****

**Software Requirements Specification**

**<Pain and Gain>**

**Version 1.0 Approved**

**Presented by**

**<Abdullah Moiz (FA20-BSE-078)>**

**<Muhammad Ghous (FA20-BSE-091)>**

**<Awaab Khan (FA20-BSE-092)>**

**<Haider Ali (FA20-BSE-095)>**

**<Organization>:** CUI WAH

**<Date>: 20th December, 2022**

|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | | | |
| Date | **Revision** | **Description** | **Author** |
| 20th December, 2022 | **1.0** | **Initial Version** | **Abdullah Moiz**  **Muhammad Ghous**  **Awaab Khan**  **Haider Ali** |

Contents

[1. Introduction 3](#_Toc122647969)

[1.1 Purpose: 3](#_Toc122647970)

[1.2 Intended Audience and Reading Suggestion: 3](#_Toc122647971)

[1.3 Product Scope 3](#_Toc122647972)

[2.Overall Description 4](#_Toc122647973)

[2.1 Product Perspective: 4](#_Toc122647974)

[2.2 Operating Environment: 4](#_Toc122647975)

[2.3 Design and Implementation Constraints: 5](#_Toc122647976)

[2.4 User Documentation: 5](#_Toc122647977)

[3.External Interface Requirement 5](#_Toc122647978)

[3.1 User interface: 5](#_Toc122647979)

[3.2 Hardware Interface: 5](#_Toc122647980)

[3.3 Software interface: 5](#_Toc122647981)

[3.4 Communication interface: 6](#_Toc122647982)

[4. System Features 6](#_Toc122647983)

[4.1 Functional Requirements 6](#_Toc122647984)

[5.Other Non-Functional Requirements 7](#_Toc122647985)

[5.1 Performance Requirements: 7](#_Toc122647986)

[5.2 Safety Requirements: 8](#_Toc122647987)

[5.3 Security Requirements: 8](#_Toc122647988)

[5.4 Software Quality Attributes: 8](#_Toc122647989)

[5.5 Business Requirements: 8](#_Toc122647990)

[6. Use Case Model Survey 8](#_Toc122647991)

[7. Actor Survey 10](#_Toc122647992)

[8. Diagram 11](#_Toc122647993)

[8.1 Class Diagram: 11](#_Toc122647994)

[8.2 ER Diagram: 12](#_Toc122647995)

[8.3 Pseudo Code: 13](#_Toc122647996)

[8.4 Flow Chart: 16](#_Toc122647997)

[8.5 Decision Tree: 19](#_Toc122647998)

[8.6 User Interface: 22](#_Toc122647999)

# Introduction

## 1.1 Purpose:

* Scheduling online fitness classes
* Selling diet and gym products
* Online gym scheduling
* Increase concerned for over-weight
* Manage client coaching by giving personal training

**Significance of the system:**

* Admin dashboard. Which helps admin to view all the information simply and easily
* Easy and Interactive UI/UX.
* Supplier dashboard (can easily Add, update and delete products)
* Customer dashboard. For customer it is an easiest way to access the website, buy product and train.

## 1.2 Intended Audience and Reading Suggestion:

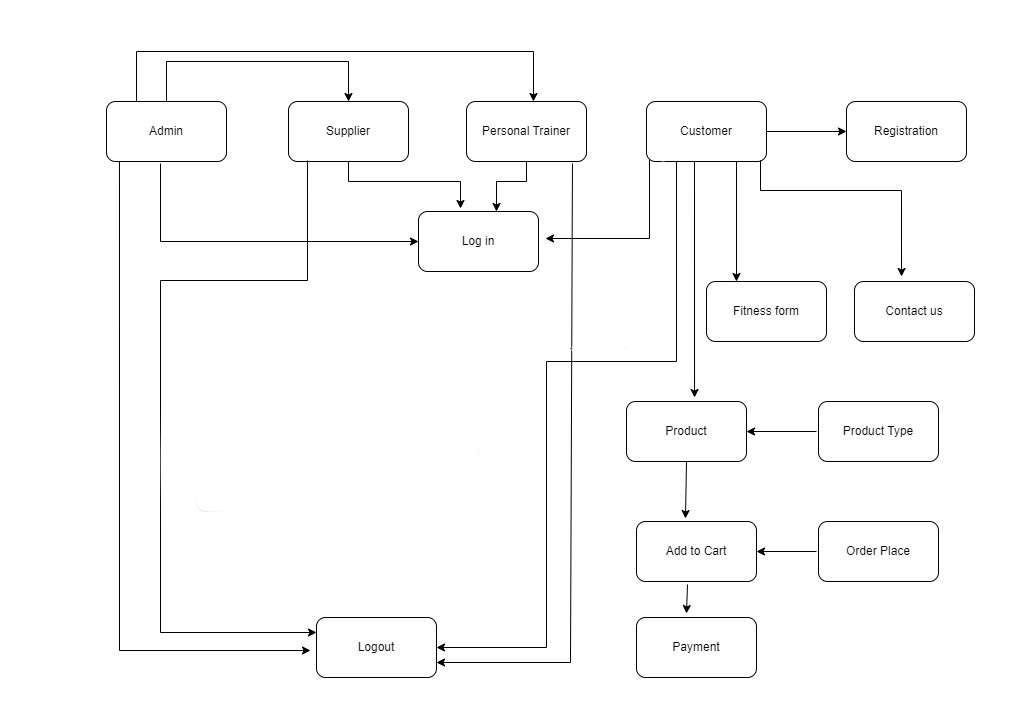
As this requirement specification document involves the necessity of having technical knowledge about the project so it is intended to be made for the technical members of the team like developer, analysis and everyone in the team. It is not made for non-technical stakeholder including customer and etc.

## 1.3 Product Scope

* **Customer**: Customer can easily perform day to day routine exercise without any complexity, training by professional.
* **Supplier**: Provides a platform for the supplier to sell his products easily and earn money.
* **Personal Trainer**: Provides a platform for personal trainer to train immature customer routine exercise and get benefit from that.

# 2.Overall Description

## 2.1 Product Perspective:



## 2.2 Operating Environment:

**Designing Tools:**

* Draw.io
* Microsoft Visio

**Frontend:**

* React JS
* Tailwind CSS

**Backend:**

* Rest API’s
* Node JS
* Express JS

**Architecture:**

* Model-view-controller (MVC)

**Database:**

* Mongo DB
* Firebase

**Browser:**

* Google chrome
* Firefox

## 2.3 Design and Implementation Constraints:

* **Design constraints:**

Developer is limited to work on visual studio code (vs code) and use git version control

* **Implementation constraints:**

MERN Stack

M-> Mongo DB

E-> Express JS

R-> React JS

N-> Node JS

## 2.4 User Documentation:

* Contact us
* Meeting arrangements
* Professional Trainer

# 3.External Interface Requirement

## 3.1 User interface:

Discount voucher are displayed on the landing page whenever a new user or a guest user visit the website this strategy attracts the user.

