**Artisan Thrive Technology (ATT) System Specification Design**

Celena Flores

CAMR Incorporated

December 5, 2017

**Table of Contents**

[Executive Summary 2](#_Toc500313253)

[1. Introduction 3](#_Toc500313254)

[1.1 Problem Statement and Project Vision 3](#_Toc500313255)

[1.2 System Services 3](#_Toc500313256)

[1.3 Nonfunctional Requirements and Design Constraints 3](#_Toc500313257)

[1.4 System Evolution 4](#_Toc500313258)

[1.5 Document Outline 4](#_Toc500313259)

[2. Structural Model 5](#_Toc500313260)

[2.1 Introduction 5](#_Toc500313261)

[2.2 Class Diagram 5](#_Toc500313262)

[2.3 Metadata 6](#_Toc500313263)

[3. Architecture Design 17](#_Toc500313264)

[3.1 Introduction 17](#_Toc500313265)

[3.2 Infrastructure Model 17](#_Toc500313266)

[Deployment Diagram 1 - Architecture Overview 17](#_Toc500313267)

[Deployment Diagram 2 – Nodes and Artifacts 18](#_Toc500313268)

[3.3 Hardware and Software Requirements 18](#_Toc500313269)

[3.4 Security Plan 19](#_Toc500313270)

[4. User-Interface 20](#_Toc500313271)

[4.1 User-Interface Requirements and Constraints 20](#_Toc500313272)

[4.2 Window Navigation Diagram 20](#_Toc500313273)

[4.3 Forms: Screen and User- Interaction Design 21](#_Toc500313274)

[4.4 Reports: “Printed Outputs” Design 24](#_Toc500313275)

[Appendices 25](#_Toc500313276)

[Bibliography 25](#_Toc500313277)

[Supporting Documentation and Recommendations 25](#_Toc500313278)

# **Executive Summary**

Ms. Weltz from Artisan’s Co-op North has hired CAMR Incorporated to plan, implement, and maintain a new system for her company. This system will bring Artisan’s Co-op North to the twenty first century with its renovated way to take and track inventory, improve sales, and provide a more active role for the artisans.

CAMR Incorporated examined the System Proposal Request by Ms. Weltz and decided the proposed system was practicable and low risk. The System Proposal highlighted the unified vision of CAMR Incorporated and Artisan’s Co-op North to build a database system in version 1 and in version 2 of the system, to build a contemporary website to modernize, expand, and simplify Artisan’s Co-op North.

This document will now move away from the analysis and towards the development of Artisan Thrive Technology (ATT). This document will begin by reintroducing the system, then show the structural model, architecture design, and the user-interface experience.

The System Proposal along with this document will be resourceful documents to refer back to during the implementation phase to ensure CAMR Incorporated produces a complete, functional, and expected system for Artisan’s Co-op North.

# **1. Introduction**

## 1.1 Problem Statement and Project Vision

Artisan’s Co-op North is a small company that notices the world is moving to a modern era. Currently, Artisan’s Co-op North keeps track of their inventory through a simple Excel document and everything is done by hand. This current system is outdated and tedious for Ms. Weltz and other members.

The vision for ATT (Artisan Thrive Technology) is to automate inventory, support craft fair sales, and reach a greater audience via the web. To achieve this vision, CAMR Incorporated plans to create a database system and website to meet the needs of Artisan’s Co-op office staff, artisans, and customers. CAMR Inc. has made the decision to develop the ATT system in phases instead of delivering a final system in one piece. First, CAMR Inc. will focus on automating Artisan’s Co-op North by delivering a database system that will track inventory, sales, and a payroll system for the artisans. In the next version, CAMR Inc. will develop a website to improve sales for art fairs and the online market. For additional information, please refer to Sections 1.1, 1.2, and 1.4 of the System Proposal).

## 1.2 System Services

The following list briefly describes the functional requirements for Artisan Thrive Technology (ATT). For additional information please refer to System Proposal. Specifically, Section 4 for a full explanation of functional requirements and Section 5 for use-case descriptions.

1. Artisan Members Requirements

* Manage account login credentials (use case 0)
* Insert, remove, and view product information from the database (use case 2 and 8)

1. Management and other office staff Requirements

* Manage system: includes creating and deleting accounts (use case 0)
* Inserting, removing, and viewing products and their information from the database. (use case 2, 4, 8)
* Generating and printing reports including payroll system. (use case 6 and 7)
* Storing receipts manually or using Square Up into the database (use case 5)

1. Customers’ Requirements

* Make account to view and purchase products (use case 0 and 3)

## 1.3 Nonfunctional Requirements and Design Constraints

The following list briefly describes the nonfunctional requirements for Artisan Thrive Technology (ATT). For additional information please refer to System Proposal. Specifically, Section 1 for constraints, Section 5 for the feasibility assessment, and Section 4 for the nonfunctional requirements.

1. Development Requirements: ATT must be completed on schedule and on budget.
2. Operational Requirements: ATT must be able to run on multiple operating systems.
3. Performance Requirements: ATT must have a familiar user interface experience and will be fast and efficient.
4. Efficiency Requirements: ATT must have consistency control and remember information.
5. Security and Control Requirements: ATT must provide Artisan's Co-op North with a secure system to protect all personal and confidential information.
6. Cultural, Political, and Legal Requirements: ATT must be available 24/7 except for schedule maintenance.
7. Service Requirements: ATT must be accessible through an external link download or a URL.

## 1.4 System Evolution

As discussed in Section 6 of the System Proposal, Artisan Thrive Technology (ATT) will be deliver in phases. In Version 1, CAMR Inc. will deliver the basic components of ATT. In Version 2, we will adopt new technology to support credit card transaction at craft fairs. In Version 3, we will create an interactive website for customers to have their own accounts to view and purchase objects. Through the implementation of the different versions, CAMR Inc. will continue to work closely with both the Artisan's Co-op North members and their customers to create a product that will be both efficient, easy to use, and physically appealing to the eye.

## 1.5 Document Outline

This document currently includes the following sections:

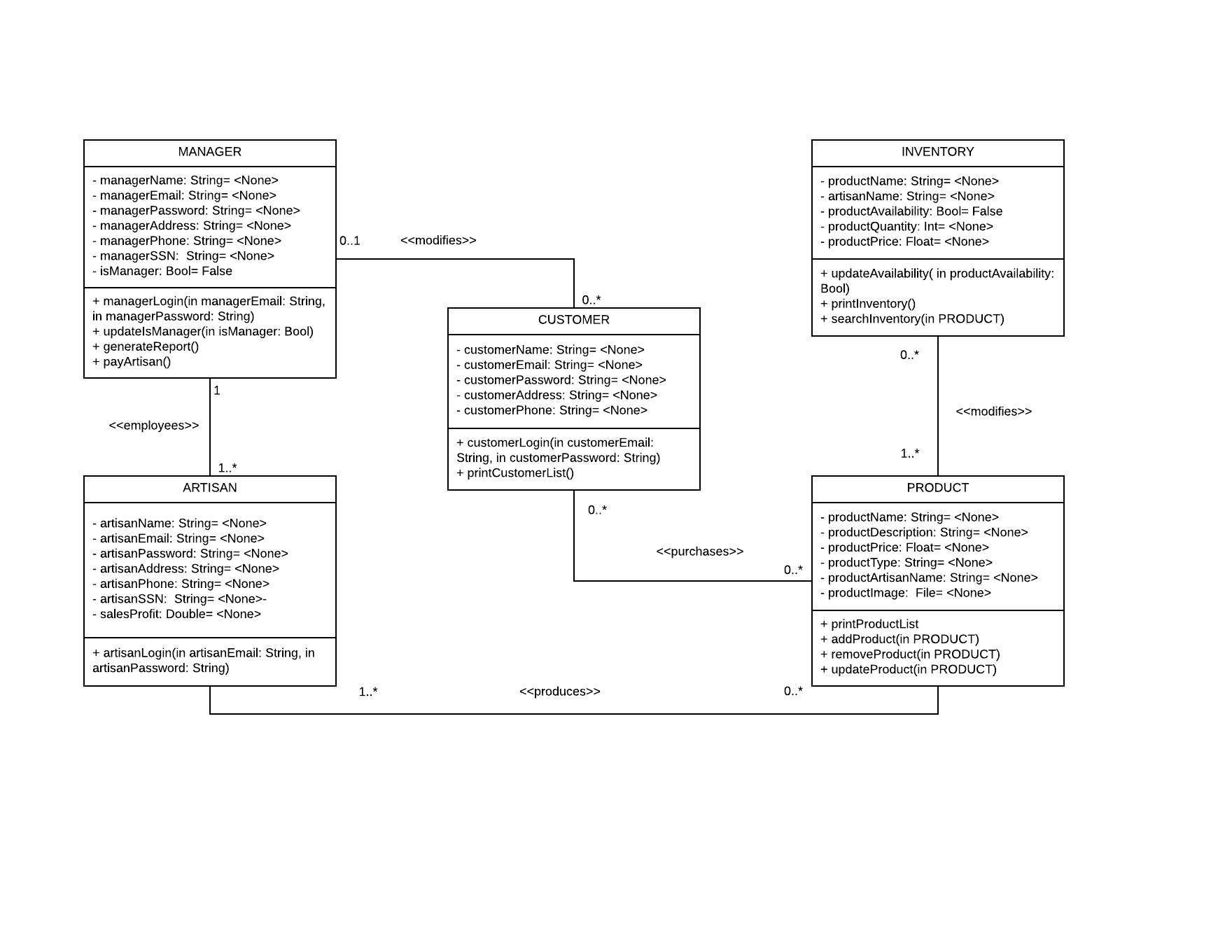
* Structural Model: This section includes class diagram and descriptions of the attributes and operations associated with each class entity.
* Architecture Design: The section includes the two deployment diagrams to represent the infrastructure model. Lists any necessary hardware and software requirements and describes a security plan to protect the system.
* User-Interface: This section includes the user- interface requirements, constraints, window navigation, forms, and the printed output design.
* Appendices: A bibliography and relevant supporting documentation and recommendations.

