OKTAVIANUS FACUN BAUR
Github: https://github.com/facunbaur/Project1_COMP7700

Project 1
COMP 7700 Software Architecture

1. Write two possible use cases for each user story: one is the common case and one is the exception.
   a. As a user, I want to add a new product into the system.
      - Common case
        1) Click button “Add Product” on Main Screen
        2) Display “Add Product” screen
        3) Input data on each field of the screen form
        4) Display alert “Product added succesfully!”
        5) Clik “OK”
        6) Display main screen

      - The exception
        For input products field:
        1) When productid is empty, display "ProductID could not be EMPTY!!!"
        2) When productid is not a number, display "ProductID is INVALID (not a number)!!!"
        3) When product name is empty, display "Product Name could not be EMPTY!!!"
        4) When price is empty, display "Price could not be EMPTY!!!")
        5) When price is not a number, display "Price is INVALID (not a number)!!!"
        6) When quantity is empty, display "Quantity could not be EMPTY!!!"
        7) When quantity is empty, not a number "Quantity is INVALID (not a number)!!!"

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

- Common case
    1) Click button “Add Customer” on Main Screen
    2) Display “Add customer” screen
    3) Input data on each field of the screen form
    4) Display alert “Customer added succesfully!”
    5) Clik “OK”
    6) Display main screen

- The exception

    For input customers field:
    1) When productid is empty, display "CustomerID could not be EMPTY!!!"
    2) When CustomerID is not a number, display "CustomerID is INVALID (not a number)!!!"
    3) When Customer name is empty, display "Customer Name could not be EMPTY!!!"
    4) When Address is empty, display "Address could not be EMPTY!!!")
    5) When Phone is empty, display "Phone could not be EMPTY!!!"

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

- Common case
    1) Click button “Add Purchase” on Main Screen
    2) Display “Add Purchase” screen
    3) Input data on each field of the screen form
    4) Display alert “Purchase added succesfully!”
    5) Clik “OK”
    6) Display main screen

- The exception

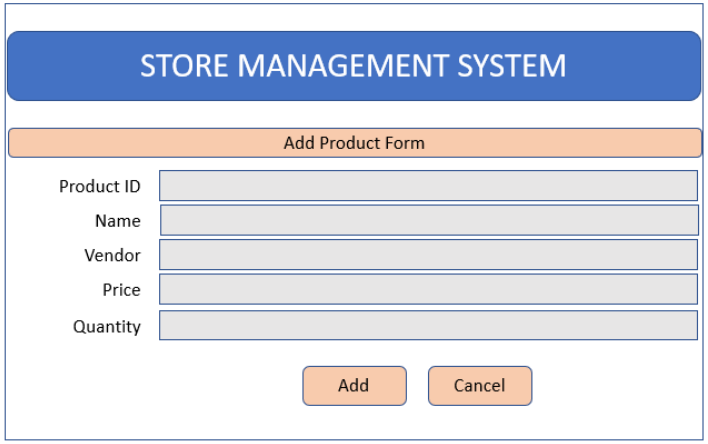
  For input purchase field:

