We now implement 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:

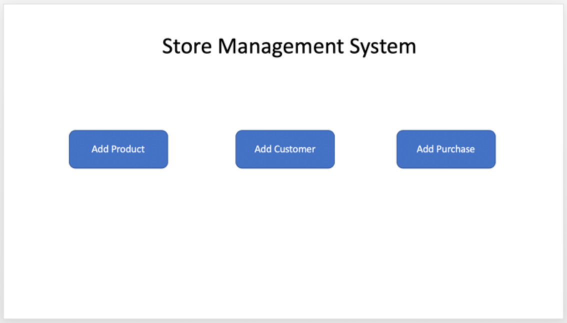
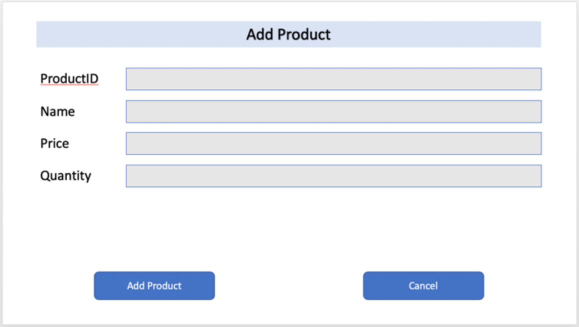
1. Write two possible use cases for each user story: one is the common case and one is the exception. Design the screens (UI windows and widgets) the system should display in each use case.

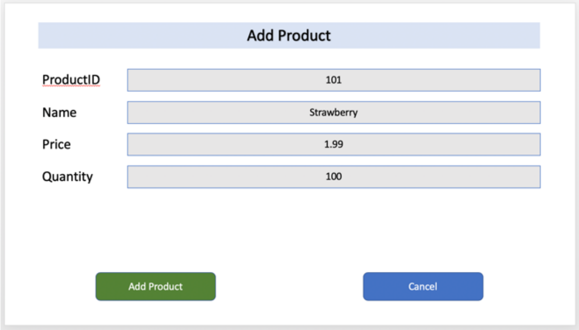
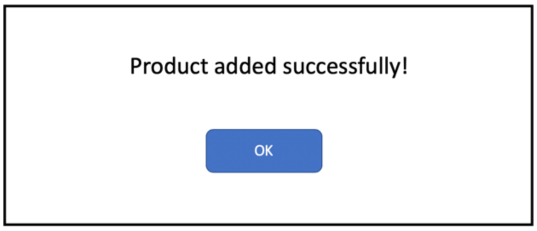
Common case:

a). - As a user, I want to add a new product into the system.

A. Name: Adding a new product
B. Actors: store manager or employee
C. Goals: to add a new product into the system
D. Preconditions: the administrator has logged in the system and entered into admin interface.
E. Summary: the user chooses 'Add product' command, and then type the product information into all the input box. After that, the user confirms the content he entered in. Finally, the new product will be added into the system.
F. Related use cases:
Adding a new customer
Adding a purchase

