

## 2021-2022

### Câu 1: Tên chương trình: timtu

Viết chương trình nhập vào một chuỗi, sau đó tìm và xuất ra từ ngắn nhất cuối cùng và từ dài nhất đầu tiên trong chuỗi. Ví dụ nhập “co thanh lap trinh do” xuất “do thanh”.

Ngôn ngữ lập trình có thể dùng: C/C++, C#, Java, nhưng chỉ nộp 1 file mã nguồn “timtu.cpp” nếu dùng C++ hay “timtu.cs” nếu dùng C#

```
import java.util.*;

public class timtu{
    public static void main(String[] args){

        Scanner scanner = new Scanner(System.in);
        System.out.print("Input:");

        String input = scanner.nextLine();
        String[] words = input.split(" ");

        String shortestLastWord = words[0];
        String longestFirstWord = words[0];

        for (String word : words) {
            if (word.length() > longestFirstWord.length()) {
                longestFirstWord = word;
            }
            if (word.length() <= shortestLastWord.length()) {
                shortestLastWord = word;
            }
        }

        System.out.println("Output:" + shortestLastWord + " " +
longestFirstWord);
        scanner.close();
    }
}
```

### Câu 3: Mail SV

- Đoạn giả sử repos nằm trên máy có thể 200 cách  
Thủ công -> tạo Cau\_3 xong di chuyển repos qua Cau\_3
- Git bash thì giả sử Repos.zip đang nằm ở F: đi thì

↑ This PC > DATA (F:) Search DATA (F:)

	Name	Date modified	Type	Size
loads	3第二课_明大见.pptx	4/2/2024 8:25 PM	Microsoft PowerPoin...	8,354 KB
ments	4第四课_你去哪儿.pptx	4/6/2024 5:08 PM	Microsoft PowerPoin...	5,494 KB
res	AI Developer-DuongTrongChi.pdf	8/4/2024 1:49 PM	Microsoft Edge PDF ...	73 KB
(F:)	Backup of DangNguyen.wbk	7/30/2024 11:33 PM	Microsoft Word Back...	1,572 KB
ase	Bi_kip_git (2).docx	7/30/2024 8:37 PM	Microsoft Word Doc...	195 KB
atphan mem	bci.jpg	6/3/2024 1:57 PM	JPG File	40 KB
2020-2024	DangNguyen.docx	7/30/2024 11:34 PM	Microsoft Word Doc...	1,573 KB
ve - Personal	Docker Desktop Installer.exe	11/28/2023 3:35 PM	Application	601,709 KB
:	Dự án CNTT - CLC.pdf	4/20/2024 2:50 PM	Microsoft Edge PDF ...	93 KB
:	Dự án CNTT - ĐHTA.pdf	4/20/2024 2:50 PM	Microsoft Edge PDF ...	89 KB
:	Dự án CNTT - Tiêu chuan.pdf	4/20/2024 2:47 PM	Microsoft Edge PDF ...	115 KB
jects	food_c101_n1000_r384x384x3.h5.zip	5/26/2024 5:32 PM	Tập tin nén ZIP WinR...	350,922 KB
op	Git-2.46.0-64-bit.exe	8/4/2024 12:40 PM	Application	66,622 KB
ments	Git2022-2023.docx	8/4/2024 12:59 PM	Microsoft Word Doc...	336 KB
loads	HSK1 Sách bài tập.pdf	3/26/2024 2:24 PM	Microsoft Edge PDF ...	21,171 KB
:	HSK1 Sách giáo khoa.pdf	3/24/2024 10:27 PM	Microsoft Edge PDF ...	35,662 KB
:	idealC-2023.3.4.exe	2/21/2024 12:59 AM	Application	569,200 KB
res	in màu 1 bộ.pdf	4/2/2024 1:16 AM	Microsoft Edge PDF ...	1,458 KB
s	LicenseServer_9.1.3.15_x64.exe	9/19/2022 6:09 PM	Application	98,706 KB
:)	lich-thi-ca-nhan-knthcm-dot-t08-2024--1-(202...	7/29/2024 10:57 PM	Microsoft Edge PDF ...	4,307 KB
(F:)	nox_setup_v7.0.5.8_full_intl.exe	7/12/2023 11:18 PM	Application	592,203 KB
rk	repos.zip	4/7/2022 4:16 AM	Tập tin nén ZIP WinR...	12 KB
	TestArchitect_9.1.3.006_x64.exe	9/15/2022 12:04 PM	Application	450,717 KB
	tik.txt	6/23/2024 3:44 AM	Text Document	9 KB
	wel update v6d mci	11/30/2023 3:07 PM	Windows Installer Da	16,704 KB