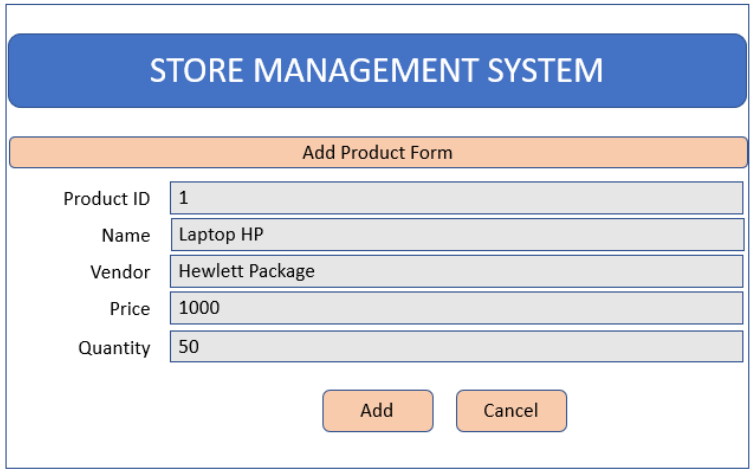
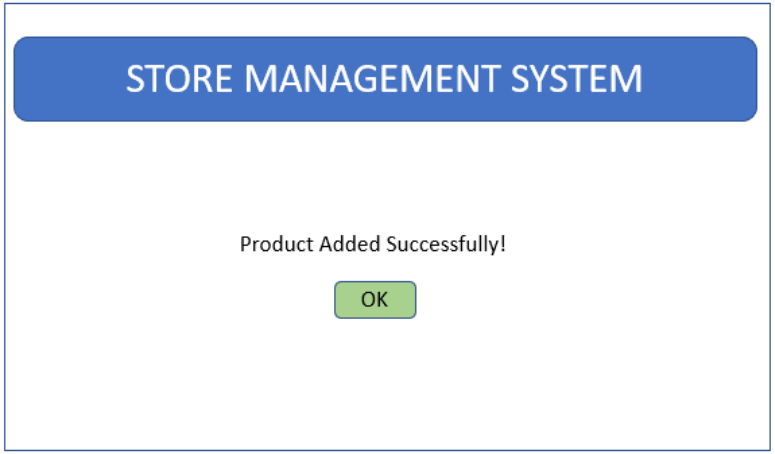
  1) When PurchaseID is empty, display " PurchaseID could not be EMPTY!!!"
  2) When PurchaseID is not a number, display " PurchaseID is INVALID (not a number)!!!"
  3) When CustomerID is not a number, display " CustomerID is INVALID (not a number)!!!"
  4) When CustomerID is NULL or not found , display " No Customer with ID...!!!"
  5) When ProductID is not a number, display " ProductID is INVALID (not a number)!!!"
  6) When ProductID is NULL or not found , display " No Products with ID...!!!"

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

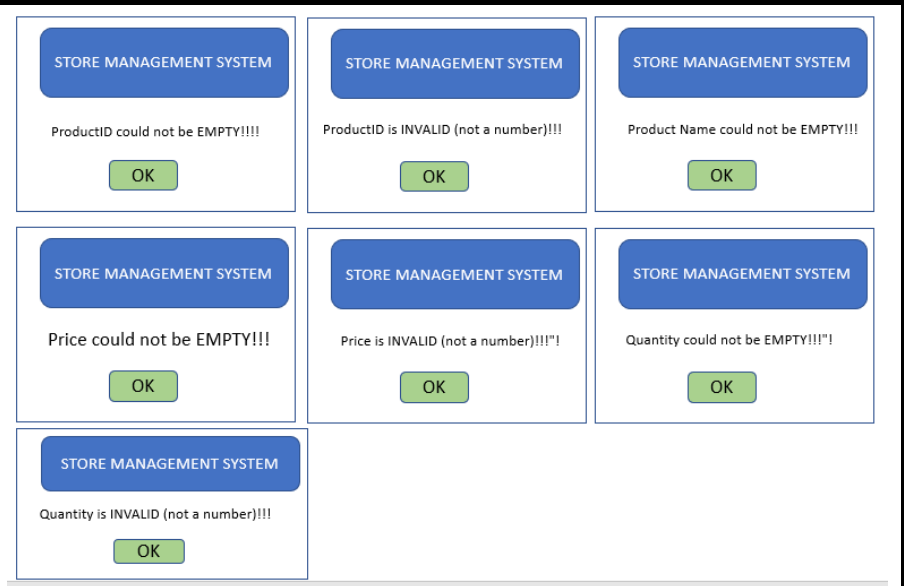
❖ As a user, I want to add a new product into the system.

Common Case

Actor	System
1. Click button “Add Product” on Main Screen 	2. Display “Add Product” screen 

