

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



## HỆ CƠ SỞ DỮ LIỆU (CO2013)

---

### BÁO CÁO BÀI TẬP LỚN 2

LỚP TN01 - NHÓM 666

---

#### ĐỀ TÀI:

### HỆ THỐNG QUẢN LÝ PHÒNG NET

Giảng viên hướng dẫn:	TS. Nguyễn Thị Ái Thảo
Sinh viên thực hiện:	Phạm Thế Hiếu - 2111213
	Đỗ Tín Nghĩa - 2111837
	Huỳnh Thái Học - 2113443
	Tạ Đình Tiến - 2110583

# Mục lục

<b>1</b>	<b>Tổng quan hệ thống</b>	<b>4</b>
1.1	Mô tả nghiệp vụ . . . . .	4
1.2	Thiết kế EERD và Schema . . . . .	5
1.2.1	EERD . . . . .	5
1.2.2	Thiết kế schema . . . . .	6
<b>2</b>	<b>Thiết kế hệ thống</b>	<b>7</b>
2.1	Kiến trúc hệ thống . . . . .	7
2.2	Kỹ thuật . . . . .	8
2.2.1	Hệ quản trị cơ sở dữ liệu (DBMS) . . . . .	8
2.2.2	Session Management Service . . . . .	8
2.2.3	Reception service . . . . .	9
2.2.4	Admin service . . . . .	11
<b>3</b>	<b>Đặc tả cơ sở dữ liệu</b>	<b>13</b>
3.1	Cấu trúc file . . . . .	13
3.2	Bảng dữ liệu (table) . . . . .	13
3.3	Các trigger . . . . .	18
3.3.1	Trigger thay đổi số dư tài khoản hội viên ( update_account_balance) . . . . .	18
3.3.2	Trigger đăng xuất tài khoản hội viên (sign_out) . . . . .	19
3.3.3	Trigger kiểm tra thêm sản phẩm được áp dụng khuyến mãi (add_product_for_discount) . . . . .	20
3.4	Hàm và thủ tục . . . . .	21
3.4.1	Các thủ tục (procedure) . . . . .	21
3.4.1.a	Đăng nhập (signIn) . . . . .	21
3.4.1.b	Thêm sản phẩm vào hóa đơn (addProductToBill) . . . . .	22
3.4.1.c	Thông kê khuyến mãi (thongKeKhuyenMai) . . . . .	24
3.4.2	Các hàm (function) . . . . .	25
3.4.2.a	Kiểm tra mã máy tính (checkPCID) . . . . .	25
3.4.2.b	Lấy giá máy tính (getPCPrice) . . . . .	25
3.5	Dữ liệu mẫu . . . . .	26
<b>4</b>	<b>Hiện thực và các tính năng</b>	<b>27</b>
4.1	Session Management Service (dành cho hội viên) . . . . .	27

4.1.1	Dăng nhập . . . . .	27
4.1.1.a	Mô tả . . . . .	27
4.1.1.b	Thủ tục . . . . .	27
4.1.2	Thay đổi thông tin . . . . .	27
4.1.2.a	Mô tả . . . . .	27
4.1.2.b	Thủ tục . . . . .	29
4.1.3	Dăng xuất . . . . .	29
4.1.3.a	Mô tả . . . . .	29
4.1.3.b	Thủ tục . . . . .	29
4.2	Reception Service . . . . .	29
4.2.1	Xem danh sách hội viên . . . . .	29
4.2.1.a	Mô tả . . . . .	29
4.2.1.b	Thủ tục . . . . .	30
4.2.2	Thêm hội viên . . . . .	30
4.2.2.a	Mô tả . . . . .	30
4.2.2.b	Thủ tục . . . . .	31
4.2.3	Thêm hóa đơn . . . . .	33
4.2.3.a	Mô tả: . . . . .	33
4.2.3.b	Thủ tục . . . . .	33
4.3	Admin Service . . . . .	37
4.3.1	Giao diện quản lý máy tính . . . . .	37
4.3.1.a	Mô tả . . . . .	37
4.3.1.b	Thủ tục . . . . .	37
4.3.2	Thông tin máy tính . . . . .	37
4.3.2.a	Mô tả . . . . .	37
4.3.2.b	Thủ tục . . . . .	38
4.3.3	Chỉnh sửa thông số máy tính . . . . .	38
4.3.3.a	Mô tả . . . . .	38
4.3.3.b	Thủ tục . . . . .	39
4.3.4	Thêm máy tính . . . . .	39
4.3.4.a	Mô tả . . . . .	39
4.3.4.b	Thủ tục . . . . .	40
4.3.5	Thống kê sản phẩm khuyến mãi . . . . .	40
4.3.5.a	Mô tả . . . . .	40



4.3.5.b	Thủ tục . . . . .	41
---------	-------------------	----

# 1 Tổng quan hệ thống

## 1.1 Mô tả nghiệp vụ

Xây dựng hệ thống quản lý phòng net giúp quản lý net một cách dễ dàng, thân thiện.

