



Arts and Management System

System Specification
December 6, 2017

Author
Project Manager – Sion Park

PenSoulutions

Table of Contents

| | Pages |
|---|----------|
| Executive Summary..... | 1 |
| | |
| 1.0 Introduction | |
| 1.1 Problem statement/ Project Vision..... | 2 |
| 1.2 System Services..... | 2 |
| 1.3 Nonfunctional Requirements and Design Constraints | 3 |
| 1.4 System Evolution..... | 3 |
| 1.5 Document Outline | 4 |
| | |
| 2.0 Structural Model | |
| 2.1 Introduction..... | 5 |
| 2.2 Class Diagram..... | 5 |
| 2.3 Metadata..... | 6 - 16 |
| | |
| 3.0 Architecture Design | |
| 3.1 Introduction..... | 17 |
| 3.2 Infrastructure Model | |
| 3.2.1 Deployment Diagram 1 – Architecture Overview..... | 18 |
| 3.2.2 Deployment Diagram 2 – Nodes and Artifacts..... | 19 |
| 3.3 Hardware and Software Requirements | |
| 3.3.1 Required Hardware Components..... | 20 |
| 3.3.2 Required Software Components..... | 20 |
| 3.4 Security Plan..... | 21 |
| | |
| 4.0 User-Interface | |
| 4.1 User-interface Requirements and Constraints..... | 22 |
| 4.2 Window Navigation Diagram..... | 23 |
| 4.3 Forms: Screen/ User-Interaction Design..... | 24 -27 |
| 4.4 Reports: “Printed Output” Design..... | 28-29 |
| | |
| 5.0 Appendices | |
| 5.1 Bibliography/Reference..... | 30 |



PenSoulutions

Executive Summary

Artisan's Co-op North (ACoN) has consulted PenSoulutions to design and develop a system that will automate the process of keeping track of their inventory and sales. The system will be responsible for managing all the transactions and operations of the company. The primary goal of the system is to increase productivity of ACoN and reduce the workload by implementing automation.

We at PenSoulutions created the system proposal for Arts and Sales Management System(ASMS), which includes the following:

- Introduction and Overview of ASMS
- System Request and our response to the request
- Feasibility Assessment
- System requirements
- Use-Case Diagram and its descriptions
- System Evolution

ACoN and PenSoulutions have concluded that the system is feasible and will benefit its users. We are now moving on to the design and modeling of the system. This document serves as an overview of the system design and its software and hardware architecture. It includes a brief introduction of the system, a structural model, two different infrastructure models, a security plan, and basic user-interaction designs. This document is meant to provide a foundation for the development of ASMS.



PenSoulutions

1.0 Introduction

1.1 Problem statement/ Project Vision

PenSoulutions is designing the Arts and Sales Management System (ASMS) that will reduce workload and support the business management in a more computerized way. We will be implementing automation which will give access to each artisan of ACoN to track their inventory and generate reports. ASMS will also automate data entry with the use of a barcode system and make sales available online. ASMS is meant to improve workflow efficiency by removing unnecessary delay and provide convenient environment managing the business.

The purpose of ASMS is to provide an efficient way to manage inventory, sales, and accounting for both customers and the members of ACoN. PenSoulutions envisions to develop the system that meets all the requirements listed by ACoN and benefits everyone who uses the system.

The system will not be expanding globally in the near future. Nonetheless, we hope to build a system that will help ACoN run its business in the most efficient way within a reasonable budget and timeline.

1.2 System Services

This section briefly describes the functional requirements, which are written in more detail in the System Proposal. Please refer to the section 4 of the System Proposal.

- **Account Management**
 - Create account to purchase products online, manage sales and inventory.
(Refer to Use Case 1,1.1)
- **Inventory Management**
 - Generate barcode for each item to add item and manage the flow of the inventory. (Refer to Use Case 2, 3, 4)
- **Report Management**
 - Generate report for inventory and payment. (Refer to Use Case 5)
- **Payment Management**
 - Deposit payment to seller's account every first day of the month. (Refer to Use Case 6)
- **Customer online order (Website Management)**
 - Browse and purchase products online through the ACoN's website. (Refer to Use Case 7)



PenSoulutions

1.3 Nonfunctional Requirements and Design Constraints

The following is a summary of major nonfunctional requirements and design constraints to consider when implementing ASMS.

- The system must be compatible with major browsers including Google Chrome, Microsoft edge, Apple Safari, Mozilla Firefox and Internet Explorer.
- The development and maintenance of ASMS must be cost effective. It should focus on its functionality and requirements. We will make sure there is no need to rework or extra programs that are not necessary to the system.
- ASMS needs to be available for all the users without additional costs, which means ASMS must be compatible with as many devices as possible.
- ASMS must be user friendly. Since the members of ACON have different technological background, we must consider developing a system that is simple and easy to learn, use, and troubleshoot.
- The system must be portable, meaning it should be able to move to different hardware or operating system environment since ACoN will be opening booths at events or fairs.
- All user data must be protected and passwords must be encrypted.
- All the payment information must be confidential and only accessible to those who are associated with it.
- The system must be reliable (have back up plans) and consistently perform according to its specification.

1.4 System Evolution

The initial version of the system will have all the core and require features and functions incorporated. As we release the first version of the system, we will be working on the second version of the system which will include online shopping feature. This will enable customers not only view the items available but also order items without having to visit events or craft fairs. We will also be responsible for making the website available on smartphones and tablets. We want to deliver the second version of Arts and Sales Management System (ASMS) immediately after the first version.

We value our users' comments and feedback and we will seriously take those to improve our system. It could be adding new features or altering the way the system functions. Our goal is to provide a system that is user-friendly and useful for the system users.



PenSoulutions

1.5 Document Outline

This document currently contains the following sections:

- Structural Model: Includes A class diagram, and associated class metadata, that shows how objects and stored data will interact in ASMS
- Architecture Design: Two deployment diagrams to show the physical architecture of CCS. Also included are hardware and software requirements and a system security plan.
- User-Interface: Forms, screens and basic requirements and constraints for the user-interface design of CCS. Also included are layout designs for printed reports.
- Appendices: Includes bibliography and supporting documentation.

