**Fall 2019 COMP 7706 Software Architecture**

**Project 1**

1. Write two possible use cases for each user story: one is the common case and one is the exception.
   1. As a user, I want to add a new product into the system.

|  |  |
| --- | --- |
| Actor actions | System Response |
| 1. Click the “Add Product” button on the main menu. | 2. A window will appear displaying the “Add Product” form. |
| 3. Complete the “Add Product” form with the desired product information. Then click the “Add Product” button. | 4. The information provided by the user will be used to create a new product entry in the database. Display a popup informing the user that the action was successful. |
| 5. Click “Ok” on the confirmation window. |  |

|  |  |
| --- | --- |
| Actor actions | System Response |
| 1. Click the “Add Product” button on the main menu. | 2. A window will appear displaying the “Add Product” form. |
| 3. Do not complete the “Add Product” form. Click the “Add Product” button. | 4. The appropriate exceptions should be thrown alerting the user of the invalid form. |
| 5. Click “Ok” on the confirmation window. |  |

1. As a user, I want to add a new customer into the system.

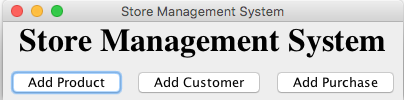
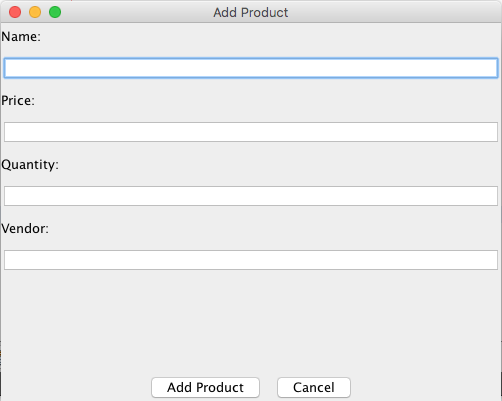
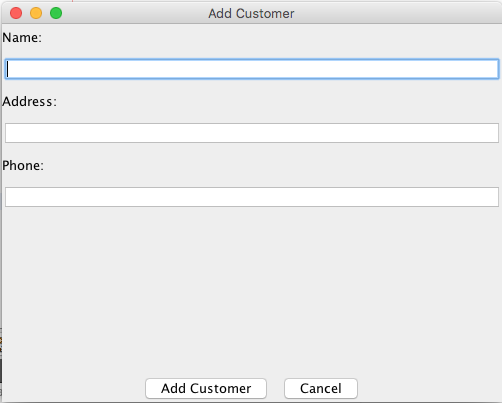
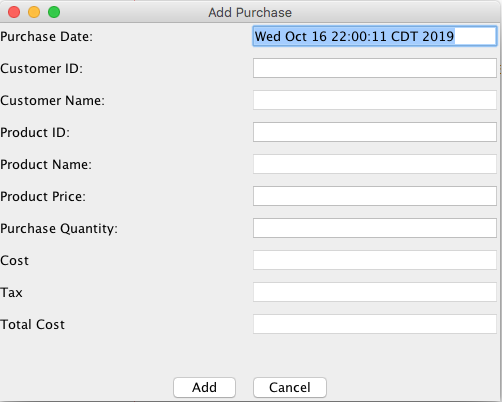
|  |  |
| --- | --- |
| Actor actions | System Response |
| 1. Click the “Add Customer” button on the main menu. | 2. A window will appear displaying the “Add Customer” form. |
| 3. Complete the “Add Customer” form with the desired product information. Then click the “Add Customer” button. | 4. The information provided by the user will be used to create a new customer entry in the database. Display a popup informing the user that the action was successful. |
| 5. Click “Ok” on the confirmation window. |  |

|  |  |
| --- | --- |
| Actor actions | System Response |
| 1. Click the “Add Customer” button on the main menu. | 2. A window will appear displaying the “Add Customer” form. |
| 3. Do not complete the “Add Customer” form. Click the “Add Customer” button. | 4. The appropriate exceptions should be thrown alerting the user of the invalid form. |
| 5. Click “Ok” on the confirmation window. |  |

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

|  |  |
| --- | --- |
| Actor actions | System Response |
| 1. Click the “Add Purchase” button on the main menu. | 2. A window will appear displaying the “Add Purchase” form. |
| 3. Complete the “Add Purchase” form with the desired product information. Then click the “Add” button. | 4. The information provided by the user will be used to create a new purchase entry in the database. Display a popup informing the user that the action was successful. |
| 5. Click “Ok” on the confirmation window. |  |

|  |  |
| --- | --- |
| Actor actions | System Response |
| 1. Click the “Add Purchase” button on the main menu. | 2. A window will appear displaying the “Add Purchase” form. |
| 3. Do not complete the “Add Purchase” form. Click the “Add” button. | 4. The appropriate exceptions should be thrown alerting the user of the invalid form. |
| 5. Click “Ok” on the confirmation window. |  |

1. Design the screens (UI windows and widgets) the system should display in each use case.
   1. Initial Main Menu  
      
   2. Add Product Form  
      
   3. Add Customer Form  
      
   4. Add Purchase Form  
      
2. Design the database physically and prepare data for the tables, with at least 5 products, 5 customers, and 10 purchases.
   1. Customer(**customer\_id**, name, address)