Một chủ cửa hàng phòng net cần quản lý phòng net của mình. Một phòng net bao gồm 200 máy tính. Mỗi máy tính cần được lưu thông tin về, mã máy tính (mỗi máy tính có một mã máy tính riêng), thuộc cấu hình nào (bao gồm mã cấu hình, loại CPU, loại card đồ họa, ram, kích thước màn hình, tần số quét màn hình, giá, thông tin RAM), ngày mua máy tính, hãng máy tính.

Phòng net này phân làm 3 cấp độ máy và giá riêng biệt. Cấp độ máy bình dân, mức giá 6 nghìn đồng cho 1 giờ chơi với cấu hình máy là loại cpu core i3, ram 16gb, màn hình tần số quét 120 Hz với kích thước màn hình 24 inch. Cấp độ máy trung cấp với mức giá 15 nghìn đồng cho 1 giờ chơi với cấu hình máy là cpu core i5, ram 16gb, màn hình tần số quét 144 Hz với kích thước màn hình 27 inch. Cấp độ cao cấp với mức giá 30 nghìn đồng cho 1 giờ chơi với cấu hình máy core i7, 32 gb ram, tần số màn hình 240 Hz, màn hình cong kích thước 33 inch.

Phòng net được thiết kế với 3 khu vực riêng biệt bao gồm, khu vực hút thuốc, bao gồm các máy bình dân và trung cấp. Khu vực máy lạnh, có đầy đủ 3 loại máy và khu vực sự kiện, chỉ gồm các máy mức độ cao cấp. Mỗi khu vực khác nhau sẽ có mức phụ thu thêm khác nhau.

Vì phòng net có quy mô lớn, phòng net cần quản lý nhân viên. Mỗi nhân viên bao gồm một mã số nhân viên riêng biệt, ngày tháng năm sinh, họ và tên, giới tính, hợp đồng lao động (gồm ngày kí kết hợp đồng lao động, ngày hết hạn hợp đồng lao động). Nhân viên bao gồm nhân viên kĩ thuật, nhân viên lễ tân và nhân viên tạp vụ. Đối với nhân viên kĩ thuật, cần lưu trữ chuyên môn của nhân viên đó (mạng, thiết bị, ứng dụng, điện) và cách thức hoạt động (trực tiếp tại quán hoặc từ xa). Mỗi nhân viên kĩ thuật có thể có nhiều chuyên môn. Đối với nhân viên lễ tân, cần lưu trữ lịch trực lễ tân, ngoài ra mỗi lễ tân còn được cấp tài khoản (gồm tài khoản và mật khẩu) để thực hiện các thao tác quản lý. Đối với nhân viên tạp vụ gồm các công việc là bảo vệ, đầu bếp và lao công.

Quản chỉ phục vụ cho hội viên (khách vắng lai muốn được sử dụng dịch vụ phải tạo tài khoản hội viên). Hội viên lưu trữ gồm tài khoản hội viên, mật khẩu hội viên, tên hội viên, số điện thoại hội viên, email hội viên, level(khách thường hoặc khách VIP) và số dư tài khoản hội viên. Mỗi tài khoản hội viên là khác nhau và số điện thoại ứng với mỗi tài khoản hội viên khác nhau là khác nhau. Hội viên sẽ nạp tiền vào tài khoản hội viên. Khi sử dụng dịch vụ tại máy, hội viên sẽ đăng nhập tài khoản hội viên để sử dụng, vì vậy cần có cơ chế lưu trữ thông tin máy nào đang được hội viên nào sử dụng(Session gồm giờ bắt đầu, mã session, giờ kết thúc).