PenSoulutions

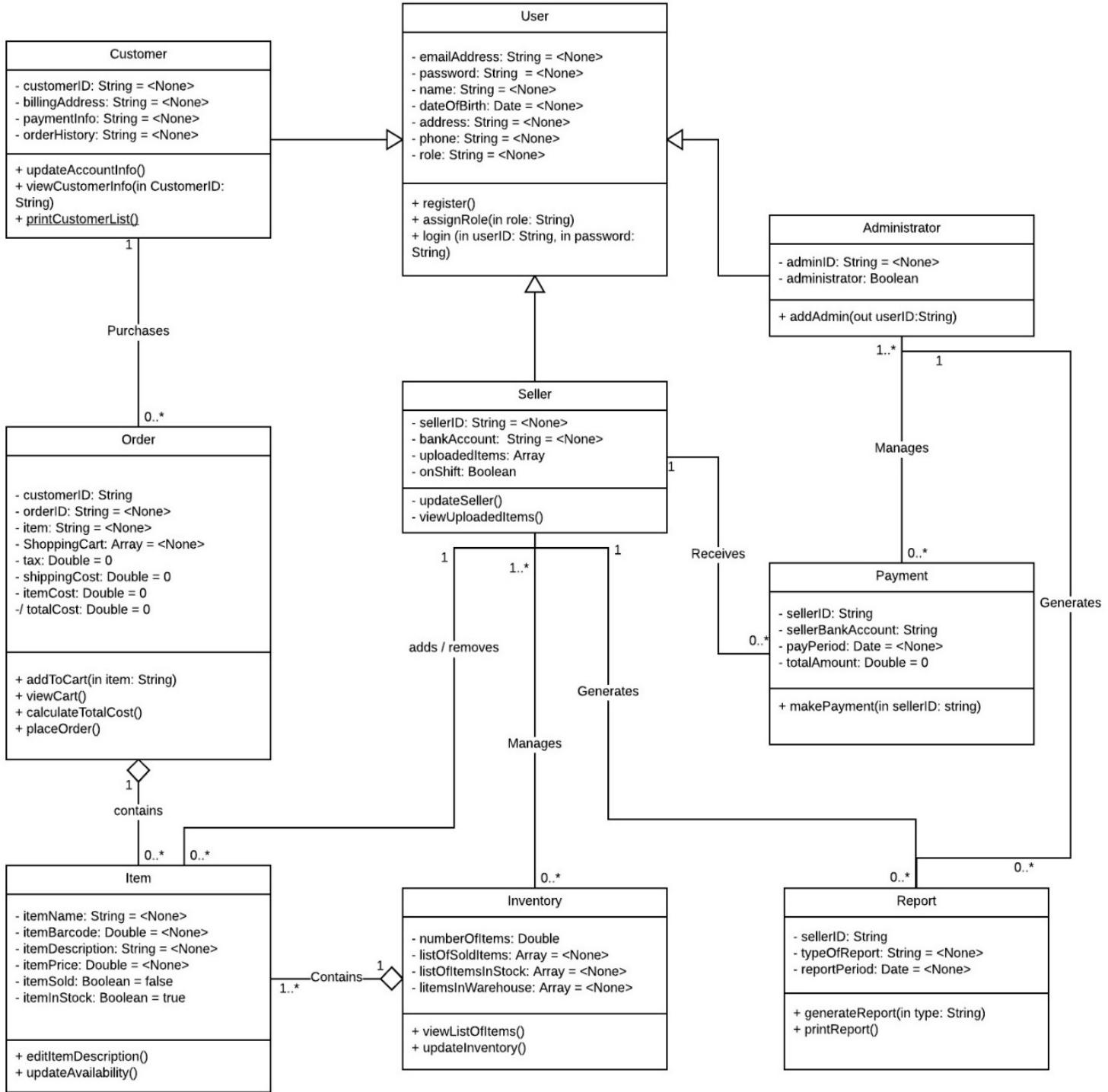
2.0 Structural Model

2.1 Introduction

This section contains a class diagram for ASMS and metadata for each class in the diagram.

Each class is described with more detail about their attributes, operations, and processing outline.

2.2 Class Diagram



PenSoulutions

2.3 Metadata

Table 1: User Class

| User |
|--|
| - emailAddress: String = <None> - password: String = <None> - name: String = <None> - dateOfBirth: Date = <None> - address: String = <None> - phone: String = <None> - role: String = <None> |
| + register() + assignRole(in role: String) + login (in userID: String, in password: String) |

Description: Represents a generic account for customer, seller, and administrator for Artisan's Co-op North

Visibility: Public

Is Abstract: Yes

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|--------------|---|------------|--------------|
| emailAddress | User email address | No | 1 |
| password | User password | No | 1 |
| name | Last, First name | No | 1 |
| dateOfBirth | User date of birth | No | 1 |
| address | User local address | No | 1 |
| phone | Phone number with area code | No | 1..* |
| role | User role (customer, seller, administrator) | No | 1..* |

PenSoulutions

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|------------|---|-----------|-----------------|
| register | The system registers a user using their email address and password. | No | No |
| assignRole | Assigns a role to a user based on the user's choice. | No | No |
| login | Logs into user's account. | Yes | No |

Processing Outlines ->

Register ()

Validate all the information (attributes) user entered
Confirm user's emailAddress and password
Register user information into the system and send confirmation link to user's entered email

assignRole (in role: String)

Given a role that user selected,
IF Customer or Seller
Display information and rules of their role and save user's chosen role

EISE

Send an alert to existing administrators for permission and display "confirmation in the process"

login(in userID: String, in password: String)

Validate entered userID (customerID, sellerID, adminID or email Address) and Password
IF valid

User accesses their account

EISE

Return an incorrect username and password alert message

PenSoulutions

Table 2: Administrator Class

| |
|--|
| Administrator |
| - adminID: String = <None> - administrator: Boolean |
| + addAdmin(out userID:String) |

Description: Represents an Artisan's Co-op North Administrator

Visibility: Public

Is Abstract: No

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|---------------|--|------------|--------------|
| adminID | Unique identifier assigned to each administrator | Yes | 1 |
| administrator | Checks for eligibility to be an administrator | No | N/A |

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|----------|--|-----------|-----------------|
| addAdmin | Allows administrator to add an administrator or give permission to be an administrator | No | No |

Processing Outlines ->

addAdmin(out userID: String)

 Update userID to an adminID

 Add the updated adminID to the system

 Send an email to the user notifying that their account has been updated

PenSoulutions

Table 3: Seller Class

| Seller | |
|---|---|
| - sellerID: String = <None> - bankAccount: String = <None> - uploadedItems: Array - onShift: Boolean | Description: Represents an Artisan's Co-op North seller Visibility: Public Is Abstract: No |
| - updateSeller() - viewUploadedItems() | |

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|---------------|--|------------|--------------|
| sellerID | Unique identifier assigned to each seller | Yes | 1 |
| bankAccount | Seller's bank account to receive monthly payment | No | 1 |
| uploadedItems | List of items that seller has added to the system | No | 0..* |
| onShift | Whether the seller is in charge of Artisan's Co-op North booth at a fair | No | N/A |

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|-------------------|---|-----------|-----------------|
| updateSeller | Update seller information | No | No |
| viewUploadedItems | Displays a list of items uploaded by seller | No | No |

Processing Outlines ->

updateSeller()

Seller entered new information or edits the existing information
 IF valid and user clicks “confirm update”
 Remove the old information and save the updated seller information
 Else
 Display the “Account can’t be updated” message

viewUploadedItems()

Find all the items that are uploaded by the seller
 Display the list of all items uploaded by the seller