## 3.2 Hardware Interface:

We don’t have any external hardware use in our project.

## 3.3 Software interface:

Customer can sign in to fitness management system and after signing in he can access two modules.

* **Ecommerce**

Now if the person wants to buy fitness product he can visit the ecommerce module and can add his favourites products to add to cart, and after this he can buy this product a payment form will appear on screen and after filling the payment form he will receive the email notification of confirmation that the order had been placed.

* **Personal Trainer:** 
  + Personal Trainer need’s first apply on our platform through an Application Form (that is related to a specific detail including Professional information’s).
  + After that he can begin training customers and can communicate with them through zoom and chat.

## 3.4 Communication interface:

If the customer has any problem regarding anything there is a help option on the navigation bar which will automatically render the contact us page on this form you can fill the whatever query you have and the customer will receive email that his request has been submitted.

We are using HTTP (Hyper Text Transfer Protocol) protocols to serve website, and we use SMTP (Simple Mail Transfer Protocol) for email communication and verification.

# 4. System Features

## 4.1 Functional Requirements

* **Login System:**

Admin, Supplier, Personal Trainer, Customer can login-via Email and Password

* **Admin Module:** 
  + - Admin can Add Supplier (Mean an admin have authority to provide pre-defined credentials once supplier agreed to our Terms and conditions).
    - Admin Can Remove Supplier in case of any violation of Terms and Services or any valid complain from customer.
    - Admin can access the Contact details or profile of Supplier.
    - Admin can provide a pre-defined credentials to Supplier after validating his/her application.
* **Supplier Module:** 
  + - Supplier need’s first apply on our platform through an Application Form (that is related to a specific detail including personal or Professional information’s).
    - Supplier can do following:
      * Supplier can add Products.
      * Supplier can Update product info.
      * Supplier can view products.
      * Supplier can Delete products.
      * Supplier can login with pre-defined Credentials and can Logout.
* **Customer Module:**
  + - Customer can Register, once register he can login using the registered credentials.
    - Customer can purchase different product from our e-commerce Store.
    - Customer can train with our professional trainer.
* **Personal Trainer:** 
  + - Personal Trainer need’s first apply on our platform through an Application Form (that is related to a specific detail including Professional information’s).
    - After that he can begin training customers and can communicate with them through zoom and chat.
* **E-commerce Module functionalities:** 
  + Add to cart.
  + Wish list.
  + Payment method (PayPal, Venmo).
  + Search Products.

# 5.Other Non-Functional Requirements

## 5.1 Performance Requirements:

Performance should be high speed to save the time of customer for this purpose we limit the developer to work on only React JS and Node JS so that the website loading for the customer does not take long and his not an issue.

## 5.2 Safety Requirements:

We give first priority to our customer and the information that they share with us admin and managements is responsible to keep that information secure. Similarly, if someone from outside tries to engage with our website or try to steal data from our website for this purpose we use Firebase to authenticate our data.

## 5.3 Security Requirements:

* Email verification and authentication
* Pin code

## 5.4 Software Quality Attributes:

* Portability

## 5.5 Business Requirements:

* Provides a platform for personal trainer to train immature customer routine exercise and get benefit from that
* Our website provides an online fitness service as well as a personal trainer and also provide an opportunity to buy gym products
* Our product will be in React JS along with MongoDB database
* Will generate us greater revenue by the addition of two modules including personal trainer and e-commerce (selling product)

# 6. Use Case Model Survey

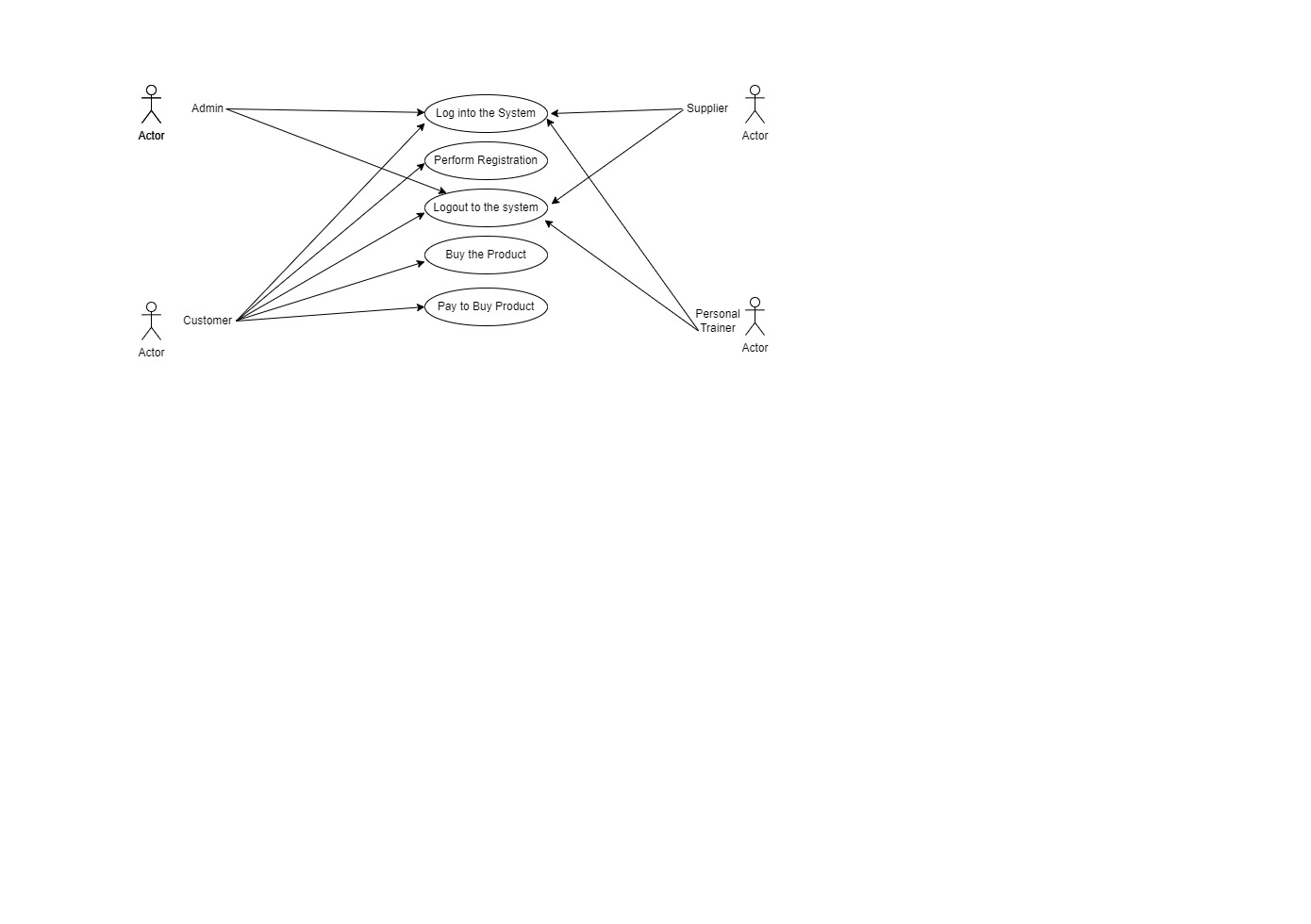
|  |  |
| --- | --- |
| Use case ID | **01** |
| Use Case Name | Login |
| Description | Personal Trainer, Admin, Customer and supplier can enter their email and password to login to the website to perform their operations. |
| Post-Condition | After login into the website ,he will be directed to the landing page. |
| Pre-condition | The actors must have an account before signing in. Actor must enter their email and password to sign in to the website. |
| Exception | The same email cannot be exist for one user |