G. 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 ProductID Name Price Quantity Add Product Cancel

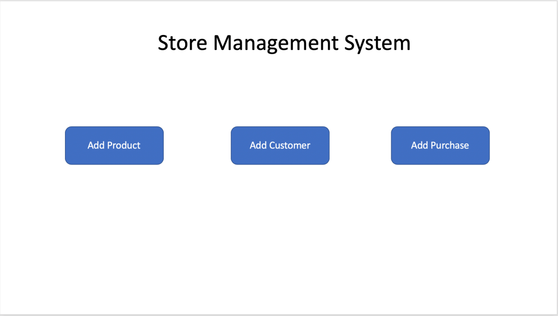
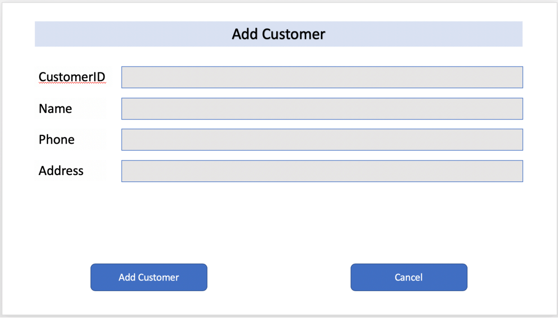
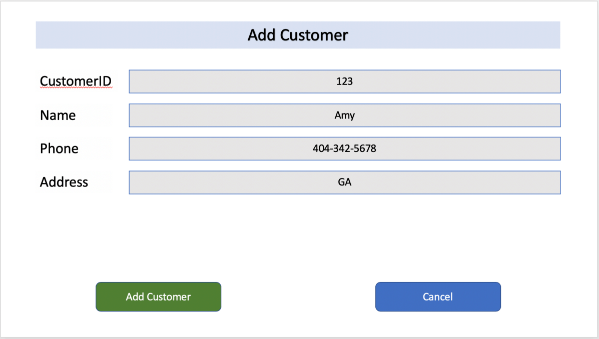
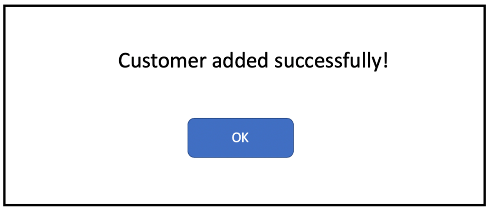
3. input data then click button "Add Product" "Add Product" screen with input data:  Add Product ProductID 101 Name Strawberry Price 1.99 Quantity 100 	4. Hide "Add Product" screen and display an alert "Product added successfully!" The "Product added successfully!" alert:  Product added successfully!
5. Click "OK"	6. Display main screen

H. Postconditions: the new product will be added into the system and then back to main screen.

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

A. Name: Adding a new customer
B. Actors: store manager or employee
C. Goals: to add a new customer into the system
D. Preconditions: the administrator has logged in the system and entered into admin interface.
E. Summary: the user chooses 'Add Customer' command, and then types the customer information into all the input box. After that, the user confirms the content he entered in. Finally, the new customer will be added into the system.
F. Related use cases:
Adding a new product
Adding a purchase

G. Steps:

Actor	System
1. Click button “Add Customer” Main screen:  Store Management System Add Product Add Customer Add Purchase	2. Display “Add Customer” screen “Add Customer” screen:  Add Customer CustomerID Name Phone Address Add Customer Cancel
3. input data then click button “Add Customer” “Add Customer” screen with input data:  Add Customer CustomerID 123 Name Amy Phone 404-342-5678 Address GA 	4. Hide “Add Customer” screen and display an alert “Customer added successfully!” The “Customer added successfully!” alert:  Customer added successfully!
5. Click “OK”	6. Display main screen

H. Postconditions: the new customer will be added into the system and then back to main screen.

c). - As a user, I want to record a purchase from a customer into the system.

A. Name: Adding a new purchase order
B. Actors: store manager or employee
C. Goals: to add a new purchase into the system

D. Preconditions: the administrator has logged in the system and entered into admin interface.

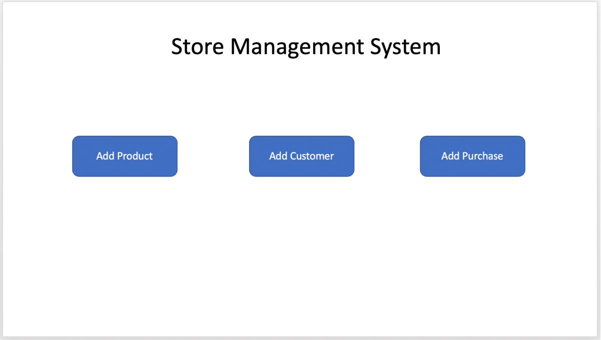
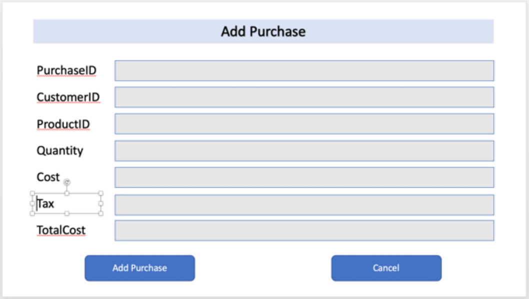
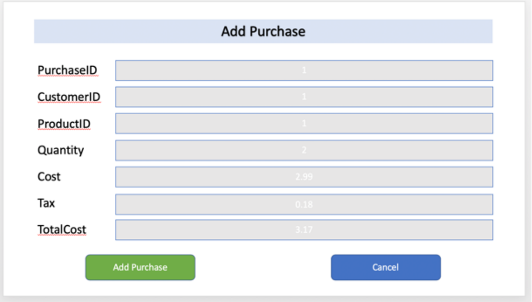
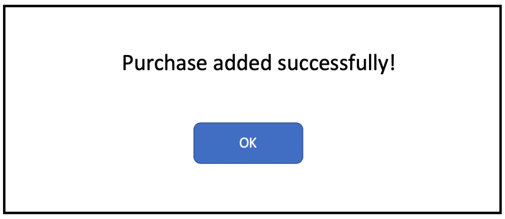
E. Summary: the user chooses ‘Add Purchase’ command, and then types the purchase order information into all the input box. After that, the user confirms the content he entered in. Finally, a new purchase will be added into the system.