MINGW64:/f/Cau\_3

```

CPT - ADMIN@LAPTOP-AOMNO4EM MINGW64 ~
$ cd F:

CPT - ADMIN@LAPTOP-AOMNO4EM MINGW64 /f
$ mkdir Cau_3

CPT - ADMIN@LAPTOP-AOMNO4EM MINGW64 /f
$ cd Cau_3

CPT - ADMIN@LAPTOP-AOMNO4EM MINGW64 /f/Cau_3
$ cp -r /F/repos.zip /F/Cau_3

CPT - ADMIN@LAPTOP-AOMNO4EM MINGW64 /f/Cau_3
$ ls
repos.zip

CPT - ADMIN@LAPTOP-AOMNO4EM MINGW64 /f/Cau_3
$ |

```

<https://docs.google.com/document/d/19q5up-1X-GcVZmupV4HR1USL0Hktc4n2hQBD77WDqeI/edit>

Tạo 1 thư mục có tên là <Cau\_3> và thực hiện mọi thao tác bên trong thư mục này. Dùng gitbash để thực hiện các yêu cầu bên dưới và chụp lại màn hình thể hiện kết quả.

**Cần nộp lại:**

- File word chứa hình các bước thực hiện và diễn giải (nếu có).
- Thư mục (nén lại) đã tạo ở trên gồm 3 thư mục con **repos**, **xuly1**, **xuly2** sau khi đã thực hiện các bước theo yêu cầu.

### **Yêu cầu thực hiện:**

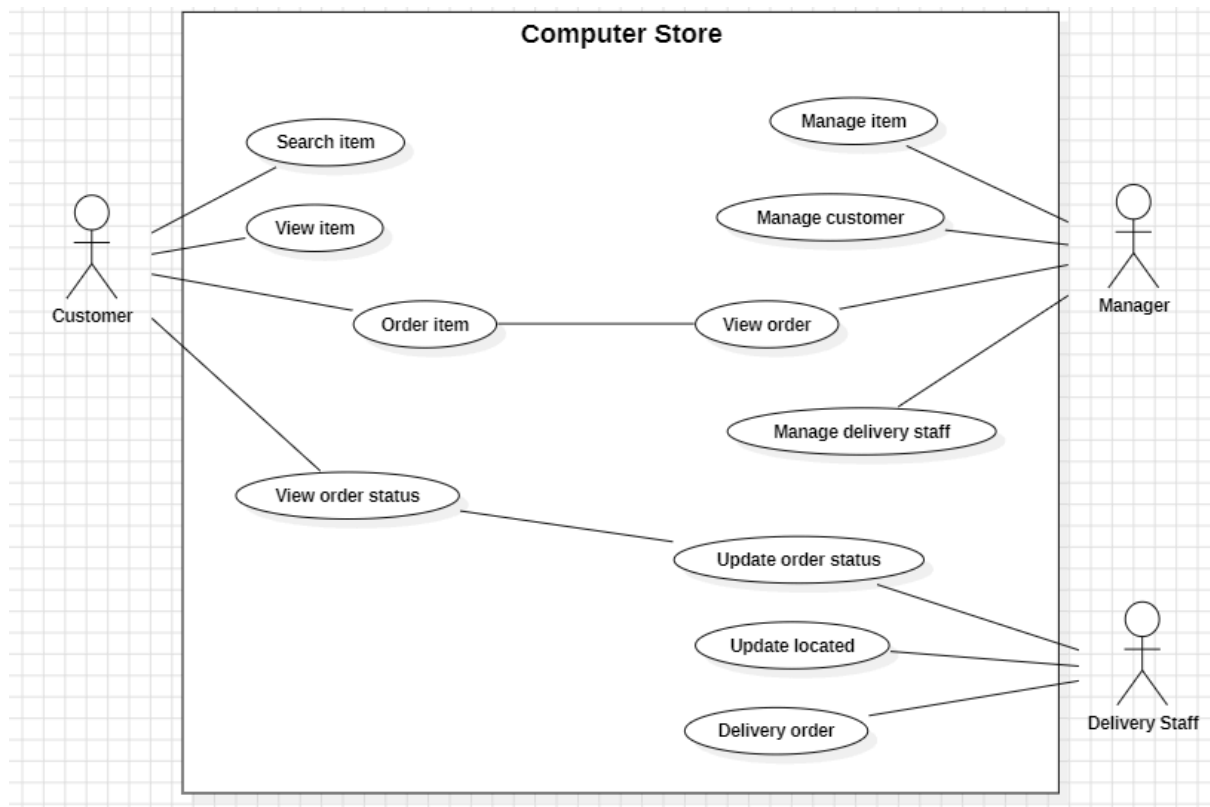
- Tải và nén file dữ liệu **repos.zip** và giải nén vào thư mục bài làm của **câu 3**. Sử dụng các lệnh của git để sao chép nội dung từ kho chứa trung tâm (**repos**) thành các thư mục làm việc và đặt tên là **xuly1** và **xuly2** (không copy/paste như bình thường). Lúc này thư mục repos sẽ đóng vai trò là kho chứa trung tâm, còn **work1** và **work2** đóng vai trò là nơi chứa source code của developers khác nhau tham gia vào chung 1 dự án.
- Trong thư mục **xuly1** tạo thêm tệp tin có tên là **Draw.java** và **readme.txt** có nội dung là chuỗi ngẫu nhiên nào đó và đồng bộ lên kho chứa trung tâm.
- Tạo tệp tin **Shape.java** trong **work1** và **work2** có nội dung như sau:

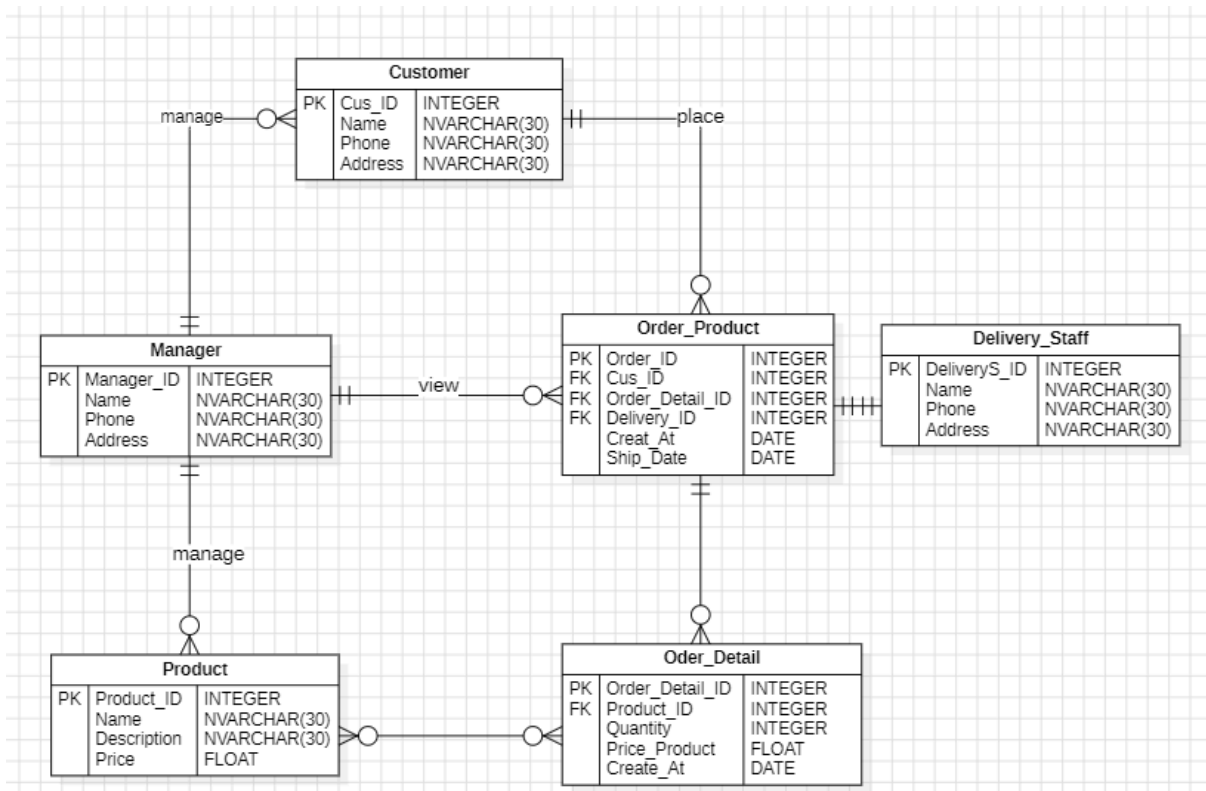
Public class Shape{ Private String name; Private double perimeter; }	Public class Shape{ Private String name; Private double area; }
<b>Work1/Shape.java</b>	<b>work2/Shape.java</b>