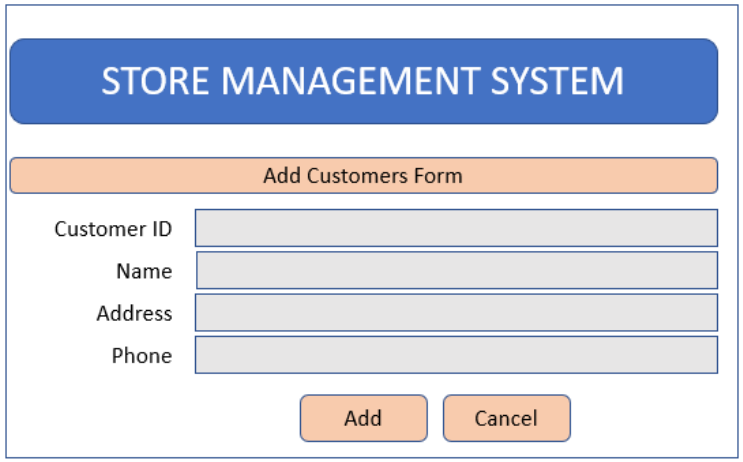
3. Input data then click add button "Add product" screen with input data:  	4. Hide "add product" screen and display a alert "Product added Succesfully!" 
5. Klick "OK"	6. Display main screen

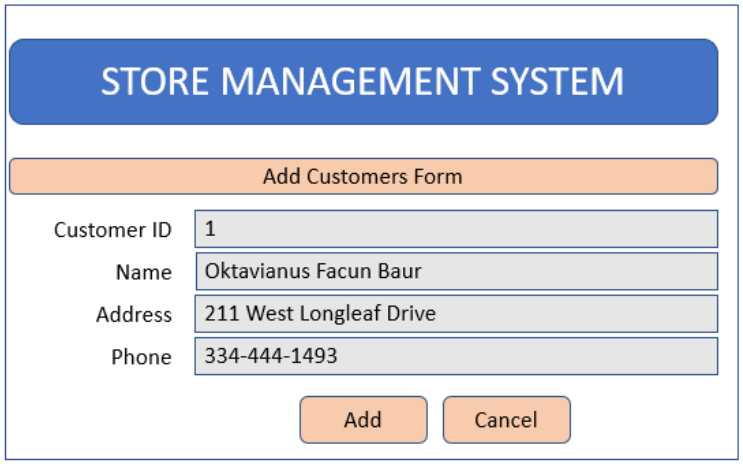
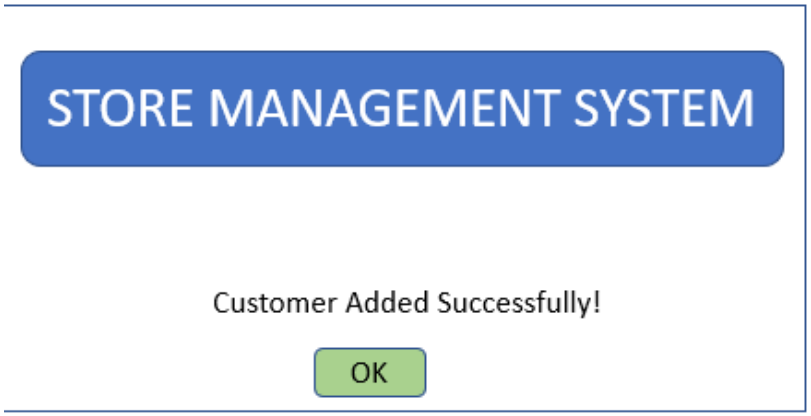
Exception

Actor	System
1) When productid is empty, display "ProductID could not be EMPTY!!!" 2) When productid is not a number, display "ProductID is INVALID (not a number)!!!" 3) When product name is empty, display "Product Name could not be EMPTY!!!" 4) When price is empty, display "Price could not be EMPTY!!!") 5) When price is not a number, display "Price is INVALID (not a number)!!!" 6) When quantity is empty, display "Quantity could not be EMPTY!!!" 7) When quantity is empty, not a number "Quantity is INVALID (not a number)!!!"	

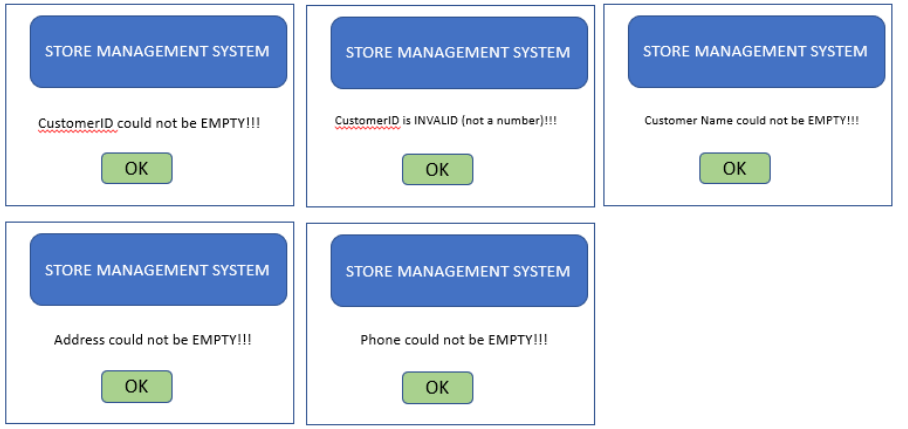
❖ As a user, I want to add a new customer into the system.