# **2. Structural Model**

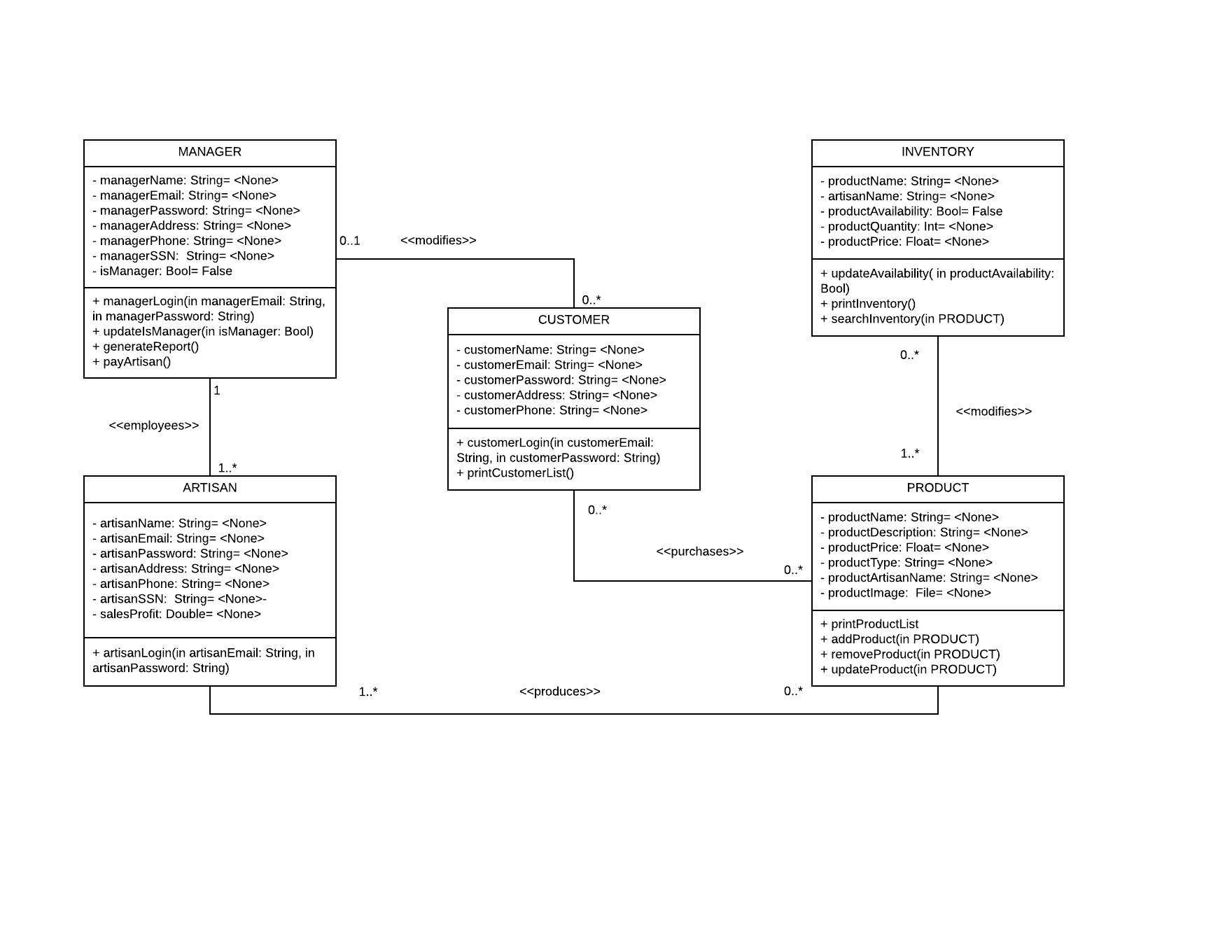
## 2.1 Introduction

This section gives a class diagram for Artisan Thrive Technology (ATT). Section 2.2 includes a UML Class Model diagram. Section 2.3 will provide a more detailed description of a class’s attributes and operations.

## 2.2 Class Diagram



## 2.3 Metadata



Description: Represents an artisan member

Visibility: Private

IsAbstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| artistName | Last, First Name | String | No | No | Private | 1 | None |
| artisanEmail | Used to log into ATT | String | No | No | Private | 1 | None |
| artisanPassword | Security measure to log into ATT | String | No | No | Private | 1 | None |
| artisanAddress | Street 1, Street 2, City, State, Zip Code | String | No | No | Private | 1 | None |
| artisanPhone | Phone Address | String | No | No | Private | 1..\* | None |
| artisanSSN | Member’s SSN to keep track of payroll | String | No | No | Private | 1 | None |
| salesProfit | Amount made by products | String | No | No | Private | 0..\* | None |

**Operations:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Return Type | Parameters | Visibility | Scope | In Query |
| artisanLogin | Artisans can log into their account to insert, remove, and view products in the database. | None | artisanEmail  Direction: In  String  Default: None  artisanPassword  Direction: In  String  Default: None | Public | Instance | No |

**Processing Outline:**

*artisanLogin:*

Artisan inputs email and password

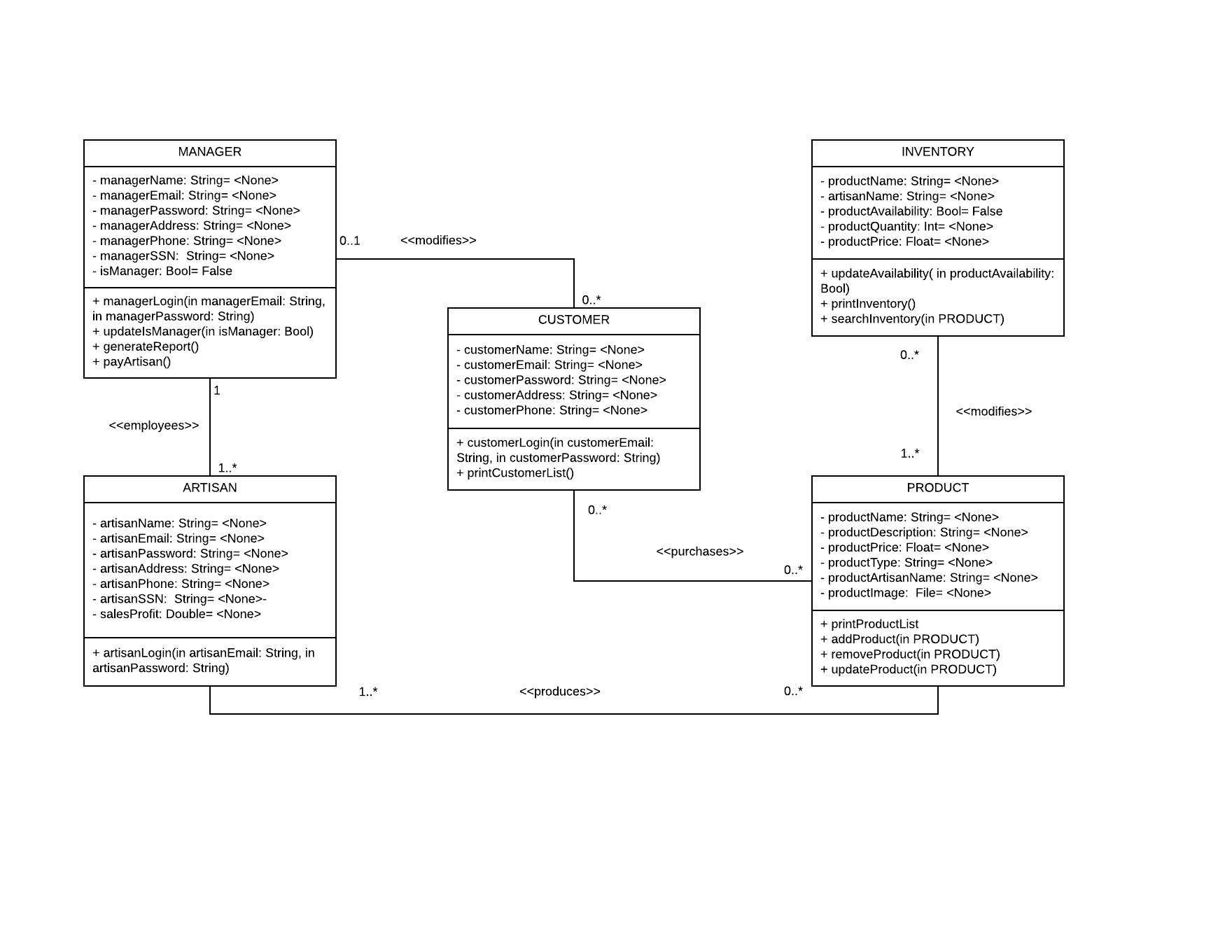
If valid

allow to login to system

display appropriate window

Else

display error message



Description: Represents a customer

Visibility: Private

IsAbstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| customerName | Customer Name | String | No | No | Private | 1 | None |
| customerEmail | Customer Email | String | No | No | Private | 1 | None |
| customerPassword | Customer Password | String | No | No | Private | 1 | None |
| customerAddress | Customer Address | String | No | No | Private | 1 | None |
| customerPhone | Customer Phone Number | String | No | No | Private | 1..\* | None |

