COPM 7700 Project1

This project implements the first version of the store management system with 3 user stories:

1. As a user, I want to add a new product into the system.
2. As a user, I want to add a new customer into the system.
3. As a user, I want to add a purchase from a customer into the system.

Tasks:

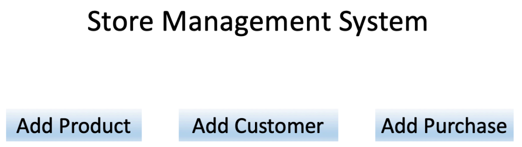
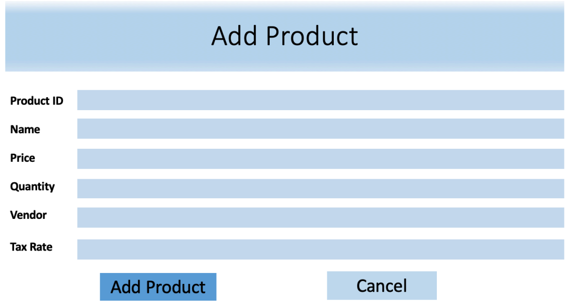
1. Write two possible use cases for each user story: one is the common case and one is the exception.

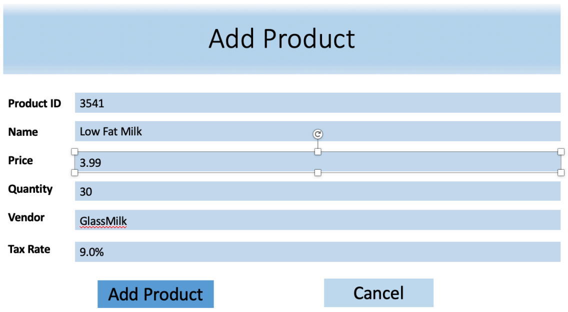
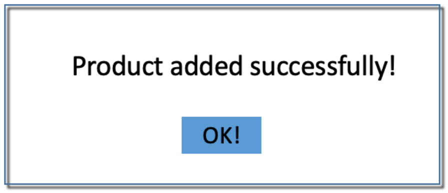
a. valid case1: add a new product with valid information of product successfully.
   invalid case: add a new product with invalid information, like with empty product ID.
b. valid case1: add a new customer with valid information of customer successfully.
   invalid case: add a new customer with invalid information, like with empty customer ID.
c. valid case1: add a new purchase with valid information of purchase successfully.
   invalid case: add a new purchase with invalid information, like with a non-existed product ID.

2. Design the screens (UI windows and widgets) the system should display in each use case.

- use case a:
    - As a user, I want to add a new product into the system. (common case)

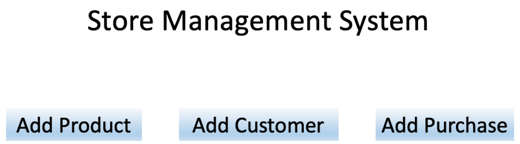
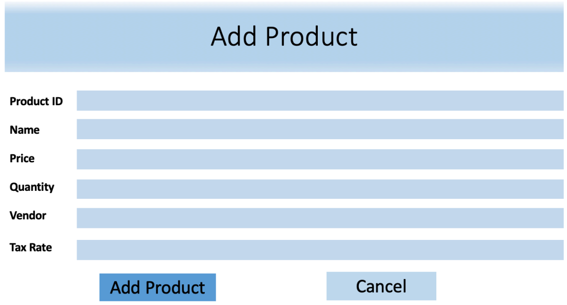
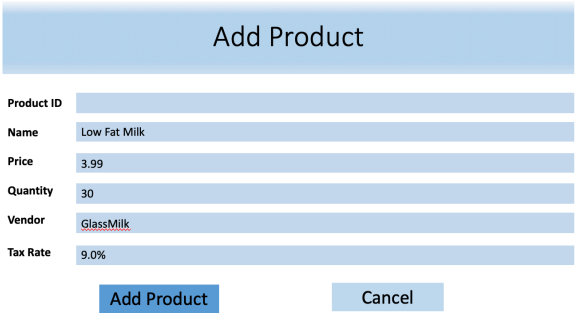
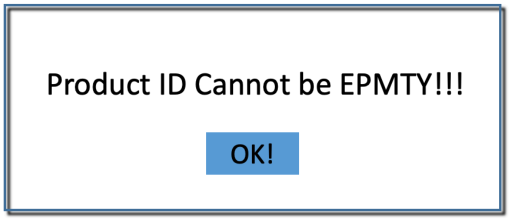
Steps:

Actor	System
1. Click button "Add Product" Main screen  Store Management System Add Product Add Customer Add Purchase	2. Display "Add Product" Screen "Add Product" screen  Add Product Product ID Name Price Quantity Vendor Tax Rate Add Product Cancel :
3. Input data then click button "Add Product" "Add Product" screen with input data:	4. Hide "Add Product" screen and display an alert "Product added successfully!"

 Add Product Product ID 3541 Name Low Fat Milk Price 3.99 Quantity 30 Vendor GlassMilk Tax Rate 9.0% 	 Product added successfully!
5. Click “OK”	6. Display Main screen

- As a user, I want to add a new product without product ID into the system. (Exception)

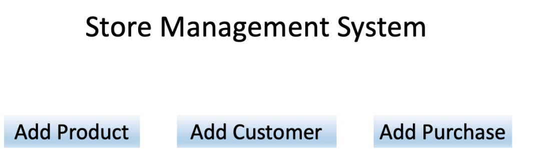
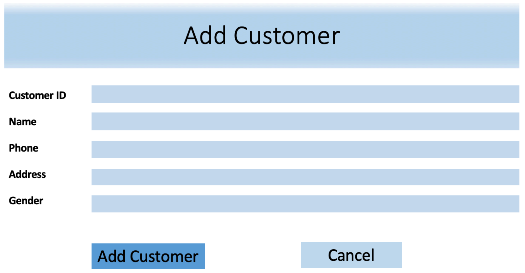
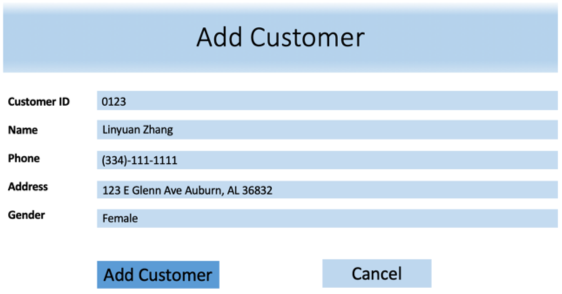
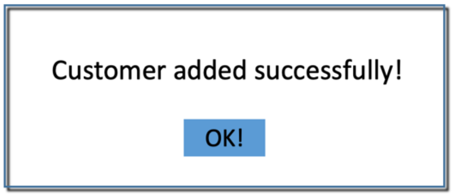
steps:

Actor	System
1. Click button “Add Product” Main screen  Store Management System 	2. Display “Add Product” Screen “Add Product” screen  Add Product Product ID Name Price Quantity Vendor Tax Rate :
3. Input data then click button “Add Product” “Add Product” screen with input data:  Add Product Product ID Name Low Fat Milk Price 3.99 Quantity 30 Vendor GlassMilk Tax Rate 9.0% 	4. Hide “Add Product” screen and display an alert with error message “Product ID Cannot be EMPTY!!!”  Product ID Cannot be EPMTY!!!
5. Click “OK”	6. Display Main screen

- use case 2:

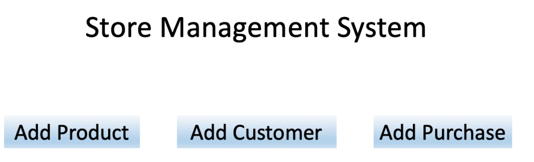
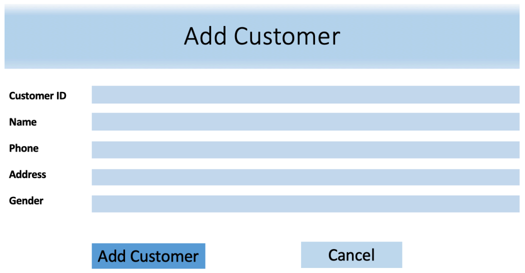
- As a user, I want to add a new customer into the system. (Common case)