Common Case

Actor	System
1. Click button “Add Customer” on Main Screen 	2. Display “Add Customer” screen 
3. Input data then click add button “Add Customer” screen with input data:	4. Hide “add customer” screen and display a alert “Customer added Succesfully!”

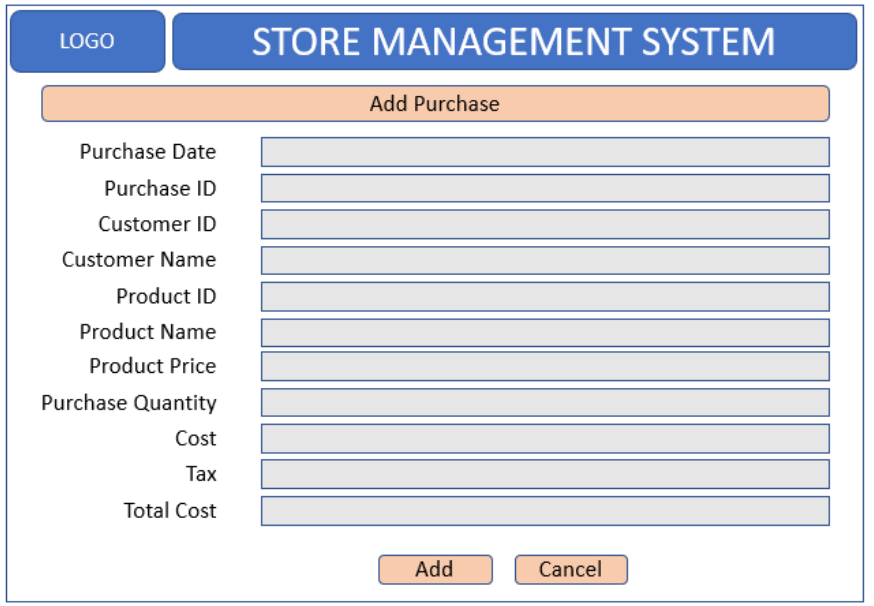
	
5. Klick “OK”	6. Display main screen

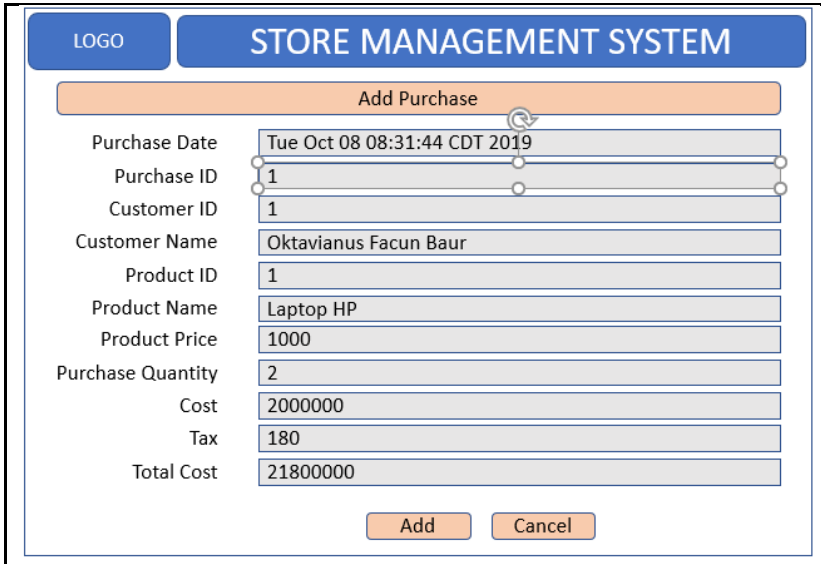
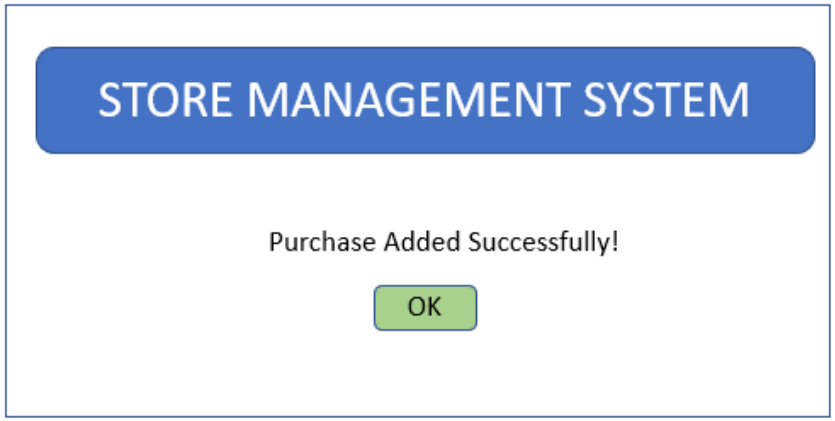
Exception

