American International University- Bangladesh
Faculty of CSE

Object oriented analysis and design

Project Name:Store Management System

Group Members NAME

NAME
MD. DIDARUL ALAM ALIF

Table of Contents

List of figures

List of tables

Abstract

The " Store Management System" project comprises sales, stock keeping, warehouse management, personnel updates, customer management, and point of sale management. As a result, this work was completed in order to address issues that are common in most businesses. The Store Management System was built utilizing one programming environments: java. For security purpose the system builds with face recognition system. Seller can log in to the system by recognizing face. It can be accessed by either the administration or the personnel. The interfaces are quite intuitive. Personal data is well-protected in database, and data processing is lightning fast.

Background Information

A store is a place where extra materials are maintained to be used as needed. "To accept materials, to preserve them from damage and unlawful removal while in storage, to issue the material in the appropriate quantities, at the right time, to the right place, and to deliver these services promptly and at the lowest cost," says store management. Our project store management system is one of the systems that performs difficult duties is a system.

Objectives

In view of the problems mentioned above, this project is aimed at implementing a Store management system which will exclusively:

- Automate every sales of all product in stock
- Manage different products and categories in store
- Manage every items details
- Manage every customers of the system
- Create bill after selling items
- Manage Supplier

Scope

In our project we were occupied in the assessment of the existing system and identifying the problems of store management system. We proposed possible alternative solutions which can help to choose the best feasible solution and design an efficient system for store management system. Generally, our system is capable of addressing the following function to the Store.

- Seller of the store, search items by type and name of items.
- Adding and Recording new items and amount into the store.
- Updating and organizing already stored items.
- Checking updated information in the store.
- Storing new and updated items.

Proposed Solutions

The motive of this Shopping Web Application is to allow the user to play with the search tool and create different combinatorial search criterion to perform exhaustive search.

- Making the application enabled gets rid of these unnecessary delays letting the seller to perform exhaustive.
- Provide Interactive interface through which a user can interact with seller easily.
- Provide an interface where customer can add item and remove item from the cart
- Provide an app where customer can do everything online without interacting with seller

Risks and Constraints

Technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes for the implementing this system.

Organizational feasibility

This study is carried out to check the economic impact will have on the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products have to be purchased.

Use case diagram

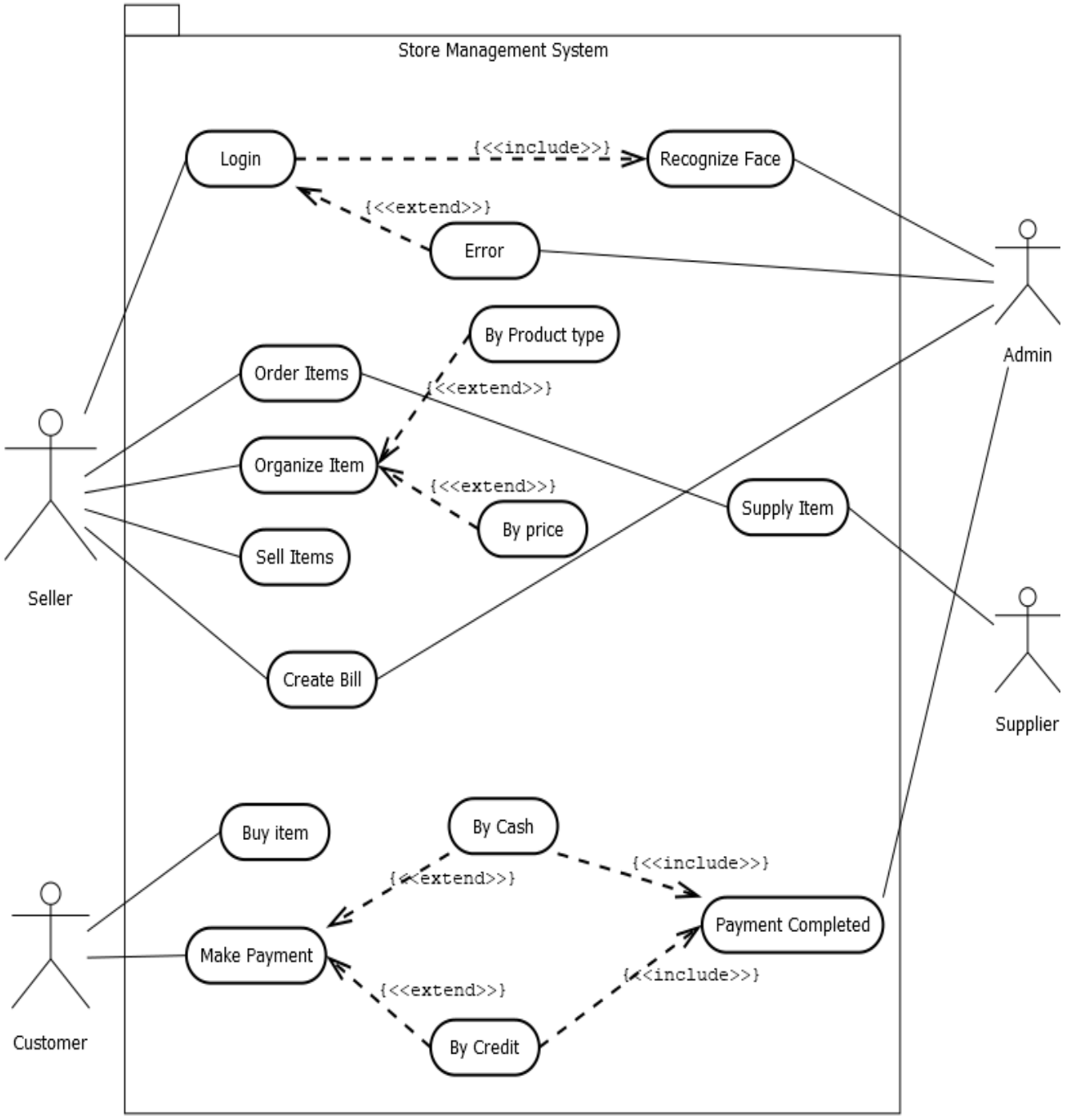

Figure 1:Use case diagram

Use Case Specification

Login

Table 1:Login(Use case specification)

Use Case Name:	login	
Actor(s):	Seller, System	
Description:	The System login is system where seller can login to the system by going through some steps for privacy of the system. Face recognition will confirm his/her identity then the seller will be taken to the home page interface if the seller is recognized.	
Reference ID:	SMS 01	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: Seller will login to the system. **Step 2:** Then seller will show the face as the login method.	**Step 3** : The system will identify sellers information from database by recognizing sellers face with face recognition system. **Step 4** : If the data matches with the database information then system will show login successful. **Step 5** : After successful login the system interface will be open.
Alternative course of events:	**Step 4a**: If the data mismatches with the database, then system will show login declined.	
Pre-condition:	Sellers must be registered in database. Sellers must remove mask before scanning.	
Post-condition:		

Order Items

Table 2:Order items(Use case specification)

Use Case Name:	Order Items	
Actor(s):	Seller, Supplier	
Description:	The Order Items is a system is used to receive order from seller to supplier and suppliers confirms the stock availability. Seller can order if the stock is not available. This use case describes the process of customer ordering items from a store.	
Reference ID:	SMS 02	
Typical course of events:	**Actor Action**	**System Response**
	Step 1: At first the seller will order the Items accordingly. **Step 2**: The supplier will receive the order **Step 3:** The supplier will check in the warehouse whether it is in stock. **Step 4:** If the items is available in stock Then the supplier will receive the product. **Step 5:** If payment is done by cash then seller will confirm it and payment will be completed. **Step 7**: The supplier will deliver items from the warehouse to the seller. **Step 8**: After receiving the item the seller will handover the item to the customer.	**Step 6:** If payment is done through card then system will show card payment interface and start the payment process.
Alternative course of events:	**Step 6a**: The fiancé sector will check the validity of the card, if the card is invalid then the payment wont be successful.	
Pre-condition:	Seller must confirm the product before payment.	
Post-condition:	Customer must check the items properly.	

Organize Items

Table 3:Organize items(use case specification)

Use Case Name:	Organize items	
Actor(s):	Seller, System	
Description:	The Organize items is a system is used to manage, order, organize and sell item. This use case describes the process of customer ordering items from a store.	
Reference ID:	SMS 03	
Typical course of events:	Actor Action	System Response
	Step 1: The seller will gather all the items from the list. Step 2: The seller can organize the items as well. Step 4: The seller will select the organization method.	Step 3: The system will display 2 different sorting method for by product type or by the price of the product. Step 5: After sorting the system will be able to display the sorted products.
Alternative course of events:		
Pre-condition:	Seller must be registered in the database.	
Post-condition:		

Selling & Buying items

Table 4:Selling & Buying items(use case specification)

Use Case Name:	Selling & Buying	
Actor(s):	Seller, Customer	
Description:	The Display item is a system is used to display items to the customers and customer can select items then system will calculate and prepare a list of items for the customer.	
Reference ID:	SMS 04	
Typical course of events:	Actor Action	System Response
	Step 1: The seller will display all the items. Step 2 : The customer will select the products. Step 3: If the customer chooses the product, then the buying interface will open. Step 5: The seller will prepare the list of selected items.	Step 4 : System will remove the amount of selected items from database.
Alternative course of events:	Step 3a: If customer wants to remove any items, then the items will be marked as rejected and those will be added to database.	
Pre-condition:	Displayed products must be available in the warehouse.	
Post-condition:	Customer should check the list before confirming.	

Creating Bill

Table 5:Creating bill(use case specification)

Use Case Name:	Creating bill	
Actor(s):	Seller, Customer, System	
Description:	The creating bill is a system is used to create a bill including the information and the amount of items and it will make a document for print.	
Reference ID:	SMS 05	
Typical course of events:	Actor Action	System Response
	Step 1: The seller will make a list of the selected items. Step 2 : The customer will check the items and proceed to buy. Step 3: The seller will enlist the items name and the items quality. Step 6: The printed bill will be handed over to the customer by the seller.	Step 4: the system will create the bill. Step 5: Then the system will print the bill.
Alternative course of events:		
Pre-condition:	A printer must be needed to print bill.	
Post-condition:	Customer must receive the printed bill.	

Make Payment

Table 6:Make payment(use case specification)

Use Case Name:	Make payment	
Actor(s):	Seller, Customer, System	
Description:	The Make payment is a system is used to complete the payment method by two different way and system is connected to the credit card authority to check card validity.	
Reference ID:	SMS 06	
Typical course of events:	Actor Action	System Response
	Step 1:The customer will make the payment by cash or by credit card. Step 2 :If the payment is done by cash the seller will confirm by receiving the payment. Step 3:If the payment is done the credit card then the card department will check the validity of the card and complete the transaction.	Stem 4: System will display Payment process successful.
Alternative course of events:	Step 4a: If the card is invalid then the authority will send transaction not successful. Step 4b: The system will show payment not successful if any other error occurs in the transaction.	
Pre-condition:	Seller must need a credit card swipe card machine to complete the process of the payment by credit card.	
Post-condition:	Customer must take back his/her credit card.	