- Đồng bộ thay đổi trong mục **xuly1** và **xuly2** lên kho chứa trung tâm sau đó xử lý đụng độ (conflict).
- Tạo một bản phát hành mới (**snapshot**) có tên là “**v2.1.1**” để lưu trữ trạng thái source code hiện tại.
- Tiến hành đổi tên file **Draw.java** trong **xuly1** thành **DrawApp.java** và đồng bộ lên kho chứa trung tâm.
- Dựa vào bản phát hành “**v2.1.1**” đã tạo ở trên, tạo một nhánh làm việc mới có tên là “**v2.1.1-bugfix**” và chuyển sang làm việc trên nhánh này và đồng bộ lên kho chứa trung tâm.
- Loại bỏ **readme.txt** khỏi danh sách theo dõi của **xuly1** (file vẫn còn trong thư mục **xuly1**, nhưng git sẽ không track file này) và đồng bộ thay đổi lên kho chứa trung tâm.
- Tạo thêm một bản phát hành mới (snapshot) có tên là “**v2.1.2**” từ nhánh đang làm việc (**v2.1.1-bugfix**) đồng thời gộp nhánh này vào nhánh chính và đồng bộ lên kho chứa trung tâm.
- Xuất ra danh sách các nhánh, và bản phát hành của toàn bộ kho chứa.

#### Câu 4:

Cửa hàng bán máy tính cần xây dựng trên di động để phục vụ khách hàng và công việc quản lý dữ liệu. Ứng dụng cho phép khách hàng tìm kiếm, xem chi tiết và đặt mua sản phẩm, xem tình trạng giao hàng, người quản lý có thể quản lý sản phẩm, khách hàng, xem các đơn hàng và ấn định nhân viên giao hàng cho đơn hàng cụ thể, nhân viên giao hàng có thể cập nhật vị trí (ứng dụng có thể tự động thực hiện khi đang chạy ngầm và nhân viên giao hàng đã đăng nhập), cập nhật trạng thái đơn hàng khi nhận giao và giao xong.





```
-- create
CREATE TABLE Customer (
  Cus_ID INT PRIMARY KEY,
  Name NVARCHAR(30) NOT NULL,
  Phone NVARCHAR(30),
  Address NVARCHAR(30),
);

CREATE TABLE Manager (
  Manager_ID INT PRIMARY KEY,
  Name NVARCHAR(30) NOT NULL,
  Phone NVARCHAR(30),
  Address NVARCHAR(30),
);

CREATE TABLE Product (
  Product_ID INT PRIMARY KEY,
  Name NVARCHAR(30) NOT NULL,
  Description NVARCHAR(30),
  Price FLOAT,
```

);

```
CREATE TABLE Order_Detail (  
    Order_Detail_ID INT PRIMARY KEY,  
    Product_ID INT,  
    Quantity INT NOT NULL,  
    Price_Product FLOAT,  
    Creat_At DATE,  
  
    FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID),  
);
```

```
CREATE TABLE Delivery_Staff (  
    DeliveryS_ID INT PRIMARY KEY,  
    Name NVARCHAR(30) NOT NULL,  
    Phone NVARCHAR(30),  
    Address NVARCHAR(30),  
)
```

```
CREATE TABLE Order_Product (  
    Order_ID INT PRIMARY KEY,  
    Cus_ID INT,  
    Order_Detail_ID INT,  
    DeliveryS_ID INT,  
    Creat_At DATE,  
    Ship_Date DATE,  
  
    FOREIGN KEY (Cus_ID) REFERENCES Customer(Cus_ID),  
    FOREIGN KEY (Order_Detail_ID) REFERENCES  
Order_Detail(Order_Detail_ID),  
    FOREIGN KEY (DeliveryS_ID) REFERENCES  
Delivery_Staff(DeliveryS_ID),  
);
```

