For the first version of the store management system, we want to start with the following user stories:

- As a user, I want to add a new product into the system or update an existing one.

- As a user, I want to add a new customer into the system or update an existing one.

- As a user, I want to record a purchase from a customer into the system or update an existing one.

Tasks:

1. Write a common use case for each user story. Sketch the screens the system should display in each use case.

- As a user, I want to add a new product into the system or update an existing one.

\* Adding/updating a product can either add successfully if there is no other product with the given productID, or may error if there is already a matching product id, or if the user fails to input all the required fields. For this project we will look at the cases where adding a product goes smoothly and then where they input a duplicate id.

1. Name: Adding a new product into the system
2. Actors:Manager (or employee with appropriate permissions)
3. Goals: add product
4. Preconditions: must be the correct user, product should not already be in the system
5. Summary: the user will log into the product management section and add a new product with the appropriate attributes such as name, type, cost, etc.
6. Related use cases- update products
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display main page |
| Select Manage Product | Display Product Screen |
| Fill out product attributes |  |
| Confirm entry | Hide the “Product” screen and display a “Product Added Successfully” screen |
| Select “OK” | Display “Main” screen again |

1. Name: Adding a new product into the system (edge case: duplicate ID)
2. Actors:Manager (or employee with appropriate permissions)
3. Goals: add product
4. Preconditions: must be the correct user, product should not already be in the system
5. Summary: the user will log into the product management section and add a new product with the appropriate attributes such as name, type, cost, etc.
6. Related use cases- update products
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display main page |
| Select ManageProduct | Display Product Screen |
| Fill out product attributes |  |
| Confirm entry | Hide the “Product” screen and display a “Duplicate Product ID warning” screen |
| Select “BACK” to navigate back to adding the product | Display previous “Product” screen. Either the user inputs a new ID and the first use case occurs, or they still enter a duplicate ID and the warning appears again. |

1. Name: Update product in the system
2. Actors:Manager (or employee with appropriate permissions)
3. Goals: add product
4. Preconditions: must be the correct user, product should not already be in the system
5. Summary: the user will log into the product management section and add a new product with the appropriate attributes such as name, type, cost, etc.
6. Related use cases- update products
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display main page |
| Select Manage Product | Display Product Screen |
| Enter productID and select load |  |
| Load stored product |  |
| Update desired attributes and select update |  |
| Confirm entry | Hide the “Product” screen and display a “Product Updated Successfully” screen |
| Select “OK” | Display “Main” screen again |

- As a user, I want to add a new customer into the system or update an existing one.

\* Adding/updating a Customer can either add successfully if there is no other customer with the given customerID, or may error if there is already a matching customer id, or if the user fails to input all the required fields. For this project we will look at the cases where adding a customer goes smoothly and then where they input a duplicate id.

1. Name: Adding a new customer into the system
2. Actors:Manager (or employee with appropriate permissions)
3. Goals: add add customer
4. Preconditions: must be the correct user, customer should not already be in the system
5. Summary: the user will log into the product management section and add a new customer with the appropriate attributes such as Name, Address, customerID, phone number, payment info, ...
6. Related use cases- update customer info
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display “Main” landing page |
| Select “Manage Customer” | Display “Customer” screen |
| Fill out customer attributes |  |
| Confirm changes | Display “Customer Added” page |
| Select “OK” | Display “Main” page again |

1. Name: Adding a new customer into the system (edge case: Duplicate customerID)
2. Actors:Manager (or employee with appropriate permissions)
3. Goals: add add customer
4. Preconditions: must be the correct user, customer should not already be in the system
5. Summary: the user will log into the product management section and add a new customer with the appropriate attributes such as Name, Address, customerID, phone number, payment info, ...
6. Related use cases- update customer info
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display “Main” landing page |
| Select “ManageCustomer” | Display “Customer” screen |
| Fill out customer attributes |  |
| Confirm changes | Display “Duplicate ID Warning” page |
| Select “BACK” to navigate back to adding the customer | Display previous “Customer” screen. Either the user inputs a new ID and the first use case occurs, or they still enter a duplicate ID and the warning appears again. |

1. Name: updating customer in the system
2. Actors:Manager (or employee with appropriate permissions)
3. Goals: add add customer
4. Preconditions: must be the correct user, customer should not already be in the system
5. Summary: the user will log into the product management section and add a new customer with the appropriate attributes such as Name, Address, customerID, phone number, payment info, ...
6. Related use cases- update customer info
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display “Main” landing page |
| Select “Manage Customer” | Display “Customer” screen |
| Fill out customer attributes |  |
| Confirm changes | Display “Customer Added” page |
| Select “OK” | Display “Main” page again |

- As a user, I want to record a purchase from a customer into the system or update an existing one.

\* Adding/updating a purchase can either add successfully if there is no other purchase with the given purchaseID AND there are existing customers and products with the given corresponding ids, or may error if there is already a matching purchase id, no existing customer or product, or if the user fails to input all the required fields. For this project we will look at the cases where adding a purchase goes smoothly and then where they input a product id that does not match any stored product.

\*\*See end of file for a different format for purchase. Both allow for purchase to be made and a transaction to be retrieved for update. I chose this method for my project because it allows for multiple items to be purchased in one transaction.

