

Project 1

AI-TE KUO

Oct 09, 2019

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

(a) Product

- i. Common case - When a new product is arrived, the system shall be able to provide a functionality that the user can add a new product to the system.

1. The user clicks on the product tab and the system shall display the add product page.

The screenshot shows a web application window titled 'Store Management System'. It has three tabs: 'Product', 'Customer', and 'Purchase'. The 'Product' tab is active. The form contains the following fields and buttons:

- ProductID:
- ProductName:
- ProductPrice:
- ProductQuantity:
- Buttons:

2. The user inputs all the product details.

The screenshot shows the same 'Store Management System' window with the 'Product' tab active. The form fields are now filled with the following data:

- ProductID: 9999
- ProductName: Pressure Washer
- ProductPrice: 99.99
- ProductQuantity: 10
- Buttons:

3. The user clicks on "Confirm" button and the system shall pop out the message showing the product has been successfully added

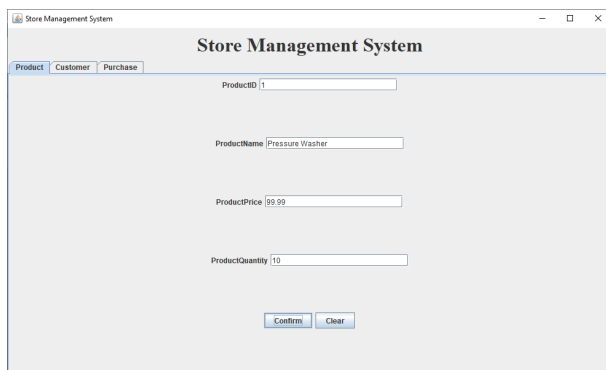
The screenshot shows a 'Message' dialog box with a close button (X) in the top right corner. It contains an information icon (i) and the following text:

Product is saved successfully!
Product ID: 9999
Product Name: Pressure Washer
Unit Price: 99.990000
StockQuantity: 10.000000

At the bottom of the dialog box is an

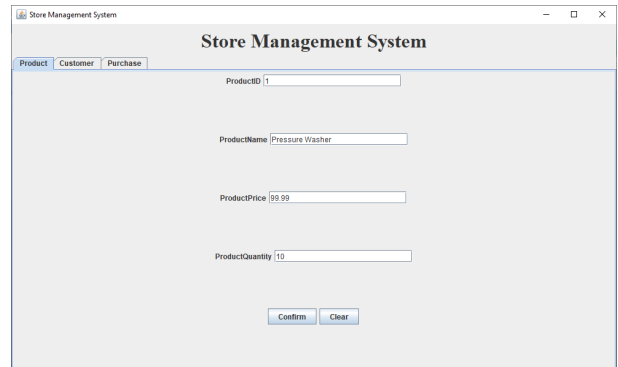
- ii. Exception - When the user wants to add a product, he/she accidentally types the product id already taken, the system shall prompt the user an error message.

1. The user clicks on the product tab and the system shall display the add product page.



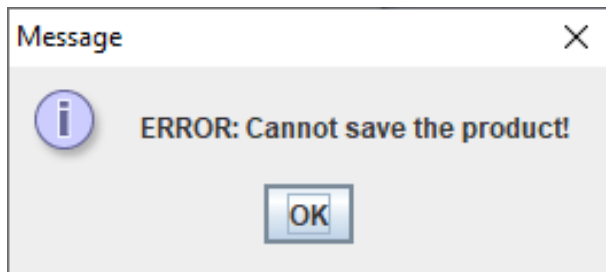
The screenshot shows a web application window titled "Store Management System". It has three tabs: "Product", "Customer", and "Purchase". The "Product" tab is active. The form contains the following fields: "ProductID" with the value "1", "ProductName" with the value "Pressure Washer", "ProductPrice" with the value "99.99", and "ProductQuantity" with the value "10". At the bottom of the form are two buttons: "Confirm" and "Clear".

2. The user types all product details but he/she cannot recall the latest product id and he/she casually types an id, which has been occupied by another product.