PenSoulutions

Table 4: Customer Class

| Customer | |
|---|--|
| <ul style="list-style-type: none"> - customerID: String = <None> - billingAddress: String = <None> - paymentInfo: String = <None> - orderHistory: String = <None> | Description: Represents an Artisan's Co-op North customer Visibility: Public Is Abstract: No |
| <ul style="list-style-type: none"> + updateAccountInfo() + viewCustomerInfo(in CustomerID: String) + printCustomerList() | |

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|----------------|---|------------|--------------|
| customerID | Unique identifier assigned to each customer | Yes | 1 |
| billingAddress | Customer's billing address | No | 1 |
| paymentInfo | Customer's credit/debit card information | No | 1 |
| orderHistory | Customer's previous and current orders list | No | N/A |

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|---------------------|---|-----------|-----------------|
| updateAccountInfo() | Update customer information | No | No |
| viewCustomerInfo | Displays customer's account information | No | No |
| printCustomerList | Outputs a sorted name list of all customers | Yes | No |

Processing Outlines ->

updateAccountInfo()

Customer entered new information or edits the existing information

IF valid and customer clicks "confirm update"

 Remove the old information and save the updated seller information

Else

 Display the "Account can't be updated" message

viewCustomerInfo(in customerID: String)

Display account information of the entered customerID

printCustomerList()

Sort customers' name by LastName, FirstName

Display the list of customers

PenSoulutions

Table 5: Order Class

| Order |
|--|
| <ul style="list-style-type: none"> - customerID: String - orderID: String = <None> - item: String = <None> - ShoppingCart: Array = <None> - tax: Double = 0 - shippingCost: Double = 0 - itemCost: Double = 0 -/ totalCost: Double = 0 |
| <ul style="list-style-type: none"> + addToCart(in item: String) + viewCart() + calculateTotalCost() + placeOrder() |

Description: Represents an order placed by an Artisan's Co-op North customer

Visibility: Public

Is Abstract: No

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|--------------|---|------------|--------------|
| CustomerID | Unique number assigned for each customer | Yes | 1 |
| OrderID | Unique number assigned for each order | Yes | 1 |
| item | Item that customer wants to buy | No | 0..* |
| ShoppingCart | Where Items that are selected by customer are stored before they place an order | No | N/A |
| tax | Tax for item | No | 1 |
| ShippingCost | Shipping cost for item | No | 1 |
| itemCost | Item price | No | 1 |
| totalCost | Total amount that customer has to pay to place an order | No | 1 |

PenSoulutions

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|--------------------|--|-----------|-----------------|
| addToCart | Adds an item to cart | No | No |
| viewCart | Displays items that customer saved in their cart | No | No |
| CalculateTotalCost | Calculates the total amount that customer has to pay to place an order | No | No |
| placeOrder | Customer submits an order | No | No |

Processing Outlines ->

addToCart(in item: String)

 Add the selected item to the cart

viewCart()

 Display all the items in the cart associated with the account (customersID)

 IF no items are in the cart

 Display “no items in the cart”

calculateTotalCost()

 totalCost = 0

 for each item in the shopping cart

 CostforItem = tax+ itemCost + shipping cost

 totalCost += costForItem

 costForItem = 0

 Once all the item’s cost is calculated, display the totalCost.

placeOrder()

 Check customer credit/debit card information

 IF exists and valid

 complete transaction by asking customer to confirm the order

 Send a confirmation email to customer (using customerID)

 Else

 Display “please reenter the card information” message and direct customer to the shopping cart page

PenSoulutions

Table 6: Item Class

| Item |
|---|
| <ul style="list-style-type: none"> - itemName: String = <None> - itemBarcode: Double = <None> - itemDescription: String = <None> - itemPrice: Double = <None> - itemSold: Boolean = false - itemInStock: Boolean = true |
| <ul style="list-style-type: none"> + editItemDescription() + updateAvailability() |

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|-----------------|-----------------------------------|------------|--------------|
| itemName | Name of Item | No | 1 |
| itemBarcode | Barcode for Item | Yes | 1 |
| itemDescription | Description of Item | No | N/A |
| itemPrice | Price of Item | No | 1 |
| itemSold | Indicates if the item is sold | Yes | N/A |
| itemInStock | Indicates if the item is in stock | Yes | N/A |

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|---------------------|--|-----------|-----------------|
| editItemDescription | Edits the item description | No | No |
| updateAvailability | Changes the status of item that tells if item is in stock or not | No | No |

Processing Outlines ->

editItemDescription() Confirm the change in the item description

If confirmed

 Removed old description and update the description

Else

 Do not make any change

updateAvailability()

If item is restocked

 Change itemInStock = true

Else

 itemInStock = false

PenSoulutions

Table 7: Inventory Class

| Inventory | |
|--|--|
| - numberOfltems: Double - listOfSoldItems: Array = <None> - listOfltemsInStock: Array = <None> - itemsInWarehouse: Array = <None> | Description: Represents inventory of items of Artisan's Co-op North Visibility: Public Is Abstract: No |
| + viewListOfltems() + updateInventory() | |

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|--------------------|---|------------|--------------|
| numberOfltems | Number of items in Inventory | No | N/A |
| listOfSoldItems | List of sold items | No | N/A |
| listOfltemsInStock | List of items in stock | No | N/A |
| ItemsInWarehouse | List of items that are sent by sellers to the warehouse | No | N/A |

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|-----------------|--|-----------|-----------------|
| viewListOfltems | View all the items in inventory | Yes | No |
| updateInventory | Makes change to status of items in Inventory | No | No |

Processing Outlines ->

viewListOfltems()

Display all the items in inventory as a sorted list

If itemName

 Display items in alphabetical order

Else if itemPrice

 Display items that cost from lowest to highest

Else if itemAvailability

 Display only items that are in stock

updateInventory()

Add/Remove items to the inventory

Confirm and save changes

If not confirmed

 Do not make any changes

PenSoulutions

Table 8: Report Class

| | |
|--|---|
| Report | Description: Represents management report generated by Artisan's Co-op North Visibility: Public Is Abstract: No |
| <ul style="list-style-type: none"> - sellerID: String - typeOfReport: String = <None> - reportPeriod: Date = <None> <ul style="list-style-type: none"> + generateReport(in type: String) + printReport() | |

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|---------------------|--|------------|--------------|
| sellerID | Unique number assigned for each seller | Yes | 1 |
| typeOfReport | Type of Report that is generated (payment report and inventory report) | No | N/A |
| reportPeriod | Start and end date of the report period | No | N/A |

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|-----------------------|-------------------------------------|-----------|-----------------|
| generateReport | Generates a specific kind of report | No | No |
| printReport | Outputs generated report | Yes | No |

Processing Outlines ->

generateReport(in type: String)

If payment report

Generate a report that includes pay period, total amount earned by Artisan's Co-op North, the amount that the seller is getting paid.