Activity Diagram

Login

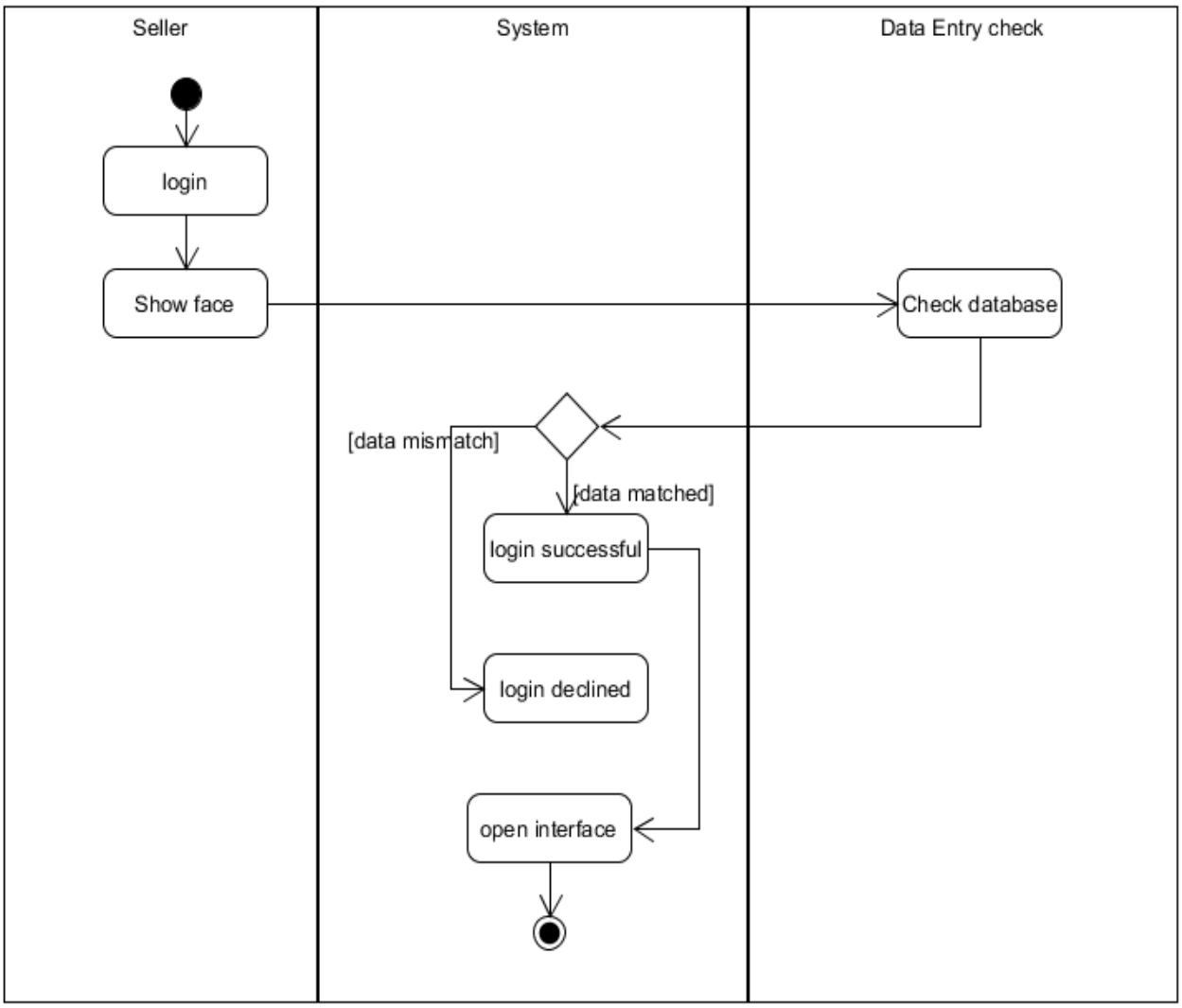

Figure 2:Login(Activity Diagram)

Description-

As illustrated in the activity diagram, the seller must first log into the system. The seller will initially scan the seller's face as a means of logging into the system. Following that, the database will save and verify the face recognition data. If the data entry check is successful and the data is correct, it will display the login dashboard and also open the system login interface. Again, if the data does not match, the information is transferred to the system. The system will then indicate that the login attempt was unsuccessful.

Order Items

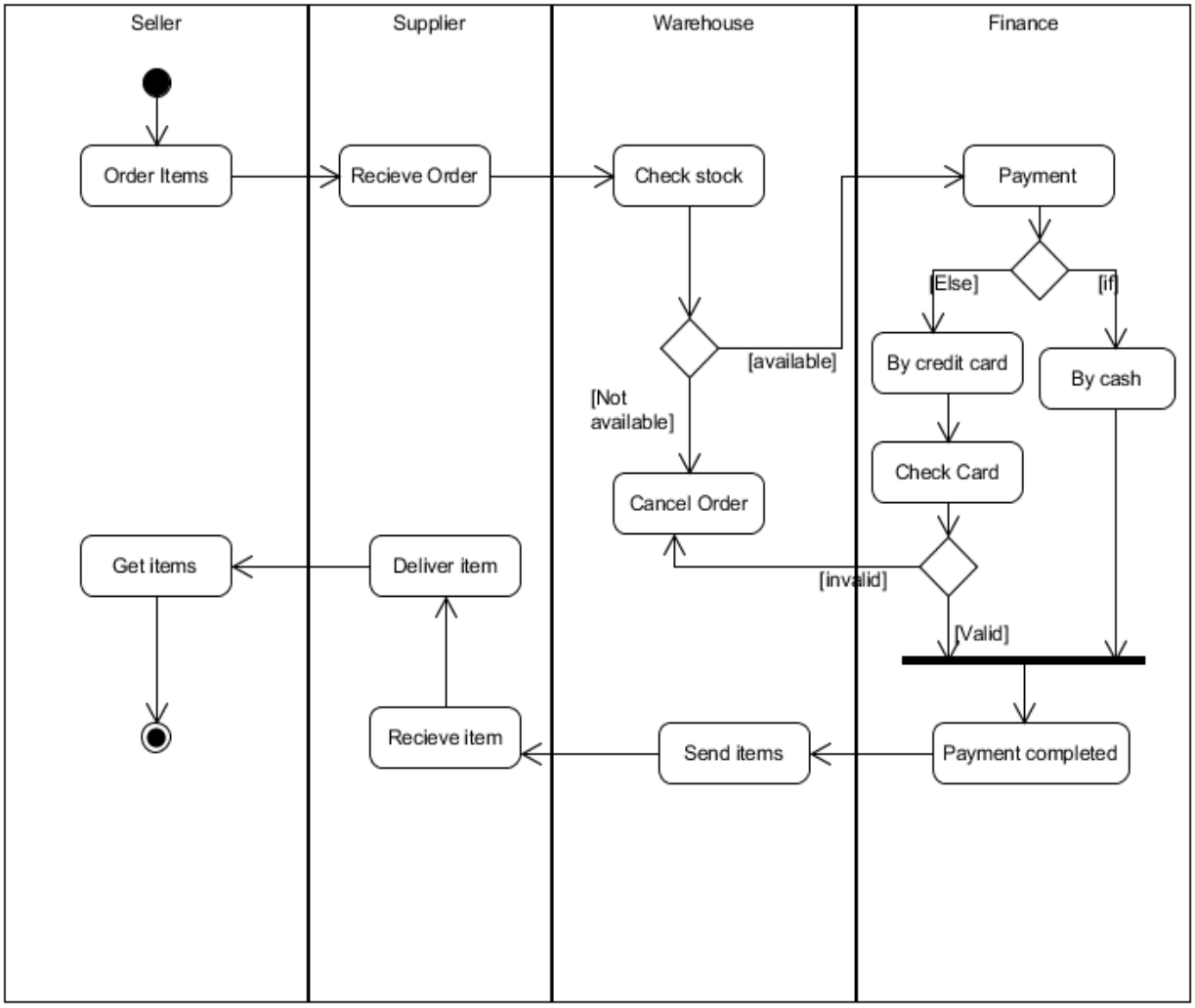

Figure 3:Order items(Activity Diagram)

Description-

In this activity diagram, we can see the seller is ordering items . Following that, the supplier receives the order. The last step is to verify that the items in the warehouse are still available. If the desired items are available, the payment system is initiated. If payment is made in cash, the payment is completed immediately and the items are sent from the warehouse to the supplier, who receives and delivers the items to the seller. On the other hand, if payment is made through credit card, the card's validity is verified; if the card is legitimate, the transaction is completed immediately. If the credit card is not legitimate, the order will be cancelled.

Organize items

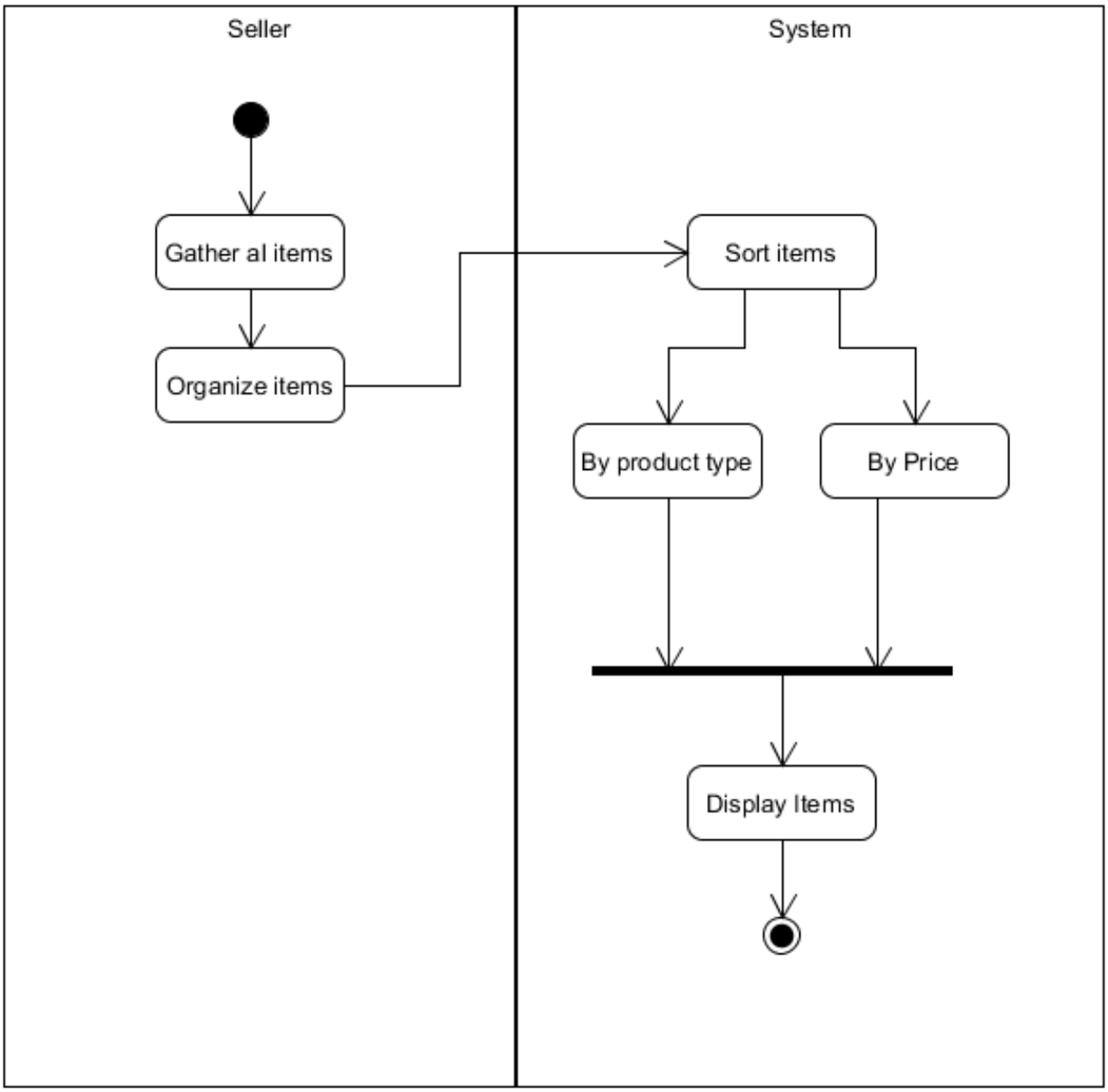

Figure 4:Organize items(Activity Diagram)

Description-

This diagram demonstrates how the seller gathers and organizes all of the items. After organizing the items, the system categorizes them in two ways. By product category and price. Finally, following sorting, the system displays the items.