F. Related use cases:

Adding a new product

Adding a new customer

G. Steps:

Actor	System
1. Click button “Add Purchase” Main screen:  Store Management System Add Product Add Customer Add Purchase	2. Display “Add Purchase” screen “Add Purchase” screen:  Add Purchase PurchaseID CustomerID ProductID Quantity Cost Tax TotalCost Add Purchase Cancel
3. input data then click button “Add Purchase” “Add Purchase” screen with input data:  Add Purchase PurchaseID: 1 CustomerID: 1 ProductID: 1 Quantity: 2 Cost: 2.99 Tax: 0.18 TotalCost: 3.17 Add Purchase Cancel	4. Hide “Add Purchase” screen and display an alert “Purchase added successfully!” The “Purchase added successfully!” alert:  Purchase added successfully! OK
5. Click “OK”	6. Display main screen

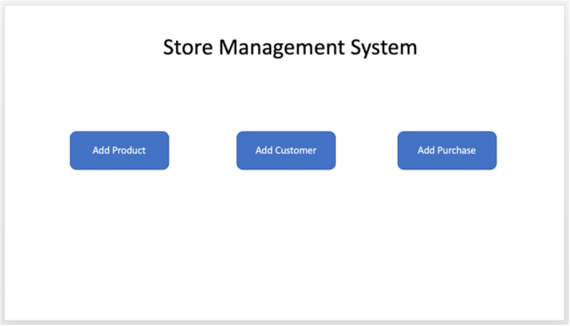
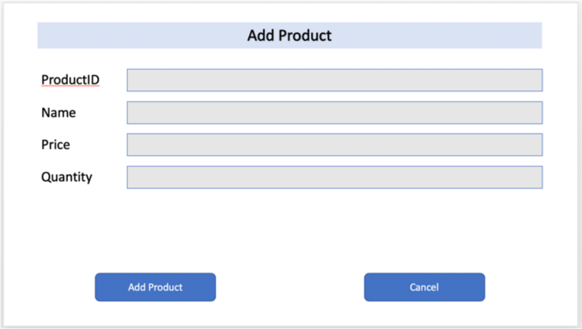
H. Postconditions: the new purchase order will be added into the system.

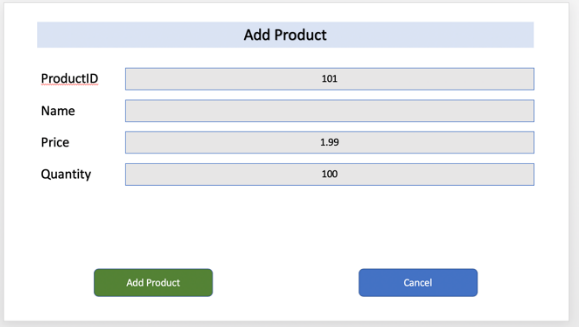
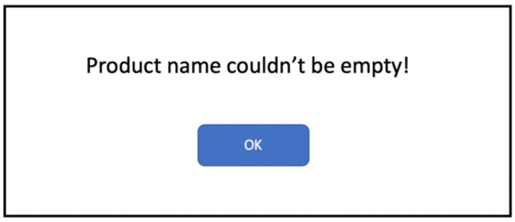
Exception case:

a). - As a user, I want to add a new product into the system.