This screenshot is identical to the one above, showing the "Store Management System" window with the "Product" tab selected and the form fields filled with "ProductID: 1", "ProductName: Pressure Washer", "ProductPrice: 99.99", and "ProductQuantity: 10". The "Confirm" and "Clear" buttons are at the bottom.

3. The user clicks on "Confirm" button and the system shall pop out the message showing the product cannot be added.

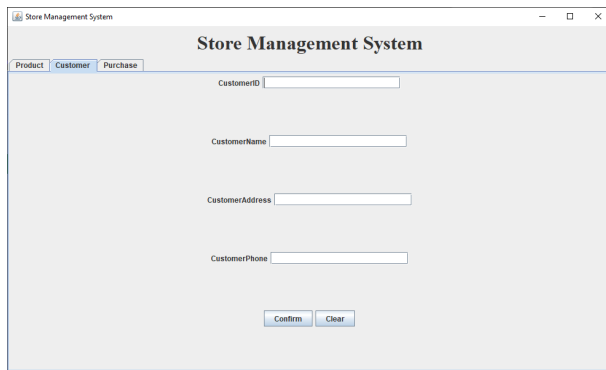


The screenshot shows a "Message" dialog box with a close button (X) in the top right corner. It features an information icon (i) on the left and the text "ERROR: Cannot save the product!" in the center. At the bottom center is an "OK" button.

(b) Customer

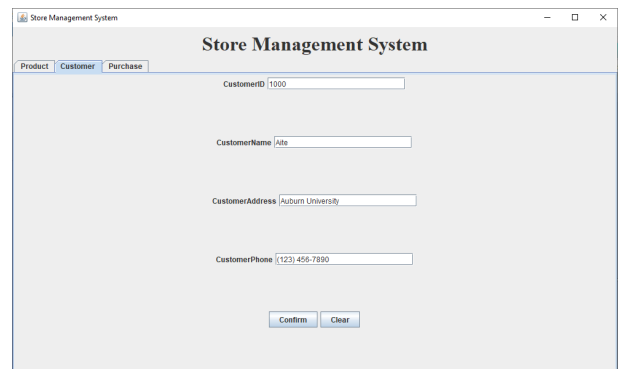
- i. Common case - When the customer wants to register a membership to get benefits at the customer service, the system shall provide a functionality that the employee can help the customer with their membership registration.

1. The user clicks on the customer tab and the system shall display the customer product page.



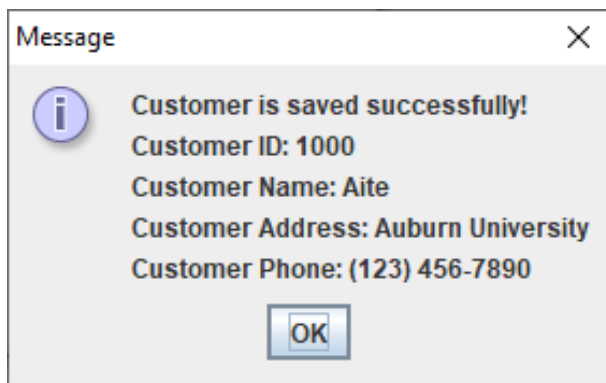
The screenshot shows the 'Store Management System' window with the 'Customer' tab selected. The interface includes input fields for 'CustomerID', 'CustomerName', 'CustomerAddress', and 'CustomerPhone'. At the bottom, there are 'Confirm' and 'Clear' buttons. The 'CustomerID' field contains the value '1000'.

2. The user inputs all the customer details.



The screenshot shows the 'Store Management System' window with the 'Customer' tab selected. The input fields are now filled with the following data: 'CustomerID' is '1000', 'CustomerName' is 'Aite', 'CustomerAddress' is 'Auburn University', and 'CustomerPhone' is '(123) 456-7890'. The 'Confirm' and 'Clear' buttons are visible at the bottom.