Selling items & Buying Items

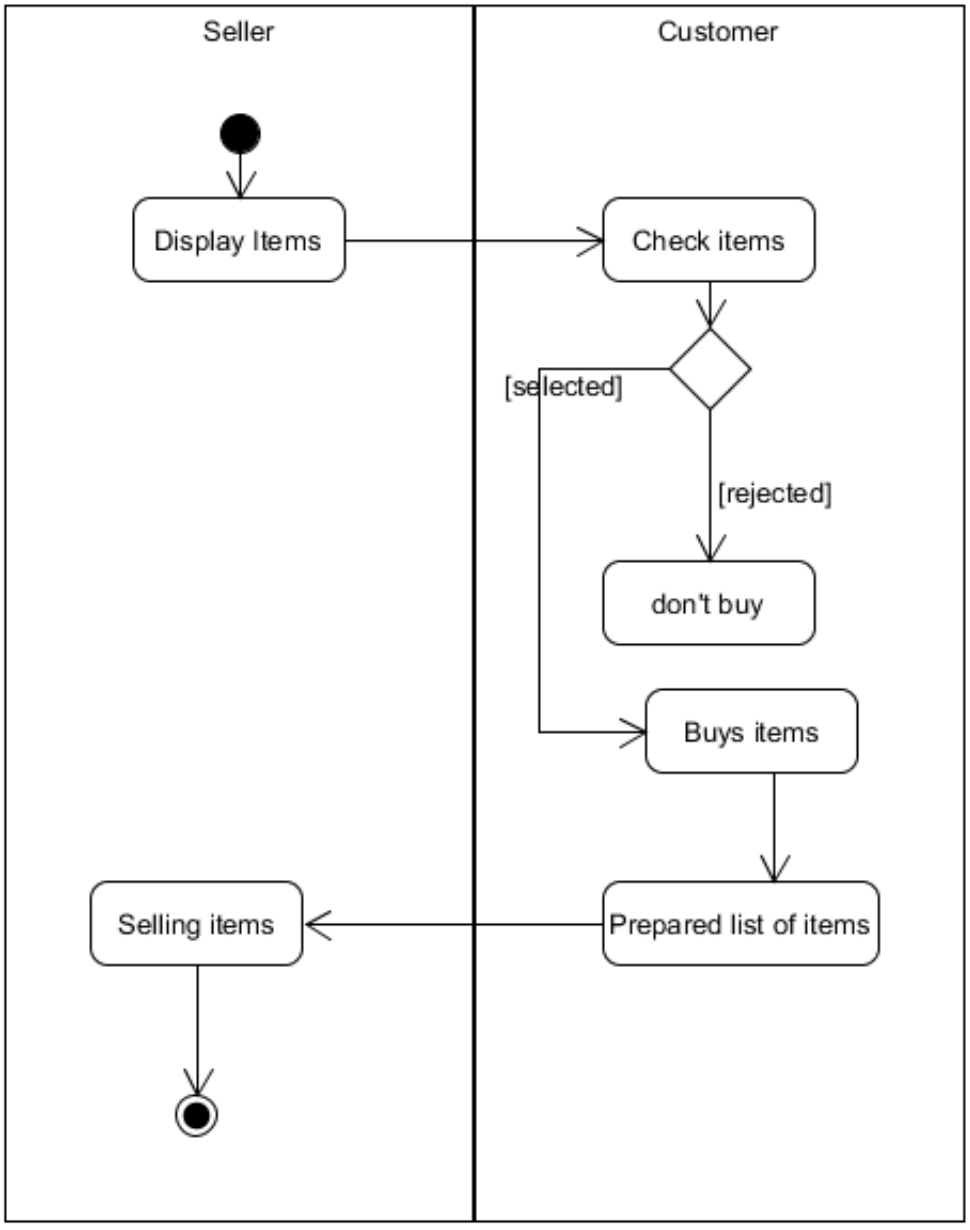

Figure 5:Selling & Buying items(Activity Diagram)

Description-

As demonstrated in the activity diagram, the seller is displaying items to the customer. Following that, the consumer inspects the items and purchases them. The customer then creates a list of items to purchase and sends it to the seller, who subsequently sells the items. On the other side, if the consumer rejects the items after inspecting them, he or she does not purchase them.

Creating bill

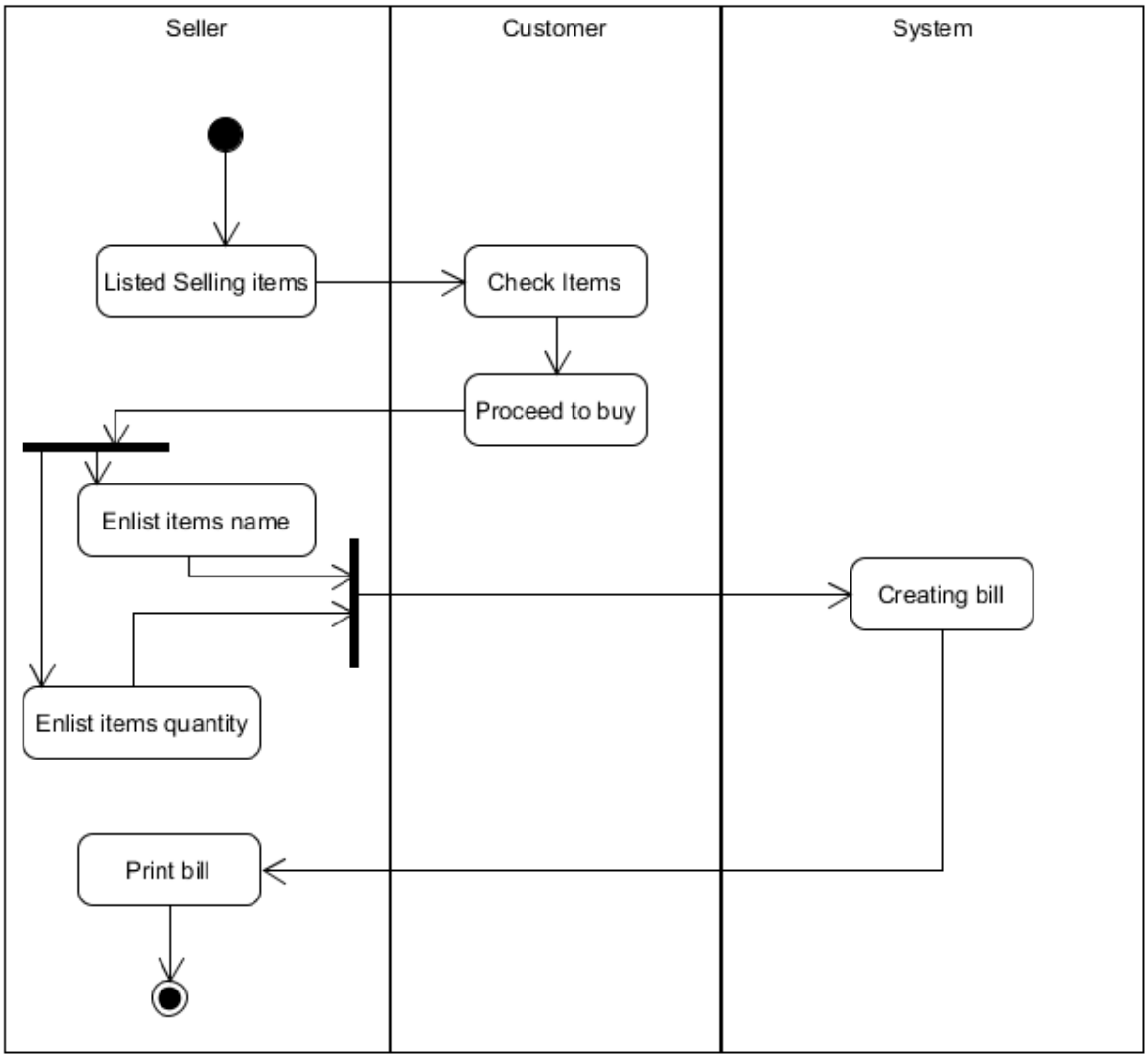

Figure 6:Creating bill(Activity Diagram)

Description-

In this ,it is seen that the seller listed all the selling items. Following that, the customer inspects the items and makes a purchase. The seller then classified the items according to their names and qualities. On the other hand, the system generates a bill for the items purchased and then sends the printout to the seller.

Make payment

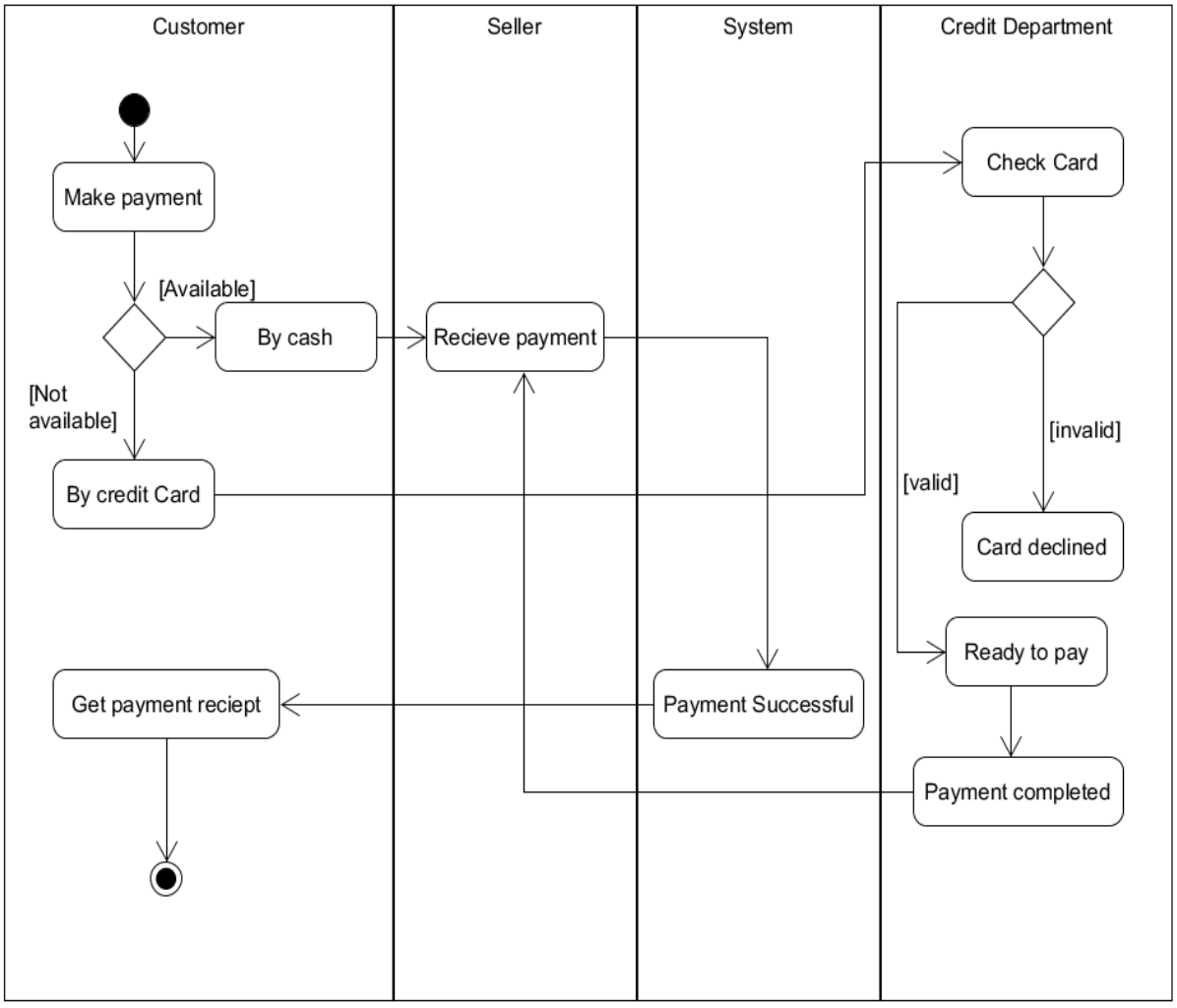

Figure 7:Make payment(Activity Diagram)

Description-

At the end of the activity diagram ,it is seen that the customer is making payment. If payment is made in cash, the seller receives the payments and the system records the transaction as successful, sending the customer a payment receipt. If, however, payment is made by credit card in term of no avaibility of cash, the legitimacy of the credit card is verified first. If the card is valid, it is eligible for payment, at which point the seller receives the payments. Once again, if the card is invalid, it will show card declined and the payment will be unsuccessful

Sequence Diagram

Login(SMS 01)