-- insert

```
INSERT INTO Customer(Cus_ID,Name,Phone,Address) VALUES  
(001, 'Nguyen','100' ,'District 1');
```

```
INSERT INTO Manager(Manager_ID,Name,Phone,Address) VALUES
```

```
(002, 'Dang','200' ,'District 2');
```

```
INSERT INTO Delivery_Staff(DeliveryS_ID,Name,Phone,Address)
VALUES (003, 'Pham','300','District 3');
```

```
INSERT INTO Product(Product_ID,Name,Description,Price)
VALUES (004, 'Apple','Red',30.00);
```

```
INSERT INTO Order_Detail(Order_Detail_ID, Product_ID, Quantity,
Price_Product, Creat_At)
VALUES (005, 004, 5, 30.50, '01-08-2024');
```

```
INSERT INTO Order_Product(Order_ID, Cus_ID, Order_Detail_ID,
DeliveryS_ID, Creat_At, Ship_Date)
VALUES (006, 001, 005, 003, '01-08-2024', '03-08-2024');
```

```
-- fetch
```

```
SELECT * FROM Customer
SELECT * FROM Manager
SELECT * FROM Delivery_Staff
SELECT * FROM Product
SELECT * FROM Order_Product
SELECT * FROM Order_Detail
```

Output:

Cus_ID	Name	Phone	Address		
1	Nguyen	100	District 1		
Manager_ID	Name	Phone	Address		
2	Dang	200	District 2		
DeliveryS_ID	Name	Phone	Address		
3	Pham	300	District 3		
Product_ID	Name	Description	Price		
4	Apple	Red	30.0		
Order_ID	Cus_ID	Order_Detail_ID	DeliveryS_ID	Creat_At	Ship_Date
6	1	5	3	2024-01-08	2024-03-08
Order_Detail_ID	Product_ID	Quantity	Price_Product	Creat_At	
5	4	5	30.5	2024-01-08	

### - Vẽ Use Case

- Design CSDL cho hệ thống dùng mô hình ER, các thuộc tính của entity sinh viên tự đề ra và phải đảm bảo mỗi thực thể có ít nhất 3 thuộc tính.
- Phát sinh ra file csdl từ mô hình ER trên (sinh viên tự viết file SQL này nhưng phải đảm bảo thực thi được và ra đúng CSDL yêu cầu.)
- **Lưu ý:** nộp gồm file word **Thietke.docx** chứa hình sơ đồ **Use Case**, **ERD** và file csdl.

## 2022-2023

### Câu 1: Tên chương trình: timso

Viết chương trình nhập vào một dãy gồm n số nguyên, xuất ra các số là số Fibonacci trong dãy vừa nhập. Ví dụ nhập dãy: 1 3 2 9 thì xuất ra các số 1 3 2

Ngôn ngữ lập trình có thể dùng: C/C++, C#, Java, nhưng chỉ nộp 1 file mã nguồn “timso.cpp” nếu dùng C++ hay “timso.cs” nếu dùng C#

```
import java.util.*;

public class timso {

    public static boolean isFibonacci(int number) {
        if (number == 0 || number == 1) {
            return true;
        }
        int a = 0, b = 1;
        while (b < number) {
            int temp = b;
            b = a + b;
            a = temp;
        }
        return b == number;
    }
}
```



```

public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);

    System.out.print("Enter the number of elements: ");
    int n = scanner.nextInt();
    int[] array = new int[n];

    System.out.println("Enter each elements:");
    for (int i = 0; i < n; i++) {
        array[i] = scanner.nextInt();
    }

    ArrayList<Integer> fibonacciNumbers = new ArrayList<>();
    for (int number : array) {
        if (isFibonacci(number)) {
            fibonacciNumbers.add(number);
        }
    }

    System.out.println("Fibonacci numbers in the array:");
    for (int number : fibonacciNumbers) {
        System.out.print(number + " ");
    }
}

```

### Câu 3:

Tạo 1 thư mục có tên là <Cau\_3> và thực hiện mọi thao tác bên trong thư mục này. Dùng gitbash để thực hiện các yêu cầu bên dưới và chụp lại màn hình thể hiện kết quả.

#### Cần nộp lại:

- File word chứa hình các bước thực hiện và diễn giải (nếu có).
- Thư mục (nén lại) đã tạo ở trên gồm 3 thư mục con **repos**, **work1**, **work2** sau khi đã thực hiện các bước theo yêu cầu.