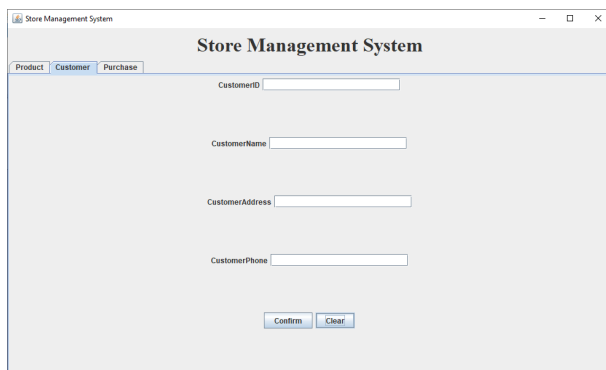
3. The user clicks on "Confirm" button and the system shall pop out the message showing the customer membership has been successfully registered.



The screenshot shows a 'Message' dialog box with a blue information icon. The text inside the dialog box reads: 'Customer is saved successfully!', 'Customer ID: 1000', 'Customer Name: Aite', 'Customer Address: Auburn University', and 'Customer Phone: (123) 456-7890'. An 'OK' button is located at the bottom center of the dialog box.

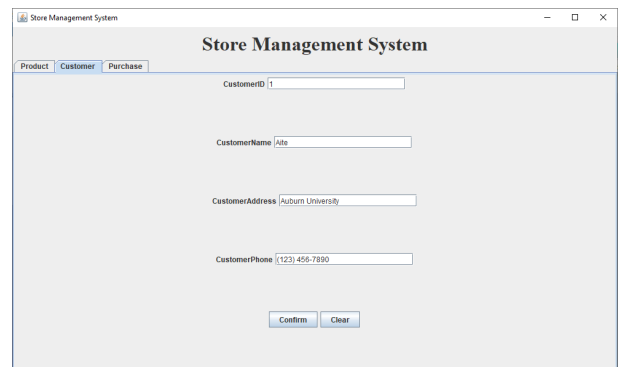
- ii. Exception - When the user wants to add a customer, he/she accidentally types the customer id already taken, the system shall prompt the user an error message.

1. The user clicks on the product tab and the system shall display the add product page.



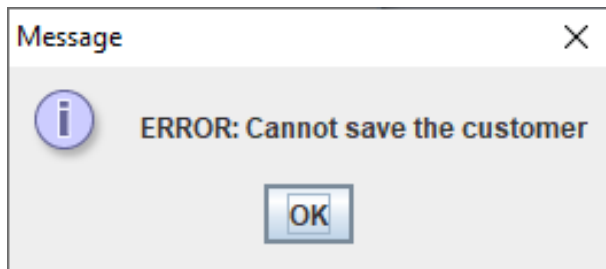
The screenshot shows the 'Store Management System' window with the 'Customer' tab selected. The form contains four input fields: 'CustomerID', 'CustomerName', 'CustomerAddress', and 'CustomerPhone'. At the bottom, there are 'Confirm' and 'Clear' buttons. The 'CustomerID' field is currently empty.

2. The user types all product details but he/she cannot recall the latest customer id and he/she casually types an id, which has been occupied by another customer.



The screenshot shows the 'Store Management System' window with the 'Customer' tab selected. The form fields are now filled: 'CustomerID' contains '1', 'CustomerName' contains 'John', 'CustomerAddress' contains 'Auburn University', and 'CustomerPhone' contains '(123) 456-7890'. The 'Confirm' and 'Clear' buttons are still at the bottom.

3. The user clicks on "Confirm" button and the system shall pop out the message showing the customer cannot be added.



The screenshot shows a 'Message' dialog box with a close button (X) in the top right corner. It features an information icon (i) on the left and the text 'ERROR: Cannot save the customer' in the center. An 'OK' button is located at the bottom center of the dialog.

(c) Purchase

- i. Common case - When the user wants to add a purchase in the system, the system shall provide a functionality that the user can record the purchase and transaction history into the database.