Ngoài phục vụ dịch vụ internet, phòng net còn phục vụ dịch vụ đồ ăn thức uống. Các loại đồ ăn thức uống cần lưu trữ, mã số sản phẩm (mỗi sản phẩm có 1 mã số riêng biệt), tên sản phẩm và giá niêm yết. Các sản phẩm này có thể mua thông qua tài khoản hội viên (không trừ tiền trực tiếp trong tài khoản, trả tiền sau khi nhận sản phẩm) hoặc đặt mua ở quầy lễ tân(có thể trả tiền tại quầy lễ tân). Lễ tân sẽ in hóa đơn cho dịch vụ này và gửi hóa đơn cho hội viên.

Để có tiền trong tài khoản hội viên, hội viên thực hiện nạp tiền vào tài khoản, hội viên thực hiện tại quầy lễ tân, và sẽ được xuất hóa đơn cho lần nạp đó.

Phòng net sẽ có các đợt khuyến mãi khác nhau nhằm tăng doanh thu. Chương trình khuyến mãi được chia làm hai loại: khuyến mãi theo sản phẩm nhất định và khuyến mãi trên tổng hóa đơn. Thông tin chung của chương trình khuyến mãi gồm: mã chương trình (duy nhất), tên chương trình, mô tả, điều kiện áp dụng, ngày bắt đầu, ngày kết thúc, loại. Mỗi đợt khuyến mãi theo sản phẩm sẽ có loại và số lượng sản phẩm khác nhau được áp dụng theo từng đợt. Với chương trình khuyến mãi trên tổng hóa đơn, điều kiện sẽ là mức tối thiểu của hóa đơn. Có thể có nhiều chương trình khuyến mãi với nhau cùng thỏa điều

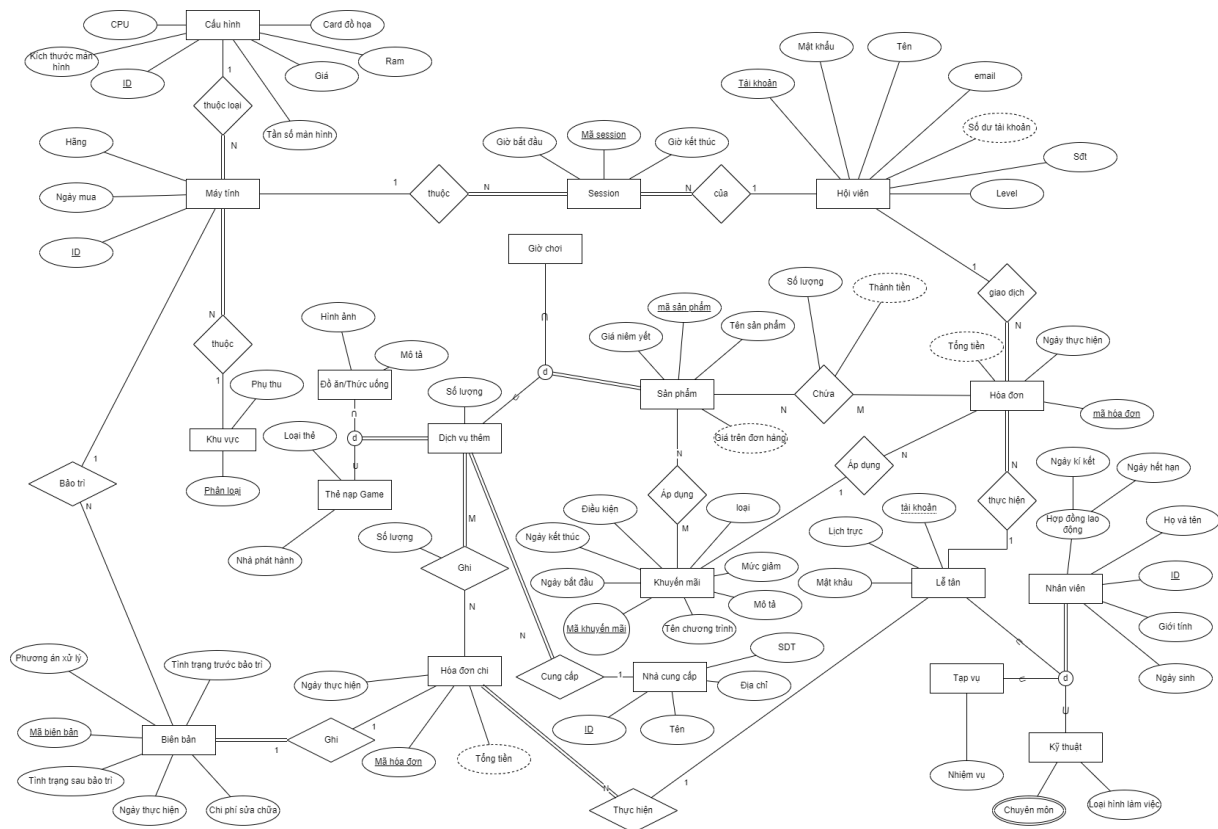
kiện, chỉ áp dụng khuyến mãi có mức giảm cao nhất.