Actor	System
1) When productid is empty, display "CustomerID could not be EMPTY!!!" 2) When CustomerID is not a number, display "CustomerID is INVALID (not a number)!!!" 3) When Customer name is empty, display "Customer Name could not be EMPTY!!!" 4) When Address is empty, display "Address could not be EMPTY!!!") 5) When Phone is empty, display "Phone could not be EMPTY!!!"	

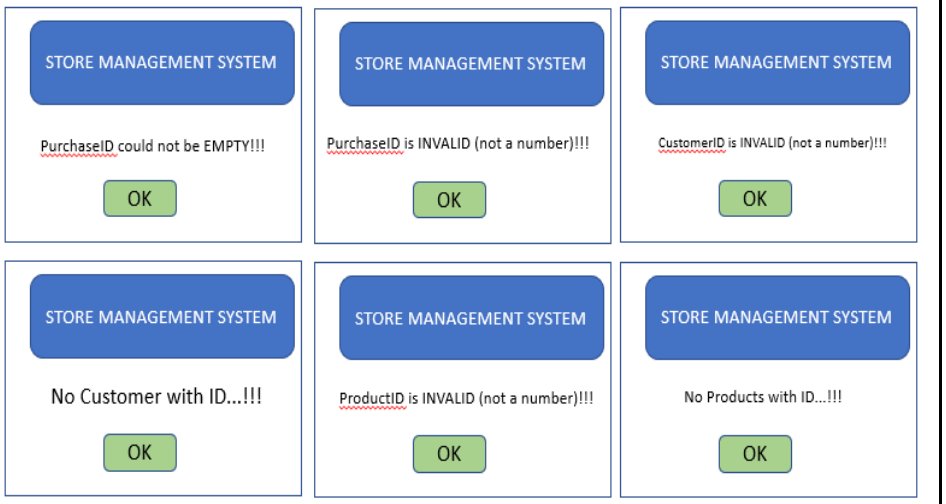
❖ As a user, I want to add a purchase from a customer into the system.

Common Case

<table>
<tr><th>Actor</th><th>System</th></tr>
<tr><td>1. Click button “Add Purhase” on Main Screen
 
</td><td>2. Display “Add Purchase” screen
 
</td></tr>
<tr><td>3. Input data then click add button
“Add purchase” screen with input data:</td><td>4. Hide “add purchase” screen and display a alert “Purchase added Succesfully!”</td></tr>
</table>

	
5. Klick “OK”	6. Display main screen

Exception

Actor	System
1. When PurchaseID is empty, display " PurchaseID could not be EMPTY!!!" 2. When PurchaseID is not a number, display " PurchaseID is INVALID (not a number)!!!" 3. When CustomerID is not a number, display " CustomerID is INVALID (not a number)!!!" 4. When CustomerID is NULL or not found , display " No Customer with ID...!!!" 5. When ProductID is not a number, display " ProductID is INVALID (not a number)!!!" 6. When ProductID is NULL or not found , display " No Products with ID...!!!"	