**Operations:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Return Type | Parameters | Visibility | Scope | In Query |
| customerLogin | Customers can log into their account to purchase products from the website. | None | customerEmail  Direction: In  String  Default: None  customerPassword  Direction: In  String  Default: None | Public | Instance | No |
| printCustomerList | Print a list of customers and its data information | None | None | Public | Instance | No |

**Processing Outline:**

*customerLogin:*

Customer inputs email and password

If valid

allow to login to system

display appropriate window

Else

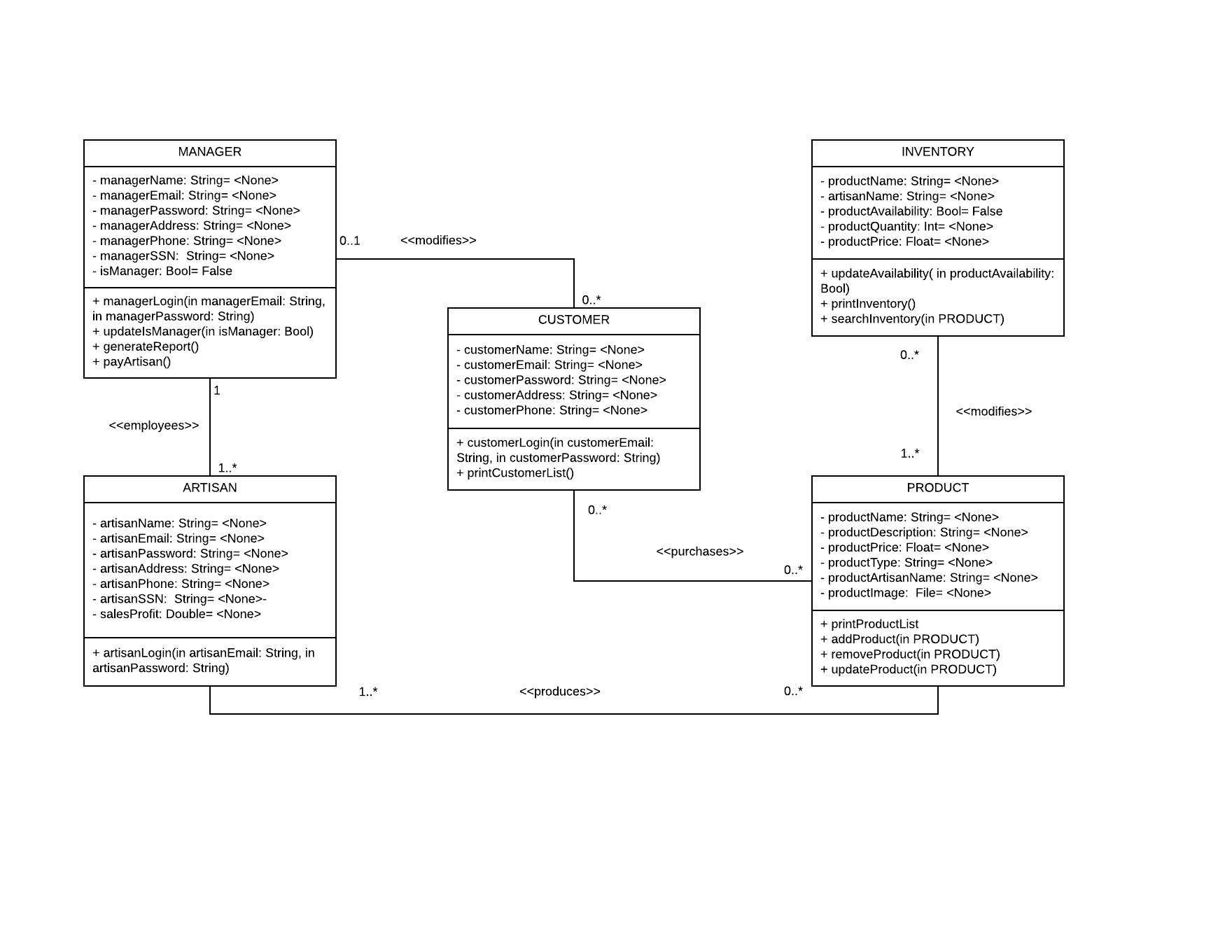
display error message

*printCustomerList:*

While there are more customers to print

for each customer

output customer name, email, password, address, and phone number



Description: Represents a manager of Artisan’s Co-op North

Visibility: Private

IsAbstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| managerName | Manager Name | String | No | No | Private | 1 | None |
| managerEmail | Manager Email | String | No | No | Private | 1 | None |
| managerPassword | Manager Password | String | No | No | Private | 1 | None |
| managerAddress | Manager Address | String | No | No | Private | 1 | None |
| managerPhone | Manager Phone | String | No | No | Private | 1..\* | None |
| managerSSN | Manager SSN to ensure employee | String | No | No | Private | 1 | None |
| isManager | Manager Status | Bool | No | Yes | Private | 0..1 | False |

**Operations:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Return Type | Parameters | Visibility | Scope | In Query |
| managerLogin | Managers can log into their account to manage all users and create reports. | None | managerEmail  Direction: In  String  Default: None  mangerPassword  Direction: In  String  Default: None | Public | Instance | No |
| updateIsManager | Manager is able to update status about whether they are a manager or not. | None | isManager  Direction: In  Bool  Default: None | Public | Instance | No |
| generateReport | Manager is able to generate repots from templates and forms. | None | artisanName  Direction: In  String  Default: None | Public | Instance | No |
| payArtisan | Manager’s are able to calculate how much an artisan should be paid every month. | Float | artisanName  Direction: In  String  Default: None  salesProfit  Direction: In  Float  Default: None  fee  Direction: In  Float  Default: None | Public | Instance | No |