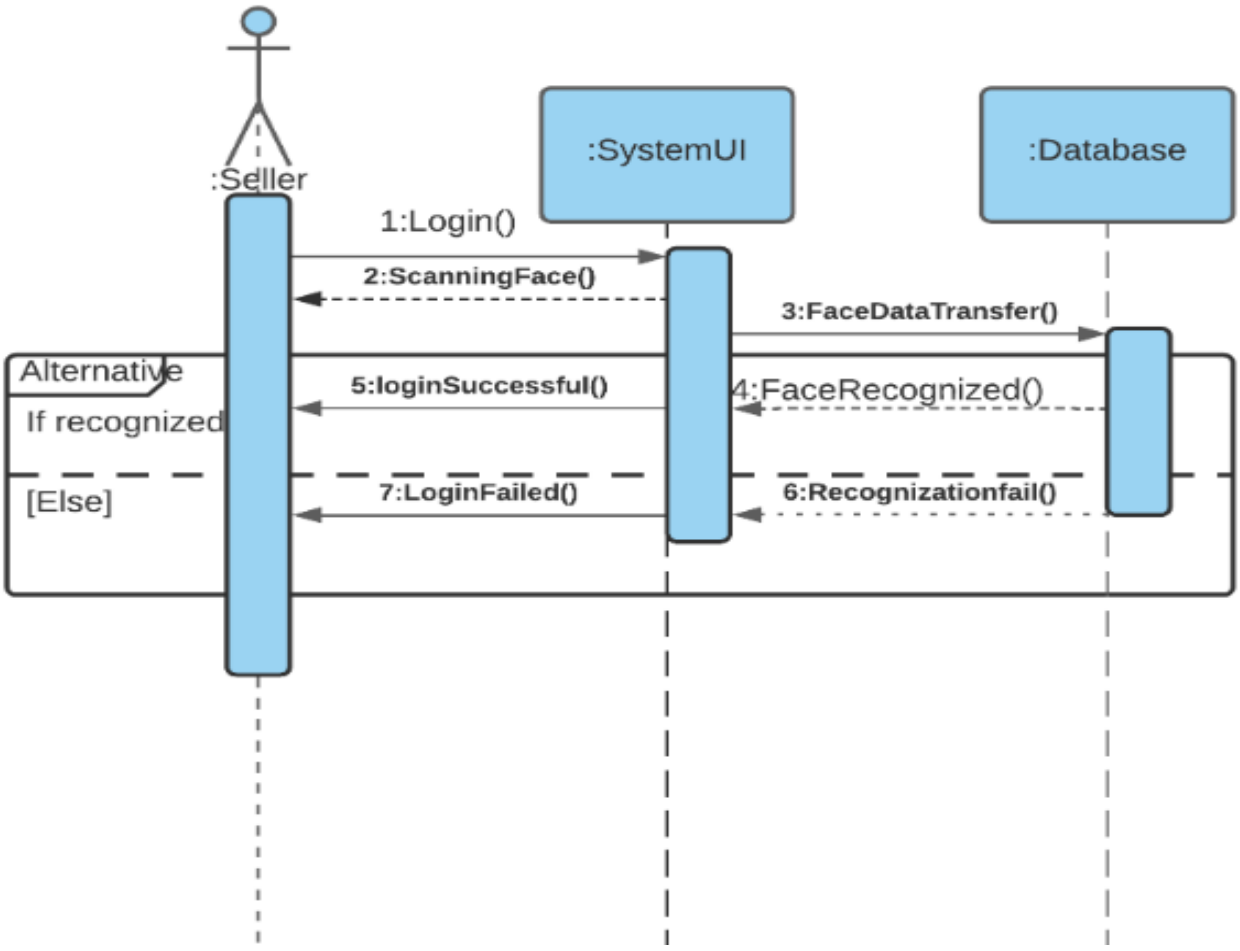

Figure 8:Login(Sequence diagram)

Description-

The sequence diagram that seller requires to login to the systemUI.At the initial stage the seller will scan the face as a password to the systemUI.After that the data of the face recognition will be saved and checked by the database.If the database verify the face then it will pass the information to the system UI and then systemUI will show login successful to the seller.Again,if the database is failed to recognized the desired face then it will transfer the information to the system .Then it will be impossible for the seller to Login in the systemUI.

Order Items(SMS02)

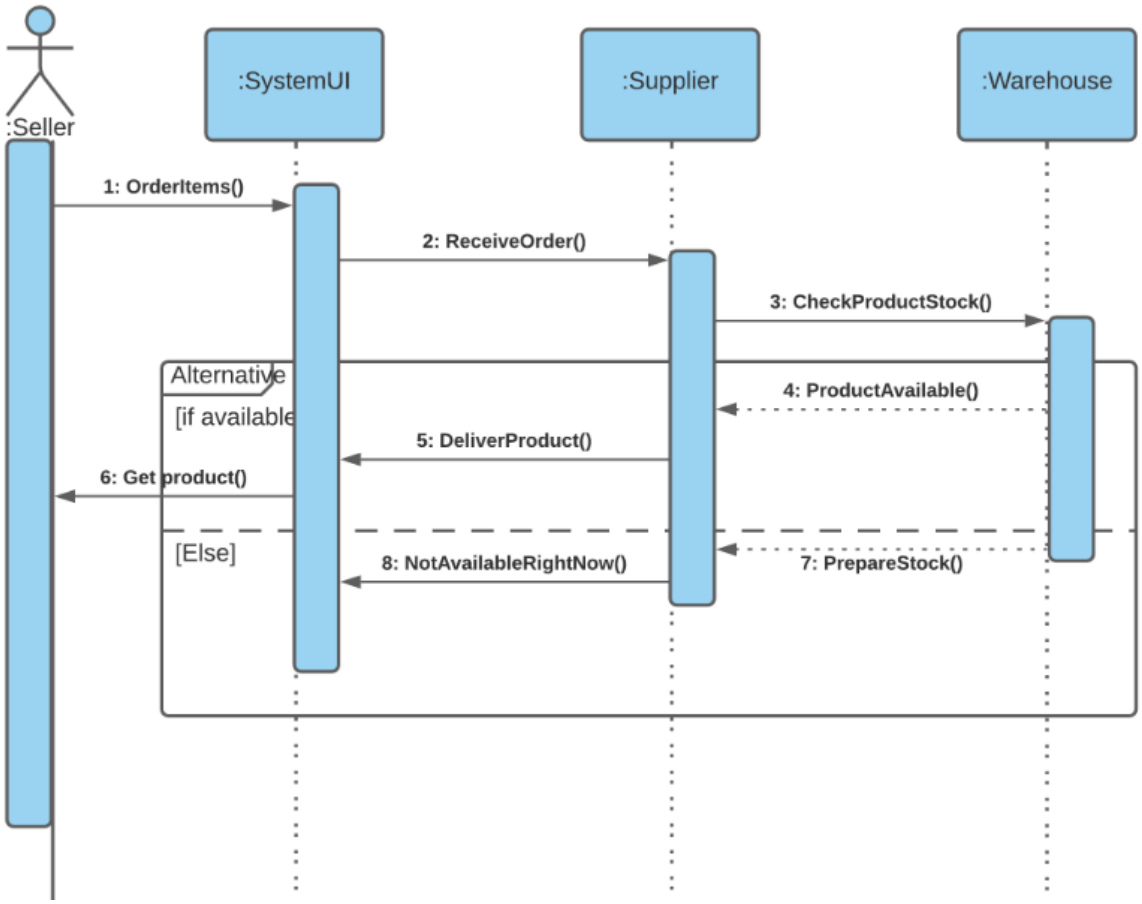

Figure 9:Order items(Sequence diagram)

Description-

In this sequence diagram, we can see the seller order items to the supplier. After that the order is received by the supplier. Then the next step is to check the product in the warehouse whether it is in stock or not. If the product is available, then it is carried by the supplier from the warehouse and delivered to the system UI delivered to the system UI caught. And finally then the systemUI directly handed over the product to the seller. Alternate is that when the product is out of stock then from the warehouse it directly informs through the systemUI.

Organize Items(SMS 03)

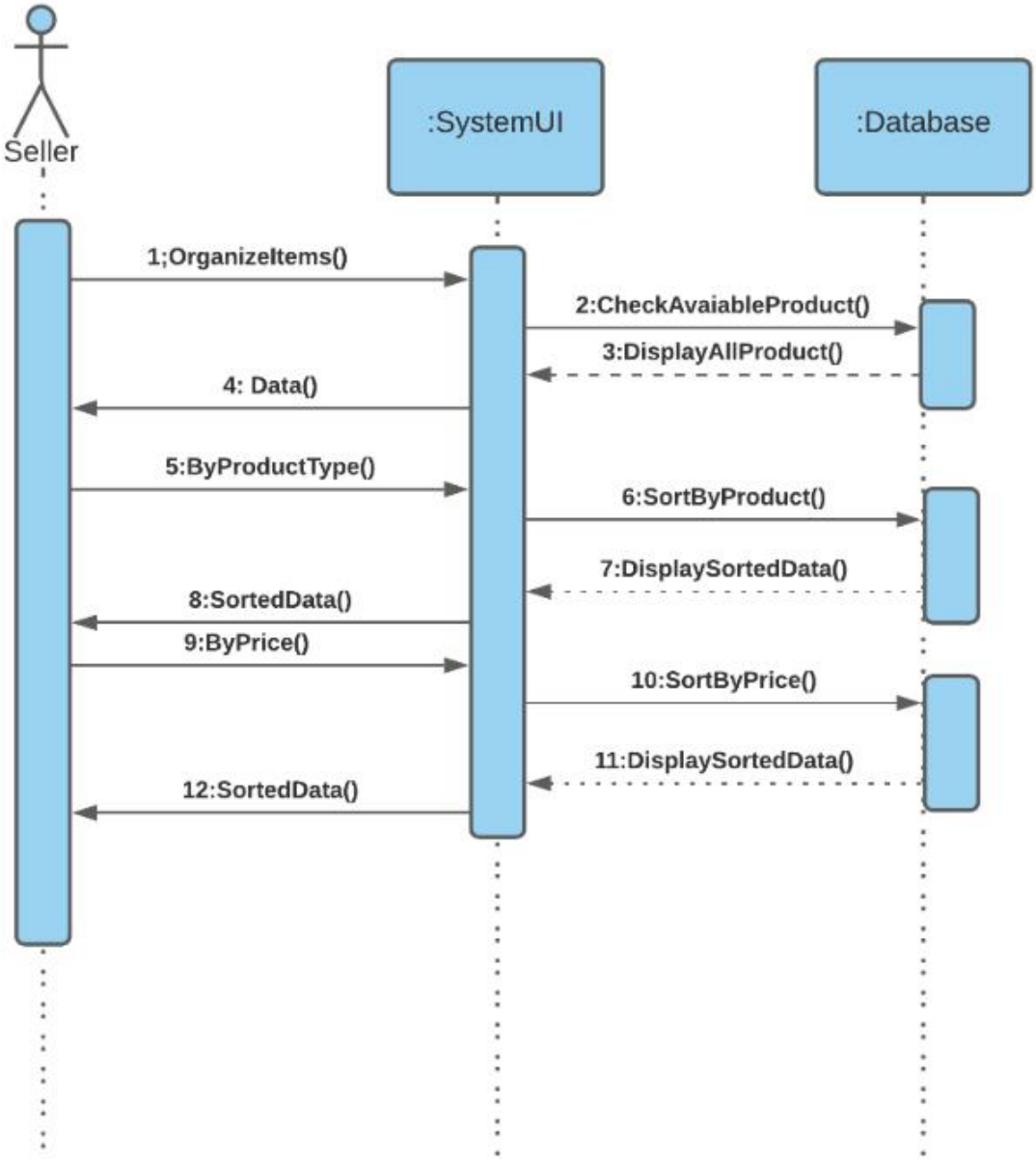

Figure 10:Organize items(Sequence diagram)

Description-

In this ,it is seen that the seller organizes the items it the systemUI. The systemUI checks the availability of the product in the database. Then after checking displays all the products and transform the data to the seller. Again the seller search product by its type and share it to the systemUI .The systemUI sort the product in the database and displays the sorted data of the product to the seller. Finally the Sorted Items will displayed to the seller.