|  |  |
| --- | --- |
| Use case ID | **02** |
| Use Case Name | Registration |
| Description | Customer can enter their email, password, confirm password to login to the website to perform their operations. |
| Post-Condition | After login into the website ,he will be directed to the landing page. |
| Pre-condition | Actor must enter their email and password to sign in to the website. |
| Exception | The same email cannot be exist for one user |

|  |  |
| --- | --- |
| Use case ID | **03** |
| Use Case Name | Buy Product |
| Description | Actors can view the product and supplier can update while the customer can buy the product and the product they like they can add them to their cart. |
| Post-Condition | Actors can view the product and supplier can update while the customer can buy the product and the product they like they can add them to their cart. |
| Pre-condition | The product already exist on the website. |
| Exception | Customer cannot buy the product without adding the product to their cart. |

|  |  |
| --- | --- |
| Use case ID | **04** |
| Use Case Name | logout |
| Description | Actors can logout the website. |
| Post-Condition | After logout the actors will be redirected to the sign in. |
| Pre-condition | Actors can logout the website. |
| Exception | Null |

|  |  |
| --- | --- |
| Use case ID | **05** |
| Use Case Name | Payment |
| Description | After selecting the product in order to buy that product the customer has to make the payment and will have a payment id when he will fulfill the payment form . |
| Post-Condition | After selecting the product in order to buy that product the customer has to make the payment and will have a payment id when he will fulfill the payment form. |
| Pre-condition | The customer have their payment method like(master and visa card) before make payment. |
| Exception | Without fulfilling the payment form the customer cannot buy the product. |