**Processing Outline:**

*managerLogin:*

Manger inputs email and password

If valid

allow to login to system

display appropriate window

Else

display error message

*updateIsManager:*

Update user manually when manager status is active

*generateReport:*

Manager presses generate report button

Manager fills out all required fields

Manager hits submit

*payArtisan:*

Manager presses generate report button

If manager hits the pay roll button

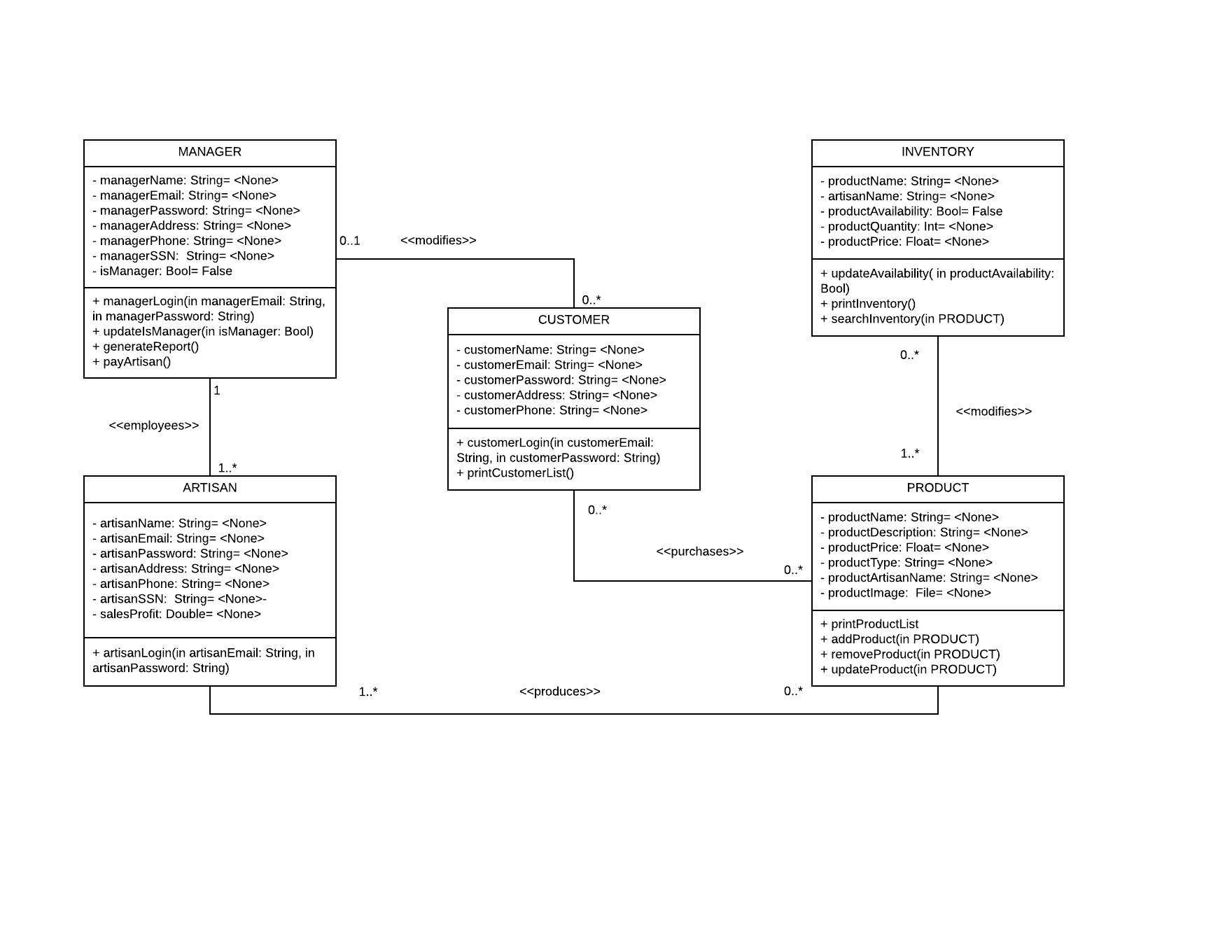
Manager fills out all required fields

Manager hits submit

Payroll has been successful

Else

Generate normal report



Description: Represents a product made by an artisan member

Visibility: Private

IsAbstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| productName | Product Name | String | No | No | Private | 1 | None |
| productDescription | Short Product Description | String | No | No | Private | 1 | None |
| productPrice | Product Retail Price | Float | No | No | Private | 1 | None |
| productType | Product Type | String | No | No | Private | 1..\* | None |
| productArtisanName | Artisan Name who made the product | String | No | No | Private | 1..\* | None |
| productImage | Product Image | File | No | No | Private | 0..\* | None |

**Operations:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Return Type | Parameters | Visibility | Scope | In Query |
| printProductList | Print a list of products and its data information | None | None | Public | Instance | No |
| addProduct | Add Product to the inventory | None | Product  Direction: In  Object of type Product  Default: None | Public | Instance | No |
| removeProduct | Remove Product from the inventory | None | Product  Direction: In  Object of type Product  Default: None | Public | Instance | No |
| updateProduct | Update Product fro the inventory | None | Product  Direction: In  Object of type Product  Default: None | Public | nstance | No |

**Processing Outline:**

*printProductList:*

While there are more products to print

for each product

output product name, description, price, type, artisan name, and images

*addProduct:*

While (Member wants to add a product)

add product

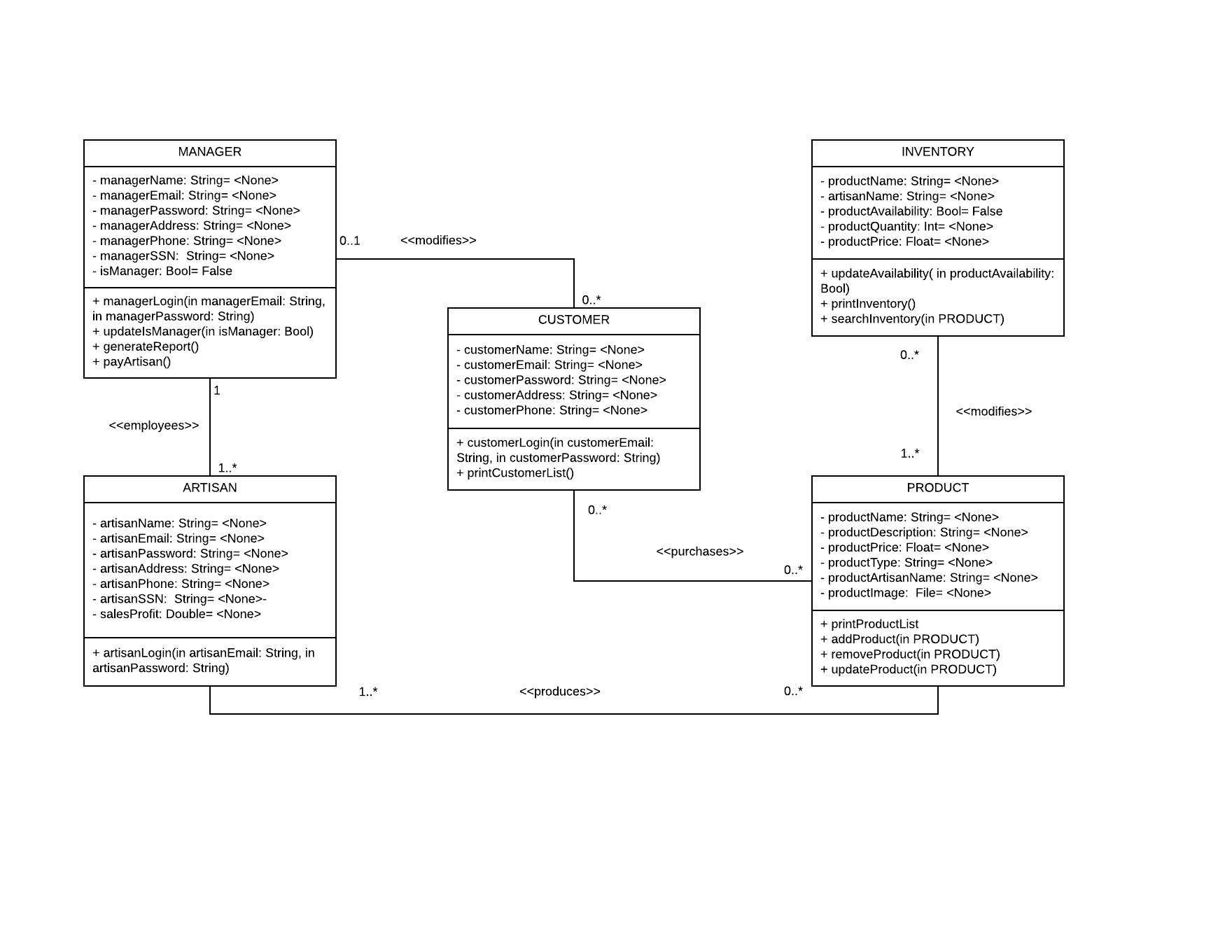
*removeProduct:*

While (Member wishes to remove a product)

delete product

*updateProduct:*

Update availability with parameter passed in



Description: Represents whether a product is available or not

Visibility: Private

IsAbstract: No

**Attributes:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Data Type | Is Derived | Is Read Only | Visibility | Multiplicity | Default Value |
| productName | Product Name | String | No | No | Private | 1 | None |
| artisanName | Artisan’s name who made the product | String | No | No | Private | 1 | None |
| productAvailability | Bool value if product is available or not | Bool | Yes | Yes | Private | 0..1 | False |
| productQuantity | Value of how many are available | Int | Yes | No | Private | 0..\* | None |
| productPrice | Price of product | Float | No | No | Private | 1 | None |

**Operations:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Name | Description | Return Type | Parameters | Visibility | Scope | In Query |
| updateAvailability | Update the availability of a product in the database | None | Product  Direction: In  Object of type Product  Default: None | Public | Instance | No |
| printInventory | Print a list of inventory items and its data information | None | None | Public | Instance | No |
| searchInventory | Search the inventory for a specific product | None | Product  Direction: In  Object of type Product  Default: None | Public | Instance | No |

**Processing Outline:**

*updateAvailability:*

Update availability with parameter passed in

*printInventory:*

While there are more inventory items to print

for each inventory item

output product name, artisan name, product availability, product quantity, and price

*searchInventory:*

While(inventory item not found)