Selling & Buying Items(SMS 04)

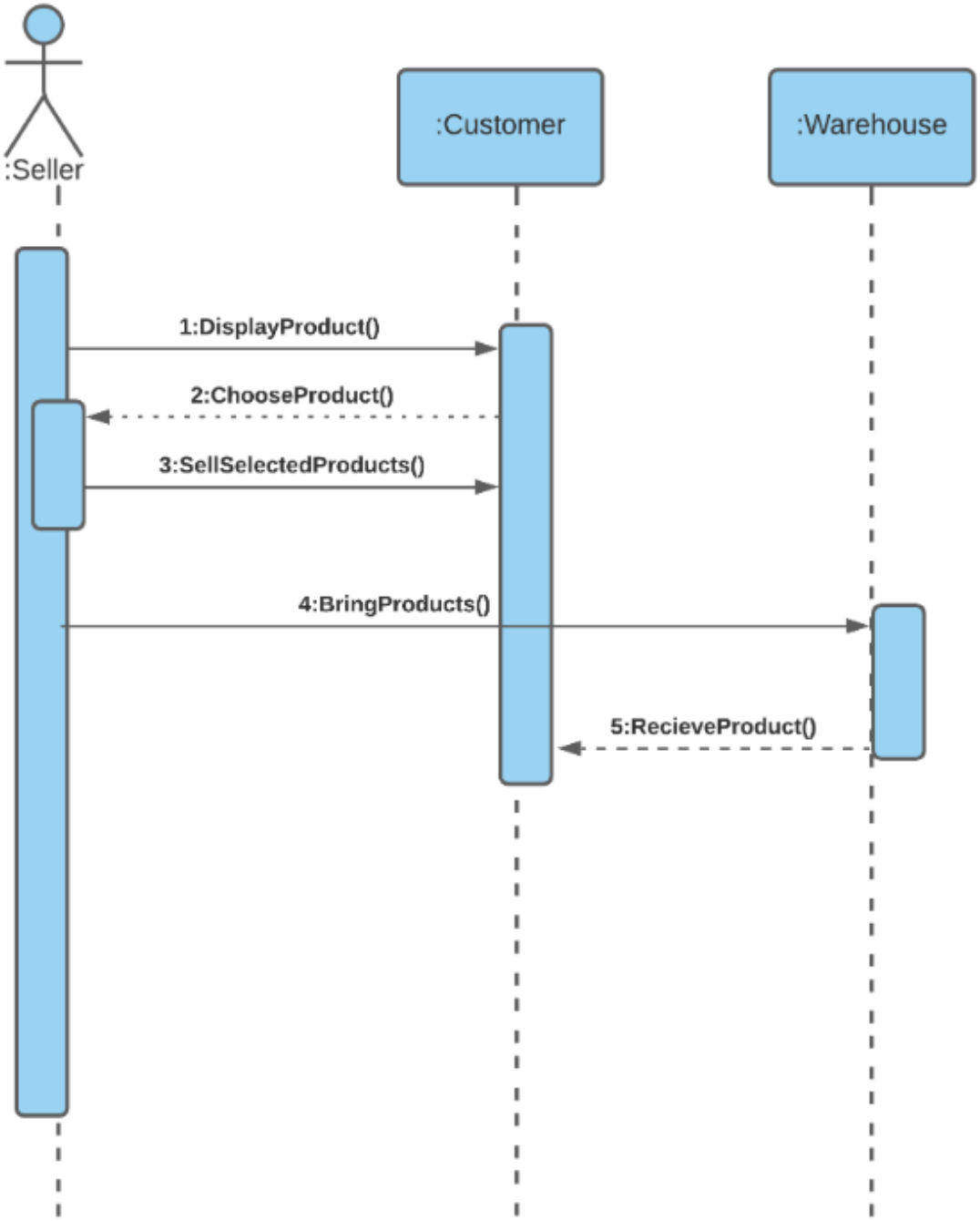

Figure 11:Selling & Buying items(Sequence diagram)

Description-

The sequence diagram shows that the seller displays the product to the customer. Customer can chose product from the store and select many items and many amount of items from the store. When the customer chooses the product, then the seller brings the product from the warehouse and sells to the customer.

Create Bill(SMS 05)

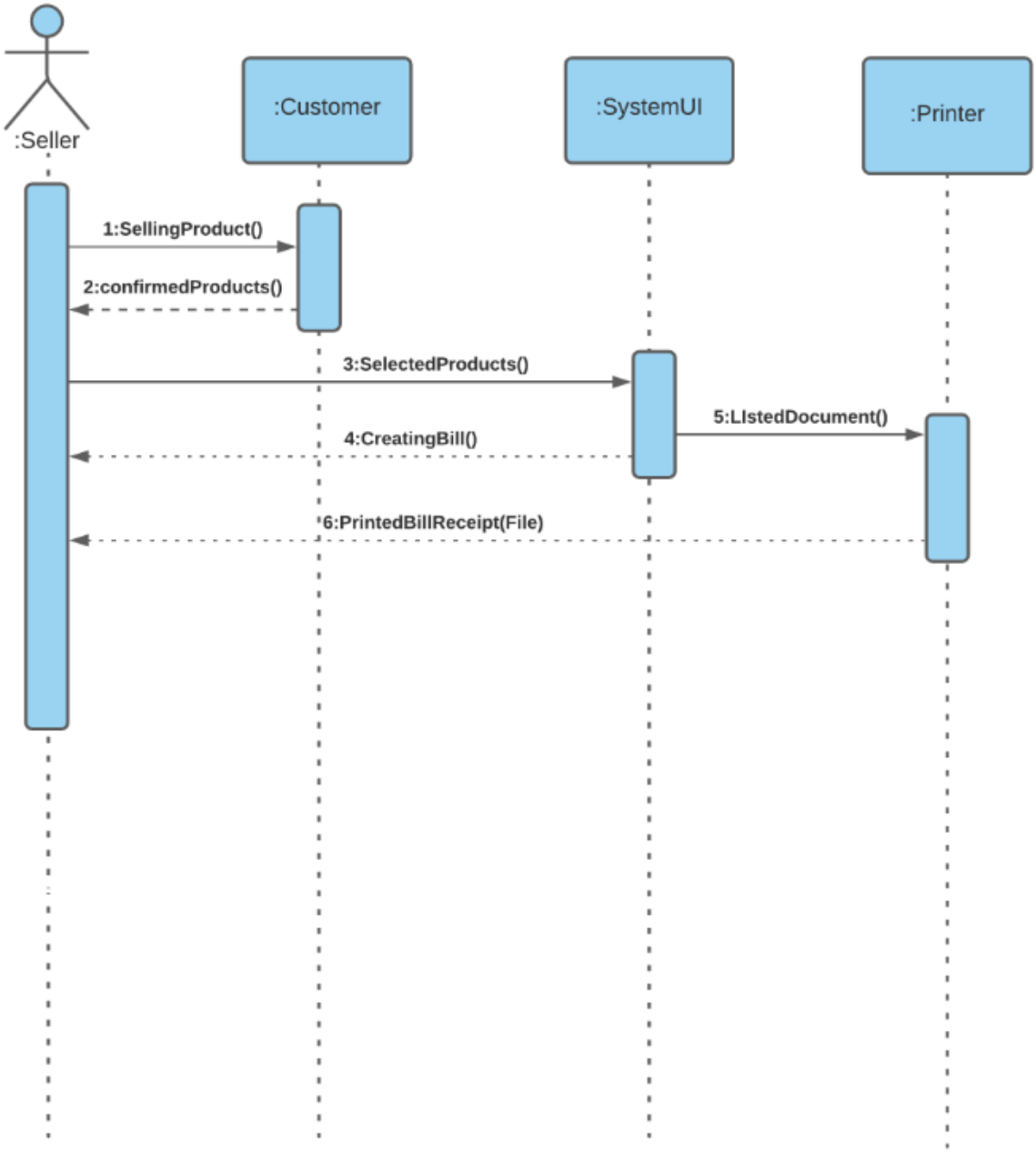

Figure 12:Create bill(Sequence diagram)

Description-

As in the sequence diagram it is seen that the seller is selling the product to the customer.And after confirming the product the customer notify it to the seller .After that the seller pass the selected product to the systemUI for creating the bill of the product. Again the systemUI listed the documents for printing and then the bill in printed and receive by the seller .

Make Payment(SMS 06)

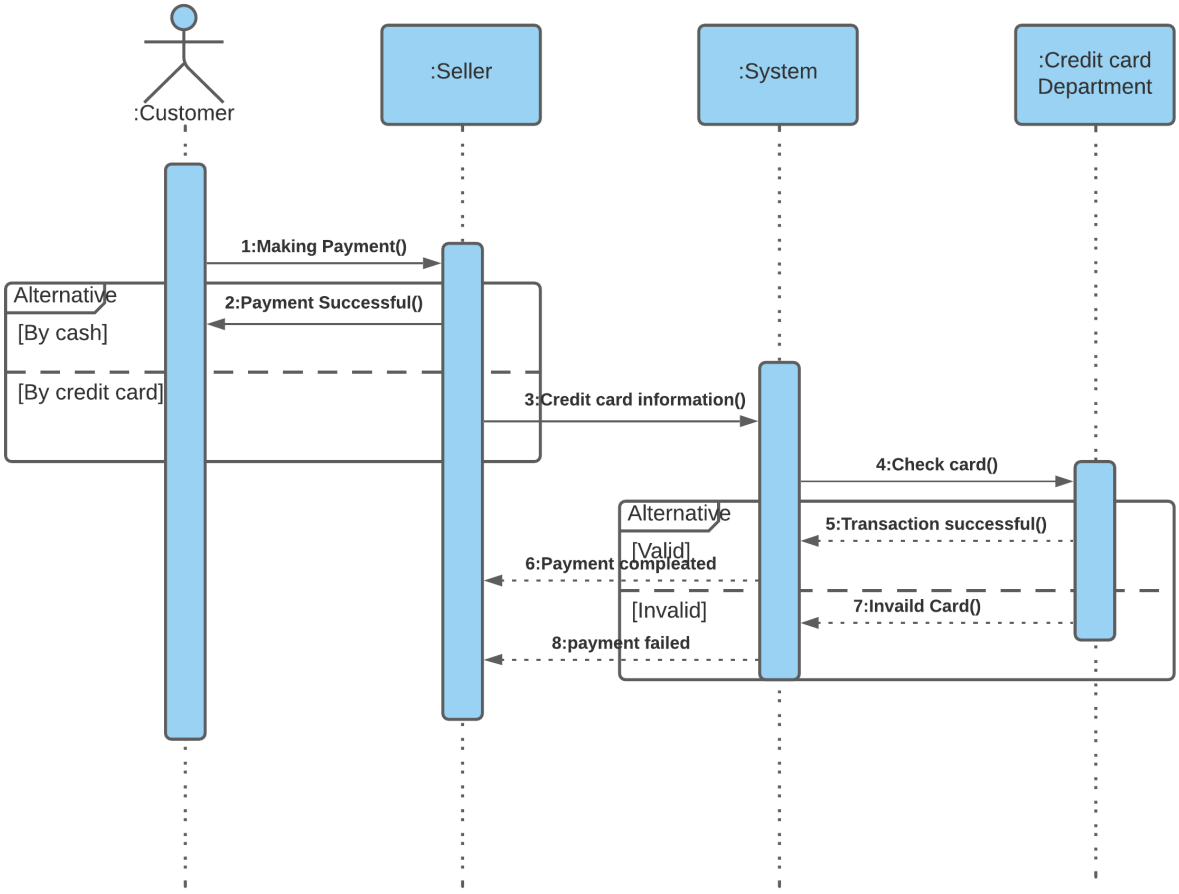

Figure 13:Make payment(Sequence diagram)

Description-

At the end part of the sequence it is seen that the customer is making the payment to the seller.If the payment is done by cash then it will show directly to the customer that the payment is successful.Again if the customer pays the payment with credit card then seller will pass the information of the credit card to the systemUI. Then the card information will be checked by the credit card department ,if the card is valid it will receive the payment and pass the information to the systemUI and shows transaction successful.Alternate, if the card in invalid the credit card department will show payment failed .After that the systemUI will show transaction is not possible to the seller.