Else if inventory report

Generate a report that includes all the items associated with the account. It also contains information about how many, what, and when items are sold

printReport()

output the generated report

PenSoulutions

Table 9: Payment Class

| Payment | |
|--|---|
| - sellerID: String - sellerBankAccount: String - payPeriod: Date = <None> - totalAmount: Double = 0 | Description: Represents payment that are made by Artisan's Co-op North Visibility: Public Is Abstract: No |
| + makePayment(in sellerID: string) | |

Additional Information

Attributes:

| Name | Description | Read Only? | Multiplicity |
|--------------------------|---|------------|--------------|
| sellerID | Unique number assigned for each seller | Yes | 1 |
| sellerBankAccount | Seller's bank account number that receives payment from Artisan's Co-op North | No | 1 |
| payPeriod | Start and end date of the pay period | No | N/A |
| totalAmount | Amount that seller gets paid given the period | No | N/A |

Operations:

| Name | Description | Is Query? | Is Polymorphic? |
|--------------------|--------------------------------------|-----------|-----------------|
| makePayment | Sends money to seller's bank account | No | No |

Processing Outlines ->

makePayment(in SellerID: String)

Search for seller's bank account in the secured database given sellerID
 Send a request to the Arisan's Co-op North bank to transfer the totalAmount to the seller's bank account
 If transaction goes through
 Send a confirmation email to the seller
 Else if
 Report the problem to the bank, administrator, and seller.



PenSoulutions

3.0 Architecture Design

3.1 Introduction

This section includes two infrastructure models (architecture overview and nodes and artifacts diagram), lists of hardware and software requirements, and a security plan for Arts and Sales Management System (ASMS).

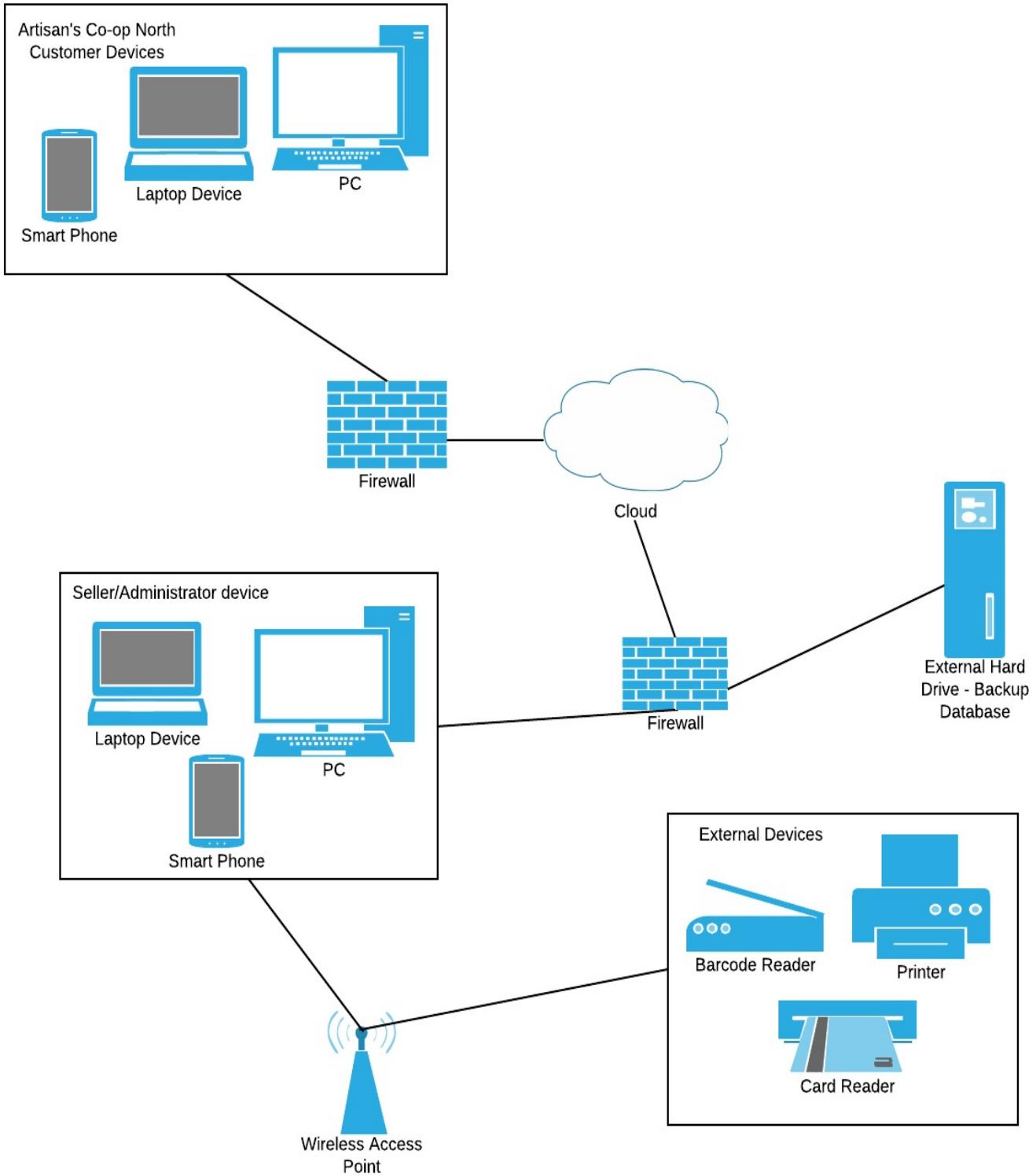
The system will be a thin-client architecture with the use of cloud computing. The main server will be “in the cloud” and the client will be on the desktop (or other devices). The cloud server will manage business application, data access, and presentation logic, while the client only performs the presentation logic.

In terms of the required hardware for ASMS, ACoN will need to purchase an external hard drive, barcode scanner, card reader, and possibly a printer.