Search for item

# **3. Architecture Design**

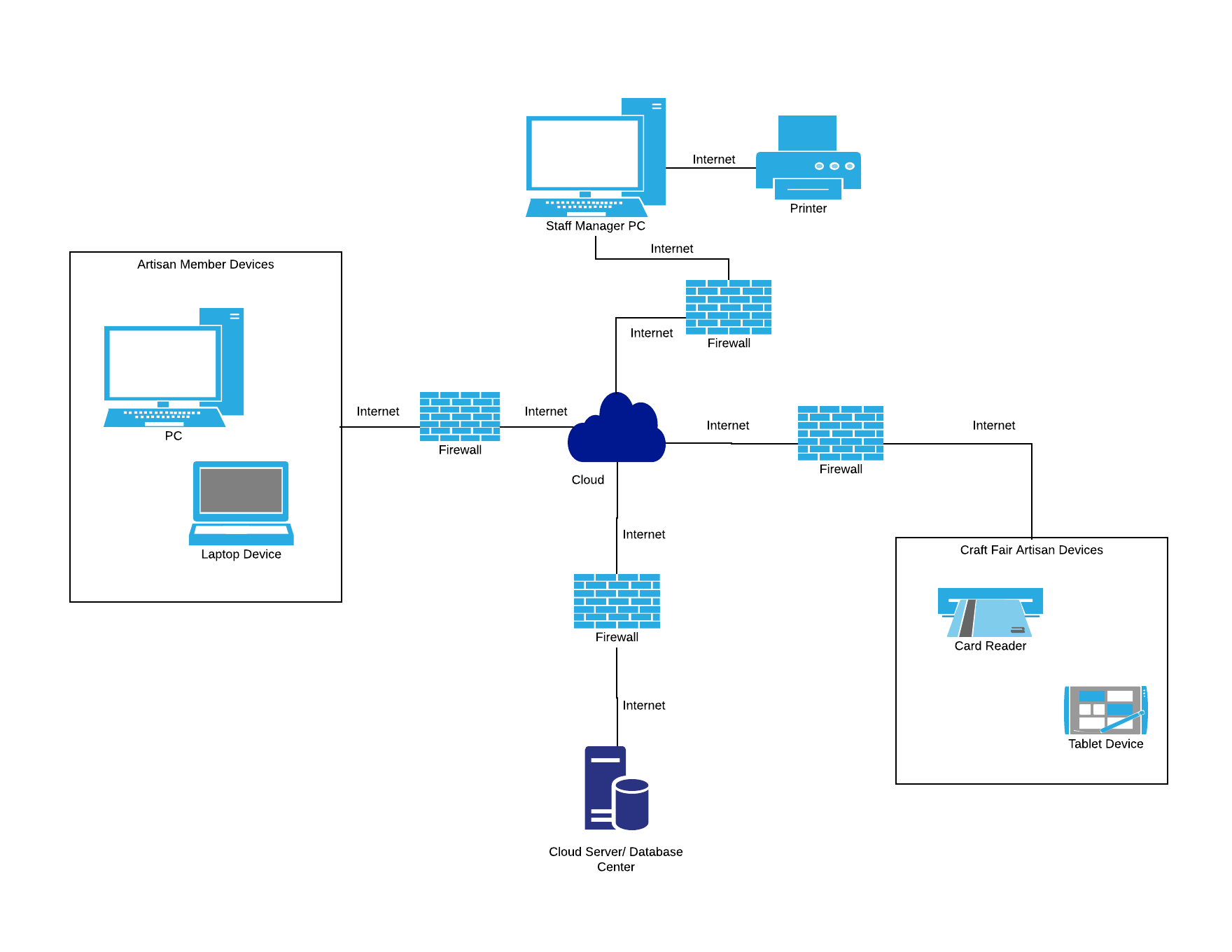
## 3.1 Introduction

This section includes two infrastructure models, descriptions of any necessary hardware and software systems, and an initial security plan for Artisan Thrive Technology (ATT). The first infrastructure model will be an architecture overview and the second will be over the nodes and artifacts of the system. Both infrastructure diagrams will represent how the hardware and software components of ATT will work together to produce a reliable system.

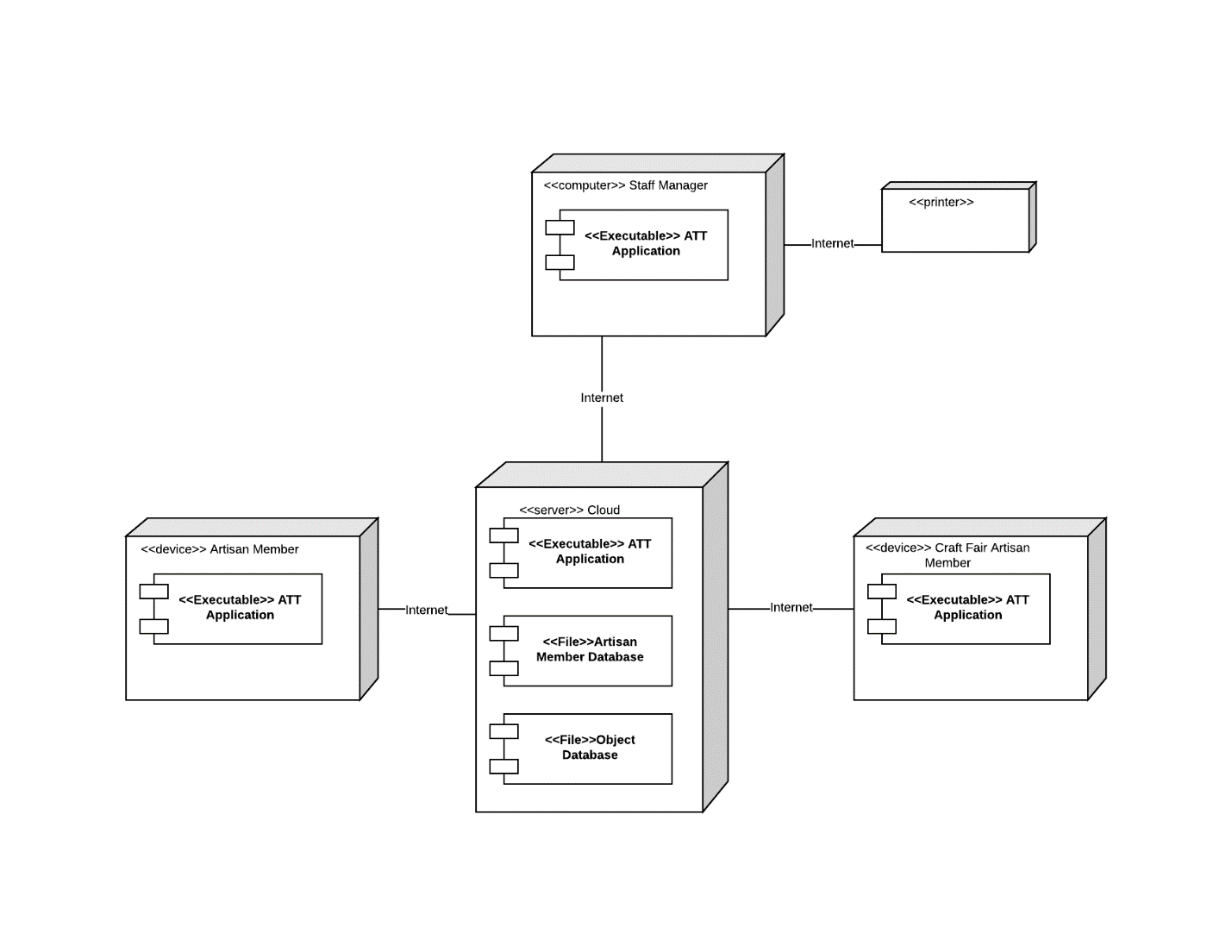
CAMR Inc. recommends a third party cloud service provider. The cloud service will handle all logic, data access and storage. This is an ideal architecture for Artisan’s Co-op North because data will be readily available for all users whether he or she is in the warehouse/basement, in the privacy of their own homes, or at a local café.

## 3.2 Infrastructure Model

Deployment Diagram 1 - Architecture Overview



### Deployment Diagram 2 – Nodes and Artifacts



## 3.3 Hardware and Software Requirements

Required Hardware Components

* Computer for manager and authorized staff in basement/warehouse
* Any computer or portable devices for all artisan members that will have access to either the database or website
* Cloud Server subscription
* Printer to print reports

Required Software Components

* ATT must support a number of operating systems including Windows(7 or later), Mac(El Capitan or later), Linux, Android (5.0 or later), and iOS (7 or later)
* ATT must support a number of web browsers, including Chrome, Safari, Firefox, and Microsoft Edge.
* Square Software must be implemented with the database software and website
* Website must be able to read and write to the database software
* Cloud service will need to include database management and be able to work with ATT software.