Class Diagram

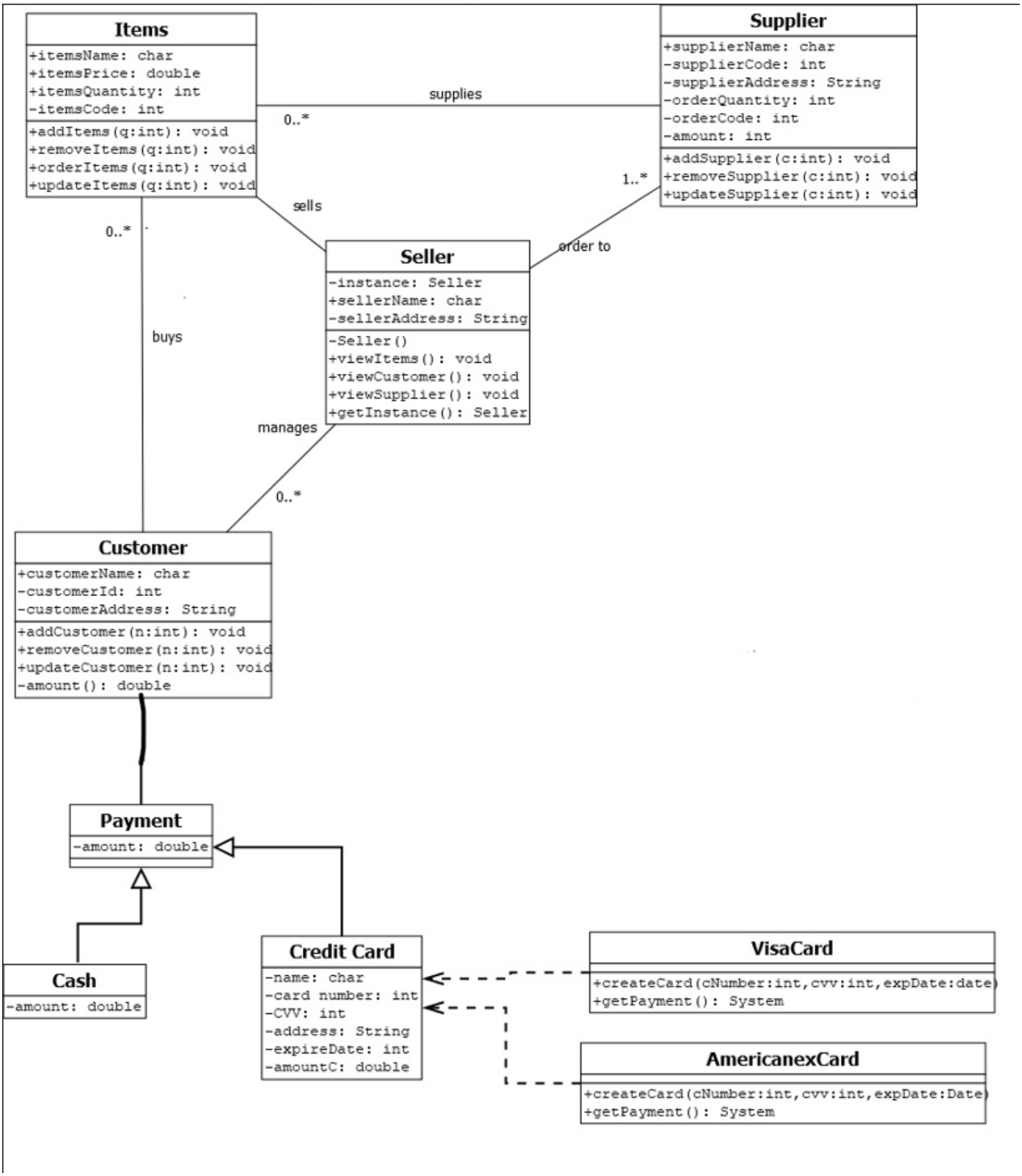

Figure 14:Class Diagram

Description-

In our system, we utilize a class diagram to visualize object-oriented classes and their interactions. This diagram shows 8 classes with relationships. Most of the class has their attributes and operations. And every class has a unique name. In a class '-' indicates private form of class and '+' indicates public form of class.int ,double,char,String are datatype. A supplier can supplies no item to many items. Customer can buys no item to many items as his /her wish. Seller can order items form supplier at least 1 number of quantity. Seller can sells items as well. Seller can manages customer too. He/she can manage no customer too many customer. In this diagram, cash and credit card is the child of payment class. Payment associate with customer. Then VisaCard and MasterCard interface CreditCard.

Utilization of Software Design Principles and Patterns

We have implemented a number of principles and patterns in our system. The Open-Closed Principle (OCP), Liskov Substitution Principle (LSP), Dependency Inversion Principle (DIP), and Law of Demeter are the principles we adopted (LOD). Singleton Pattern, State Pattern, and Factory Pattern are the patterns we implemented.

- Open-Closed Principle (OCP): The entity is open for extension but closed for alteration, according to the statement. The Payment class has been configured to be an interface. When a new type of payment method is added to the system's support list, the new class can extend the Payment Class. Only a minor modification is required. When all attribute assessors are set to private. The characteristics are not accessible to other classes. Then, when we need to edit the code, we can do so within the class itself.
- Liskov Substitution Principle (LSP):The following two subclasses (class VisaCard and class AmericanexCard) are fully interchangeable with its parent class (class creditCard). The methods in the interface class creditCard, for example, are completely implemented in their derived subclasses VisaCard and AmericanexCard. As a result, even if we replace the creditCard interface with either VisaCard or AmericanexCard, our system will continue to function normally. As a result, LSP is applicable in our system since the derived subclasses VisaCard and AmericanexCard have the same interface implementation characteristics like creditCard.
- Dependency Inversion Principle (DIP): One payment class depends on the creditCard Interface rather than the concrete subclass of creditCard in the creditCard class. This may be accomplished through the use of polymorphism. DIP offers a better option to enhance the function using this architecture.
- Law of Demeter (LOD): It will maintain its own creditCard class under the payment class. The creditCard object's assessor is private. There are also no getter and setter methods for the object. Object encapsulation is the term for this. Other classes can't directly access the creditCard class because of encapsulation. Object encapsulation can be used to accomplish LoD.

Singleton

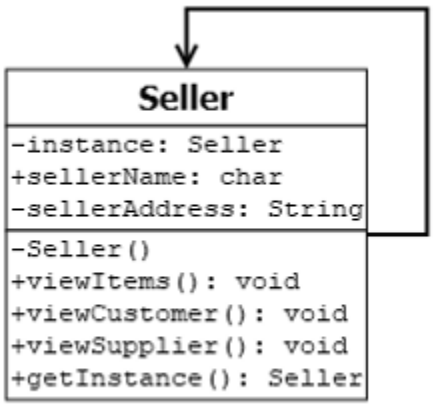

Figure 15:Singleton pattern

It can be useful to have only one instance of a class at times. In a system, for example, there should only be one system manager. Singletons are commonly used for centralized management of internal or external resources, as well as providing a single point of access to themselves. One of the design patterns is the singleton pattern. In our case, we can see that there is only one class involved, which is responsible for instantiating itself, ensuring that it does not create more than one instance, and providing a global point of access to that instance. As a result, the same instance can be used everywhere and the constructor can be called directly each time.

State Pattern

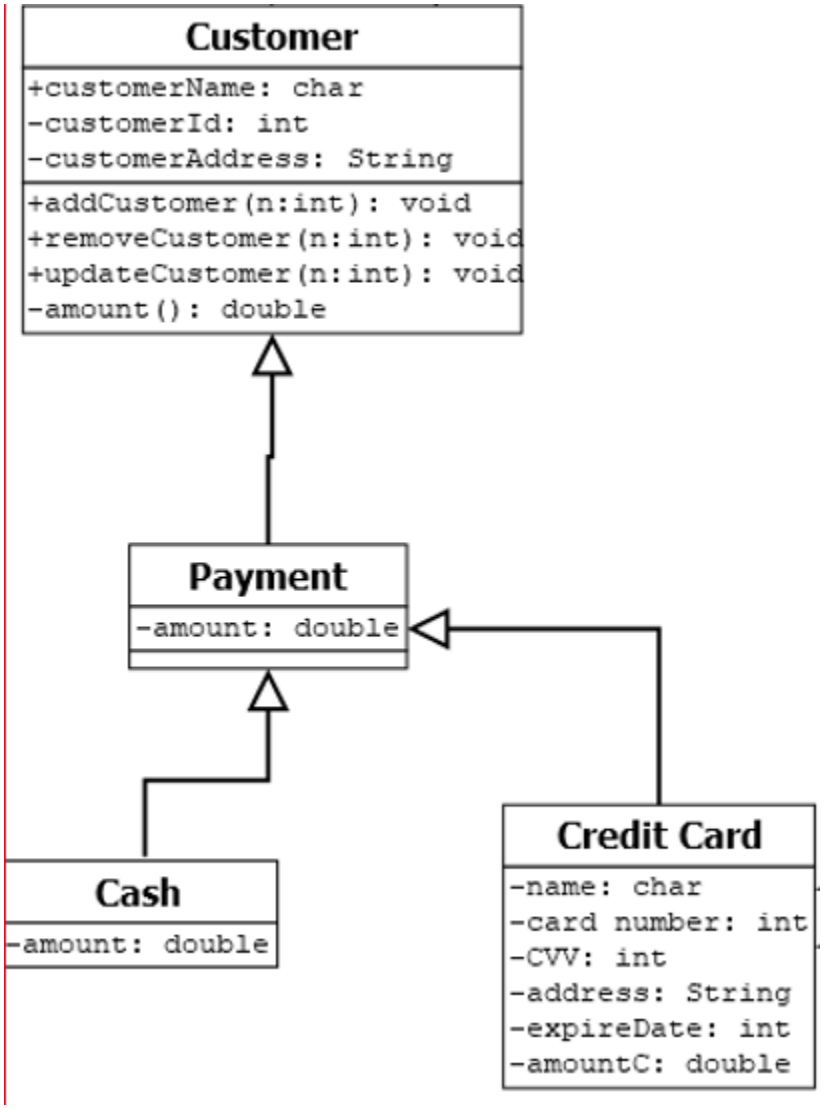

Figure 16:State patterns

In terms of the state pattern, as shown in the diagram above, each buy item payment method will have a payment method state that is either cash or credit card. The state pattern was chosen because it allows us to change the payment method state quickly and easily, and it allows us to define different things to deal with for each type of payment method state.

