1. Use Cases
   1. New Product
      1. Use Case Name: Add new product
      2. Actors: Any employee of the store (cashier, manager, owner)
      3. Goals: Have new product added to system so it can later be used in transactions between the store and customer as well as having its inventory tracked
      4. Preconditions: No entry for the new product exists in the store database
      5. Summary: Once the product is added all of its attributes will exist in the store database and it can later be used in transactions as well as stock checks
      6. Related Use Cases: Add product to cart, check price of product, check stock of product
      7. Steps:

|  |  |
| --- | --- |
| Step | Description of System Response |
| Choose add new product command | Dialog opens for new product add |
| Fill in attributes for the new product | Systems asks for product id, product name, price, quantity in stock |
| Confirm information | Dialog closes if information confirmed as correct |

* + 1. Postcondition: Entry for product now exists for product in database. Can later be added to other store functions such as transactions and inventory checks



Cancel Product Entry

Enter Product into System

Quantity:

Product Price:

Product Name:

Product ID:

New Product Entry

* 1. New Customer
     1. Use Case Name: Add new customer
     2. Actors: Any employee of the store (cashier, manager, owner), and customer
     3. Goals: Have new customer added to system so it can later be used in transactions
     4. Preconditions: No entry for the customer exists in the store database
     5. Summary: Once the customer is added all of its attributes will exist in the store database and it can later be used in transactions
     6. Related Use Cases: Check customer entry
     7. Steps:

|  |  |
| --- | --- |
| Step | Description of System Response |
| Choose add new customer command | Dialog opens for new customer add |
| Fill in attributes for the new customer | Systems asks for ID, last and first name, phone number |
| Confirm information | Dialog closes if information confirmed as correct |

* + 1. Postcondition: Entry for product now exists for customer in store database





Cancel Product Entry

Enter Customer into System

Phone Number

Customer First Name

Customer Last Name:

New Customer Entry

Customer ID:

* 1. Record Purchase from customer
     1. Use Case Name: Record purchase
     2. Actors: Any employee of the store (cashier, manager, owner), and customer
     3. Goals: Have transaction added to system with total cost, customer name and phone number
     4. Preconditions: No transaction exists in database
     5. Summary: Once the purchased is added it can be found in the history of transactions for the store
     6. Related Use Cases: Check purchase history
     7. Steps:

|  |  |
| --- | --- |
| Step | Description of System Response |
| Choose new transaction command | Dialog opens for new purchase |
| Fill in attributes for the purchase | Systems asks for purchase ID, customerID, productID and quantity |
| Confirm information | Dialog closes if information confirmed as correct |

* + 1. Postcondition: Entry for purchase now exists for customer in store database



Cancel Purchase Entry

Enter Purchase into System

PurchaseID:

Quantity:

Customer ID:

New Purchase Entry

ProductID:

1. Entity-Relationship Diagram

Products

Serves



Employee



Purchases

Customer

1. Database Design
   1. Relations and attributes
      1. Customer(CustomerID, LastName, FirstName, phone)
      2. Product(ProductID, Name, Quantity, Price)
      3. Purchase(PurchaseID, customerID, ProductID, Quantity, Total)
   2. Keys
      1. Customer key: customerID
      2. Product key: ProductID
      3. Purchase key: PurchaseID
2. SQL Code for tables

CREATE TABLE "Customers" (

"CustomerID" INTEGER NOT NULL UNIQUE,

"LastName" TEXT,

"FirstName" TEXT,

"Phone" INTEGER,

PRIMARY KEY("CustomerID")

)

CREATE TABLE "Products" (

"ProductID" INTEGER NOT NULL UNIQUE,

"Name" TEXT,

"Quantity" INTEGER,

"Price" NUMERIC,

PRIMARY KEY("ProductID")

)

CREATE TABLE "Purchases" (

"PurchaseID" INTEGER NOT NULL UNIQUE,

"CustomerID" INTEGER,

"ProductID" INTEGER,

"Quantity" INTEGER,

"Total" NUMERIC,

PRIMARY KEY("PurchaseID")

)

1. SQL code for inserts: see database file for actual records

INSERT INTO "main"."Customers" ("CustomerID", "LastName", "FirstName", "Phone") VALUES ('1', 'Doe', 'John', '1234567890');

INSERT INTO "main"."Customers" ("CustomerID", "LastName", "FirstName", "Phone") VALUES ('2', 'Smith', 'Sara', '1234567891');

INSERT INTO "main"."Customers" ("CustomerID", "LastName", "FirstName", "Phone") VALUES ('3', 'Murray', 'Bill', '5555555555');

INSERT INTO "main"."Customers" ("CustomerID", "LastName", "FirstName", "Phone") VALUES ('4', 'Doe', 'Jack', '1234567892');

INSERT INTO "main"."Customers" ("CustomerID", "LastName", "FirstName", "Phone") VALUES ('5', 'Doe', 'Jill', '1234567895');

INSERT INTO "main"."Products" ("ProductID", "Name", "Quantity", "Price") VALUES ('1', 'Bananas', '6', '1.5');

INSERT INTO "main"."Products" ("ProductID", "Name", "Quantity", "Price") VALUES ('2', 'Water', '6', '2.5');

INSERT INTO "main"."Products" ("ProductID", "Name", "Quantity", "Price") VALUES ('3', 'Ground Beef', '18', '4.5');

INSERT INTO "main"."Products" ("ProductID", "Name", "Quantity", "Price") VALUES ('4', 'Potato', '48', '0.5');

INSERT INTO "main"."Products" ("ProductID", "Name", "Quantity", "Price") VALUES ('5', 'Chips', '9', '3.5');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('1', '1', '1', '2', '3.21');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('2', '1', '2', '2', '5.35');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('3', '2', '3', '1', '4.815');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('4', '2', '4', '1', '0.535');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('5', '3', '3', '1', '4.815');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('6', '3', '5', '1', '3.745');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('7', '4', '2', '2', '5.35');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('8', '4', '4', '1', '0.535');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('9', '5', '5', '5', '18.725');

INSERT INTO "main"."Purchases" ("PurchaseID", "CustomerID", "ProductID", "Quantity", "Total") VALUES ('10', '5', '1', '2', '3.21');