1. The user clicks on the product tab and the system shall display the add product page.

Store Management System

Product Customer Purchase

Purchase Date: Tue Oct 08 22:48:41 CDT 2019

Purchase ID:

Customer ID:

Customer Name:

Product ID:

Product Name:

Product Price:

Purchase Quantity:

Cost:

Tax:

Total Cost:

Confirm Clear

2. When the user types the customer id, the system shall automatically display the corresponding customer name.

Store Management System

Product Customer Purchase

Purchase Date: Tue Oct 08 22:48:41 CDT 2019

Purchase ID:

Customer ID: 1

Customer Name: Lee

Product ID:

Product Name:

Product Price:

Purchase Quantity:

Cost:

Tax:

Total Cost:

Confirm Clear

3. When the user types the product id, the system shall automatically display the corresponding product name.

Store Management System

Product Customer Purchase

Purchase Date: Tue Oct 08 22:48:41 CDT 2019

Purchase ID:

Customer ID: 1

Customer Name: Lee

Product ID: 3

Product Name: 65 inch TV

Product Price: 749.99

Purchase Quantity: 1

Cost:

Tax:

Total Cost:

Confirm Clear

4. When the user types the purchase quantity, the system shall automatically show the cost, tax and total cost.

Store Management System

Product Customer Purchase

Purchase Date: Tue Oct 08 22:48:41 CDT 2019

Purchase ID:

Customer ID: 1

Customer Name: Lee

Product ID: 3

Product Name: 65 inch TV

Product Price: 749.99

Purchase Quantity: 2

Cost: 1499.98

Tax: 134.9982

Total Cost: 1634.9782

Confirm Clear

5. The user clicks on "Confirm" button and the system shall pop out the message showing the customer cannot be added.

Message

Purchase is saved successfully!
STORE MANAGEMENT SYSTEM

Customer ID: 3
Customer Name: Lee
Customer Address: Auburn, AL
Customer Phone: (382)289-1928
Product ID: 3
Product Name: 65 inch TV
Unit Price: 749.990000
StockQuantity: 2.000000
Purchase ID: 1
Customer ID: 3
Product ID: 3
Cost: 1499.980000
Tax: 134.998200
TotalCost: 1634.978200
Price: 749.990000
Quantity: 2.000000
Date: Tue Oct 08 22:48:41 CDT 2019

OK

- ii. Exception - When the user wants to add a purchase in the system, he/she accidentally types the purchase id already taken, the system shall prompt the user an error message.

1. The user clicks on the product tab and the system shall display the add product page.

Store Management System

Product Customer **Purchase**

Purchase Date: Tue Oct 08 22:48:41 CDT 2019

Purchase ID:

Customer ID:

Customer Name:

Product ID:

Product Name:

Product Price:

Purchase Quantity:

Cost:

Tax:

Total Cost:

2. The user types all purchase details but he/she cannot recall the latest purchase id and he/she casually types an id, which has been occupied by another purchase.

Store Management System

Product Customer **Purchase**

Purchase Date: Tue Oct 08 22:48:41 CDT 2019

Purchase ID: 1

Customer ID: 3

Customer Name: Lee

Product ID: 3

Product Name: 65 inch TV

Product Price: 749.99

Purchase Quantity: 2

Cost: 1499.98

Tax: 134.9982

Total Cost: 1634.9782

3. The user clicks on "Confirm" button and the system shall pop out the message showing the purchase cannot be added.

Message

ERROR: Cannot save the purchase!

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

```
CREATE TABLE "Customers" (  
  "customerID" INTEGER NOT NULL,  
  "name" TEXT DEFAULT 'Guest',  
  "phone" TEXT DEFAULT '(000)000-0000',  
  "address" TEXT DEFAULT 'Auburn, AL',  
  PRIMARY KEY("customerID")  
);
```

```
CREATE TABLE "Products" (  
  "productID" INTEGER NOT NULL,  
  "productName" TEXT NOT NULL,  
  "unitPrice" REAL NOT NULL,  
  "stockQuantity" INTEGER NOT NULL,  
  PRIMARY KEY("productID")  
);
```

```
CREATE TABLE "Purchases" (  
  "purchaseID" INTEGER NOT NULL,  
  "customerID" INTEGER NOT NULL,  
  "productID" INTEGER NOT NULL,  
  "price" REAL NOT NULL,  
  "quantity" REAL NOT NULL,  
  "cost" REAL NOT NULL,  
  "tax" REAL NOT NULL,  
  "totalCost" REAL NOT NULL,  
  "date" INTEGER,  
  FOREIGN KEY("customerID") REFERENCES "Customers"("customerID"),  
  FOREIGN KEY("productID") REFERENCES "Products"("productID"),  
  PRIMARY KEY("purchaseID")  
);
```