Ngoài ra, phòng net còn yêu cầu quản lý doanh thu theo từng ngày. Cần lưu trữ nhân viên trực, thu nhập từ hội viên nạp tiền, thu nhập từ dịch vụ đồ ăn thức uống. Thống kê giờ chơi theo từng ngày lưu trữ, tổng số giờ chơi ở từng khu vực theo từng loại cấp độ máy.

Cửa hàng sẽ cần lưu trữ lại tất cả các khoản thu chi đã thực hiện. Các thông tin bao gồm ID của khoản (duy nhất), tên khoản, loại khoản (thu hay chi), ngày giờ thực hiện, số tiền. Các hoá đơn được coi là một khoản thu mới và được tự động ghi lại. Ngoài ra, chỉ có nhân viên lễ tân được phép tạo các khoản thu chi mới. Đối với các khoản chi liên quan đến nhập sản phẩm, mỗi loại sản phẩm chỉ có thể nhập từ một nhà cung cấp duy nhất, mỗi nhà cung cấp sản phẩm có một mã số riêng biệt, tên, địa chỉ và số điện thoại. Mỗi lần có bảo trì máy móc, sẽ cần ghi một biên bản cần ghi chi phí sửa chữa, tình trạng trước bảo trì, tình trạng sau bảo trì, ngày thực hiện, mã biên bản. Chi phí này sẽ được xuất hóa đơn chi.