## 3.4 Security Plan

A system with like Artisan Thrive Technology (ATT), will have potential threats from unauthorized users and loss of sensitive information from any physical factors. Some precautionary measures to reduce the risk of threats can be taken by implementing firewalls and using reputable payment and cloud services. Loss of sensitive information due to physical disasters can be prevented with maintenance backups. Theft can be prevented with anti-virus software, encryption, and alarms. The following table will give specific controls that will help prevent threats on ATT components.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Threats | | | | | |
| Components | Power Loss | Fire | Flood | Virus | Physical Theft/ Loss | Unauthorized Access |
| Server | 2, 14, 15 | 2, 14, 15 | 2, 14, 15 | 4, 5, 6, 7, 14 | 14, 15 | 5, 6, 7, 13, 14 |
| Database | 2, 14, 15 | 2, 14, 15 | 2, 14, 15 | 4, 5, 6, 7, 14 | 8, 9, 14, 15 | 5, 6, 7, 13, 14 |
| Artisan's Co-op North Devices | 1, 2, 3, 8, 15 | 1, 2, 8, 10, 15 | 1, 2, 8, 15 | 4, 6 | 2, 8, 9 | 6, 11, 12, 13, 14 |
| Individual Artisan's Devices | X | X | X | 4, 7 | X | 7, 11, 12, 13, 14 |
| Customer Devices | X | X | X | 4, 5 | X | 5, 12, 13, 14 |

Controls

1. Host computer in the basement
2. Disaster recovery and emergency plans
3. Back-up power generator for basement computer
4. Anti-Virus Software
5. Firewall between client devices and cloud server
6. Firewall between Artisan’s Co-op North devices and cloud server
7. Firewall between artisan's devices and cloud server
8. Insurance for hardware purchased by Artisan's Co-op North
9. Basement security alarms
10. Fire extinguishers and sprinklers in basement
11. Extensive driver training on preventing loss and theft of devices and sensitive information
12. User training on password and account security
13. Strong encryption software for password and all credentials
14. Hire a reputable and secure payment method and cloud service providers
15. Scheduled backup saved to the cloud for restoring database

# **4. User-Interface**

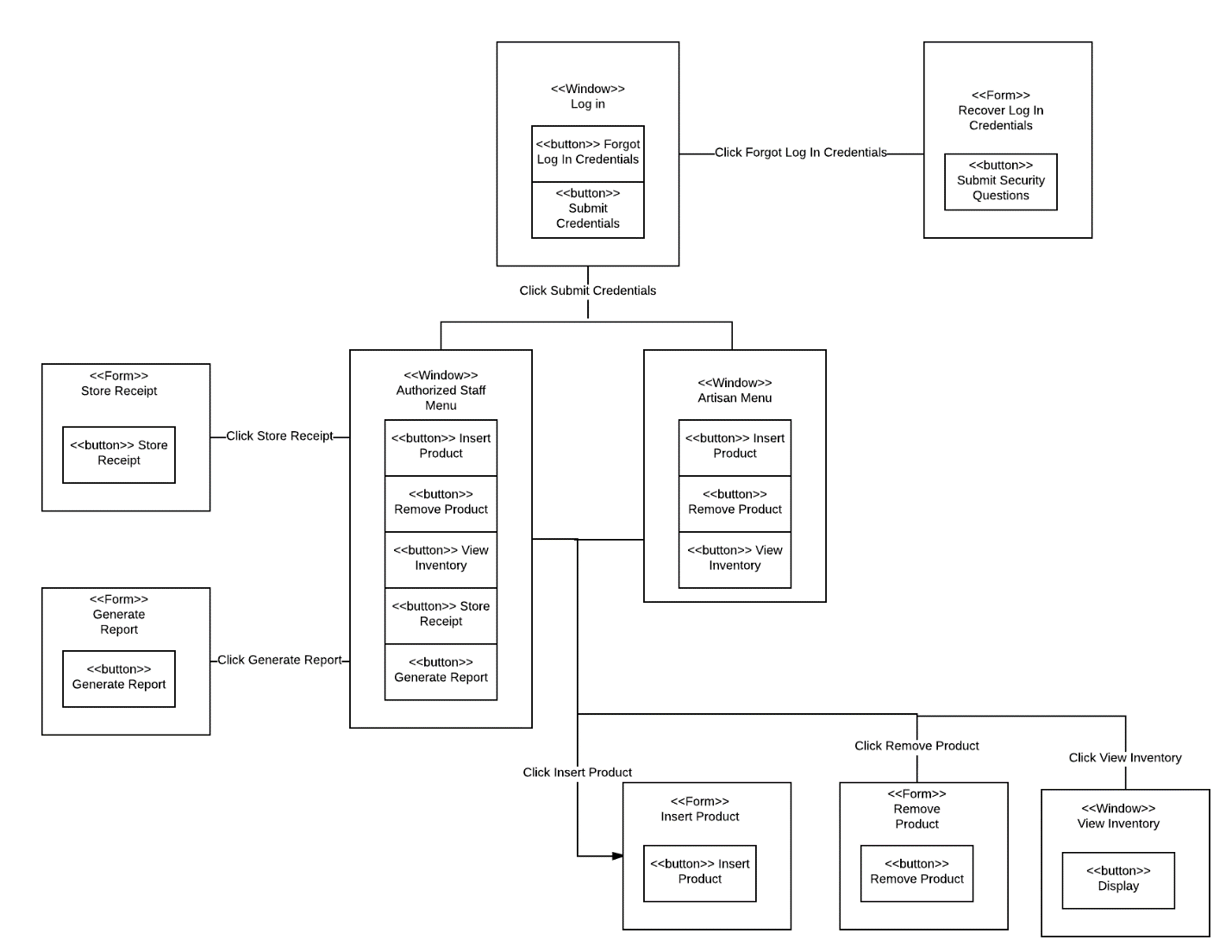
## 4.1 User-Interface Requirements and Constraints

The following section gives a basic layout for the user-interface design. Section 4.2, will include a window navigation diagram of all the interface elements and how they all fit together. Section 4.3, is a more detailed version of a few general windows in Artisan Thrive Technology (ATT). Lastly, Section 4.4, will include the basic printed output of a report (pay roll).

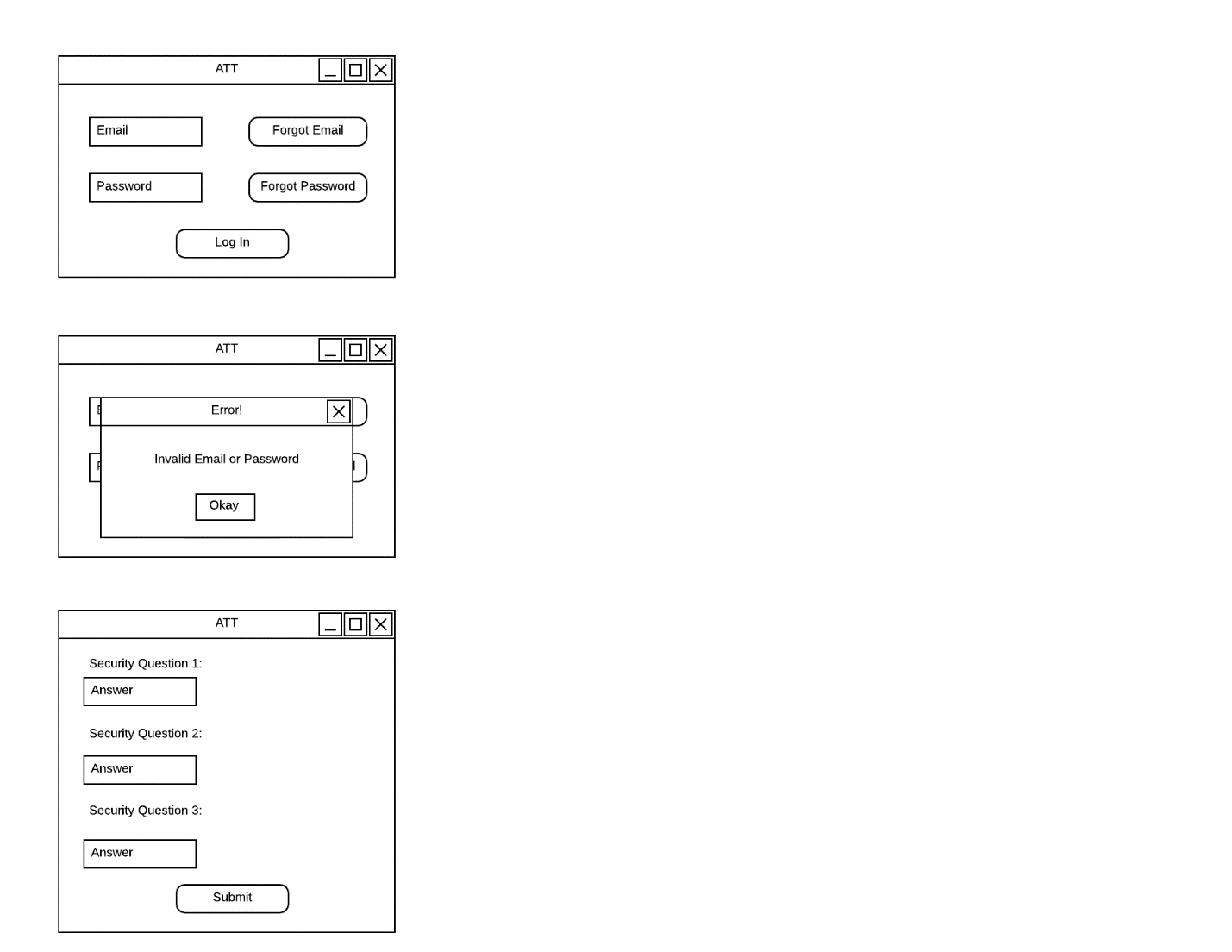
It is important to remember ATT is a simple application program that allows users to log in and will display an appropriate window based on the user credentials entered. The requirements for the application would be to provide ease of use for all artisan members while maintaining its efficiency capabilities.