A. Name: Adding a new product
B. Actors: store manager or employee
C. Goals: to add a new product without name into the system
D. Preconditions: the administrator has logged in the system and entered into admin interface.
E. Summary: the user chooses ‘Add product’ command, and then type the product information into the input box without name. After that, the system will prevent the user from adding the product and mention the user’s mistake.
F. Related use cases:
Adding a new customer
Adding a purchase

G. 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 ProductID Name Price Quantity Add Product Cancel
3. input data then click button “Add Product” “Add Product” screen with input data:	4. error message prompted:

 Add Product ProductID 101 Name Price 1.99 Quantity 100 	 Product name couldn’t be empty!
5. Click “OK”	6. Display main screen

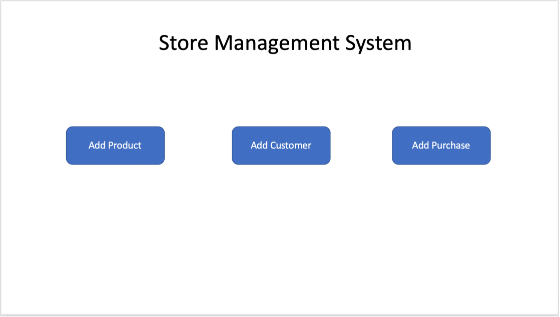
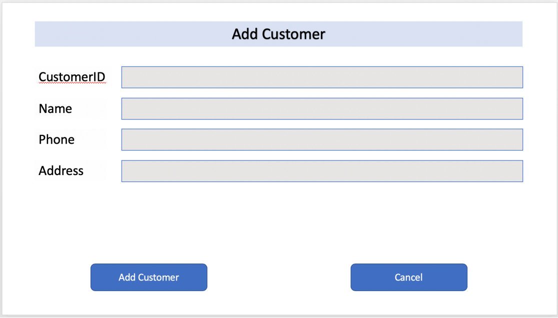
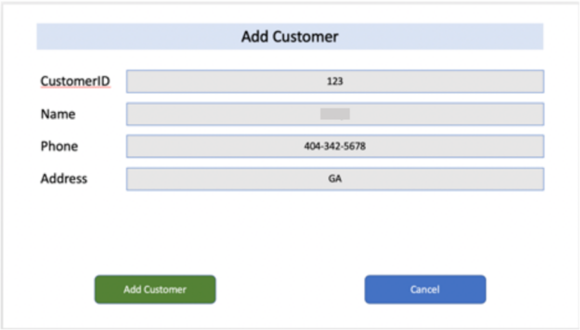
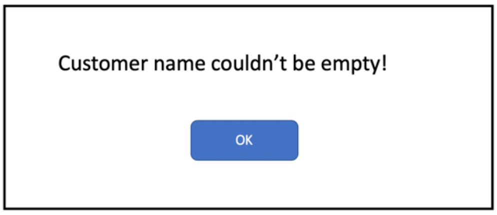
H. Postconditions: the new product will be refused to be added to the database.

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

A. Name: Adding a new customer
B. Actors: store manager or employee
C. Goals: to add a new customer without name into the system
D. Preconditions: the administrator has logged in the system and entered into admin interface.
E. Summary: the user chooses ‘Add Customer’ command, and then types the customer information into all the input box. After that, the system will prevent the user from adding the customer and mention the user’s mistake.
F. Related use cases:
Adding a new product
Adding a purchase

G. Steps:

Actor	System
1. Click button “Add Customer”	2. Display “Add Customer” screen

Main screen:  Store Management System Add Product Add Customer Add Purchase	“Add Customer” screen:  Add Customer CustomerID Name Phone Address Add Customer Cancel
3. input data then click button “Add Customer” “Add Customer” screen with input data:  CustomerID 123 Name Phone 404-342-5678 Address GA	4. error message prompted:  Customer name couldn’t be empty!
5. Click “OK”	6. Display main screen