#### Yêu cầu thực hiện:

- Tải và nén file dữ liệu **repos.zip** và giải nén vào thư mục bài làm của **câu 3**. Sử dụng các lệnh của git để sao chép nội dung từ kho chứa trung tâm (**repos**) thành thư mục làm việc và đặt tên là **work1** và **work2** (không copy/paste như bình thường). Lúc này thư mục **repos** sẽ đóng vai trò là kho chứa trung tâm, còn **work1** và **work2** đóng vai trò là nơi chứa source code của developers khác nhau tham gia vào chung 1 dự án.
- Trong thư mục **work1** tạo thêm tệp tin có tên là **Draw.java** và **readme.txt** có nội dung là chuỗi ngẫu nhiên nào đó và đồng bộ lên kho chứa trung tâm
- Tạo tệp tin **Shape.java** trong **work1** và **work2** có nội dung như sau:

Public class Shape{ Private String name; Private double perimeter; }	Public class Shape{ Private String name; Private double area; }
<b>Work1/Shape.java</b>	<b>work2/Shape.java</b>

Đồng bộ thay đổi trong mục **work1** và **work2** lên kho chứa trung tâm sau đó xử lý đụng độ (conflict).

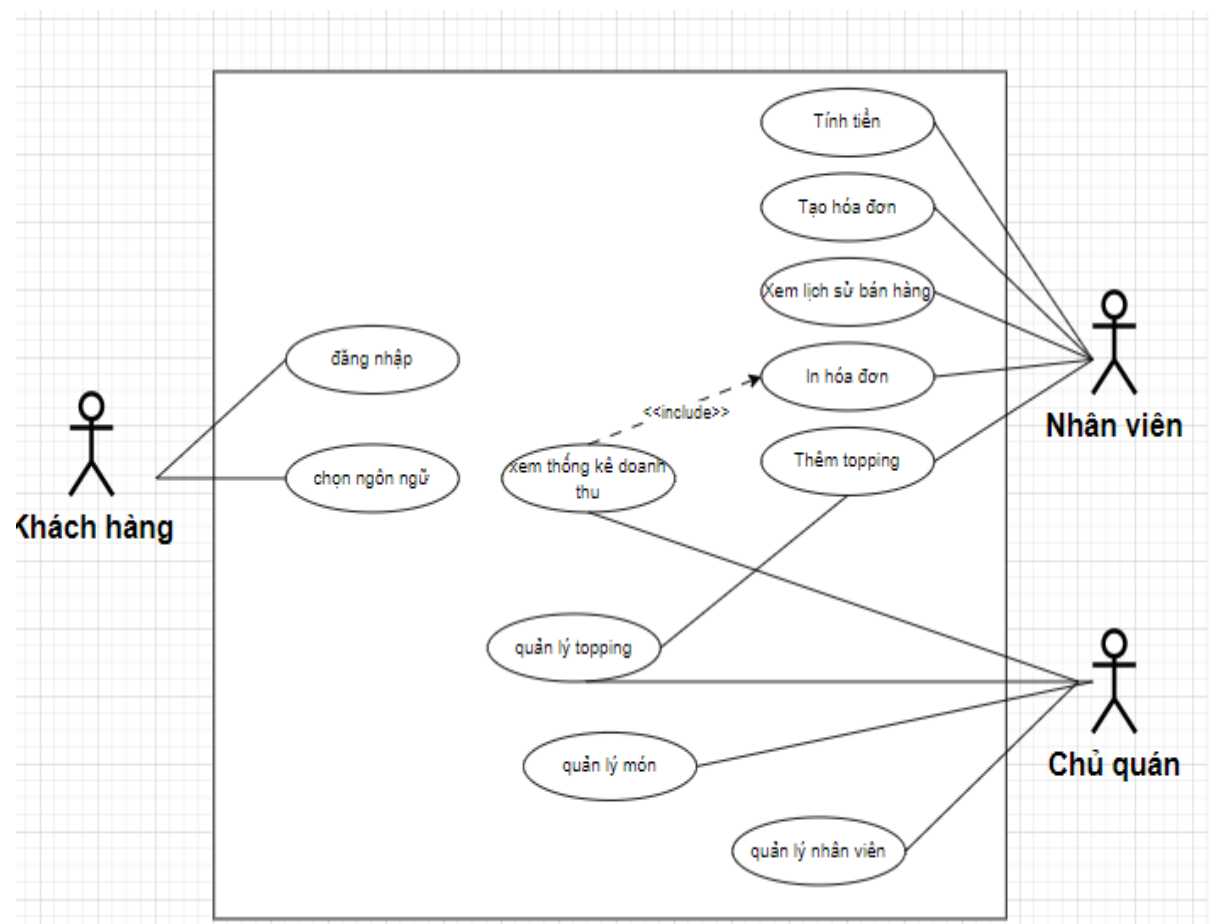
- Tạo một bản phát hành mới (**snapshot**) có tên là “**v2.1.1**” để lưu trữ trạng thái source code hiện tại.
- Tiến hành đổi tên file **Draw.java** trong **work1** thành **DrawApp.java** và đồng bộ lên kho chứa trung tâm.
- Dựa vào bản phát hành “**v2.1.1**” đã tạo ở trên, tạo một nhánh làm việc mới có tên là “**v2.1.1-bugfix**” và chuyển sang làm việc trên nhánh này và đồng bộ lên kho chứa trung tâm.
- Loại bỏ **readme.txt** khỏi danh sách theo dõi của **work1** (file vẫn còn trong thư mục **work1**, nhưng git sẽ không track file này) và đồng bộ thay đổi lên kho chứa trung tâm.
- Tạo thêm một bản phát hành mới (**snapshot**) có tên là “**v2.1.2**” từ nhánh đang làm việc (**v2.1.1-bugfix**) đồng thời gộp nhánh này vào nhánh chính và đồng bộ lên kho chứa trung tâm.
- Xuất ra danh sách các nhánh, và bản phát hành của toàn bộ kho chứa.