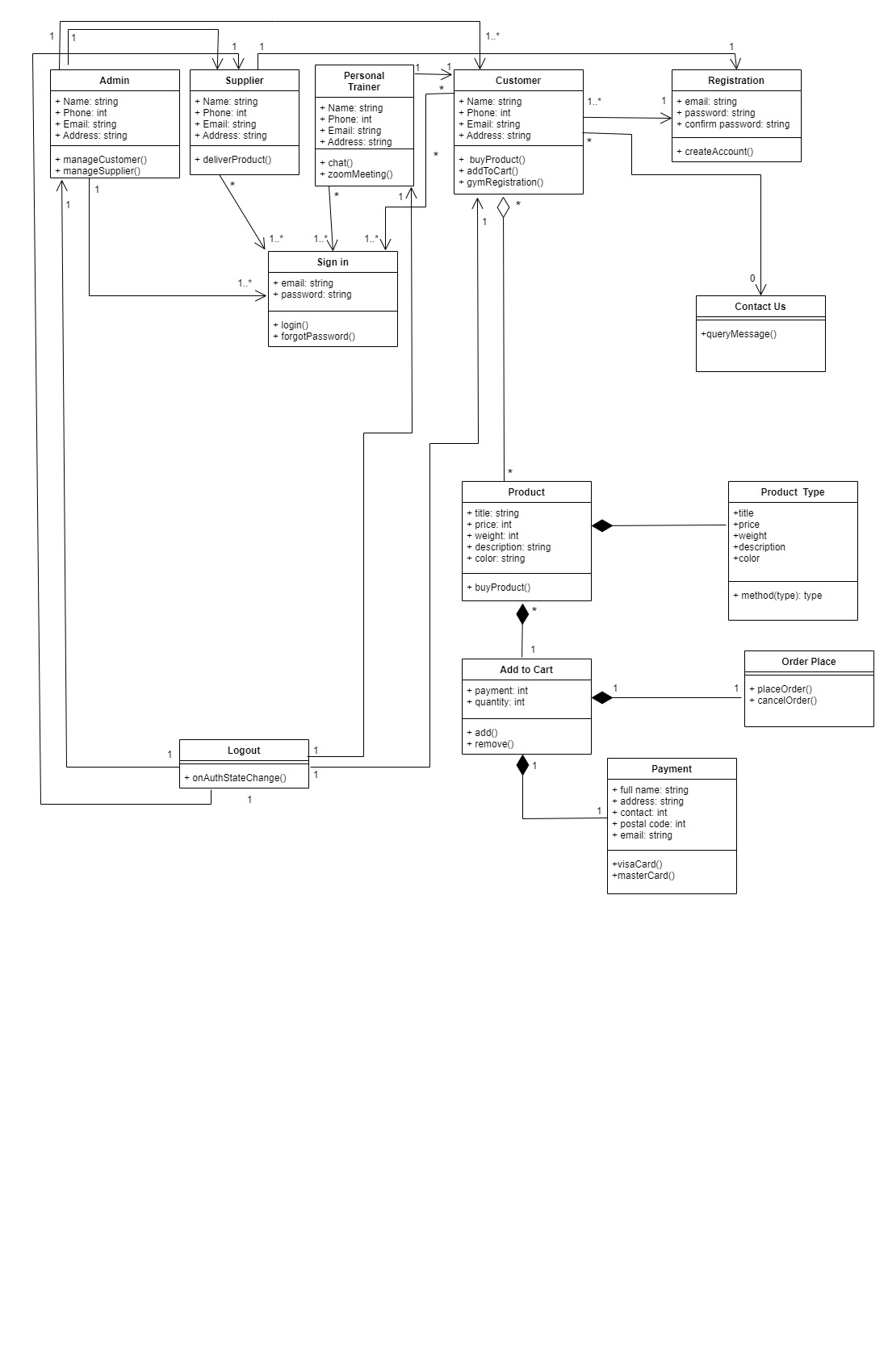


# 7. Actor Survey

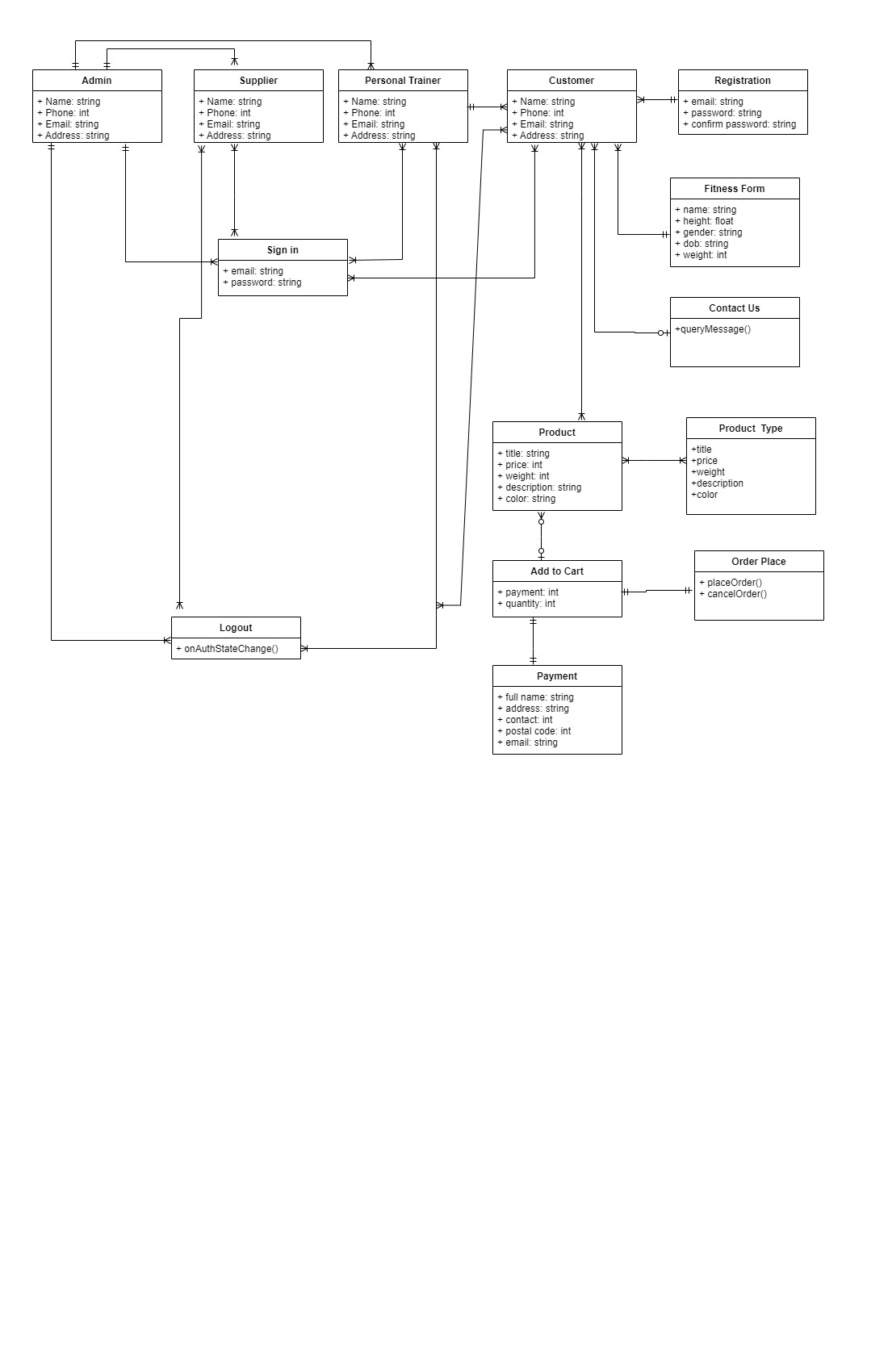
|  |  |
| --- | --- |
| Admin | Manages supplier and customer. |
| Customer | Buy product, Add to card, Gym registration, train from professional train. |
| Supplier | Send his profile to admin and then after selection set his profile and update products. |
| Personal Trainer | Send his profile to admin and then after selection set his profile and starts training customer personally |

# 8. Diagram

## 8.1 Class Diagram:



## 8.2 ER Diagram:



## 8.3 Pseudo Code:

**E-commerce module**

**1: Register**

start

Get userEmail

Get password

Get confirmPassword

IF (password == confirmPassword)

Start

Find userEmail

if exists

Return email already exist

Else

Store data

Return Registered successfully

End

Else

Start

Return Password doesn’t match

End

End IF

**2: Login**

Get userEmail

Get password

IF (loginEmail == enteredEmail && loginpassword == enteredPassword)

Return login successful

Else

Return Login Failed

End IF

**3: Forget Password**

Get userEmail

IF (userEmail == registeredEmail)

Start

Send reset verification link to registered email

Then

get new password

Return password changed Successfully

End

Else

Return emails isn’t registered

End IF

**4: Verified Email**

Send verification link on email with hashed Token

IF (verified through link)

Return account verified

Else

Show alert (unverified account)

End IF

**5: Add to Cart**

IF (userEmail != verified)

Return First verify your Email

Else

if ( email && product id exists )

Return product is already in cart

Else

Cart= [ ]

Cart.push(product)

Return product added to cart

End IF

End IF

**6: Shopping Cart**

**6.1 Increase/decrease Product quantity**

Let itemquantity = productQuantity

Increase product quantity

IF (supplierQuantity – itemquantity != 0 )

itemQuantity = itemQuantity + 1

supplierQuantity = supplierQuantity – itemquantity

else

return out of stock

decrease product quantity

IF(itemqunatity > 1)

Itemquantity = itemQuantity - 1

supplierQuantity = supplierQuantity + itemquantity

else

itemQuantity = 1

**6.2 calculate cart total Price**

cartPrice = array[]

cartPrice = cart.map(item =>

eachItemPrice = item.productQuantity \* item.productPrice

return eachItemPrice)