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

*Purchase(**PurchaseID**,CustomerID,ProductID,Cost,Tax,TotalCost,Price,Quantity,Date);*
*Customer(**CustomerID**,Name,Address,Phone);*
*Products(**ProductID**,Name,Vendor,Price,Quantity);*

BEGIN TRANSACTION;
CREATE TABLE IF NOT EXISTS "Purchase" (

```
	"PurchaseID"	INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
	"CustomerID"	INTEGER,
	"ProductID"	INTEGER,
	"Cost"	REAL,
	"Tax"	REAL,
	"TotalCost"	REAL,
	"Price" REAL,
	"Quantity"	INTEGER,
	"Date" TEXT
);
CREATE TABLE IF NOT EXISTS "Customer" (
	"CustomerID"	INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
	"Name"	TEXT,
	"Address"	TEXT,
	"Phone"	TEXT
);
CREATE TABLE IF NOT EXISTS "Products" (
	"ProductID"	INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
	"Name"	TEXT,
	"Vendor"	TEXT,
	"Price" REAL,
	"Quantity"	INTEGER
);
INSERT INTO "Purchase" VALUES (1,1,5,1200.0,216.0,2616.0,1200.0,2,'Sat Oct 05 06:59:53 CDT 2019');
INSERT INTO "Purchase" VALUES (2,2,4,4000.0,360.0,4360.0,1000.0,4,'Sat Oct 05 06:59:53 CDT 2019');
INSERT INTO "Purchase" VALUES (3,3,3,6.0,0.54,6.54,2.0,3,'Sat Oct 05 06:59:53 CDT 2019');
```

INSERT INTO "Purchase" VALUES (4,4,2,4.0,0.36,4.36,4.0,1,'Sat Oct 05 06:59:53 CDT 2019');
INSERT INTO "Purchase" VALUES (5,5,1,2000.0,180.0,2180.0,1000.0,2,'Sat Oct 05 11:59:53 CDT 2019');
INSERT INTO "Purchase" VALUES (6,1,1,5000.0,150.0,5450.0,1000.0,5,'Tue Oct 08 10:59:53 CDT 2019');
INSERT INTO "Purchase" VALUES (7,2,2,4.0,0.36,4.36,4.0,1,'Tue Oct 08 10:59:53 CDT 2019');
INSERT INTO "Purchase" VALUES (8,3,3,4.0,0.36,4.36,2.0,2,'Tue Oct 08 10:59:53 CDT 2019');
INSERT INTO "Purchase" VALUES (9,4,4,4000.0,360.0,4360.0,1000.0,4,'Tue Oct 08 10:59:53 CDT 2019');
INSERT INTO "Purchase" VALUES (10,5,5,2400.0,216.0,2616.0,1200.0,2,'Tue Oct 08 14:59:53 CDT 2019');

INSERT INTO "Customer" VALUES (1,'Oktavianus Facun Baur','The Social Apartment','334-444-1492');
INSERT INTO "Customer" VALUES (2,'Onyinye Rosemary Asogwa','Samford Apartment','334-444-1493');
INSERT INTO "Customer" VALUES (3,'Abhishek Jariwala','Auburn Housing','334-444-1494');
INSERT INTO "Customer" VALUES (4,'Kripa Shankar Muthukumar','The Vilas Apartment','334-444-1495');
INSERT INTO "Customer" VALUES (5,'Ishita Naresh Joshi','The Beacon','334-444-1496');

INSERT INTO "Products" VALUES (1,'Laptop HP','Helett Packard',1000.0,20);
INSERT INTO "Products" VALUES (2,'Mouse Logitech','Logitech',4.0,10);
INSERT INTO "Products" VALUES (3,'Keyboard','AllVendors',2.0,34);
INSERT INTO "Products" VALUES (4,'Laptop HP','Acer',1000.0,5);
INSERT INTO "Products" VALUES (5,'Mac Book Pro','Mac',1200.0,30);
COMMIT;

4. Implement the use cases.
    - Codes are on github: https://github.com/facunbaur/Project1_COMP7700
5. Test the system with each use case.

    - Testing screens are on github: https://github.com/facunbaur/Project1_COMP7700