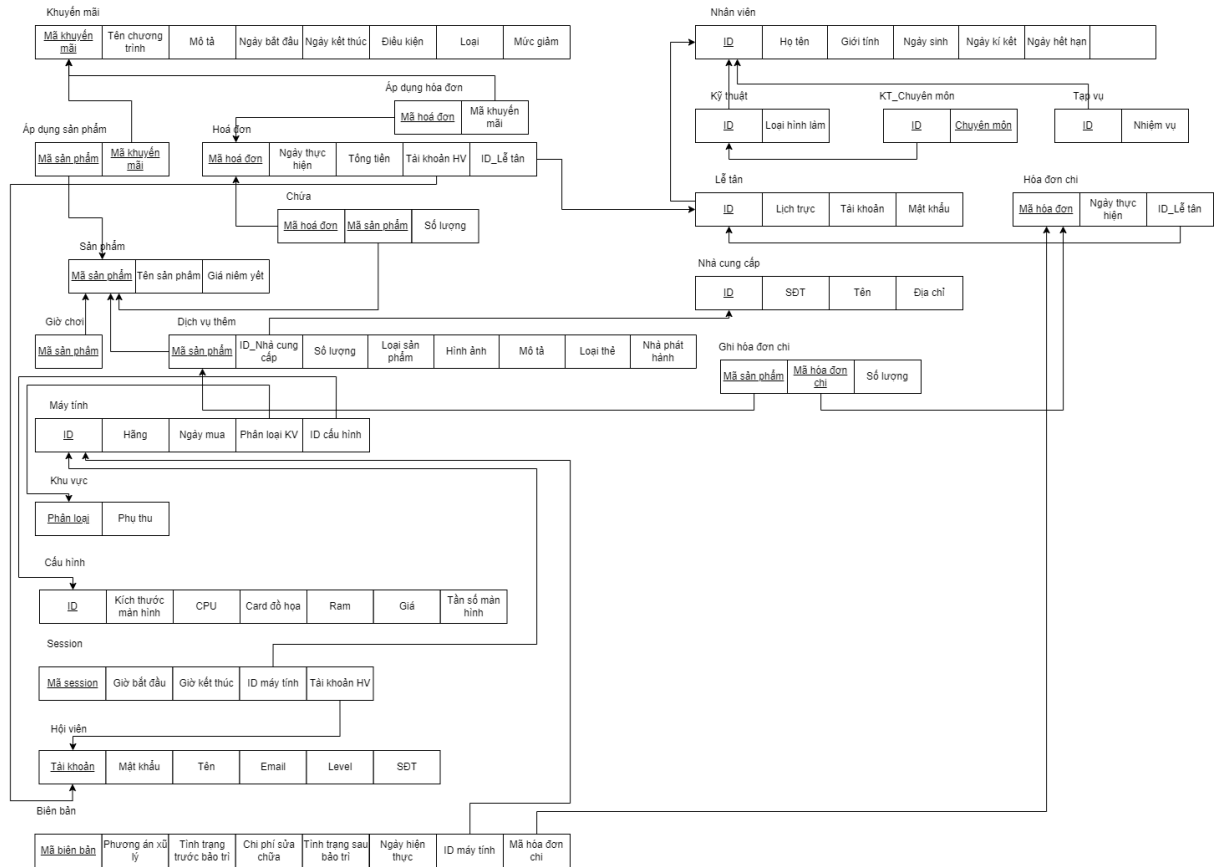
## 1.2 Thiết kế EERD và Schema

### 1.2.1 EERD



Hình 1: Biểu đồ mối quan hệ

## 1.2.2 Thiết kế schema



Hình 2: schema

## 2 Thiết kế hệ thống

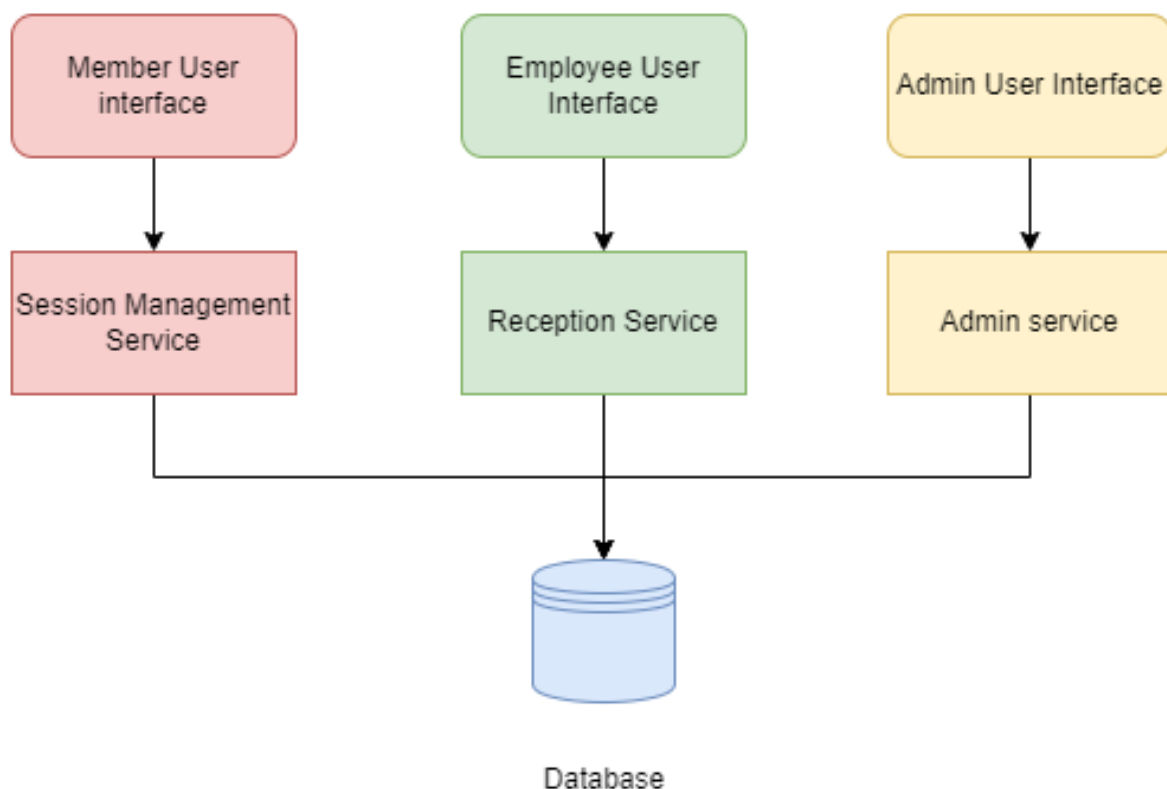
### 2.1 Kiến trúc hệ thống

Dựa theo nghiệp vụ, hệ thống được chia làm 3 nhóm người dùng chính bao gồm:

- Hội viên: Đây là người dùng dịch vụ, sử dụng dịch vụ máy tính ở phòng net, hội viên cần một ứng dụng để có thể đăng nhập máy tính và quản lý thời gian sử dụng máy.
- Nhân viên phòng net: Cần ứng dụng để dễ dàng phục vụ nhu cầu dịch vụ của hội viên, có thể thực hiện hóa đơn cơ bản với các giao dịch của Hội Viên.
- Chủ phòng net: Cần ứng dụng để dễ dàng tùy chỉnh phòng net, xem các thông kê.