totalPrice = cartprice.reduce(prevSum , itemprice =>

prevSum + itemprice , 0

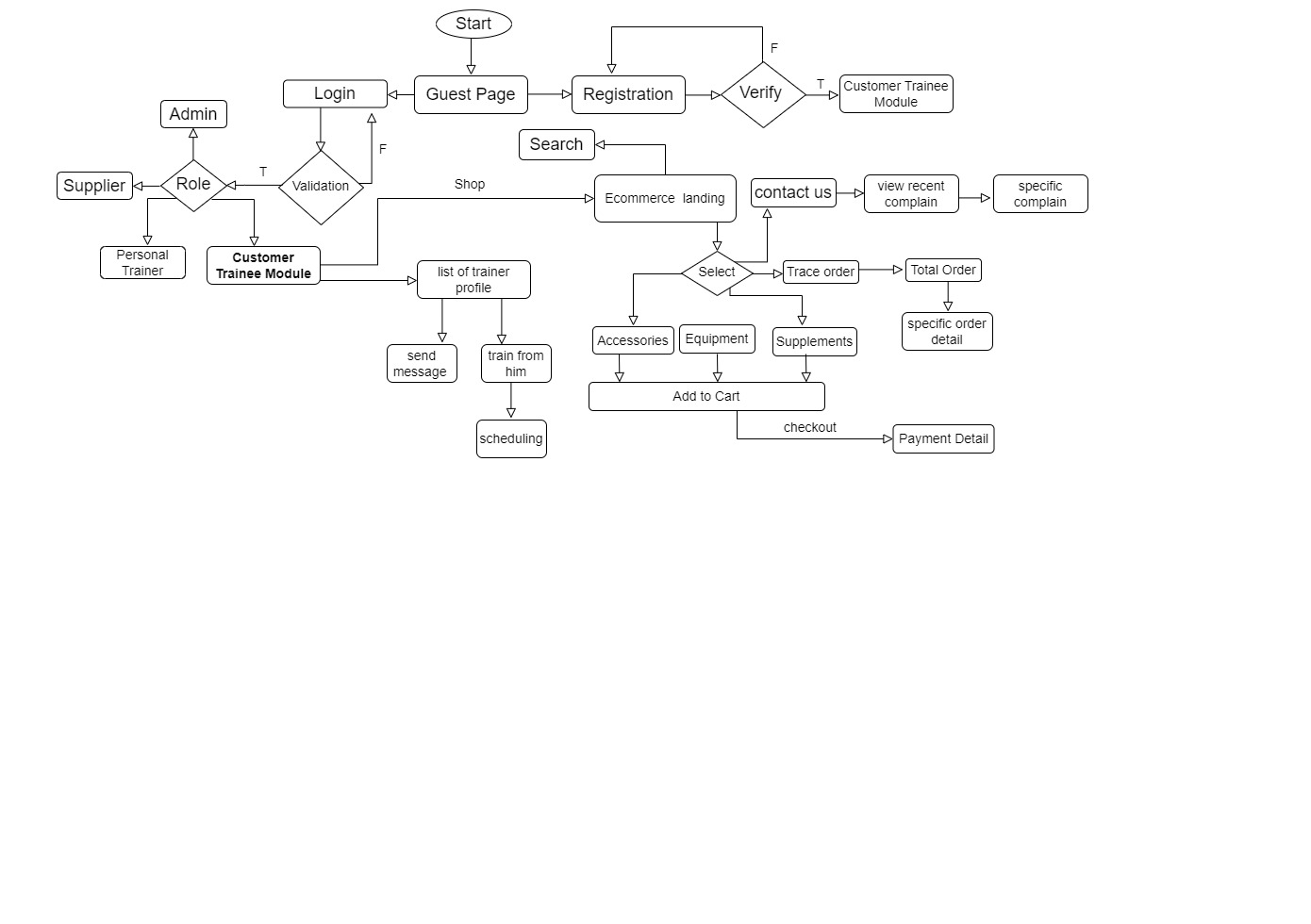
**6.3 remove cart Item**

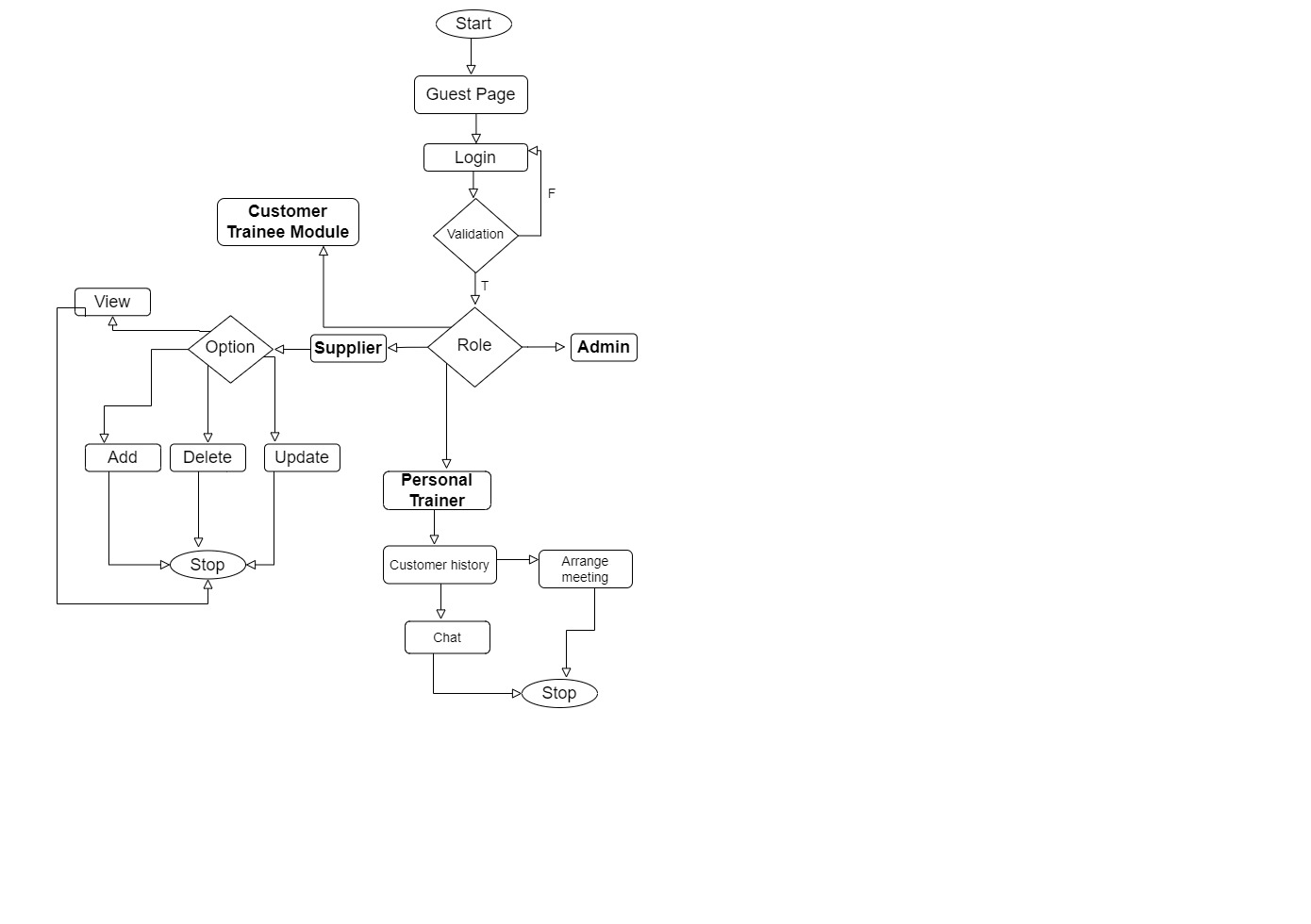
If (productID exists in cart)