Factory Pattern

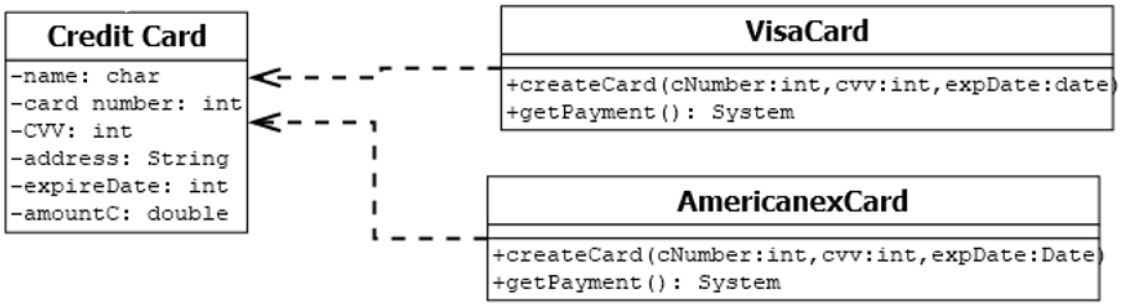

Figure 17:Factory patterns

Product(Credit card) defines the interface of objects the factory method creates. ConcreteProduct (VisaCard,AmericanexCard) implements the Product interface. Declares createCard which returns an object of type CreditCard. Also, may define get method to get a CreditCard object. The factory pattern is used here because it is one of the most popularly used Design Patterns in SE and it provides a mechanism for producing various objects