1. Name: Record a new purchase into the system
2. Actors:Manager (or cashier)
3. Goals: record a transaction
4. Preconditions: must be the correct user, customer must have selected products to buy and have a means of purchasing them
5. Summary: the user will log into the product management section and add a new purchase. All appropriate attributes must be recorded including all product information from the product database, and the customer info and purchase method from the customer database.
6. Related use cases- add/update products and add/update customer (required to access all needed info to complete transaction
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display “Main” landing screen |
| Select “Manage Purchase” | Display Purchase page |
| Enter PurchaseID |  |
| Scan products or enter manually | Populate transaction with products |
| Select “Finish And Pay” | Display “Payment Method” page |
| Enter Customer name or ID | Display available customers |
| Select Customer |  |
| Select “Select Customer” button | Populate with customer info/ prompt payment method |
| Select Confirm transaction | Display “Transaction Complete!” screen |
| Select “OK” | Display “Main” screen again |

1. Name: Record a new purchase into the system (Edge case: item not found)
2. Actors:Manager (or cashier)
3. Goals: record a transaction
4. Preconditions: must be the correct user, customer must have selected products to buy and have a means of purchasing them
5. Summary: the user will log into the product management section and add a new purchase. All appropriate attributes must be recorded including all product information from the product database, and the customer info and purchase method from the customer database.
6. Related use cases- add/update products and add/update customer (required to access all needed info to complete transaction
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display “Main” landing screen |
| Select “Manage Purchase” | Display Purchase page |
|  |  |
| Enter a productID/ scan the item to retrieve the productID |  |
| Return to “Purchase” page | Display previous “Purchase” screen. Either the user inputs a new ID and the first use case occurs, or they still enter a duplicate ID and the warning appears again. |

1. Name: Update Purchase
2. Actors:Manager (or cashier)
3. Goals: record a transaction
4. Preconditions: must be the correct user, customer must have selected products to buy and have a means of purchasing them
5. Summary: the user will log into the product management section and add a new purchase. All appropriate attributes must be recorded including all product information from the product database, and the customer info and purchase method from the customer database.
6. Related use cases- add/update products and add/update customer (required to access all needed info to complete transaction
7. Steps:

|  |  |
| --- | --- |
| Actor | System |
| Log into system | Display “Main” landing screen |
| Select “Manage Purchase” | Display Manage Purchase page |
| Enter a purchaseID |  |
| Select Confirm transaction | Display “Transaction Updated!” screen |
| Select “OK” | Display “Main” screen again |

1. Draw the entity-relationship diagram for this system. We assume the minimal requirement with two entities: products and customers, and one relationship "a customer purchases a product".



1. Design the database logically, i.e., write the relations, attributes, and define keys.

Products:

* barcode, name, productid, expiration date, price, tax rate, quantity, supplier, manufactured date,...

Products(productid, barcode, name, expiration date, price, tax rate, quantity, supplier, manufactured date )

Customers:

* Name, Address, customerID, phone number, payment info, ...

Customers(CustomerID, name, address, phone number, payment info)

Relationship:

* Buy: date, time, quality, pice, tax, total cost

purchase(transactionID, customerID, productID, date, time, quantity, price, tax, total cost)

Using table specific IDs as key.

1. Design the database physically using SQL, i.e., write SQL code to create the tables for those relations.

CREATE TABLE "Customers" (

"CustomerID" INTEGER NOT NULL,

"Name" TEXT DEFAULT 'Guest',

"Phone" TEXT DEFAULT '(000)000-000',

"Address" TEXT,

PRIMARY KEY("CustomerID")

);

CREATE TABLE "Products" (

"ProductID" INTEGER NOT NULL,

"Name" TEXT,

"Price" REAL,

"Quantity" REAL,

"Tax Rate" REAL,

PRIMARY KEY("ProductID")

);

CREATE TABLE "Purchases" (

"TransactionID" INTEGER NOT NULL,

"CustomerID" INTEGER,

"ProductID" INTEGER,

"Date" REAL,

"Time" REAL,

"Quantity" INTEGER,

"Price" REAL,

"Tax" REAL,

"Total" REAL,

PRIMARY KEY("TransactionID"),

FOREIGN KEY("CustomerID") REFERENCES "Customers"("CustomerID"),

FOREIGN KEY("ProductID") REFERENCES "Products"("ProductID"),

FOREIGN KEY("Price") REFERENCES "Products"("Price")

);

1. Insert data into the tables, with at least 5 products, 5 customers, and 10 purchases.

Products

1 Apple 0.99 100.0 0.09

2 Orange 1.99 200.0 0.09

3 Banana 0.39 150.0 0.09

7 Water 2.0 150.0 0.09

12 Chips 3.99 150.0 0.09

34 Coke 0.69 100.0 0.09

12,12,34,7,3,1,12

Customers

1 Ellie 999-999-9999 USA

2 Lynn 000-000-0000 Auburn, AL

3 Emma 000-000-0000 Japan

7 Monty 000-000-0000 America

8 Meg 314-245-3775 New York

81 Matt 202-978-1423 India

282 Sam 000-000-0000 Atlanta, GA

953 Hannah 000-000-0000 France

1 1 1 9/10/19 2:34 2 0.99 0.09 2.36

2 1 7 9/10/19 5:00 1 2.0 0.09 2.09

3 7 3 2/17/12 3:00 1 0.39 0.09 0.48

4 953 12 3/19/15 4:00 2 3.99 0.09 8.16

5 81 12 4/19/13 3:00 1 3.99 0.09 4.08

6 953 34 8/20/15 2:00 3 0.69 0.09 2.34

7 282 7 5/12/19 7:00 2 2.0 0.09 4.18

8 7 3 4/2/19 8:00 1 0.39 0.09 0.48

9 2 1 6/5/17 11:00 3 0.99 0.09 3.24

10 8 12 6/9/17 2:00 2 3.99 0.09 8.16

See .db



Below is an alternative view for adding a purchase if the purchase is only done one product at a time manually rather than the method in the case studies above that scans multiple products in one purchase and then calculates the totals and charges the customer accordingly.