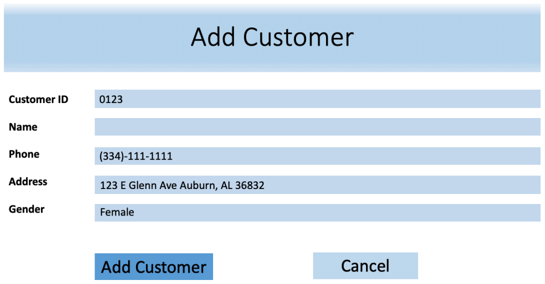
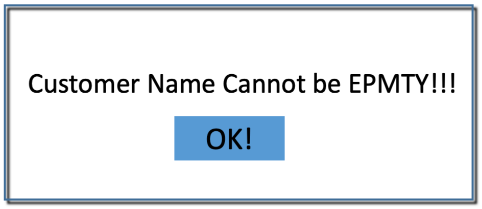
Steps:

Actor	System
1. Click button “Add Customer” Main screen  Store Management System Add Product Add Customer Add Purchase	2. Display “Add Product” Screen “Add Customer” screen  Add Customer Customer ID Name Phone Address Gender Add Customer Cancel :
3. Input data then click button “Add Customer” “Add Customer” screen with input data:  Customer ID 0123 Name Linyuan Zhang Phone (334)-111-1111 Address 123 E Glenn Ave Auburn, AL 36832 Gender Female 	4. Hide “Add Customer” screen and display an alert “Customer added successfully!”  Customer added successfully!
5. Click “OK”	6. Display Main screen

- As a user, I want to add a new customer without customer’s name into the system. (Exception)

Steps:

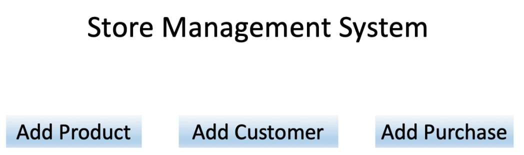
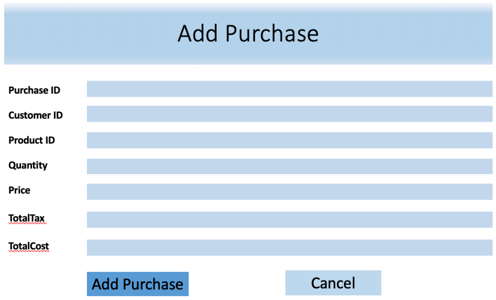
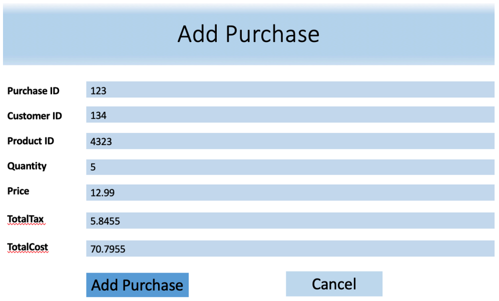
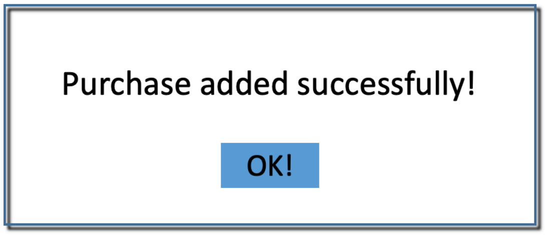
Actor	System
1. Click button “Add Customer” Main screen  Store Management System Add Product Add Customer Add Purchase	2. Display “Add Product” Screen “Add Customer” screen  Add Customer Customer ID Name Phone Address Gender Add Customer Cancel :
3. Input data then click button “Add Customer” “Add Customer” screen with input data:	4. Hide “Add Customer” screen and display an alert with the error message “Customer Name Cannot be EMPTY!!!”