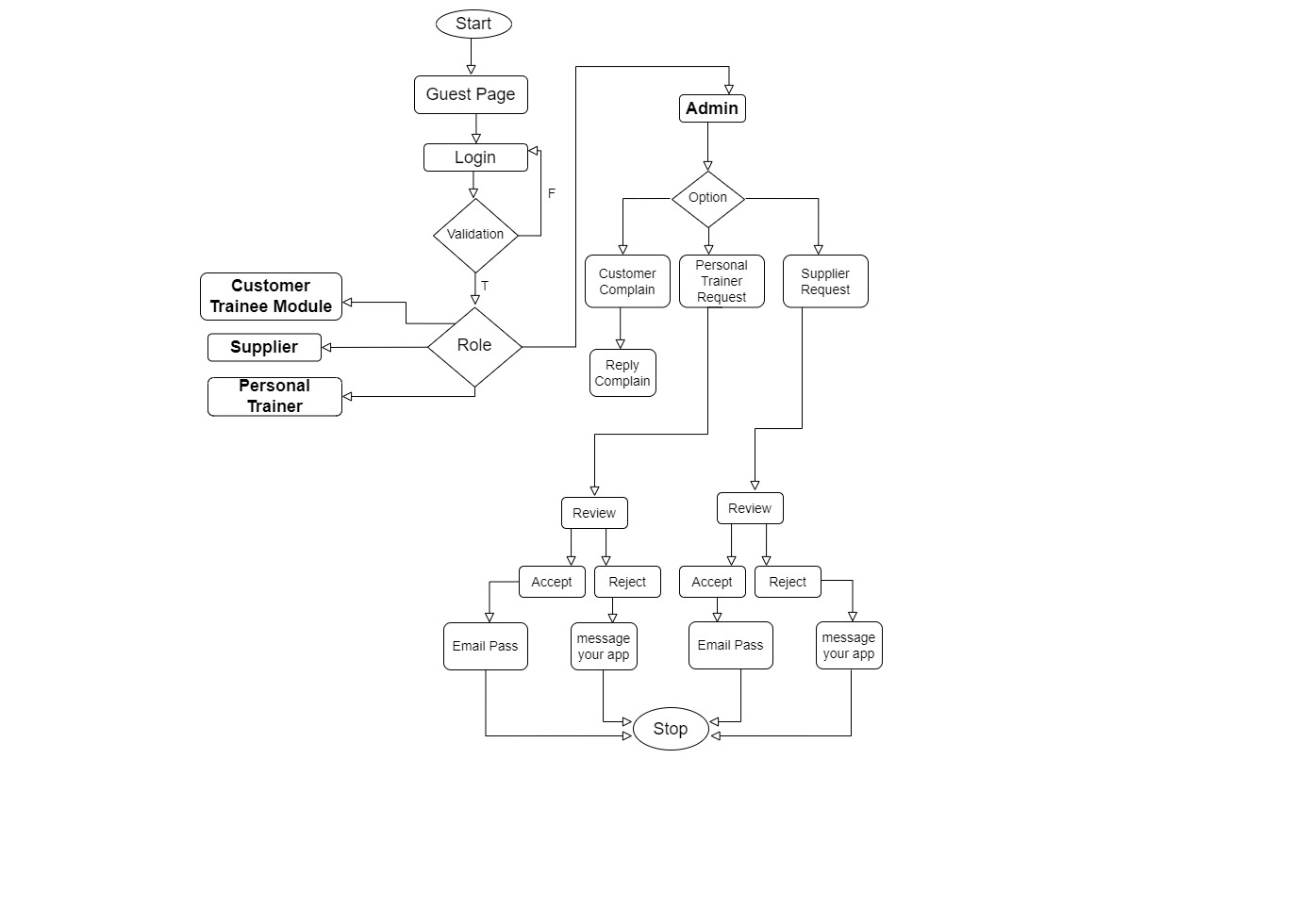
Cart.pop(product)