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

```
INSERT INTO "Products" VALUES (1,'Clay Bar',19.99,33);
INSERT INTO "Products" VALUES (2,'Heat Guns',29.99,6);
INSERT INTO "Products" VALUES (3,'65 inch TV',749.99,2);
INSERT INTO "Products" VALUES (4,'PS4',399.99,31);
INSERT INTO "Products" VALUES (5,'XBOX ONE S',299.99,26);
INSERT INTO "Products" VALUES (6,'Amazon Kindle',199.99,11);
INSERT INTO "Products" VALUES (7,'Baby Stroller',299.99,25);
INSERT INTO "Products" VALUES (8,'Baby Diaper',2.99,120);
```

```
INSERT INTO "Customers" VALUES (1,'William','(185)864-1832','Auburn, AL');
INSERT INTO "Customers" VALUES (2,'Cox','(284)968-3821','Auburn, AL');
INSERT INTO "Customers" VALUES (3,'Lee','(382)289-1928','Auburn, AL');
INSERT INTO "Customers" VALUES (4,'Daniel','(042)213-4126','Auburn, AL');
INSERT INTO "Customers" VALUES (5,'Miller','(158)239-1295','Auburn, AL');
INSERT INTO "Customers" VALUES (6,'Tayolr','(334)512-3123','Auburn, AL');
INSERT INTO "Customers" VALUES (7,'Diaz','(772)334-6041','Auburn, AL');
INSERT INTO "Customers" VALUES (8,'Brown','(341)412-5816','Auburn, AL');
INSERT INTO "Customers" VALUES (9,'Oliver','(751)285-1682','Auburn, AL');
INSERT INTO "Customers" VALUES (10,'Jacob','(681)186-2851','Auburn, AL');
```

```
INSERT INTO "Purchases" VALUES (1,1,1,19.99,12.0,239.88,21.5892,261.4692,'Tue Oct 08 23:36:45 CDT 2019');
INSERT INTO "Purchases" VALUES (2,4,1,19.99,3.0,59.97,5.3973,65.3673,'Tue Oct 08 23:36:45 CDT 2019');
INSERT INTO "Purchases" VALUES (3,6,2,29.99,4.0,119.96,10.7964,130.7564,'Tue Oct 08 23:36:45 CDT 2019');
INSERT INTO "Purchases" VALUES (4,2,2,29.99,3.0,89.97,8.0973,98.0673,'Tue Oct 08 23:36:45 CDT 2019');
INSERT INTO "Purchases" VALUES (5,6,3,749.99,1.0,749.99,67.4991,817.4891,'Tue Oct 08 23:36:45 CDT 2019');
INSERT INTO "Purchases" VALUES (6,3,5,299.99,2.0,599.98,53.9982,653.9782,'Tue Oct 08 23:36:45 CDT 2019');
INSERT INTO "Purchases" VALUES (7,4,7,299.99,1.0,299.99,26.9991,326.9891,'Tue Oct 08 23:36:45 CDT 2019');
INSERT INTO "Purchases" VALUES (8,5,8,2.99,10.0,29.9,2.691,32.591,'Tue Oct 08 23:36:45 CDT 2019');
INSERT INTO "Purchases" VALUES (9,7,3,749.99,1.0,749.99,67.4991,817.4891,'Tue Oct 08 23:40:13 CDT 2019');
INSERT INTO "Purchases" VALUES (10,9,6,199.99,2.0,399.98,35.9982,435.9782,'Tue Oct 08 23:40:13 CDT 2019');
INSERT INTO "Purchases" VALUES (11,10,2,29.99,3.0,89.97,8.0973,98.0673,'Tue Oct 08 23:47:03 CDT 2019');
```