Dựa vào đặc điểm đó, nhóm xây dựng hệ thống theo mô hình kiến trúc service-base với hệ thống được chia làm 3 service riêng biệt chính hướng tới từng đối tượng người dùng và sử dụng chung một cơ sở dữ liệu tập trung bao gồm.

- Session management service: Quản lý các phiên đăng nhập của hội viên với máy tính, hỗ trợ quản lý tài khoản người dùng.
- Admin service: Quản lý chung phòng net, hỗ trợ chủ phòng net quản lý phòng net.
- Reception service: Quản lý các giao dịch, sản phẩm, hỗ trợ lễ tân thực hiện điều hành phòng net.



Hình 3: Kiến trúc hệ thống

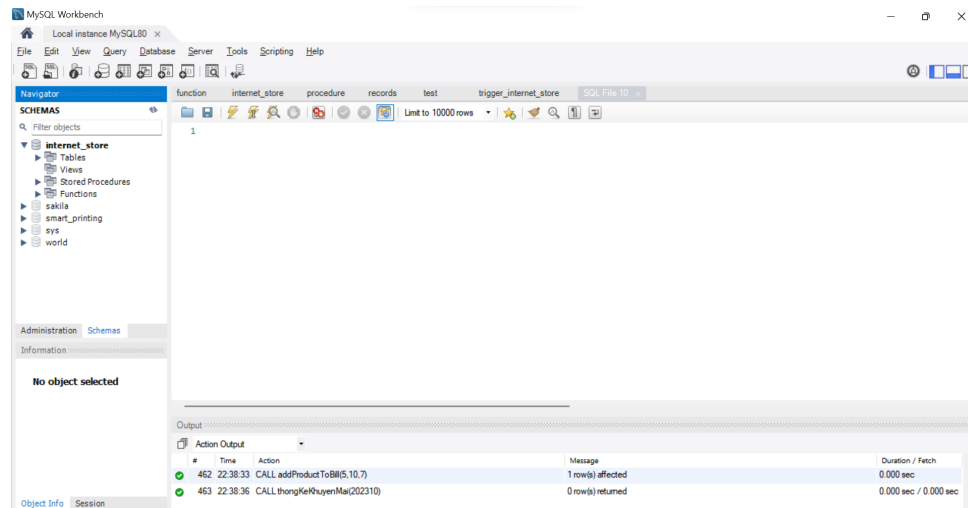
Với kiến trúc trên, nhóm xây dựng 3 ứng dụng riêng biệt dựa trên các service nhỏ, sử dụng các công nghệ khác nhau.



## 2.2 Kỹ thuật

### 2.2.1 Hệ quản trị cơ sở dữ liệu (DBMS)

Trong hệ thống quản lý phòng net, nhóm sử dụng hệ quản trị cơ sở dữ liệu MySQL và tương tác, tạo cơ sở dữ liệu thông qua môi trường MySQL WorkBench.



Hình 4: Môi trường MySQL workbench

MySQL là một hệ quản trị cơ sở dữ liệu có cấu trúc (SQL database), miễn phí, dễ sử dụng và có cộng đồng phát triển lớn. Vì thế, nhóm chọn để phát triển hệ thống trong bài tập lớn này.

Để các ứng dụng có thể giao tiếp với MySQL, MySQL workbench tạo ra một MySQL server tại localhost với port 3306, các ứng dụng giao tiếp với cơ sở dữ liệu sẽ tạo kết nối thông qua port 3306 để gọi truy vấn.