#### Câu 4:

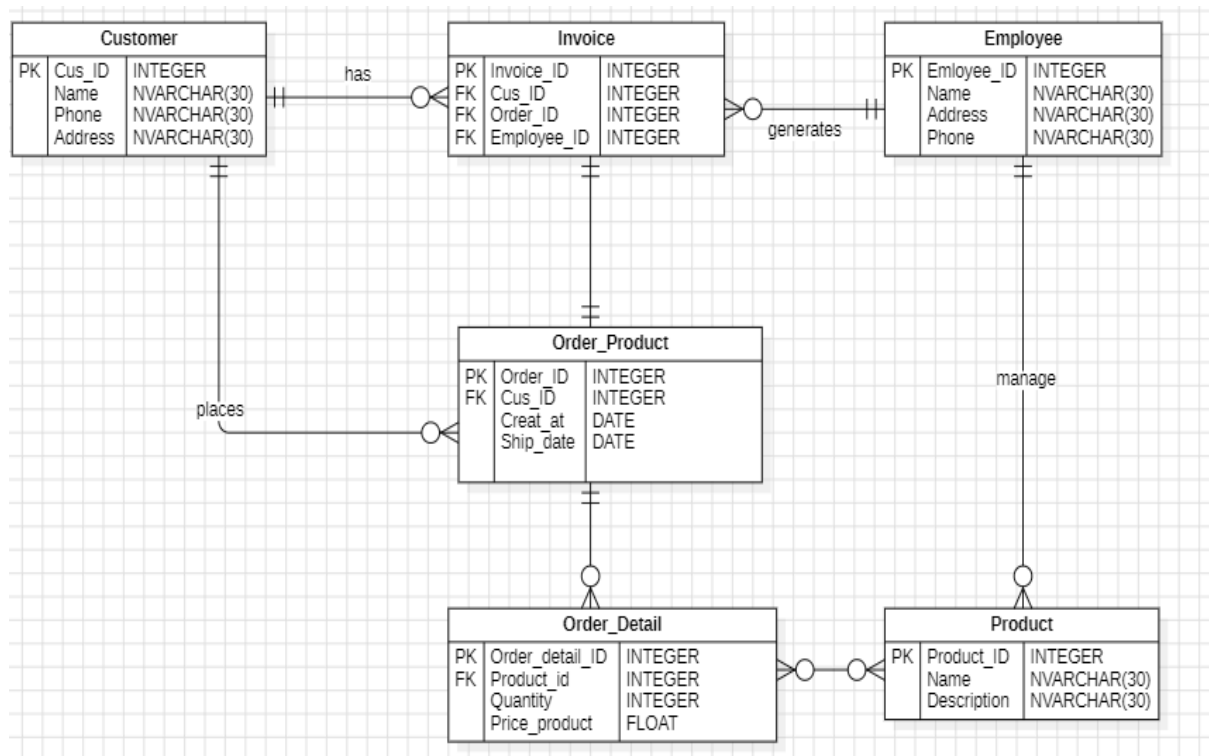
Người ta cần xây dựng ứng dụng quản lý trà sữa:

**Nhân viên** có thể **tạo hóa đơn**, khi khách đến mua, **bổ sung các loại món thêm** (trân châu, thạch dừa,..) Khi khách yêu cầu, **tính tiền** và **in hóa đơn**, **xem lại lịch sử bán hàng** của mình, với **chủ quán** có thể **quản lý nhân viên**, **quản lý món**, **topping**, **xem thống kê doanh thu theo nhân viên**, theo thời gian. Ứng dụng cho phép **người dùng** **chọn ngôn ngữ** “Tiếng Anh” hoặc “Tiếng Việt”. Người dùng cần **đăng nhập** trước khi sử dụng các chức năng của hệ thống.

- Vẽ Use Case



(Bỏ cái <<include>> chỉ nối qua bth thôi)



```

-- create table
CREATE TABLE Customer (
    Cus_ID INT PRIMARY KEY,
    Name NVARCHAR(30) NOT NULL,
    Phone NVARCHAR(30),
    Address NVARCHAR(30)
);

CREATE TABLE Employee (
    Employee_ID INT PRIMARY KEY,
    Name NVARCHAR(30) NOT NULL,
    Phone NVARCHAR(30),
    Address NVARCHAR(30),
);

CREATE TABLE Product (
    Product_ID INT PRIMARY KEY,
    Name NVARCHAR(30) NOT NULL,
    Description NVARCHAR(30)
);
  
```

```

CREATE TABLE Order_Detail (
    Order_detail_ID INT PRIMARY KEY,
    Product_ID INT,
    Quantity INT NOT NULL,
    Price_product FLOAT,

    FOREIGN KEY (Product_ID) REFERENCES Product(Product_ID)
);

CREATE TABLE Order_Product (
    Order_ID INT PRIMARY KEY,
    Cus_ID INT,
    Status DATE,
    Creat_At DATE,
    Shipe_Date DATE,

    FOREIGN KEY (Cus_ID) REFERENCES Customer(Cus_ID)
);

CREATE TABLE Invoice (
    Invoice_ID INT PRIMARY KEY,
    Cus_ID INT,
    Employee_ID INT,
    Order_ID INT,

    FOREIGN KEY (Cus_ID) REFERENCES Customer(Cus_ID),
    FOREIGN KEY (Employee_ID) REFERENCES
Employee(Employee_ID),
    FOREIGN KEY (Order_ID) REFERENCES Order_Product(Order_ID)
);

-- insert
INSERT INTO Customer(Cus_ID,Name,Phone,Address) VALUES
(001, 'Nguyen','100' ,'District 1');

INSERT INTO Employee(Employee_ID,Name,Phone,Address) VALUES
(001, 'Dang','200' ,'District 2');

INSERT INTO Product(Product_ID,Name,Description) VALUES
(101, 'Meo','Meo Meo Meo');

INSERT INTO Order_Detail(Order_detail_ID, Product_ID, Quantity,

```

```
Price_product) VALUES
(1, 101, 10, 100.0);
```

```
INSERT INTO Order_Product(Order_ID, Cus_ID, Status, Creat_At,
Shipe_Date) VALUES
(1001, 001, '01-08-2024', '02-08-2024', '03-08-2024');
```

```
INSERT INTO Invoice(Invoice_ID, Cus_ID, Employee_ID, Order_ID)
VALUES
(12345, 001, 001, 1001);
```

```
-- fetch
```

```
SELECT * FROM Customer
SELECT * FROM Employee
SELECT * FROM Product
SELECT * FROM Order_Detail
SELECT * FROM Order_Product
SELECT * FROM Invoice
```

Output:

Cus_ID	Name	Phone	Address
1	Nguyen	100	District 1

Employee_ID	Name	Phone	Address
1	Dang	200	District 2

Product_ID	Name	Description
101	Meo	Meo Meo Meo

Order_detail_ID	Product_ID	Quantity	Price_product
1	101	10	100.0

Order_ID	Cus_ID	Status	Creat_At	Shipe_Date
1001	1	2024-01-08	2024-02-08	2024-03-08

Invoice_ID	Cus_ID	Employee_ID	Order_ID
12345	1	1	1001

- Design CSDL cho hệ thống dùng mô hình ER, các thuộc tính của entity sinh viên tự đề ra và phải đảm bảo mỗi thực thể có ít nhất 3 thuộc tính.
- Phát sinh ra file csdl từ mô hình ER trên (sinh viên tự viết file SQL này nhưng phải đảm bảo thực thi được và ra đúng CSDL yêu cầu.)
- **Lưu ý:** nộp gồm file word **Thietke.docx** chứa hình sơ đồ **Use Case**, **ERD** và file csdl.