End IF

## 8.4 Flow Chart:

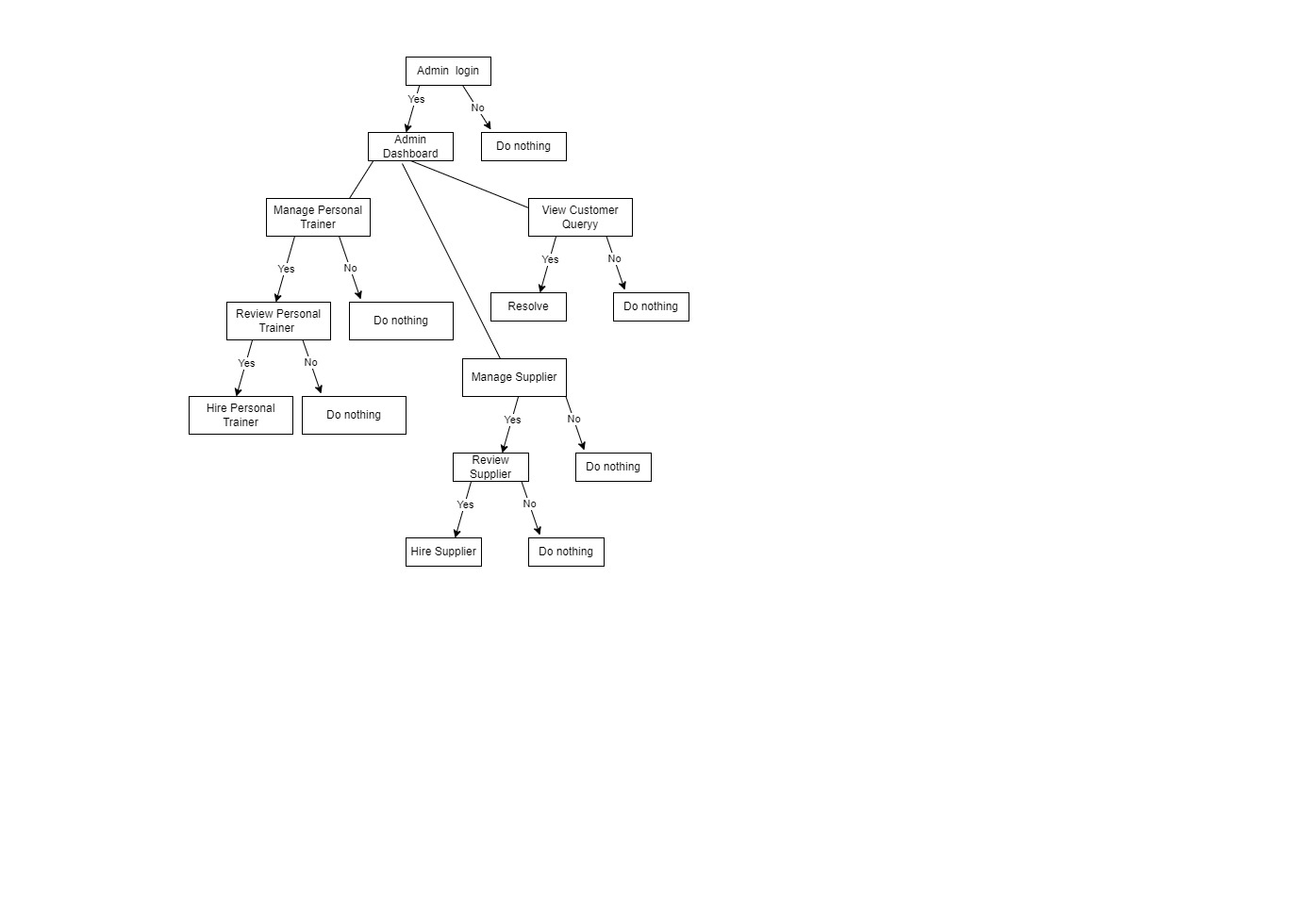




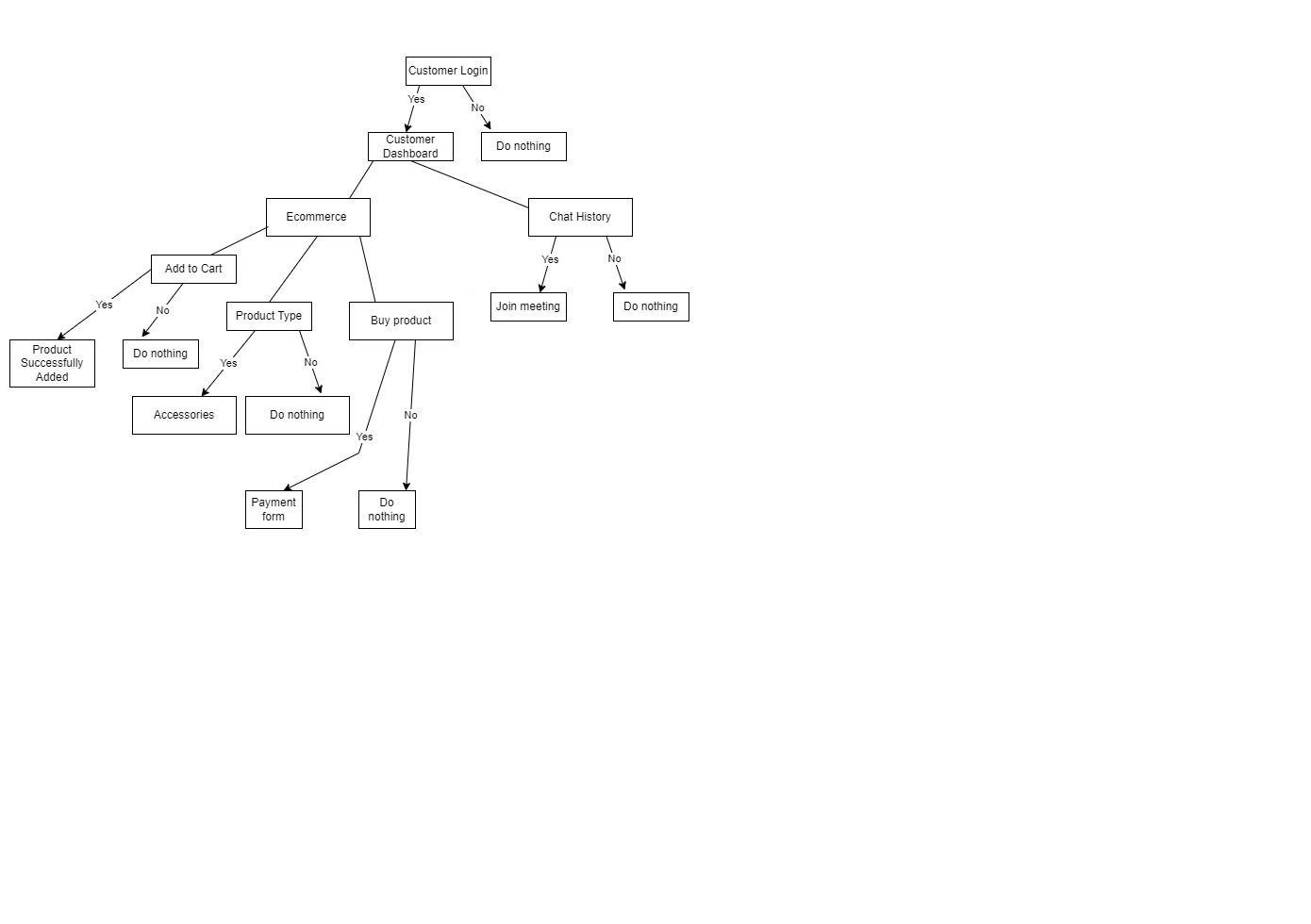


## 8.5 Decision Tree:

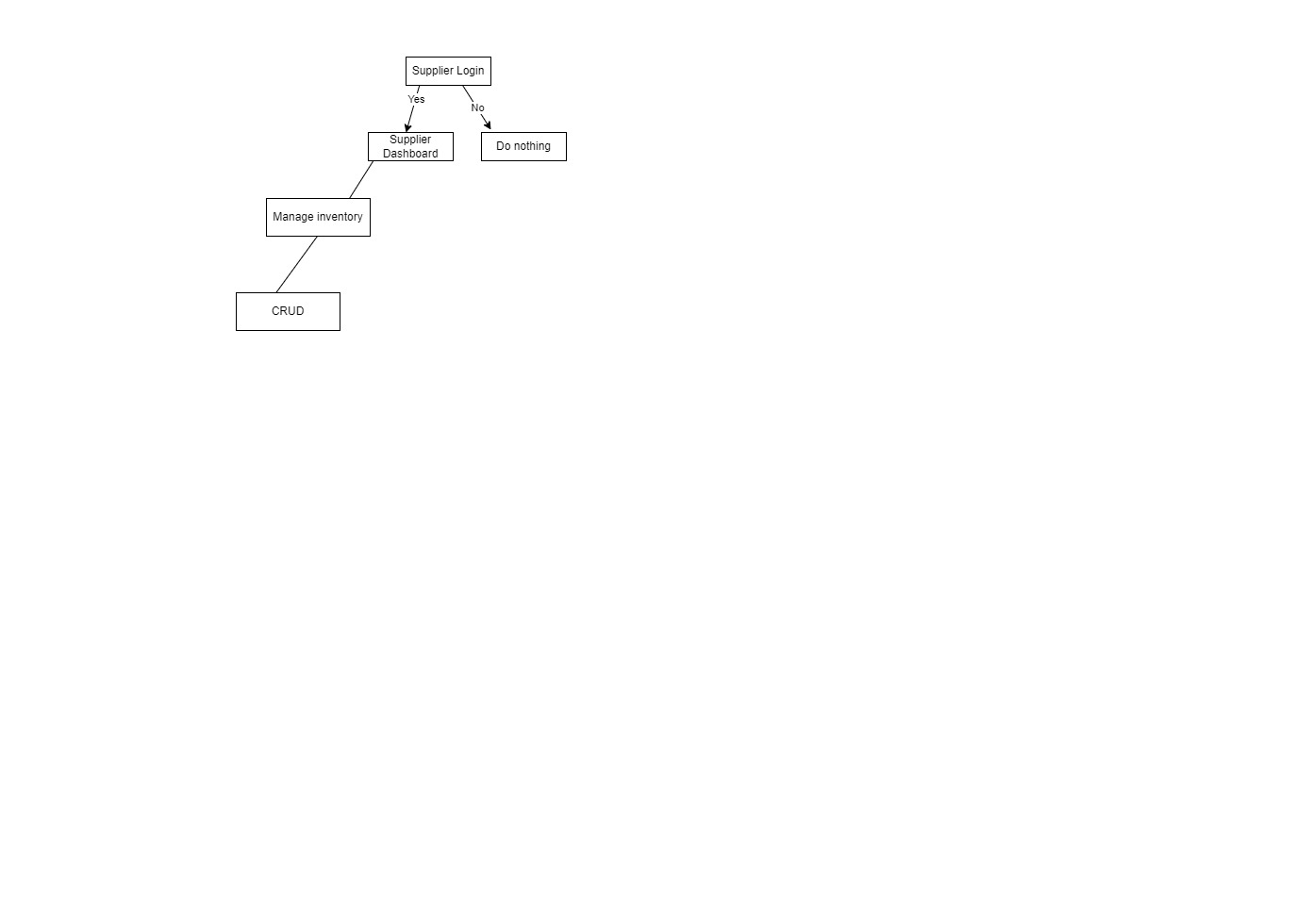
1. **Admin**