### 2.2.2 Session Management Service

Vì service này, nhằm hỗ trợ quản lý các phiên đăng nhập trên máy tính của hội viên và thay đổi giá trị số dư tài khoản của hội viên trên cơ sở dữ liệu.

```
class Computer:
> def __init__(self,pcid): ...
> def signIn(self,account,password): ...
> def signOut(self): ...
> def getPrice(self): ...
You, yesterday | 1 author (You)
class Account:
> def __init__(self,account,password): ...
> def getInfo(self): ...
> def getBalance(self): ...
> def checkBalance(self): ...
> def refetchData(self): ...
> def changeInfo(self,new_password=None,new_name=None, new_email=None,new_sdt=None,new_level=None): ...
You, yesterday | 1 author (You)
class App:
> def __init__(self,pcid): ...
> def signIn(self): ...
> def createSignInUI(self): ...
> def timing(self): ...
> def mainApp(self): ...
> def createMainUIFrame(self): ...
> def displayMain(self): ...
> def signOut(self): ...
> def openChangeInfoWindow(self): ...
> def run(self): ...
```

Hình 5: Code session management

Vì vậy, ứng dụng phục vụ Session Management được xây dựng trên nền tảng desktop application, sử dụng ngôn ngữ lập trình python và thư viện tkinter để thiết kế giao diện đơn giản. Cấu trúc code gồm

3 class chính quản lý tài khoản của hội viên, phiên đăng nhập trên một máy tính bất kì.

#### Cơ chế kết nối cơ sở dữ liệu:

- Ứng dụng kết nối với cơ sở dữ liệu theo cơ chế non-persistent nghĩa là mỗi lần truy suất cơ sở dữ liệu, ứng dụng lại tạo mới một kết nối và sử dụng kết nối đó để tương tác với cơ sở dữ liệu. Sau khi hoàn tất tác vụ sử dụng kết nối database, kết nối đó sẽ được đóng lại.

- Để kết nối ứng dụng với cơ sở dữ liệu, sử dụng thư viện mysql.connector được cung cấp bởi hệ quản trị cơ sở dữ liệu MySQL hỗ trợ kết nối với DBMS bằng ngôn ngữ lập trình Python.

- Để thiết lập kết nối với MySQL sử dụng câu lệnh:

```
1 DATABASECONFIG = {
2     'host': "localhost",
3     'user': "root",
4     'password': "password",
5     'database': 'internet_store'
6 }
7 self.__connection = mysql.connector.connect(**DATABASECONFIG)
```

Hình 6: Thiết lập kết nối với MySQL từ python

- Để truy vấn cơ sở dữ liệu sử dụng các loại câu lệnh:

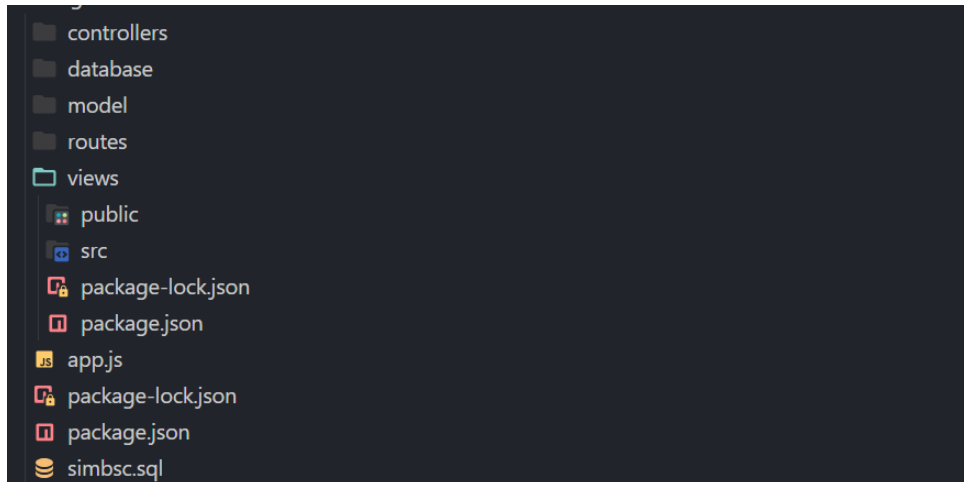
```
1 with self.__connection.cursor() as cursor:
2     cursor.execute("----query----") #execute a query
3     cursor.fetchall() #fetch all results
4     cursor.fetchone() #fetch the first result
5     cursor.callproc("----procedure name---",args=["procedure argurment"])
6     cursor.commit() #execute a procedure
7     for result in cursor.stored_results():
8         data = result.fetchall() #get procedure result
```