 Customer ID 0123 Name Phone (334)-111-1111 Address 123 E Glenn Ave Auburn, AL 36832 Gender Female 	 Customer Name Cannot be EPMTY!!!
5. Click “OK”	6. Display Main screen

- use case 3:

- As a user, I want to record a purchase of a customer into the system. (Common case)

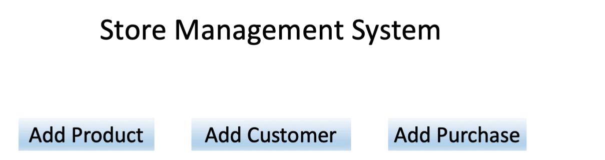
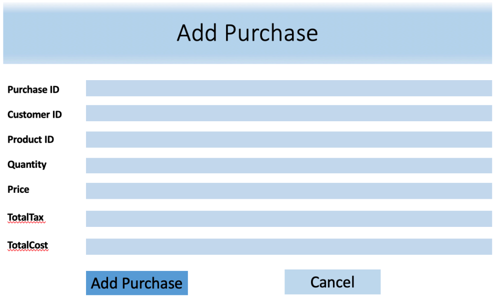
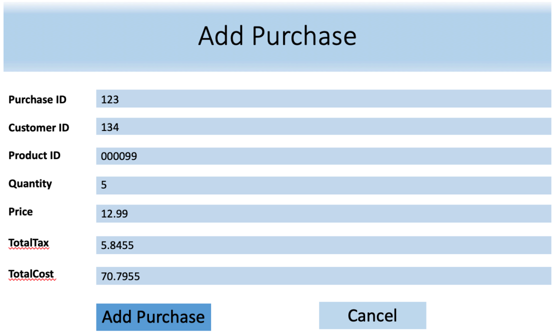
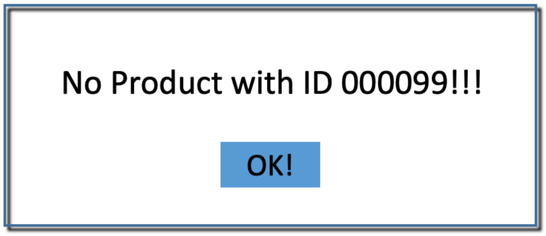
Steps:

Actor	System
1. Click button “Add Purchase” Main screen  	2. Display “Add Purchase” Screen “Add Purchase” screen:  Purchase ID Customer ID Product ID Quantity Price TotalTax TotalCost
3. Input data then click button “Add Purchase” “Add Purchase” screen with input data:  Purchase ID 123 Customer ID 134 Product ID 4323 Quantity 5 Price 12.99 TotalTax 5.8455 TotalCost 70.7955 	4. Hide “Add Purchase” screen and display an alert “Purchase added successfully!”  Purchase added successfully!
5. Click “OK”	6. Display Main screen

- As a user, I want to record a purchase of a customer with a non-existed product ID into the system. (Exception)

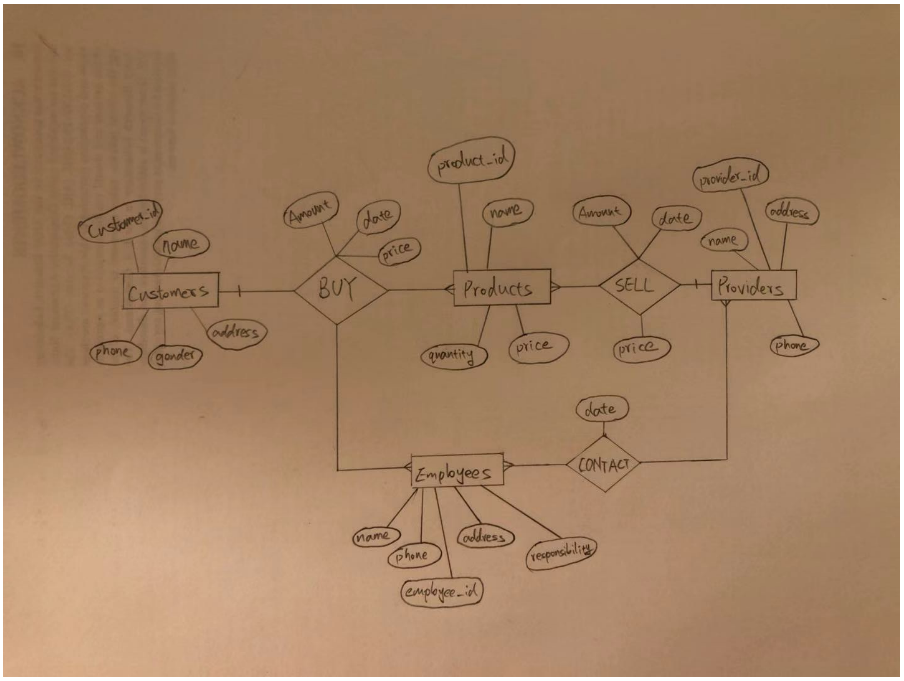
steps:

Actor	System

1. Click button “Add Purchase” Main screen  Store Management System 	2. Display “Add Purchase” Screen “Add Purchase” screen:  Add Purchase Purchase ID Customer ID Product ID Quantity Price TotalTax TotalCost
3. Input data then click button “Add Purchase” “Add Purchase” screen with input data:  Add Purchase Purchase ID 123 Customer ID 134 Product ID 000099 Quantity 5 Price 12.99 TotalTax 5.8455 TotalCost 70.7955 	4. Hide “Add Purchase” screen and display an alert with an error message “No Product with ID 000099!!!”  No Product with ID 000099!!!
5. Click “OK”	6. Display Main screen

3. Design the database physically and prepare data for the tables, with at least 5 products, 5 customers, and 10 purchases.

The entity-relationship diagram for this system is shown in the following figure.



The design the database logically, i.e., write the relations, attributes, and defined keys are listed in the following.

The attributes, and defined keys of each entity are listed in the following.

Customers: name, address, phone, customer_ID (Primary Key), gender;

Products: name, product_ID (Primary Key), quantity, price;

Providers: name, address, phone, provider_ID (Primary Key);

Employees: name, address, phone, employee_ID, responsibility;

Purchase: OrdersID (Primary Key), CustomerID (Foreign Key), ProductID (Foreign Key), Quantity, Price, TotalTax, TotalCost

The relationships between each entity in this system are shown in the following.

Customers BUY Products
Providers SELL Products
Employees contact Providers

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

The create tables' SQL codes are shown in the following.

```
CREATE TABLE "Customers" (
        "CustomerID"  INTEGER NOT NULL,
        "Name"        TEXT DEFAULT 'Guest',
        "Phone"       TEXT DEFAULT '(334)111-1111',
        "Address"     TEXT DEFAULT 'Auburn, AL',
        "Gender"      TEXT,
        PRIMARY KEY("CustomerID")
)

CREATE TABLE "Products" (
        "ProductID"     INTEGER NOT NULL,
        "Name"          TEXT,
        "Price" REAL,
        "Quantity"      REAL,
        "Vendor"        TEXT,
        "TaxRate"       REAL DEFAULT 0.9,
        PRIMARY KEY("ProductID")
);

CREATE TABLE "Providers" (
        "ProviderID"    INTEGER NOT NULL,
        "Name"          TEXT,
        "Address"       TEXT,
        "Phone"         INTEGER,
        PRIMARY KEY("ProviderID")
)

CREATE TABLE "Orders" (
        "OrderID"       INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
        "CustomerID"  INTEGER NOT NULL,
        "ProductID"     INTEGER NOT NULL,
        "Price" REAL,
        "Quantity"      REAL DEFAULT 1,
        "TotalRate"     REAL,
        "TotalCost"     REAL,
        FOREIGN KEY("ProductID") REFERENCES "Products"("ProductID"),
        FOREIGN KEY("CustomerID") REFERENCES "Customers"("CustomerID")
);
```