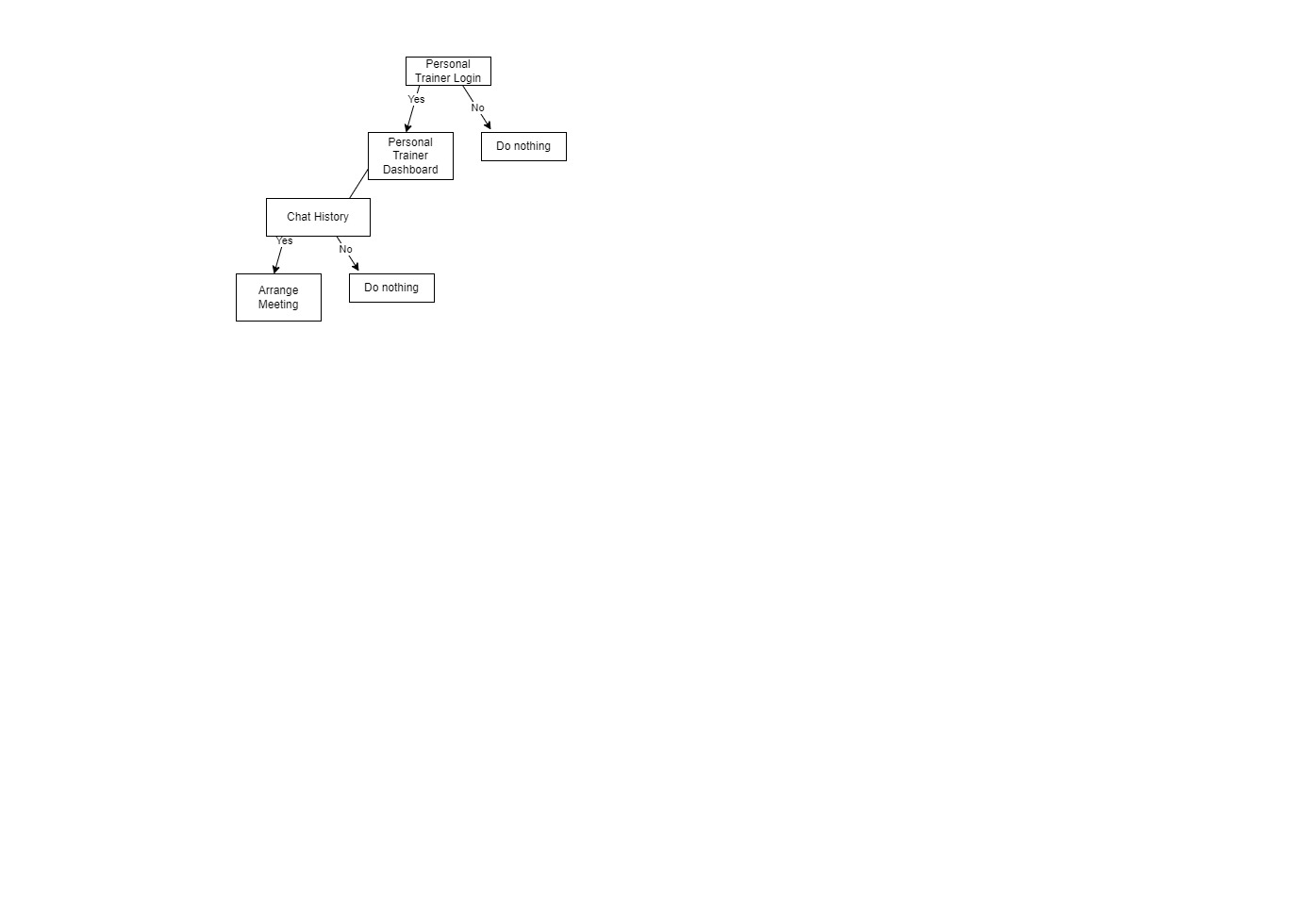
1. Customer



1. Supplier



1. Personal Trainer



## C:\Users\haider\AppData\Local\Microsoft\Windows\INetCache\Content.Word\WhatsApp Image 2022-12-20 at 12.55.41.jpg8.6 User Interface:

