

**Sneaker Online Shop System**

**Software Design Specification**

– Hanoi, April 2023 –

**Record of changeS**

| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

**Table of Contents**

[I. Overview 4](#_heading=h.gjdgxs)

[1. Code Packages 4](#_heading=h.30j0zll)

[2. Database Design 4](#_heading=h.1fob9te)

[a. Database Schema 4](#_heading=h.3znysh7)

[b. Table Description 4](#_heading=h.2et92p0)

[II. Code Designs 5](#_heading=h.tyjcwt)

[1. Login 5](#_heading=h.3dy6vkm)

[a. Class Diagram 5](#_heading=h.1t3h5sf)

[b. Class Specifications 5](#_heading=h.4d34og8)

[c. Sequence Diagram(s) 5](#_heading=h.2s8eyo1)

[d. Database queries 6](#_heading=h.17dp8vu)

[2. <Feature/Function Name2> 6](#_heading=h.3rdcrjn)

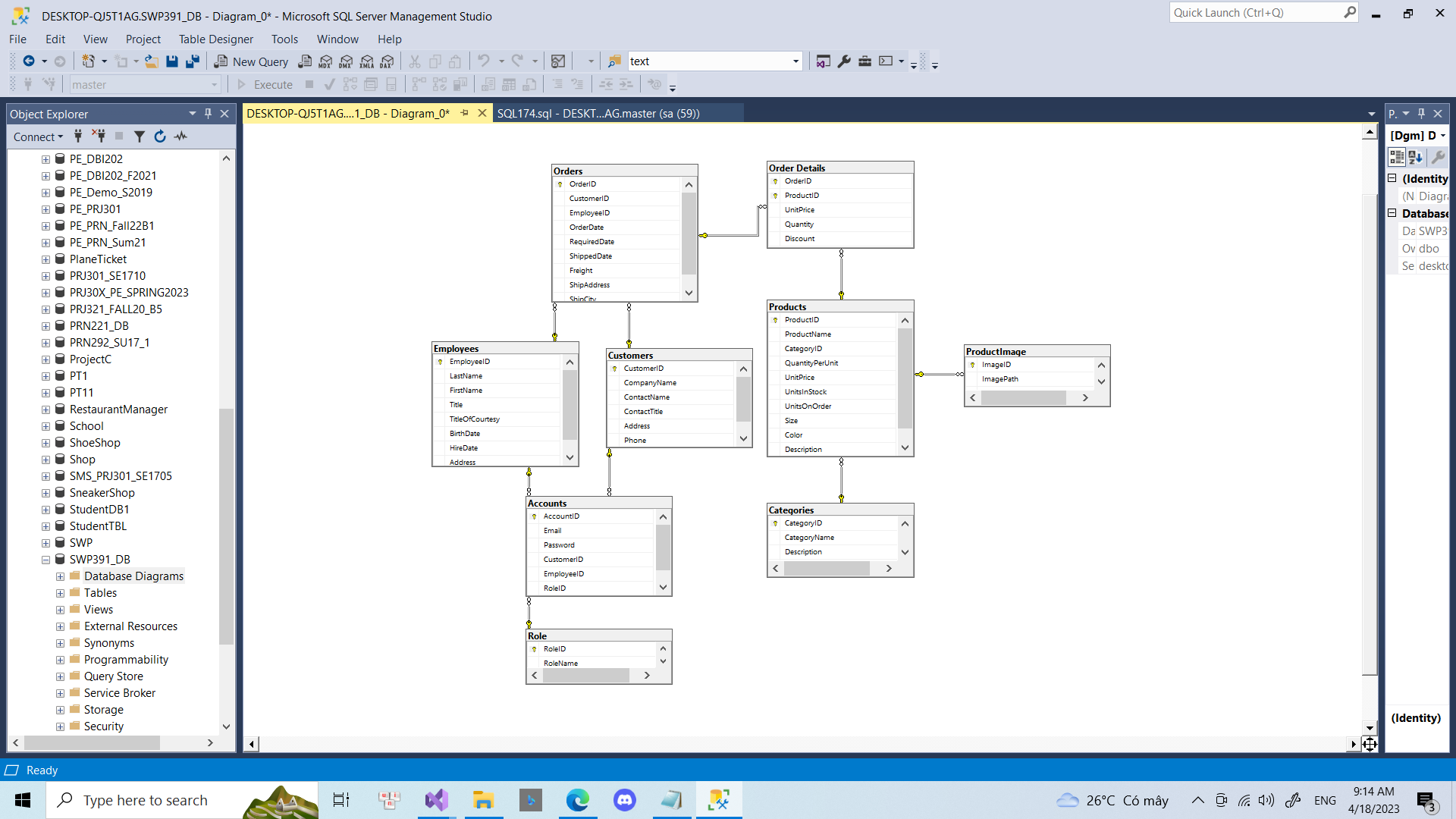
# I. Overview

## 1. Code Packages

***Package descriptions***

## 2. Database Design

### a. Database Schema



### b. Table Description

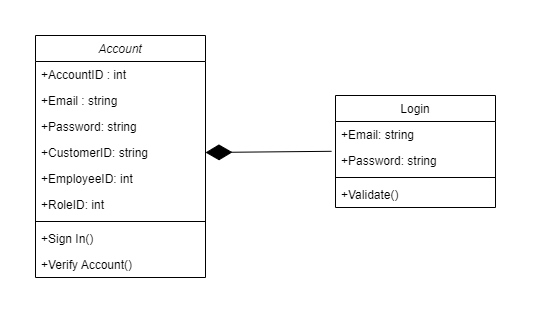