The screens on the next few pages have different symbols: ovals, text boxes, and tables. The ovals represent buttons to click, text boxes represent where a user can enter text, and tables display data in an organized manner for the user.

## 4.2 Window Navigation Diagram



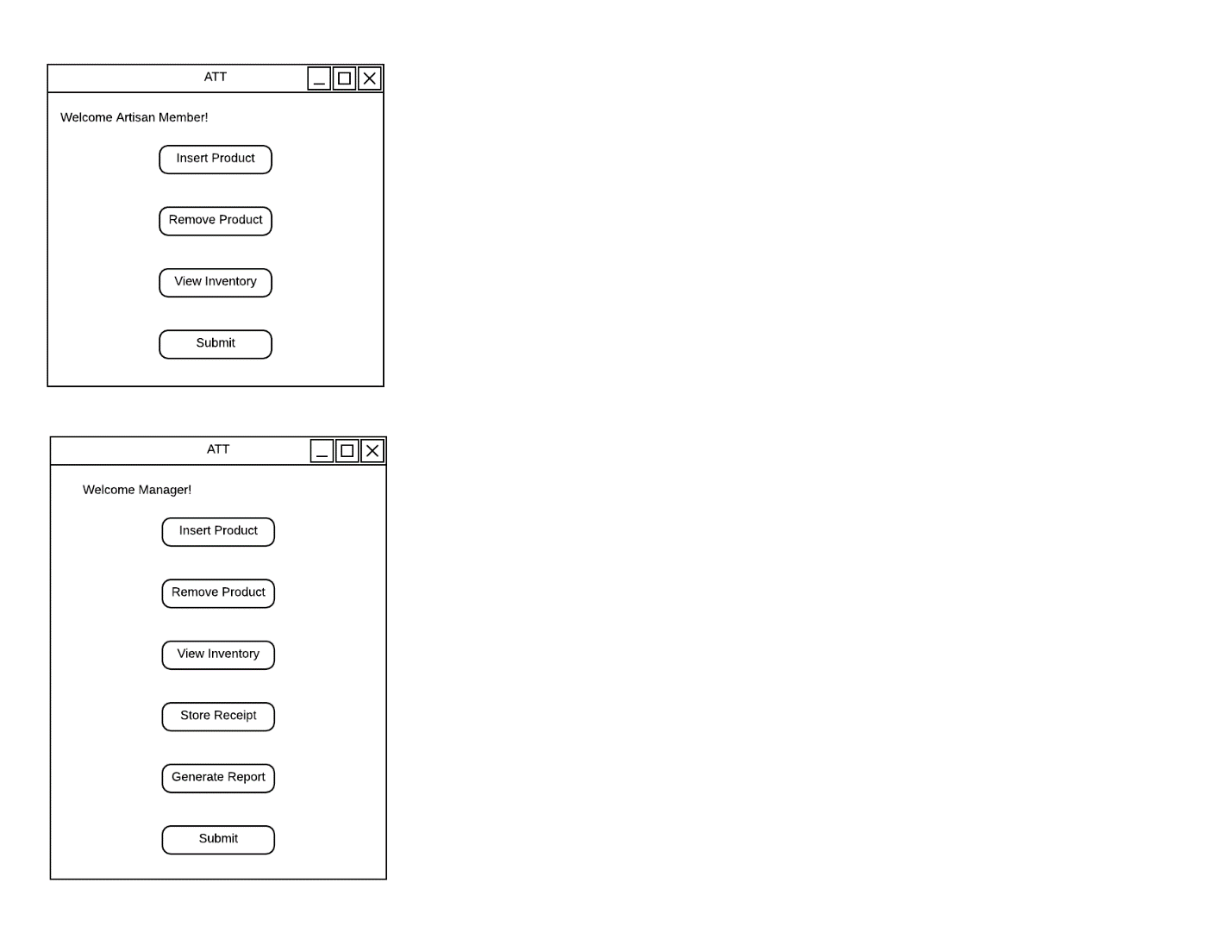
## 4.3 Forms: Screen and User- Interaction Design



Log In for all artisan members, authorized staff, and managers

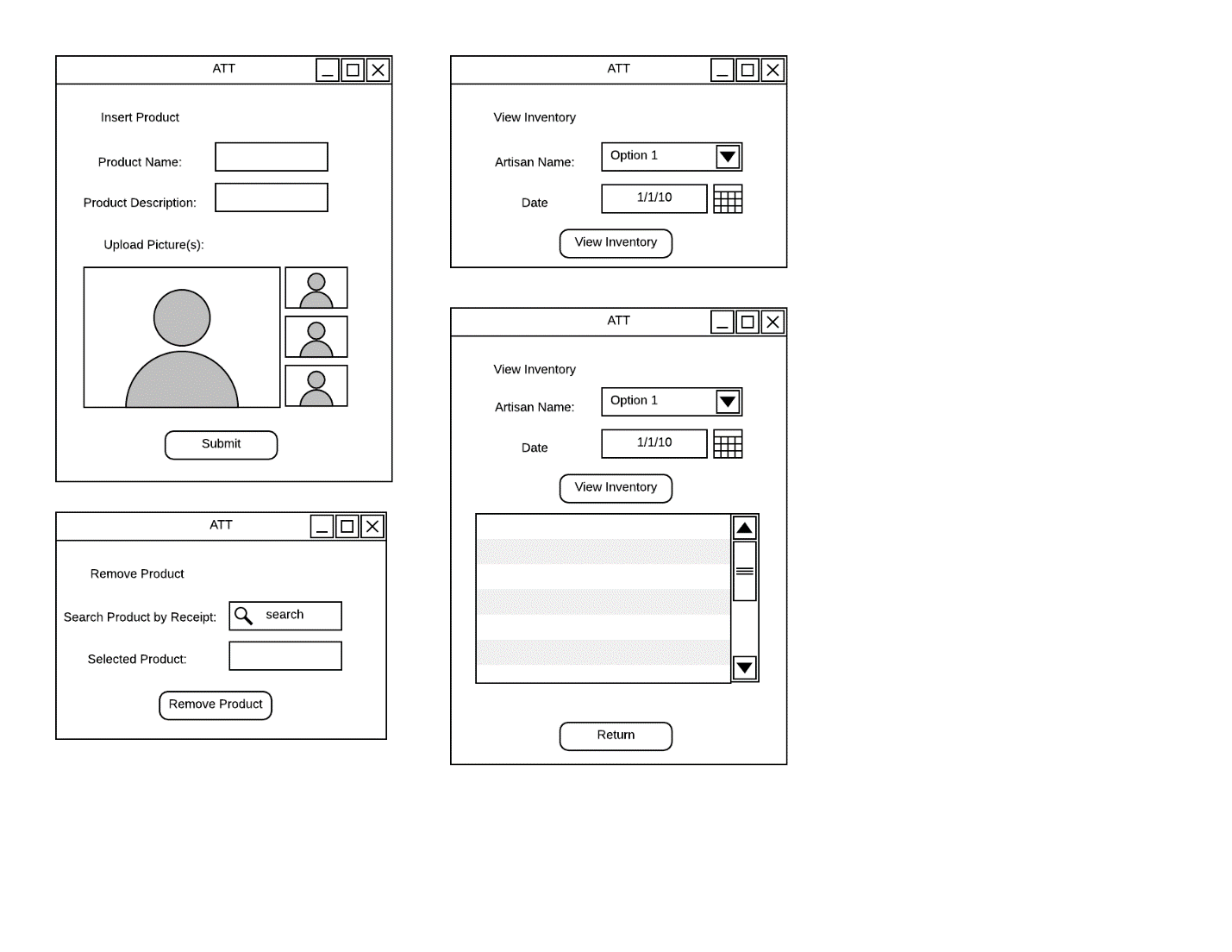
Process to Recover Log in Credentials. Includes three security questions of user’s choice that have already been saved to the database.

Error Message when log in credentials are incorrect.