PenSoulutions

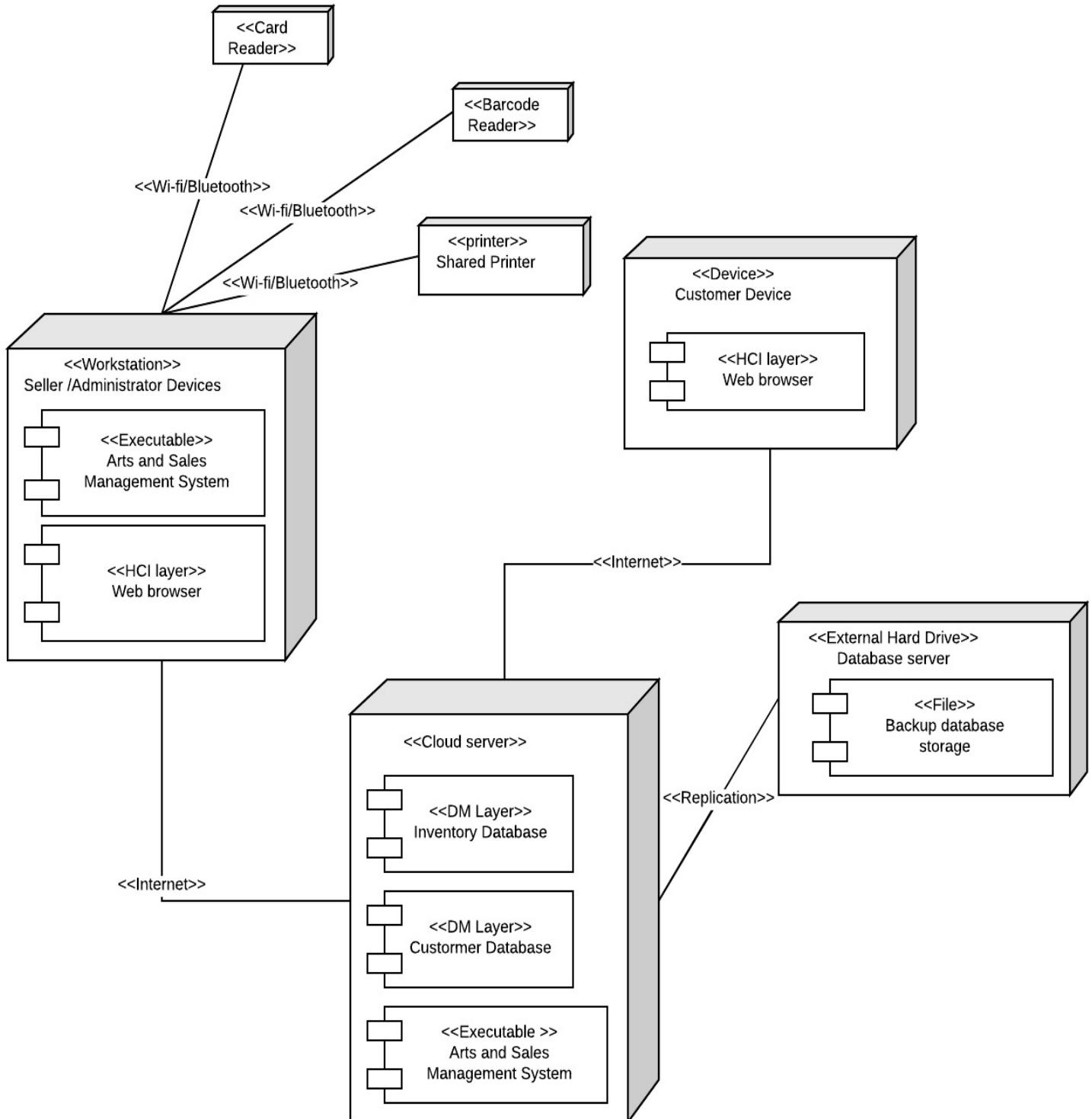
3.2 Infrastructure Model

3.2.1 Deployment Diagram 1– Architecture Overview



PenSoulutions

3.2.2 Deployment Diagram 2 – Nodes and Artifacts





PenSoulutions

3.3 Hardware and Software Recommendations

3.3.1 Required Hardware Components

- Cloud Server – Purchase from a third party that provides cloud service (yearly subscription)
- External Hard Drive – It will be used to back up and store all the data
- Printer – If sellers and administrators have a printer at their houses, there is no need to purchase a new one. This is for printing out barcodes and attaching it to items.
- Card reader – This is for receiving card payments at fairs or events. It has to portable and we recommend using Square products.
- Barcode Scanner – The purchase of a barcode scanner will be essential for automating the inventory management.
- Seller/ Administrator Devices – They don't have to buy any new device as long as they have one device (PC, laptop, phone, tablet). However, we recommend having a computer at the warehouse that can be attached to both barcode scanner and card reader.

3.3.2 Required Software Components

- Operating System (OS) – We recommend using Microsoft Windows OS since people are more familiar with using Window computers. Most of the computers that customers, sellers, and administrators own will have an operating system installed.
- Firewall – This will be crucial for preventing unauthorized access or damage to data.
- Encryption Software – It is recommended for the protection of user's information.
- Web browser – Since ACoN already owns a website, there is no need to buy another domain name.

PenSoulutions

3.4 Security Plan

This section describes potential security threats to the components of ASMS and controls for each threat. The major secure requirements for ASMS are protecting user information and preventing the system from any unauthorized access. We will be implementing a software for protection of passwords and user information. Also, in order to prevent a failure of the cloud server, we will have an external hard drive to back-up all the important management data.

The table below includes threats to ASMS and controls that can be applied to each threat.

| Threats components | Disruption, Destruction, Disaster | | | | | Unauthorized Access | |
|--|-----------------------------------|-------|------------|-----------------|-------|---------------------|-------------------|
| | Fire | Flood | Power loss | Circuit Failure | Virus | External Intruder | Internal Intruder |
| Servers (Cloud)/Hard drive | 1,2,3 | 1,3 | 1, 7 | 1,7 | 4,5,6 | 4,6,9, 10,11,12 | 4,5,8,9 10,11 |
| Devices used for ASMS (Customers, Seller, Administrator) | 1,2,3 | 1,3 | 1,7 | 1,7 | 4,5,6 | 4,8,9,10, 11,12 | 4,5,8,9, 10,11 |
| Warehouse items | 1,2,3 | 1,3 | N/A | N/A | N/A | N/A | N/A |
| Network Software | N/A | N/A | 1 | 1 | 4,5,6 | 4,10,11,12 | 4,10,11,12 |
| People | 1,2,3 | 1,3 | 1 | N/A | N/A | N/A | N/A |

Controls

1. Disaster recovery plan
2. Sprinkler system
3. Security alarms to prevent further damage
4. Virus and malware checking software on the network
5. User background checks
6. Extensive user training on phishing and other exploits
7. Save data that was in progress
8. Strong password software
9. Extensive user training on password security
10. Alert the user with a message
11. Monthly newsletter to remind customers, sellers, and administrators about best practices
12. Firewall between cloud server and devices that handle the presentation logic