#### 

| **No** | **Table** | **Description** |
| --- | --- | --- |
| *01* | *Accounts* | *Get value of Accounts* |
| *02* | *Customers* | *Get value of Customers* |
| *03* | *Employees* | *Get value of Employees* |
| *04* | *Role* | *Contain role of admin/employee/customer* |
| *05* | *Orders* | *Get value of Order* |
| *06* | *ProductImage* | *Contain url of image of product* |
| *07* | *Products* | *Get value of Products* |
| *08* | *Order Details* | *Contain details value of a order* |
| *09* | *Categories* | *Get value of Categories* |

# II. Code Designs

## 1. Login

### a. Class Diagram

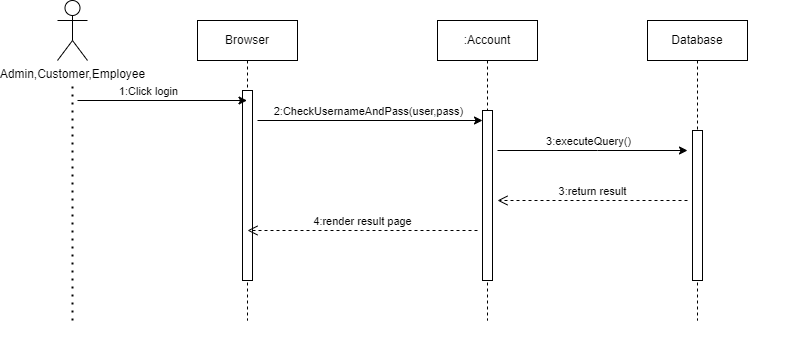


### b. Class Specifications

#### Account

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getter/setter* | *Get value of private attributes* |
|  |  |  |

### c. Sequence Diagram(s)



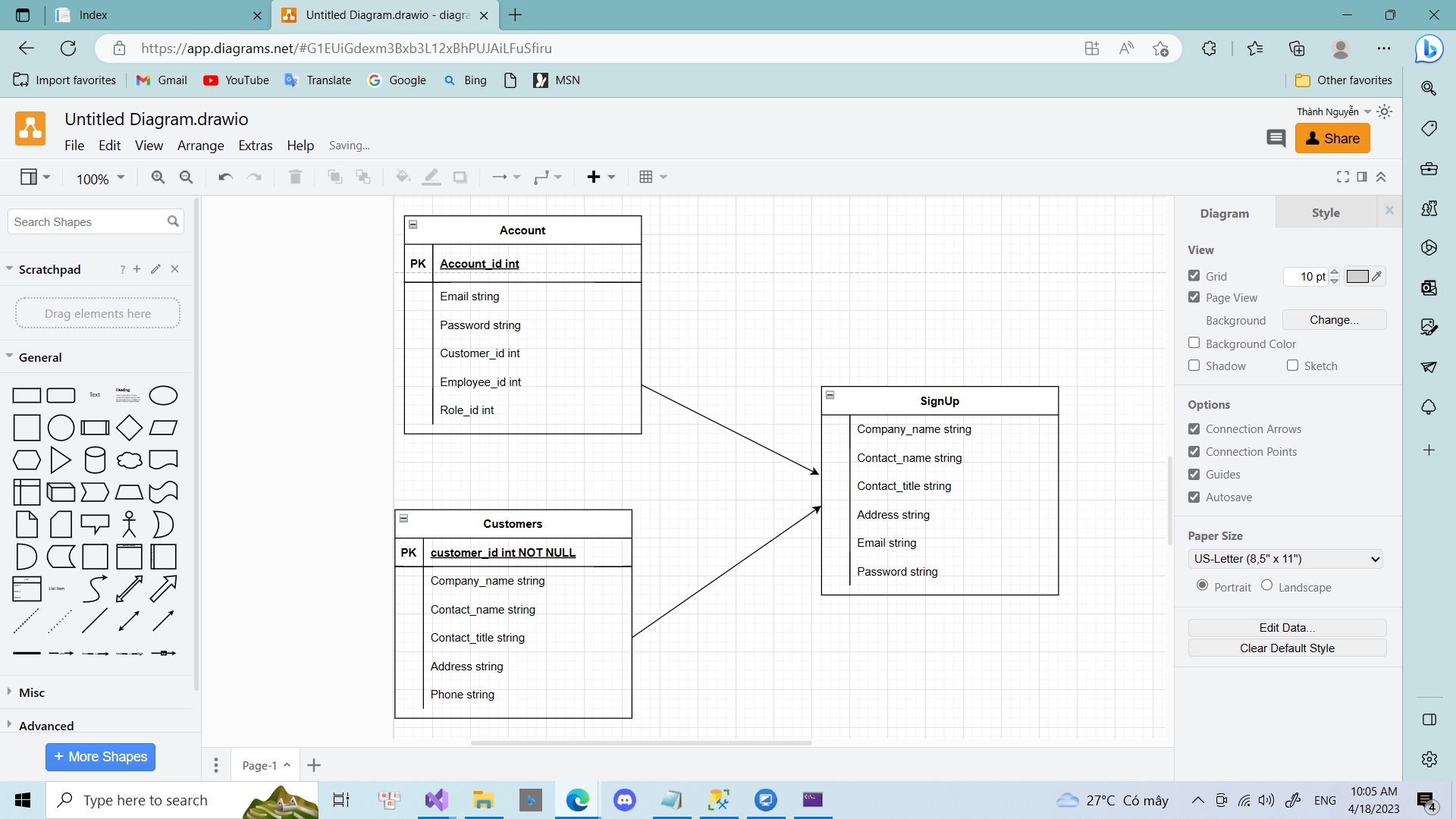
### d. Database Queries

await dBContext.Accounts.Where(

a => a.Email.Equals(account.Email) && a.Password.Equals(account.Password)).SingleOrDefaultAsync()

## 2. Sign up

### a. Class Diagram

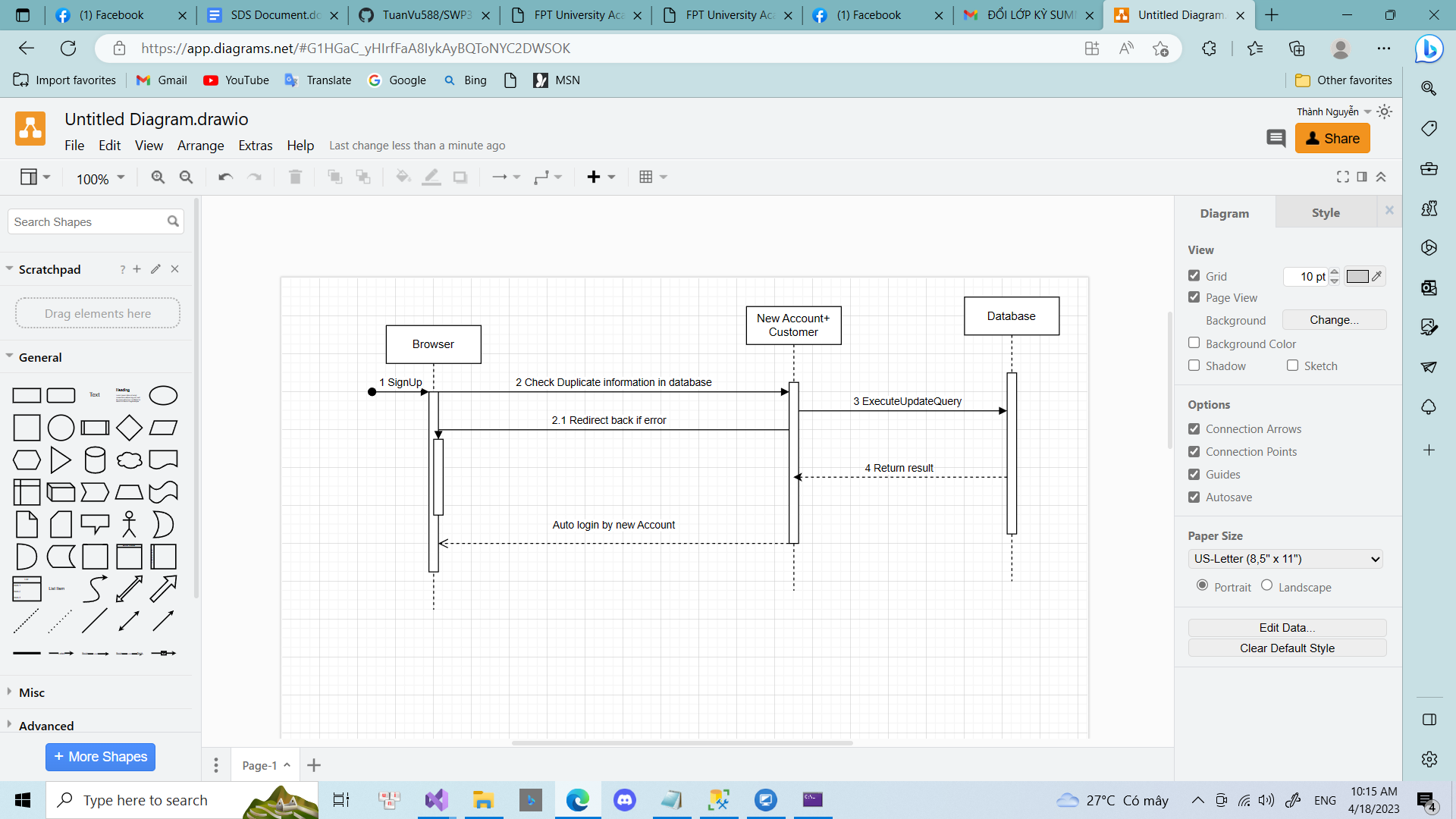


### b. Class Specifications

#### Account

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getter/setter* | *Get value of private attributes* |
|  |  |  |