﻿CREATE TABLE "Customer" (

"customer\_id" INTEGER NOT NULL DEFAULT 1 PRIMARY KEY AUTOINCREMENT UNIQUE,

"name" TEXT NOT NULL,

"phone" TEXT,

"address" TEXT NOT NULL

);

* 1. Product(**product\_id**, name, price, quantity, vendor)

﻿CREATE TABLE "Product" (

"product\_id" INTEGER NOT NULL DEFAULT 1 PRIMARY KEY AUTOINCREMENT UNIQUE,

"name" TEXT NOT NULL,

"price" NUMERIC NOT NULL,

"quantity" INTEGER NOT NULL,

"vendor" TEXT NOT NULL

);

* 1. Purchase(**purchase\_id**, customer\_id, product\_id, cost, tax, total\_cost, price, quantity, date)  
     ﻿CREATE TABLE "Purchase" (

"purchase\_id" INTEGER NOT NULL UNIQUE,

"customer\_id" INTEGER NOT NULL,

"product\_id" INTEGER NOT NULL,

"cost" REAL NOT NULL,

"tax" REAL NOT NULL,

"total\_cost" REAL NOT NULL,

"price" REAL NOT NULL,

"quantity" REAL NOT NULL,

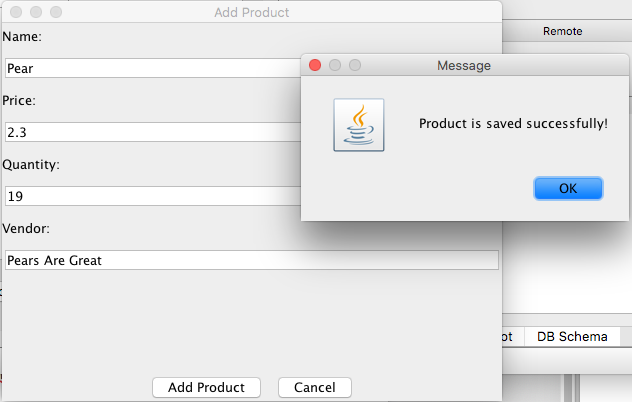
"date" TEXT NOT NULL,

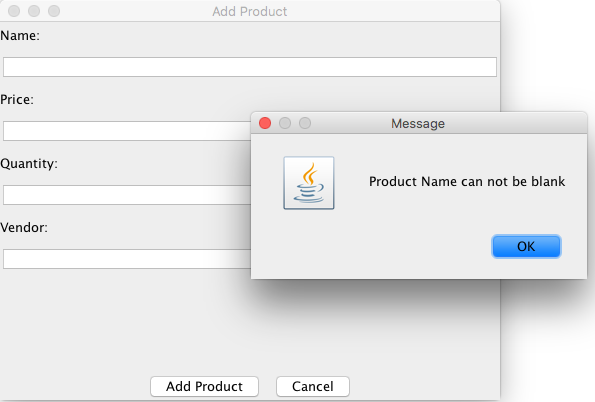
FOREIGN KEY("product\_id") REFERENCES "Product"("product\_id"),

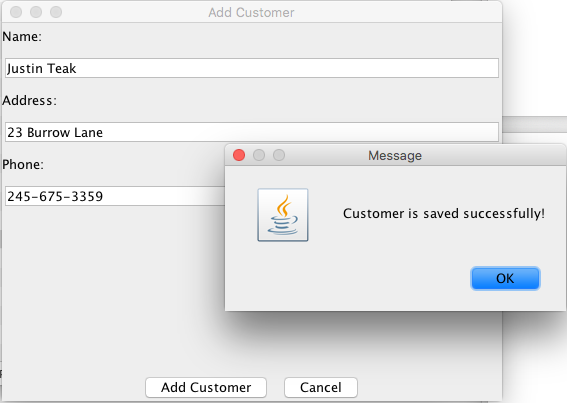
FOREIGN KEY("customer\_id") REFERENCES "Customer"("customer\_id"),

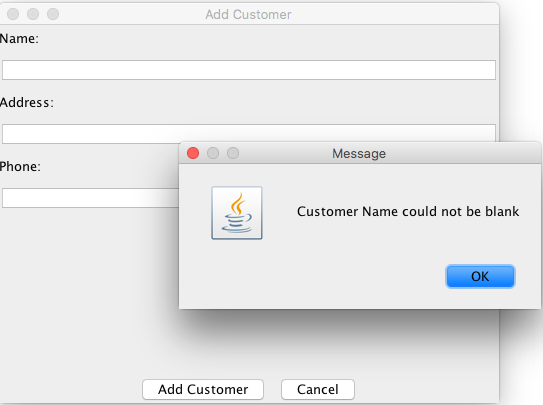
PRIMARY KEY("purchase\_id")

);

1. Implement the use cases.
   1. See Java code
2. Test the system with each use case.
   1. As a user, I want to add a new product into the system.  
      



* 1. As a user, I want to add a new customer into the system.  
     



* 1. As a user, I want to add a purchase from a customer into the system.  
     