PenSoulutions

4.0 User Interface

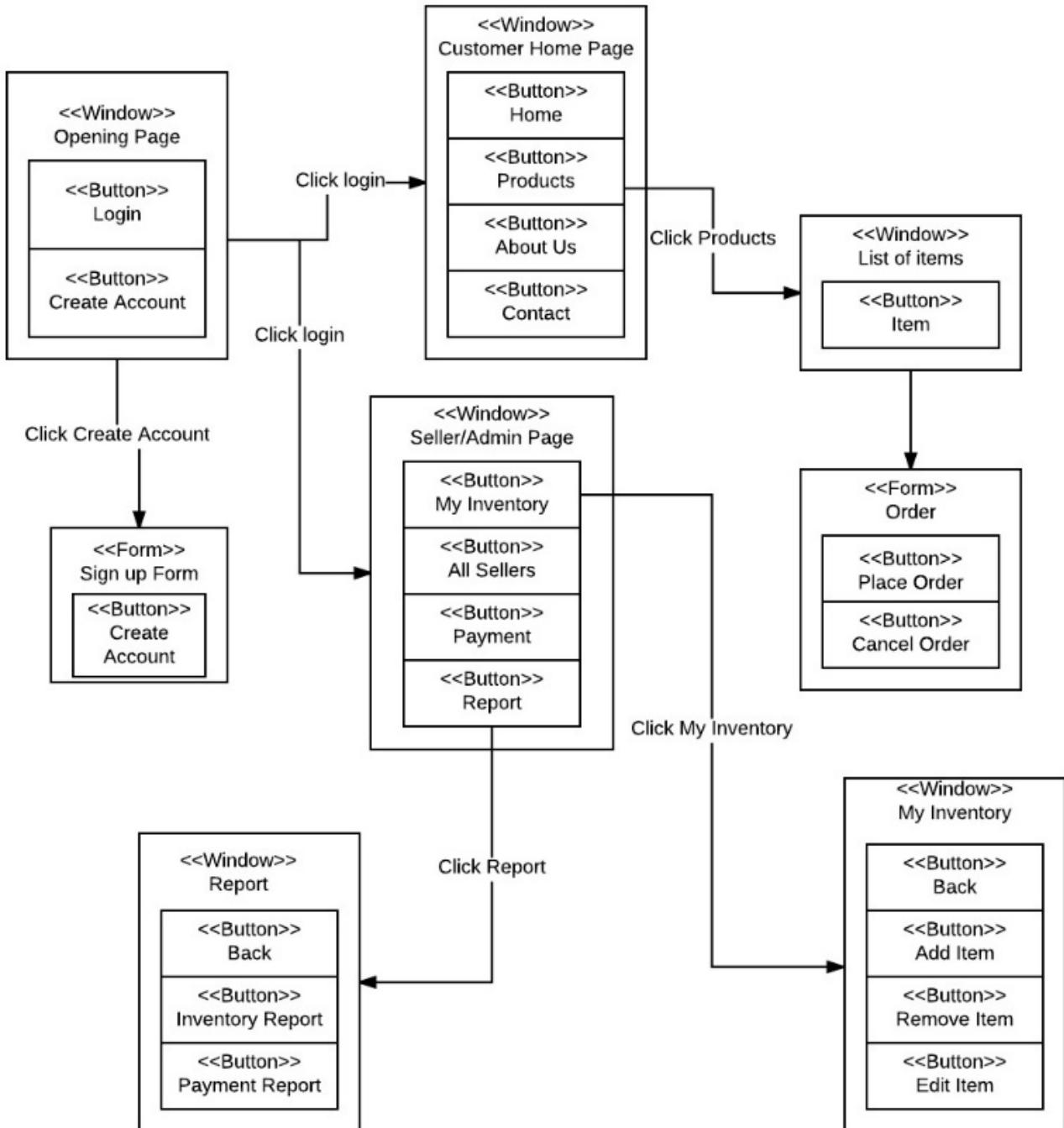
4.1 User-Interface Requirements and Constraints

This section of the document includes two sections. Section one is a window navigation diagram which shows how all the screens, forms, and reports used by the system are related and how the user moves from one to another. Section two has different mock ups for user-interface design that is very similar to what users of ASMS will be seeing on their screen. It includes screens for customers, sellers, and administrators.

Our main focus is making the design as easy to understand and follow at the same time providing a user-friendly interface. Users must be able to navigate and do what they want to do as quick and efficient as possible. Although the login page is the same for all users and the design of the system is consistent throughout, Customers and Sellers/Administrators will have different features and functions available for them. User can have more than one role, however, they will only be able to have one role at a time in the system.

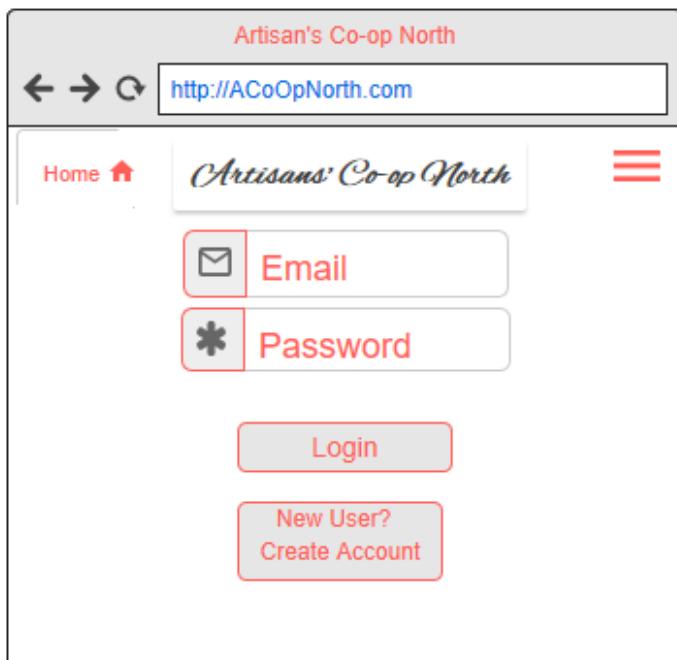
PenSoulutions

4.2 Window Navigation Diagram



PenSoulutions

4.3 Forms: Screen / User-Interaction Design



The login screen for Artisan's Co-op North features a header with the website name and URL. Below the header is a logo and navigation links for Home, Email, and Password. A central 'Login' button is followed by a link for new users to create an account.