### c. Sequence Diagram(s)



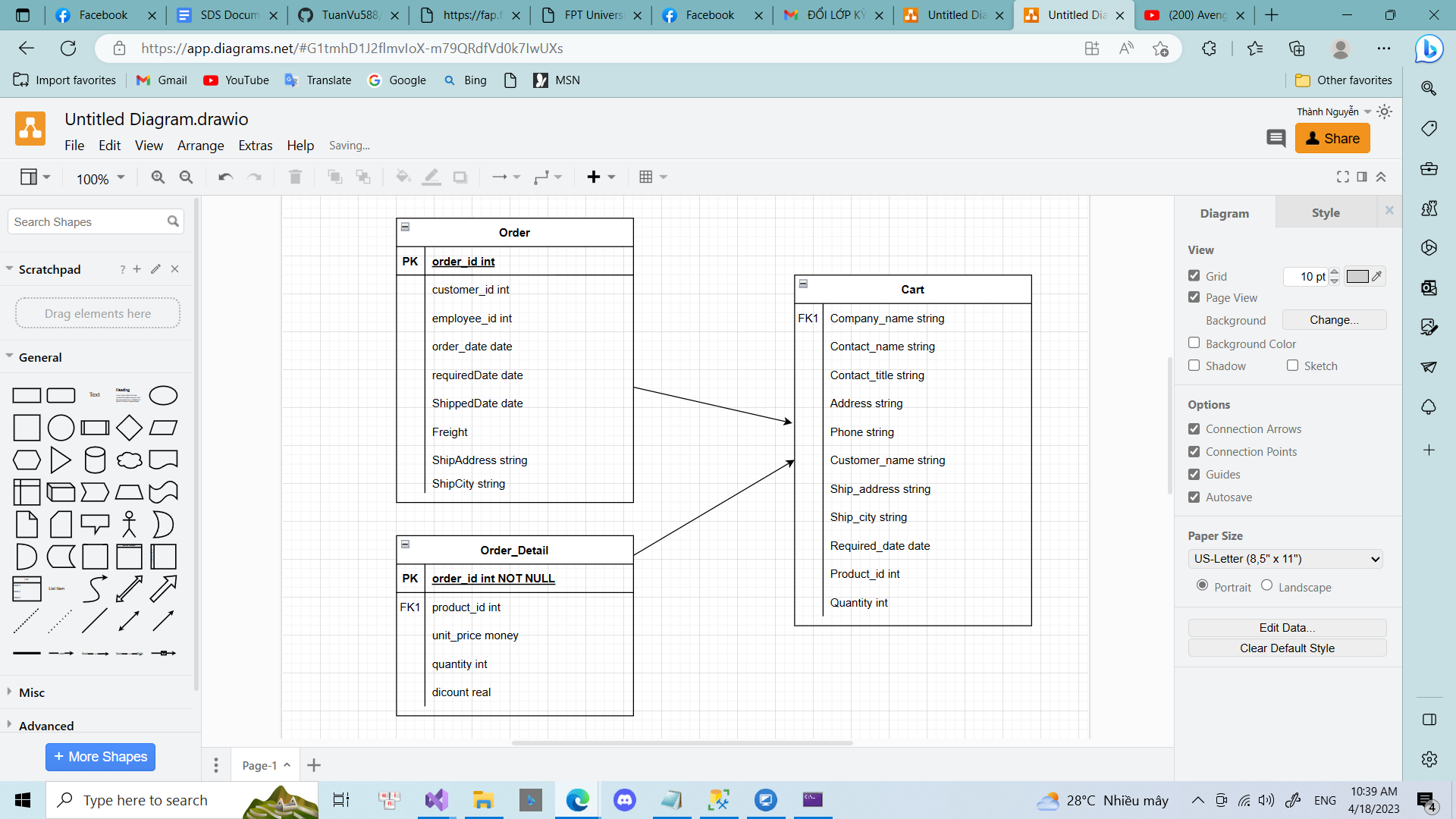
### d. Database Queries

await dBContext.Accounts.SingleOrDefaultAsync(a => a.Email.Equals(Account.Email));

await dBContext.Customers.Where(c => c.CustomerId.Equals(cusId)).FirstOrDefaultAsync();