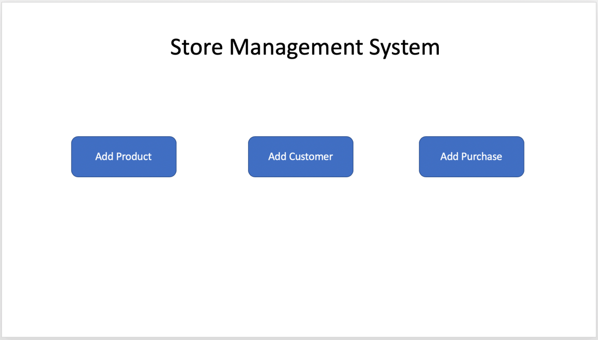
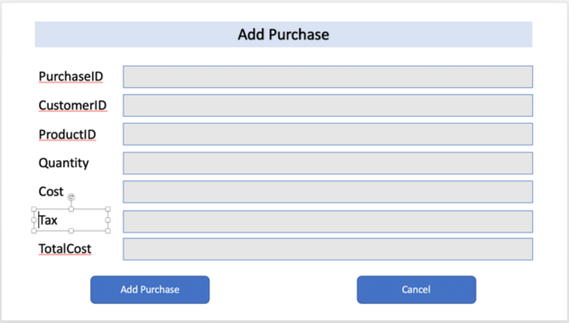
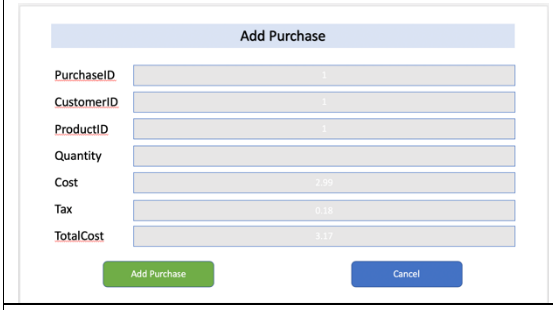
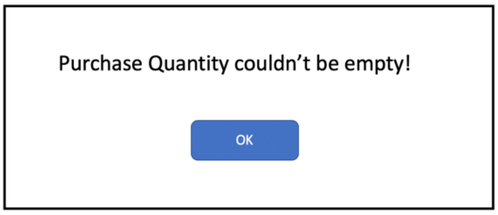
H. Postconditions: the new customer will be refused to be added to the database.

c). - As a user, I want to record a purchase from a customer into the system.

A. Name: Adding a new purchase order
B. Actors: store manager or employee
C. Goals: to add a new purchase without quantity into the system
D. Preconditions: the administrator has logged in the system and entered into admin interface.
E. Summary: the user chooses ‘Add Purchase’ command, and then types the purchase order information into all the input box. After that, the system will prevent the user from adding the purchase and mention the user’s mistake.

F. Related use cases:
Adding a new product
Adding a new customer

G. Steps:

Actor	System
1. Click button “Add Purchase” Main screen:  Store Management System Add Product Add Customer Add Purchase	2. Display “Add Purchase” screen “Add Purchase” screen:  Add Purchase PurchaseID CustomerID ProductID Quantity Cost Tax TotalCost Add Purchase Cancel
3. input data then click button “Add Purchase” “Add Purchase” screen with input data:  Add Purchase PurchaseID 1 CustomerID 1 ProductID 1 Quantity Cost 2.99 Tax 0.18 TotalCost 3.17 	4. error message prompted:  Purchase Quantity couldn’t be empty!
5. Click “OK”	6. Display main screen

H. Postconditions: the new purchase will be refused to be added to the database.

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

Customers(CustomerID*, Name, Phone, Address)

Products(ProductID*, Name, Price, Quantity)

Purchase(PurchaseID*, CustomerID, ProductID, Quantity, Cost, Tax, TotalCost)

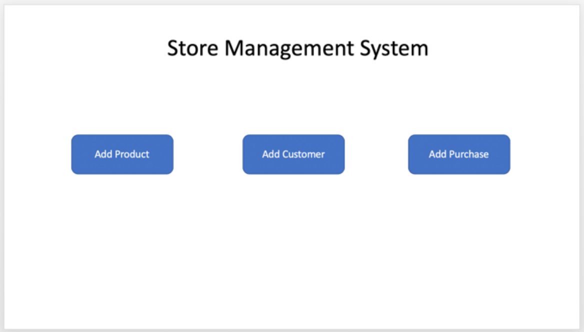
```
CREATE TABLE "Customers" (

    "CustomerID"        INTEGER,

    "Name"      TEXT NOT NULL,

    "Phone"     TEXT DEFAULT '111-111-1111',

    "Address"   TEXT DEFAULT 'US',

    PRIMARY KEY("CustomerID")

)


CREATE TABLE "Products" (

    "ProductID" INTEGER,

    "Name"      TEXT NOT NULL,

    "Price"     REAL NOT NULL,

    "Quantity"  REAL NOT NULL,

    PRIMARY KEY("ProductID")

)
```

```
CREATE TABLE "Purchase" (

    "PurchaseID"        INTEGER,

    "CustomerID"        INTEGER NOT NULL,

    "ProductID" INTEGER NOT NULL,

    "Quantity"  INTEGER NOT NULL,

    "Cost"      REAL NOT NULL,

    "Tax"  REAL NOT NULL,

    "TotalCost" REAL NOT NULL,

    PRIMARY KEY("PurchaseID")

)


INSERT INTO Customers
VALUES
(1, 'Amada', '334-898-9034', 'AL'),
(2, 'Emy', '404-890-6734', 'GA'),
(3, 'Angel', '212-234-7832', 'NY'),
(4, 'Jackson', '310-789-2839', 'LA'),
(5, 'Fiona', '415-908-2345', 'CA');


INSERT INTO Products
VALUES
(1,'Apple',0.99,300),
(2,'Noodles',1.32,500),
(3,'Pineapple',2.32,200),
(4,'Light',8.92,400),
(5,'Sunglass',23.40,50);

INSERT INTO Purchase
VALUES
(1,1,1,2,1.98,0.18,2.16),
(2,2,2,5,6.00,0.59,7.19),
(3,3,1,2,1.98,0.18,2.16),
(4,4,3,3,6.96,0.63,7.59),
(5,5,4, 4,35.68,1.78,37.46),
(6,6,2,5,6.60,0.59,7.19),
(7,7,5,1,23.50,1.18,24.68),
(8,8,3,3,6.96,0.63,7.59),
(9,9,5,1,23.5,1.18,24.68),
(10,10,2,5,6.60,0.59,7.19);
```

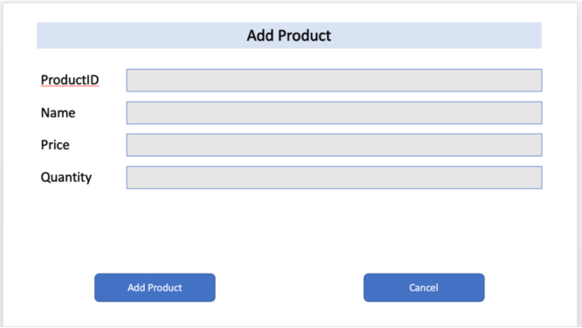
3. MVC

1. User Interface
Different views:

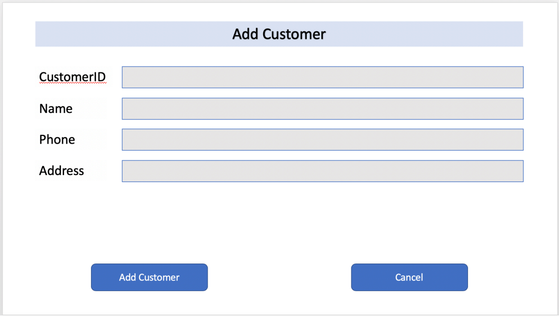
MainView class



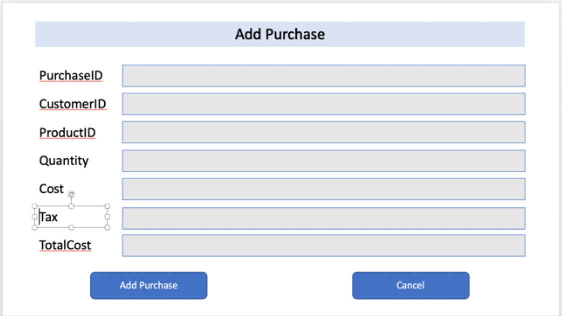
AddProductView class



AddCustomerView class



AddPurchaseView class



2. Business Logic

<u>Data models for products, orders, customers</u>

Class ProductModel: store a product object(in memory)
Class CustomerModel: store a customer object(in memory)
Class OrderModel: store a order object(in memory)

<u>Controllers for UI views</u>

Class AddProductUI: process user interactions
Class AddCustomerUI: process user interactions
Class AddOrderUI: process user interactions

<u>Main application (setup data access, create views)</u>

Class StoreManager

3. Data Access

SQLiteDataAdapter class: load/save products, customers, purchases data…