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

-Input the data to Customers table:

```
INSERT INTO "main"."Customers" ("CustomerID", "Name", "Phone", "Address", "Gender")
VALUES ('1', 'Zhang', '(334)111-1111', 'Auburn, AL', '');
INSERT INTO "main"."Customers" ("CustomerID", "Name", "Phone", "Address", "Gender")
VALUES ('2', 'Linyuan', '', '', '');
INSERT INTO "main"."Customers" ("CustomerID", "Name", "Phone", "Address", "Gender")
VALUES ('3', 'Helen', '', 'Glenn', '');
INSERT INTO "main"."Customers" ("CustomerID", "Name", "Phone", "Address", "Gender")
VALUES ('4', 'Anna', '(334)111-1234', 'Opelika, AL', 'Female');
```

```
INSERT INTO "main"."Customers" ("CustomerID", "Name", "Phone", "Address", "Gender")
VALUES ('5', 'Chris', '(334)111-4321', 'Harper', 'Male');
INSERT INTO "main"."Customers" ("CustomerID", "Name", "Phone", "Address", "Gender")
VALUES ('6', 'Evens', '(334)111-3399', '160 Ross', 'Male');
```

- Input the data to Products table

```
INSERT INTO "main"."Products" ("ProductID", "Name", "Price", "Quantity", "Vendor",
"TaxRate") VALUES ('1', 'Low Fat Milk', '3.99', '20.0', 'GLASSMILK', '0.9');
INSERT INTO "main"."Products" ("ProductID", "Name", "Price", "Quantity", "Vendor",
"TaxRate") VALUES ('2', 'Dr. Pepper', '4.99', '30.0', 'Coca-Cola Ltd', '0.9');
INSERT INTO "main"."Products" ("ProductID", "Name", "Price", "Quantity", "Vendor",
"TaxRate") VALUES ('3', 'Black Cherry', '4.99', '50.0', 'KOR Farm', '0.9');
INSERT INTO "main"."Products" ("ProductID", "Name", "Price", "Quantity", "Vendor",
"TaxRate") VALUES ('4', 'Bell Pepper', '0.99', '25.0', 'KOR Farm', '0.9');
INSERT INTO "main"."Products" ("ProductID", "Name", "Price", "Quantity", "Vendor",
"TaxRate") VALUES ('5', 'Orange', '3.19', '40.0', 'Florida Farm', '0.9');
INSERT INTO "main"."Products" ("ProductID", "Name", "Price", "Quantity", "Vendor",
"TaxRate") VALUES ('6', 'Pure water', '1.99', '20.0', 'DeerPark', '0.9');
```

-Input the data to Orders table

```
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('1', '3', '2', '4.0', '4.99', '', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('2', '3', '4', '4.0', '0.99', '', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('3', '2', '1', '2.0', '3.99', '', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('4', '5', '5', '3.19', '1.0', '', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('5', '4', '6', '1.99', '10.0', '', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('6', '4', '4', '0.99', '20.0', '', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('7', '2', '3', '4.99', '1.0', '', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('8', '1', '2', '4.99', '10.0', '0.5', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('9', '6', '1', '3.99', '1.0', '', '');
INSERT INTO "main"."Orders" ("OrderID", "CustomerID", "ProductID", "Price", "Quantity",
"TotalRate", "TotalCost") VALUES ('10', '5', '4', '0.99', '10.0', '', '');
```

- Input the data to Providers table

```
INSERT INTO "main"."Providers" ("ProviderID", "Name", "Address", "Phone") VALUES ('1', 'Coca-
Cola Ltd', '1775 Donahue Dr', '(334)-123-1111');
```

```
INSERT INTO "main"."Providers" ("ProviderID", "Name", "Address", "Phone") VALUES ('2',
'DeerPark', '531 E Glenn', '(334)-987-0001');
INSERT INTO "main"."Providers" ("ProviderID", "Name", "Address", "Phone") VALUES ('3',
'Florida Farm', '874 Cury Drive', '(334)-342-1111');
INSERT INTO "main"."Providers" ("ProviderID", "Name", "Address", "Phone") VALUES ('4', 'KOR
Farm', '', '(334)-000-1799');
INSERT INTO "main"."Providers" ("ProviderID", "Name", "Address", "Phone") VALUES ('5',
'GLASSMILK', '', '(334)-7987-1234');
```