## 3. Order

### a. Class Diagram

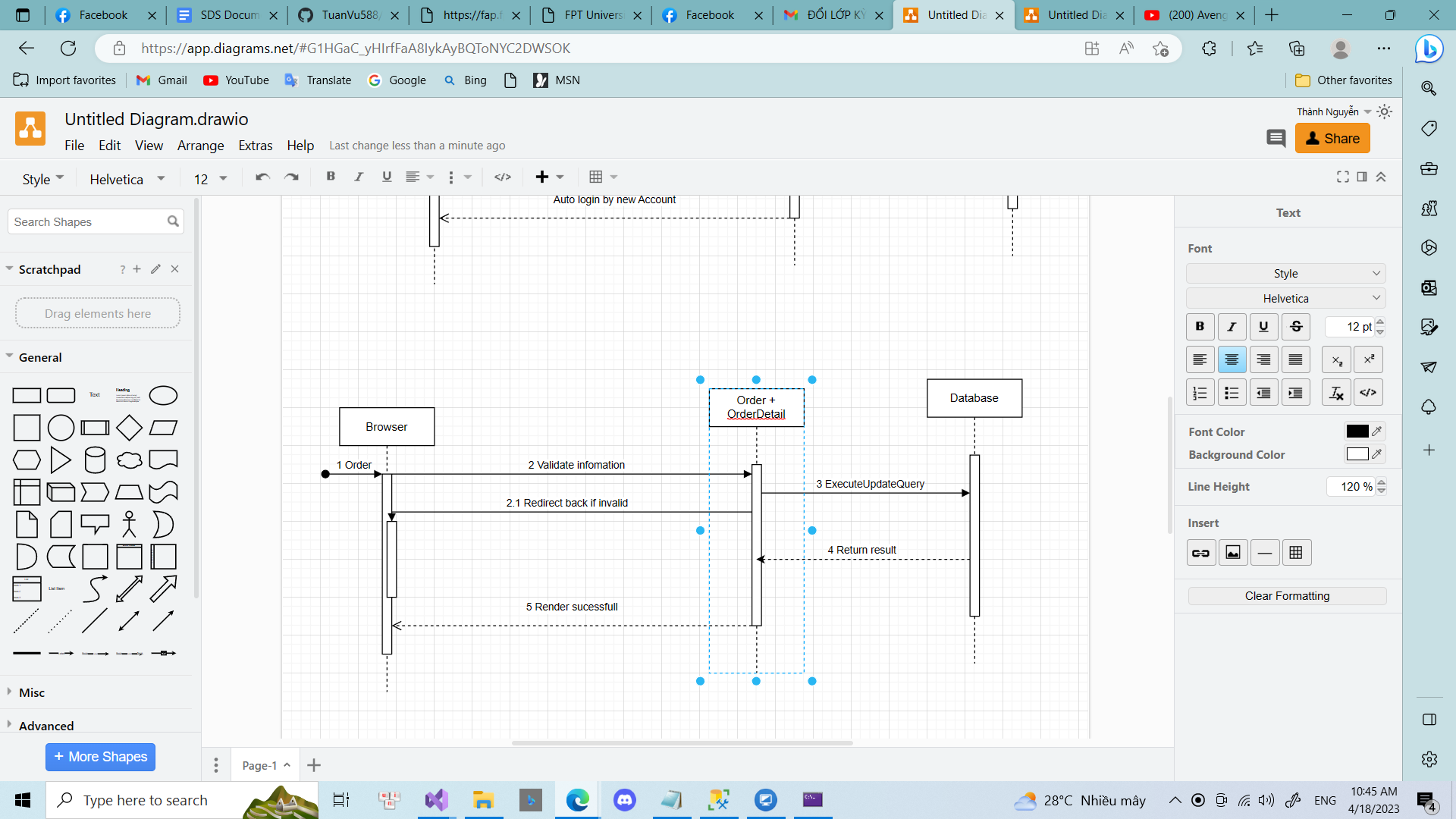


### b. Class Specifications

#### Account

| **No** | **Method** | **Description** |
| --- | --- | --- |
| *01* | *getter/setter* | *Get value of private attributes* |
|  |  |  |

### c. Sequence Diagram(s)



### d. Database Queries

\_dbContext.Customers.SingleOrDefault(cus => cus.CustomerId.Equals(Customer.CustomerId));