | Enter name, description, ingredients, and price.  Selects product type | Name, description and price must be entered.  Product type must be selected.  Generates identifier.  Data is valid.  Displays product.  Prompts to save |
|  |  | Request to save | Saves the product and returns to the main menu |
| Exception Conditions | * Owner chooses to cancel adding the product | | |

A diagram of a project

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | **Query Products by Product Type** | | |
| Triggering Event | List of products required | | |
| Brief Description | Allows the Owner to query products | | |
| Actors | Owner | | |
| Related Use Cases |  | | |
| Preconditions | Owner has opened the Main Menu. | | |
| Post Conditions | Product information provided to the actor. | | |
| Flow of activities | Actor | | System |
|  | 1. | Requests for products | Displays a list of product types and prompts for selection |
|  | 2. | Selects product type | Displays a list of products for that product type |
| Exception Conditions | * Owner chooses to cancel query products | | |

A diagram of a product

Description automatically generated

Use Case: Maintain Purchase

User Story

As the owner of this business, I would like to record my purchases so that I can quickly see how much money I’ve spent, what I’ve ordered and from where. With each purchase, I want to record which location received the product. Because I like to order in bulk, I must be able to record multiple locations for each product.

Acceptance Criteria:

1. Must be able to record purchases by supplier.
2. Must be able to query purchase details by location.

Use Case Descriptions

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | **Create Purchase** | | |
| Triggering Event | Purchase of products. | | |
| Brief Description | Allows the Owner to record a new purchase. | | |
| Actors | Owner | | |
| Related Use Cases |  | | |
| Preconditions | Owner has opened the Main Menu. | | |
| Post Conditions | Purchase is saved to the database and now can be queried. | | |
| Flow of activities | Actor | | System |
|  | 1. | Requests to add a new purchase | Displays a list of suppliers and prompts for selection. Prompts for purchase date and selection of receipt file location. |
|  | 2. | Selects a supplier.  Enters purchase date and receipt file selected. | Verifies that a supplier was selected Verifies that date was entered, and receipt selected.  Creates a unique identifier for the purchase.  Displays the purchase.  Prompts to enter purchase details. |
|  | Loop | Chooses new product | Displays a list of products, sorted by product type, and prompts for selection. |
|  | 3. | Selects a product | Display a list of locations and prompts for select.  Prompts for quantity. |
|  | 4 | Selects location and enters quantity and price | Creates a unique identifier for the detail.  Data is valid.  Extended price is calculated (price \* quantity ordered).  Taxes are calculated and displayed.  The purchase total is updated with the extended price + taxes.  Purchase tax is updated with the tax amount.  Displays purchase including totals, date and list of products and locations.  Prompts to add another location |
|  |  | Repeats above step until all locations for a product were selected | Display a list of products and prompts for selection |
|  | End | When all products are selected | Prompts to save purchase |
|  | 5. | Chooses to save | Saves the purchase and returns to the main menu |
| Exception Conditions | * Owner chooses to cancel adding the purchase | | |

A diagram of a project

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Use Case Name | **Query Purchases** | | |
| Triggering Event | Owner requires a list of purchases for a date period listing Supplier, Products quantity purchased, price paid and the extended amount (price \* quantity) | | |
| Brief Description | Allows the Owner to retrieve purchases for a specified date range | | |
| Actors | Owner | | |
| Related Use Cases |  | | |
| Preconditions | Owner has opened the Main Menu. | | |
| Post Conditions | purchases are retrieved, totaled, and displayed | | |
| Flow of activities | Actor | | System |
|  | 1. | Requests to query purchase (by date) transactions. | Displays a calendar. |
|  | 2. | Selects date range. | Verifies that dates are selected.  Retrieves purchases in the specified date range, calculating totals.  Prompts to exit |
|  | 3. | Request to cancel | returns to the main menu |
| Exception Conditions |  | | |

A diagram of a product

Description automatically generated

.h prototypes

.h files

ProductType.h

public class ProductType {

int productTypeID;

String description;

Product prodSet[];

getProductTypes();

}

Product.h

public class Product {

int productID;

String productName;

float double;

ProductType productType;

SaleDetail saleDetailSet[];

String ingredients;

String description;

PurchaseDetail purchaseDetailSet[];

getProduct();

}

Location.h

public class Location {

int locationID;

String locationName;

String address;

getLocation();

{

Supplier.h

public class Supplier {

int supplierID;

String businessName;

String contactName;

String email;

int phone;

String address;

getSupplier();

}

Purchase.h

public class Purchase {

int purchaseID;

Supplier supplier;

int purchasedate;

String receipt;

PurchaseDetail purchaseDetailSet[];

getPurchase();

}

public class PurchaseDetail {

Purchase purchase;

int lineNumber;

Product product;

Location location;

int quantity;

float price;

getPurchaseDetails();

}

UIController.h

public class UIController {

retrieveProductTypes() {

// sends requests to retrieve product types

// displays a list of product types

}

addProduct(name,description,price,ingredients) {

// sends request to add product

// displays product

}

save() { sends request to save product }

selectProductType(productTypeID) {

// sends request to select product type

// returns list of products

}

startPurchase() {

// sends request to retrieve a supplier list

// displays supplier list

}

createPurchase(supplierID,date,receipt) {

// sends request to create purchase

// displays purchase

}

createDetail () {

// sends request to retrieve a list of products

// displays product list

}

selectProduct(productID) {

// sends request to retrieve a list of locations

// displays location list

}

enterDetail(locationID,quantity,price){

// sends request to create purchase detail

// displays purchase detail and totals

}

save() { sends request to save purchase and details}

requestPurchases() { displays calendar }

selectDate(startDate,endDate) {

// sends request to retrieve purchase

// displays purchases and totals

}

}

DomainController.h

getProductTypes() {

// sends request to get product types

// returns product types

}

createProduct(name,description,price,ingredients) {

// sends request to create product

// returns product

}

createProductID()

save() {// sends request to save product}

getProducts(productTypeID) {

// sends request to retrieve products for a specific product type

// returns product list

}

getSuppliers() {

// sends request to retrieve suppliers

// returns list of suppliers

}

createPurchase(supplierID,date,receipt) {

// sends request to create purchase

// returns purchase

}

createPurchaseID()

getProducts() {

// sends request to retrieve products

// returns product list

}

getProduct(productID) {

// selects product

// sends request to retrieve locations

// returns list of locations

}

selectProduct(productID)

createDetail(locationID,quantity,price) {

// sends request to create purchase detail

// returns purchase detail and totals

}

createLineNumber()

updateTotals()

save() { // sends request to save purchase and details }

getPurchases(startDate,endDate) {

// sends requests to retrieve purchase and purchase details by date

// returns purchases and totals

}

}

EntityManager.h

public class EntityManager {

persist(object)

// saves

get(class, selection criteria)

// retrieves from the table

}