Artisan's Co-op North

Home  <http://ACoOpNorth.com>

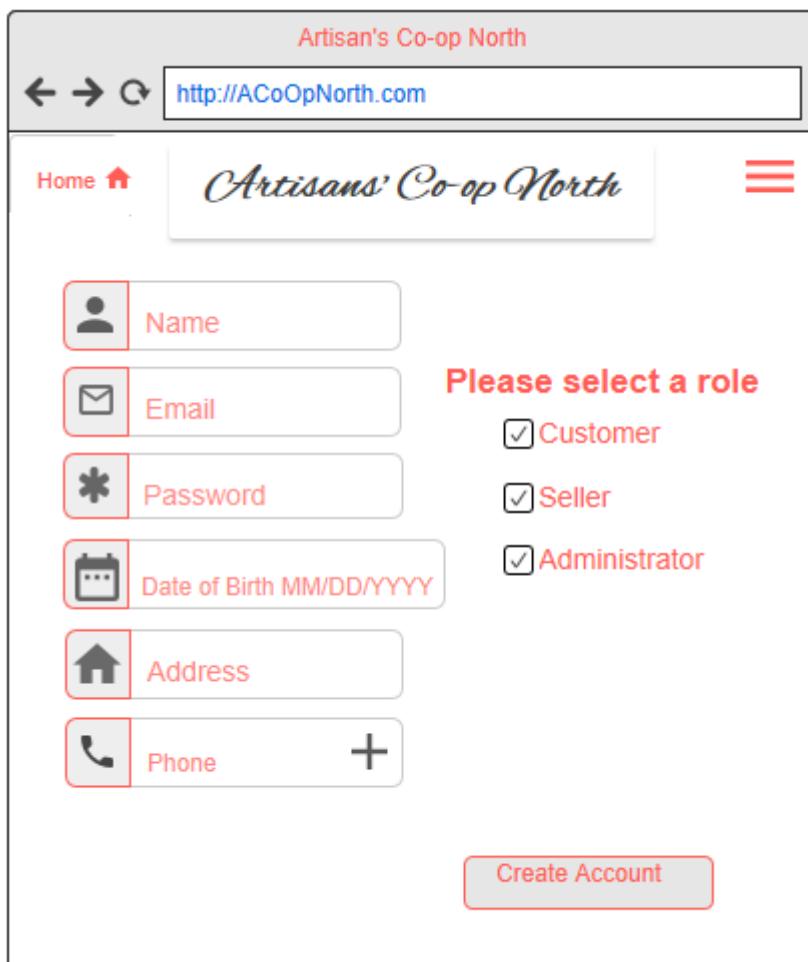
 Email

 Password

Login

New User?
Create Account

Login screen for users of the Artisan's Co-op North's management system and website.



The create account screen for Artisan's Co-op North includes fields for Name, Email, Password, Date of Birth, Address, and Phone. To the right, users are prompted to select their role (Customer, Seller, or Administrator). A 'Create Account' button is located at the bottom.

Artisan's Co-op North

Home  <http://ACoOpNorth.com>

 Name

 Email

 Password

 Date of Birth MM/DD/YYYY

 Address

 Phone 

Please select a role

Customer

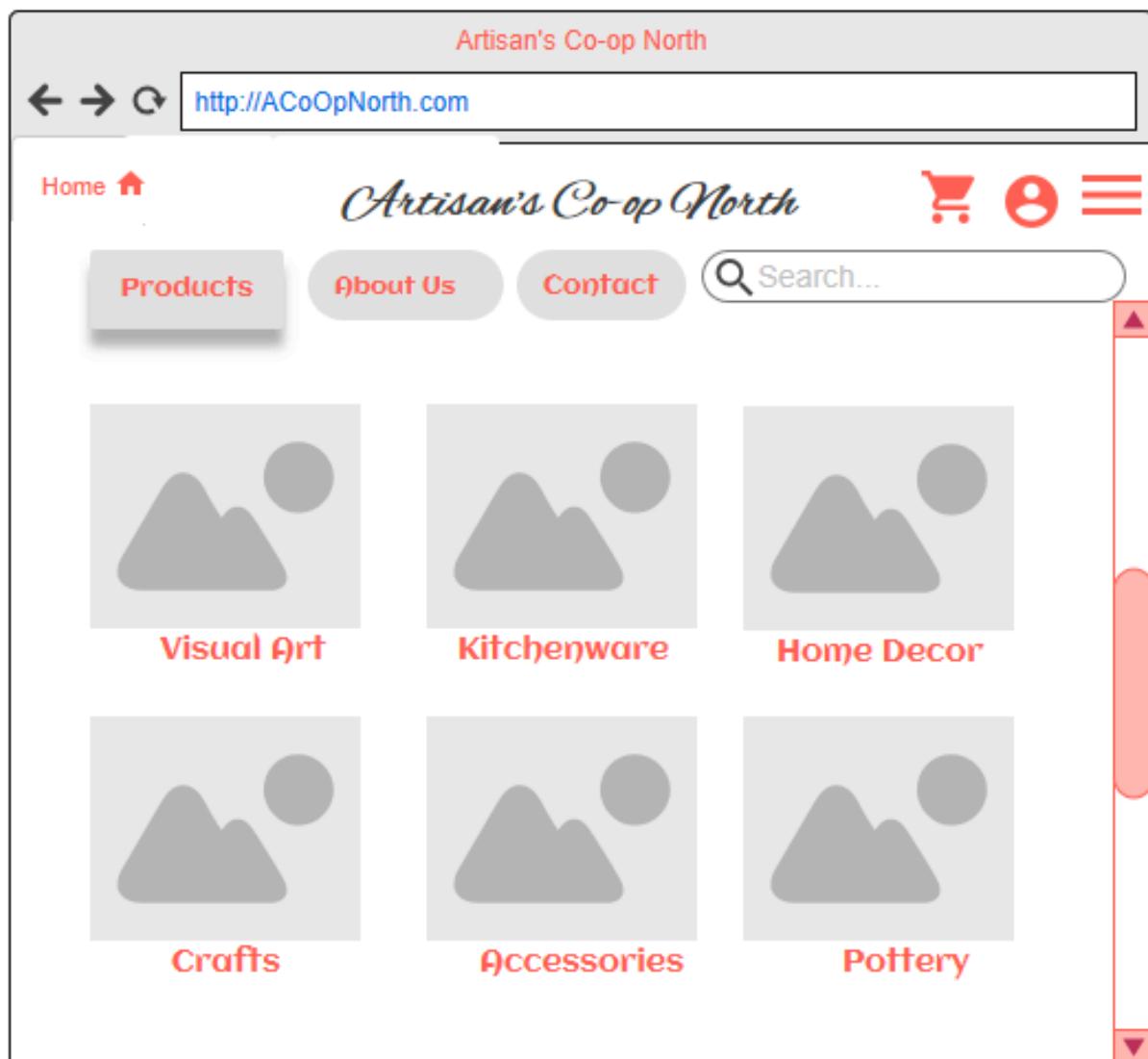
Seller

Administrator

Create Account

Screen for creating a new account. Users need to enter their information and select their role for Artisan's Co-op North.

PenSoulutions



- This is a screen when a customer clicks “Products”. Home screen will also be similar to this screen presenting main featured items with professional images.
- When a customer clicks “About Us” or “Contact”, the menu bar does not change. Only selected menu gets highlighted as shown above for “Products”. The only thing that will be different is the middle section where the pictures are.
- “About Us” – Includes a picture of main administrators and the company’s mission statement.
- “Contact” – Information about what fairs or events, when and where Artisan’s Co-op North is attending, and their contact information.