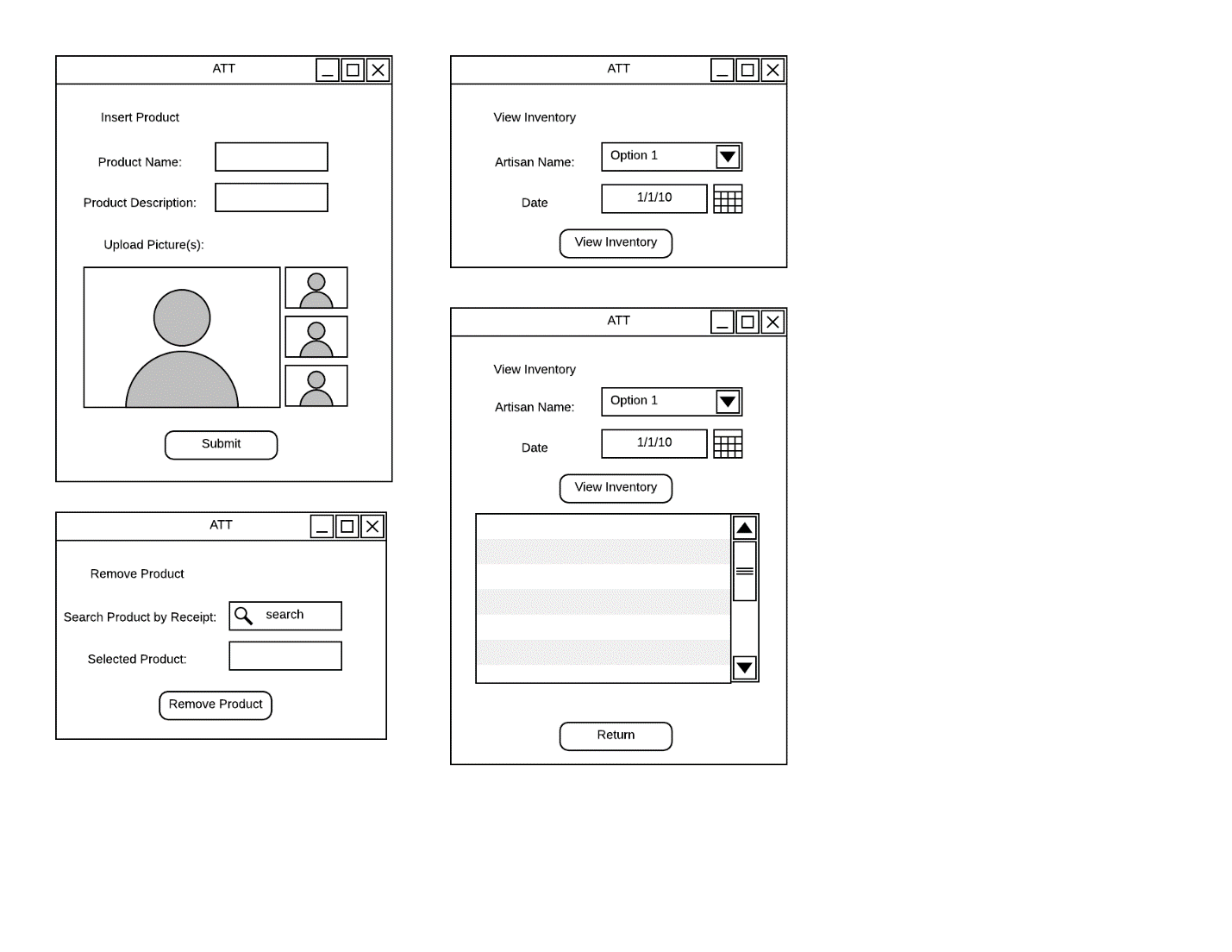
Successful Log In screen for artisan manager. Manager has the options to Insert Product, Remove Product, View Inventory (of all Members), Store Receipts, and Generate Reports.

Successful Log In screen for artisan member. Member has the options to Insert Product, Remove Product, and View Inventory.



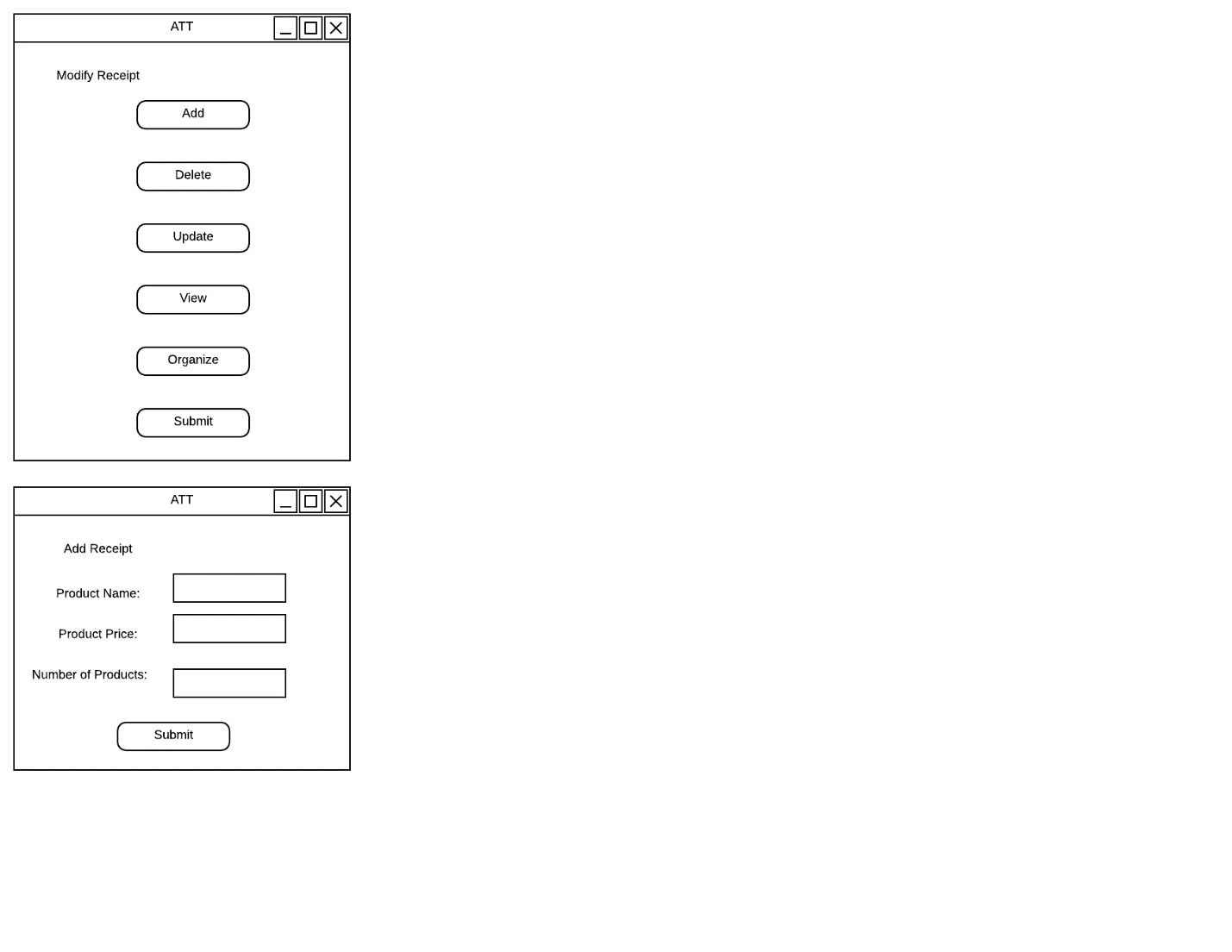
Form for Removing a Product from the database. Includes a Quick Search for the Product. Ensures right product is selected before deleting product.

Form for Inserting a Product into the database. Includes Product Name, Product Description, and Images (optional).



Form for Viewing a Product from the database. Includes an Artisan Name Field only if logged in as a Manager. Includes Date to View Inventory History.

Display for Viewing a Product from the database. Includes a table of Inventory History based on the date entered.



Display for Modify Receipt for a purchased product. Includes functions such as Add, Delete, Update, View, and Organize.

Form for Adding a Receipt into the database. Includes Product Name, Product Price, and Number of Products. This window is similar to the other functions done by Modify Receipt.

## 4.4 Reports: “Printed Outputs” Design

This is the basic printed output for a manager creating a payroll for an artisan member.

Artisan Thrive Technology

<Report Name>

<Report Type>

<Date>

Artisan Name: <Artisan Name>

Manager Name: <Manager Name>

Products Sold:

<List of all Products Sold>

Total Profit: <Profit>

<Calculation of Fees>

Total: <Amount Paid to Artisan>

<Manager Signature>

# **Appendices**

## Bibliography

*CSC 3150 Sample Book.* (2017). Seattle, WA: Seattle Pacific University.

Dennis , A., Wixon, B. H., & Tegarden, D. P. (2011). *Systems Analysis Design: An Object Oriented Approach with UML* (Fifth ed.). Hoboken, NJ: John Wiley & Sons.

Pfeiffer, W. S. (2011). *Pocket Guide to Technical Communication.* Upper Saddle River, NJ: Prentice Hall.

Weltz, E. (n.d.). *Systems Design, various Modules and Files [PowerPoint slides/Word Documents/PDFs].* Retrieved from SPU: http://canvas.spu.edu

## Supporting Documentation and Recommendations

CAMR Recommends the following cloud services for Artisan Thrive Technology:

* Atlantic.net
* Amazon Web Services
* Microsoft Azure
* Google Cloud Platform

For individual information on each cloud service, please visit <https://clutch.co/cloud>

CAMR Recommends the following secure payment services for Artisan Thrive Technology:

* Authorize.Net
* PayPal
* Google Checkout
* Amazon Payments
* Square Up

For individual information on each payment services, please visit <https://www.webpagefx.com/blog/web-design/online-payment-systems/> and <https://squareup.com/>