System Prototype

Login

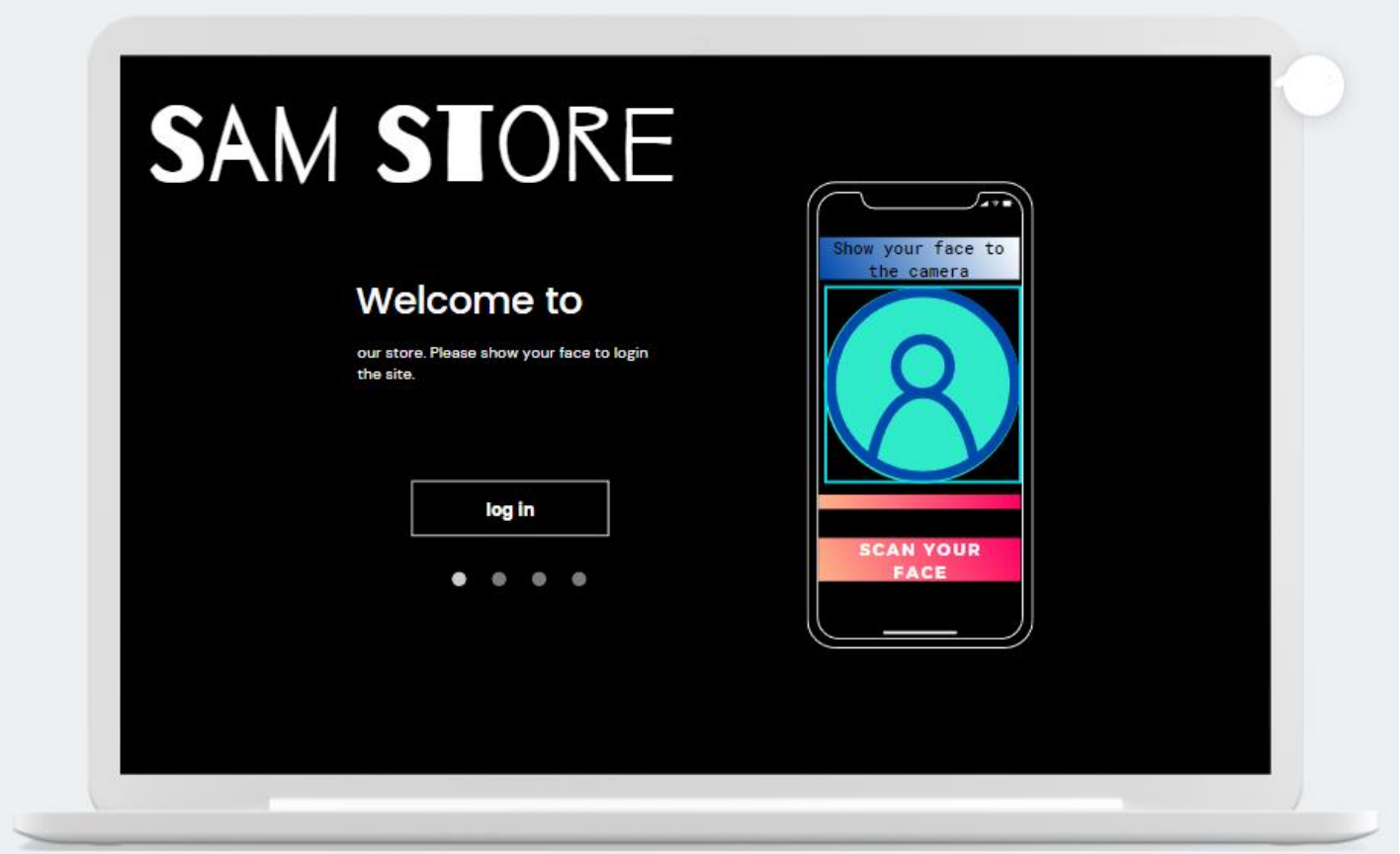

Figure 18:Login prototype

Order Items

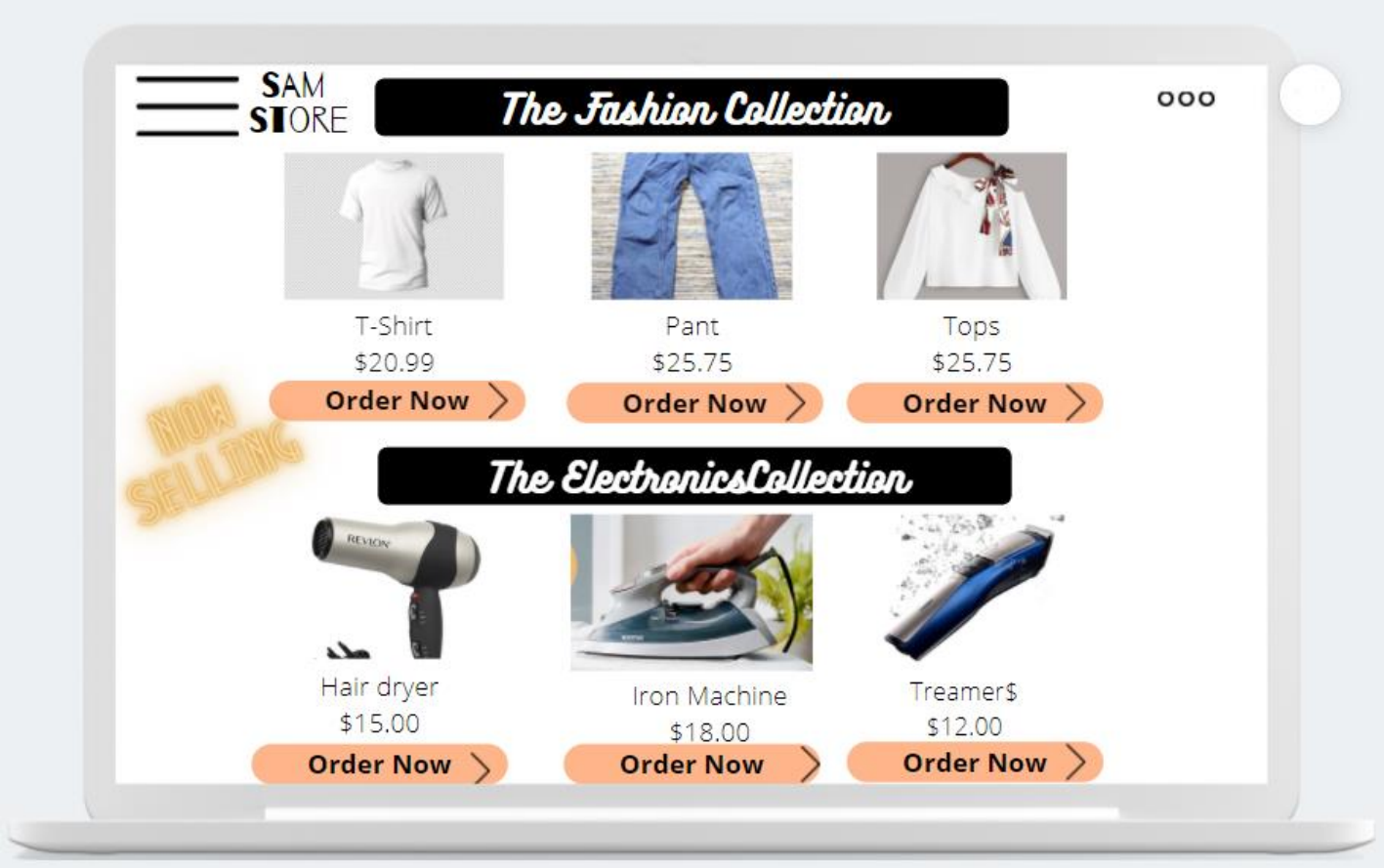

Figure 19:Order items prototype

Organize Items

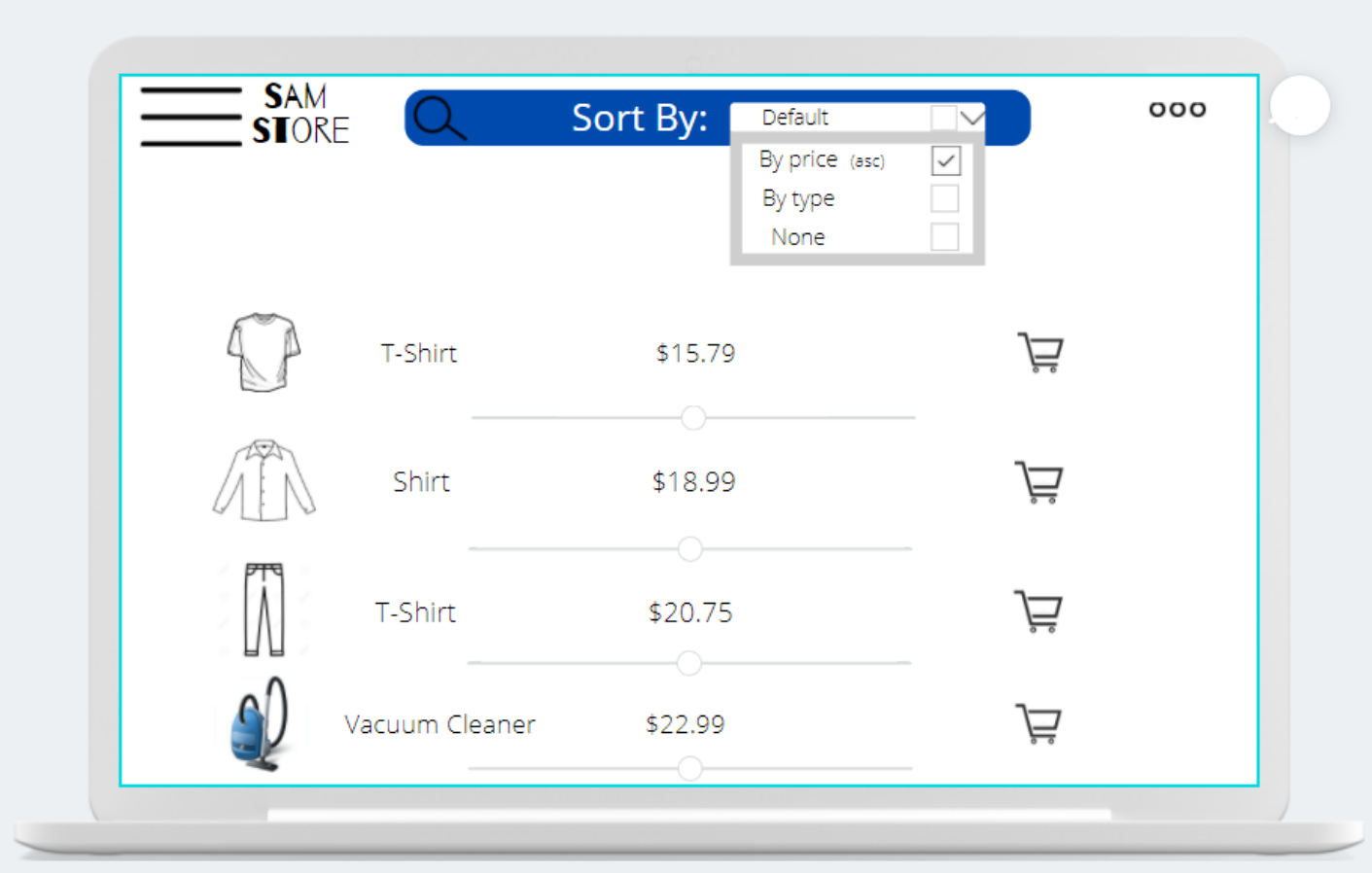

Figure 20:Organize items prototype

Selling & Buying items

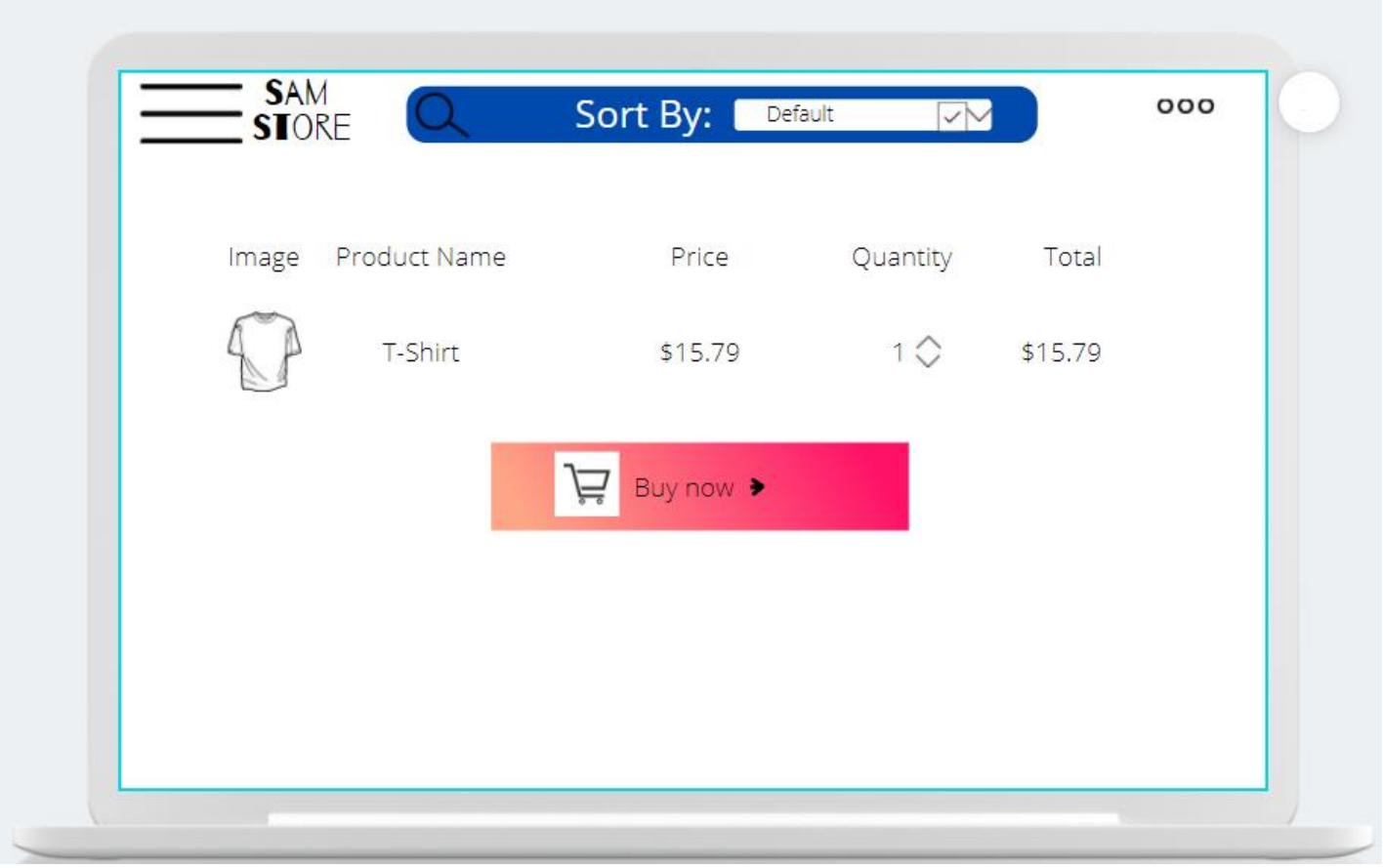

Figure 21:Selling & Buying items prototype

Create bill

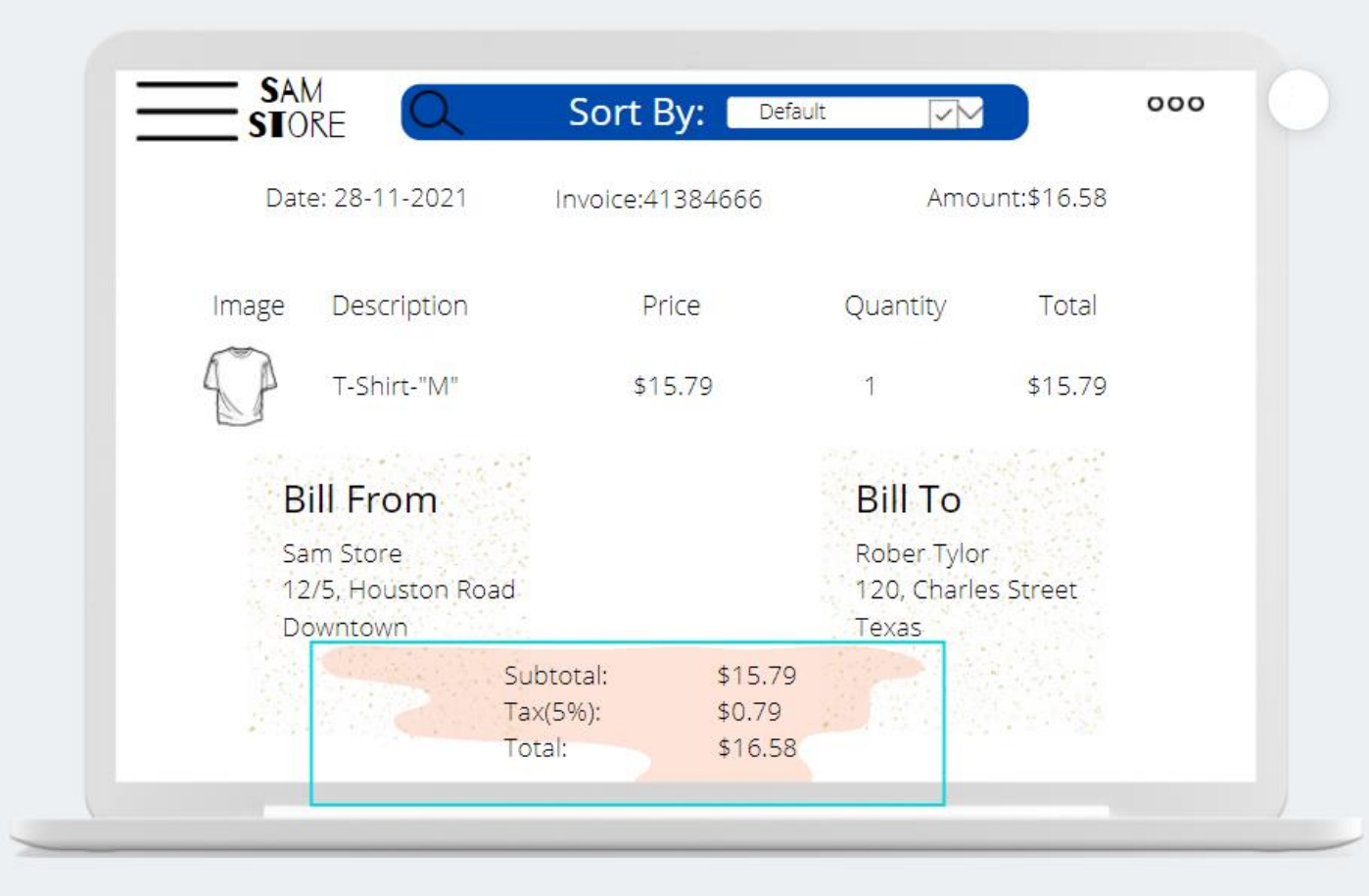

Figure 22:Create bill prototype

Make Payment

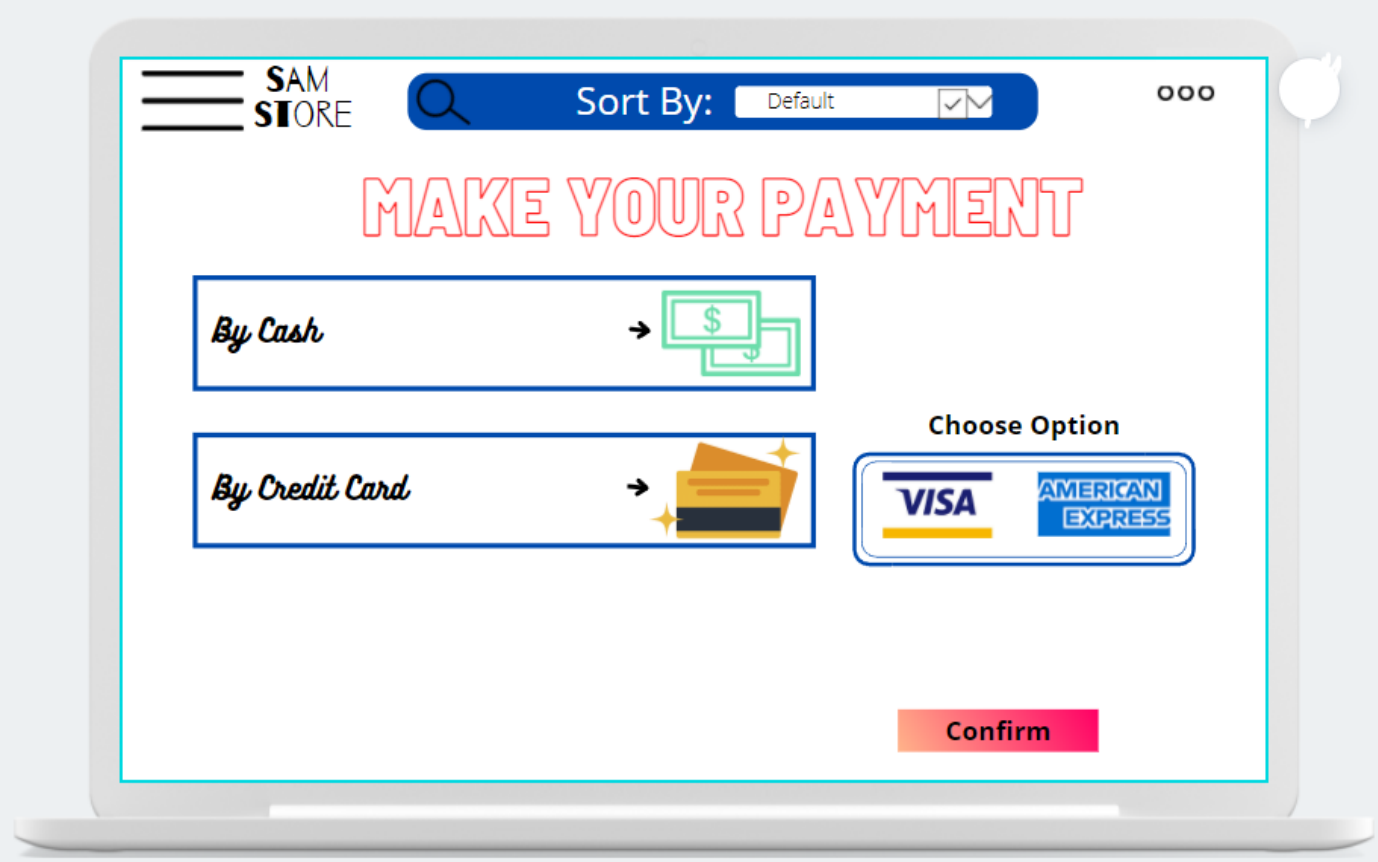

Figure 23:Make payment prototype

Conclusion

We have included the design step of developing a Store Management System in this project, which includes gathering user requirements and UML modeling.In this project report, we have discussed about the management system of store in which way it manages the items and the customer.Here the items management and items sell between the seller and customer are showed in a beneficial manner. This report displays our approach for resolving the shop's current challenges and assisting the store in generating revenue.

- For security and data privacy out system buildup with a face recognition system. Seller can only get access to the system through the face recognition method. The system will also save the face data in database for future query.
- The store management system in very useful but this have some restriction. A customer cannot buy any product in online, and the store management system don’t have any home delivery method. The Store management system don’t have any monthly EMI method which benefits an EMI for borrowers , that they know precisely how much money they will need to pay toward their loan each month, which can make personal budgeting easier.
- The system is very user friendly for the customer because seller can organize the items of the products with his/her visual comfort type.
- I think if we want to improve anything we should make everything online process where customer can buy buys online, pay online and receive the product with home delivery method. Nowadays people are attracted towards online shopping websites instead of a physical shop because it is very user friendly, and it saves people’s important time.

Finally, the goal of the 'Stores Management System' is to automate almost all of the processes mentioned above in order to reduce the clerical labor of the staff working in Stores, both technical and accounting departments, by utilizing the software industry's latest technologies and cost-effective tools, thereby giving management better control by avoiding manual errors, etc.