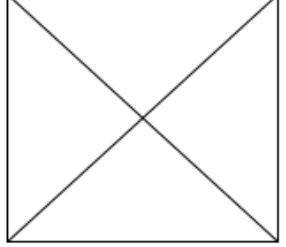
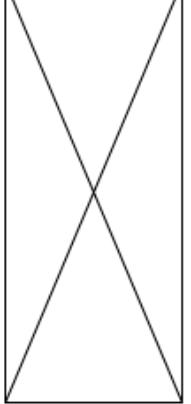
PenSoulutions

Artisan's Co-op North

http://ACoOpNorth.com

Home  *Artisan's Co-op North*   

 **Visual Art**  Search...

Information about the item
Including price and type selection

Item name:
Item Description
Details

Place Order

Add to Cart

Relevant Results

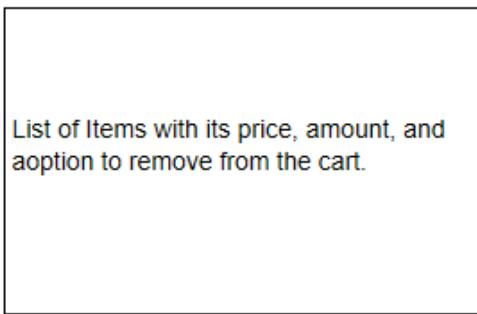
Screen for a customer viewing a specific item.

Artisan's Co-op North

http://ACoOpNorth.com

Home  *Artisan's Co-op North*   

 **Order**  Search...



Total item price
Tax
Shipping cost

Total cost

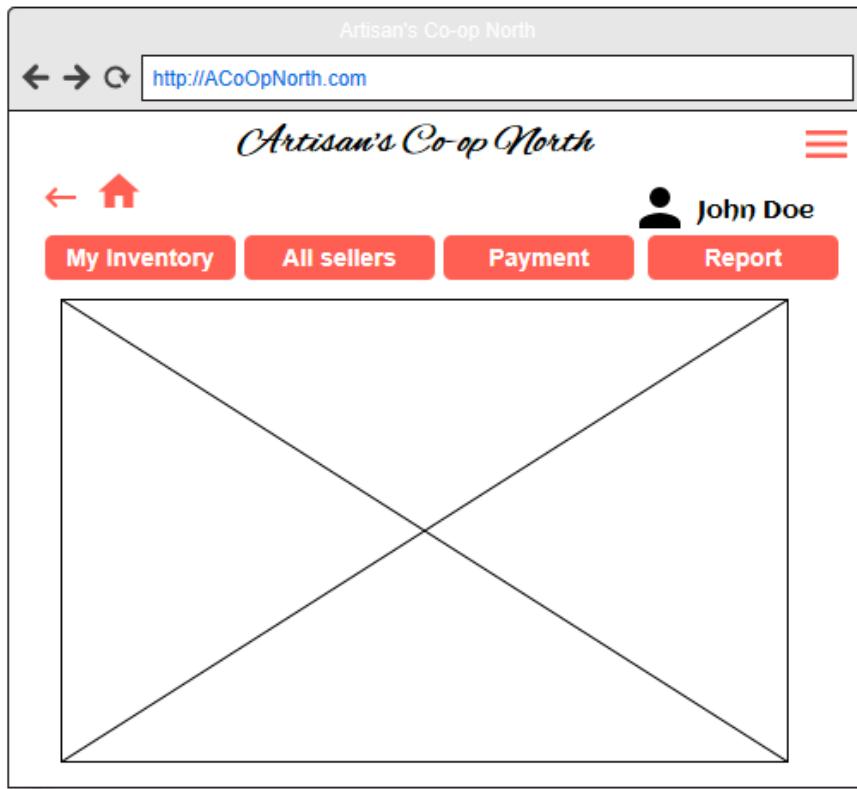
Credit card information 

Cancel Order

Confirm Payment

Screen for a customer's cart (order).

PenSoulutions



Main Screen for Seller/Administrator Account. Depending on the menu option, the middle box area will have different content.

The screenshot shows the "My Inventory" page for the seller "John Doe". The top navigation bar is identical to the main screen. The central area contains a table with four rows, each representing an item. The columns are labeled "Item Name", "Barcode", "Description", "Image", and "In Stock". The items listed are Item1 (code1), Item2 (code2), Item3 (code3), and Item4 (code4). To the right of the table are three buttons: "Add Item" (with a plus sign icon), "Remove Item" (with a minus sign icon), and "Edit Item Description" (with a pencil icon).

| ▼ Item Name | ▼ Barcode | ▼ Description | ▼ Image | ▼ In Stock |
|-------------|-----------|----------------------|---------|----------------------------------|
| Item1 | code1 | <input type="text"/> | * | <input checked="" type="radio"/> |
| Item2 | code2 | <input type="text"/> | * | <input checked="" type="radio"/> |
| Item3 | code3 | <input type="text"/> | * | <input checked="" type="radio"/> |
| Item4 | code4 | <input type="text"/> | * | <input checked="" type="radio"/> |

When Seller clicks "My inventory".

PenSoulutions

4.4 Reports: “Printed Output” Design

Payment Report

Artisan's Co-op North Report

Seller ID: Seller Name

Inventory Report

Pay Period: Start Date - End Date

Payment Report

The payment is deposited to Account number -----

Report date

PenSoulutions

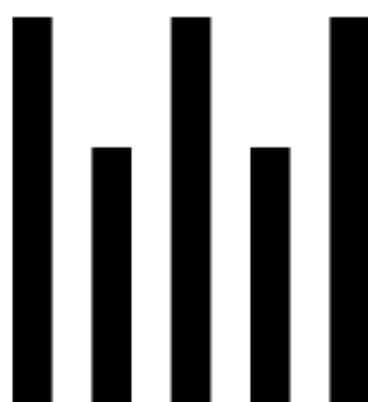
Inventory Report

Artisan's Co-op North Report

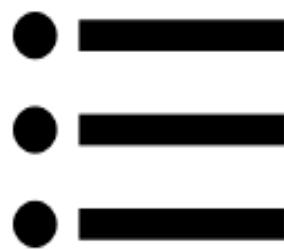
| | |
|--------------------------------------|--|
| Seller ID: Seller Name | <input checked="" type="checkbox"/> Inventory Report |
| Report Period: Start Date - End Date | <input type="checkbox"/> Payment Report |

| Date | Sold items and quantity () | New items and quantity () | Items in stock and quantity () | Items out of stock and quantity () |
|-------|-------------------------------|------------------------------|-----------------------------------|---------------------------------------|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| Total | # | # | # | # |

*It can be more than one page



A bar graph for sales for each week



List of popular and unpopular items

Report date _____



PenSoulutions

5.0 Appendices

Bibliography/Reference

Dennis, Alan, Barbara Haley Wixom, and David Paul Tegarden. Systems Analysis Design: An Object Oriented Approach with UML, Fifth Edition. Hoboken, NJ: John Wiley & Sons, 2012. Print.

Pfeiffer, William S. Pocket Guide to Technical Communication. Upper Saddle River, NJ: Prentice Hall, 2011. Print.

Weltz, Elaine (2016). Systems Design, various lectures [PowerPoint slides/Word Documents]. Retrieved from Professor Weltz and <http://canvas.spu.edu>.