Hình 7: Truy vấn MySQL từ python

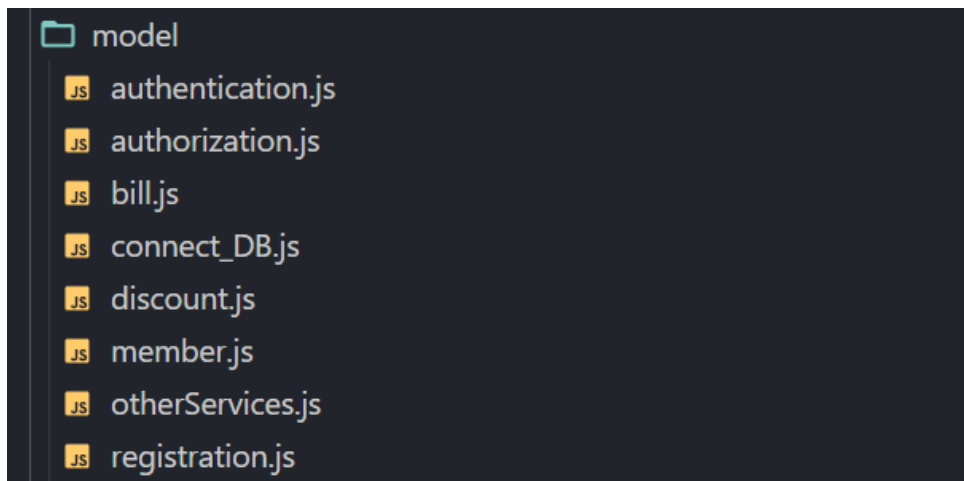
### 2.2.3 Reception service

Đây là service hỗ trợ lễ tân quản lý các giao dịch cũng như phòng net, nhóm thiết kế ứng dụng cho service này theo hướng ứng dụng web theo mô hình kiến trúc MVC. Với mô hình hoạt động theo client server. Server web được xây dựng trên thư viện NodeJS và giao diện người dùng được xây dựng trên nền tảng ReactJS.

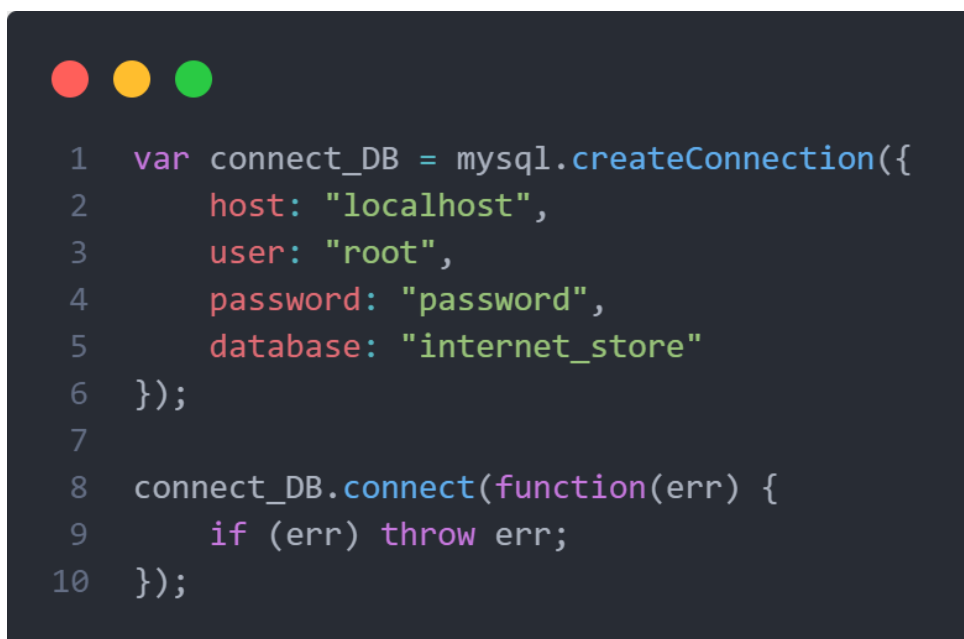
Server sẽ trực tiếp giao tiếp giao tiếp với DBMS để lấy dữ liệu từ database và sau đó gửi thông tin cho client. Với mô hình này, mọi giao tiếp giữa server với DBMS sẽ thông qua các module model (đặt trong thư mục model). Để tạo kết nối với DBMS, nhóm sử dụng thư viện mysql2.



Hình 8: Cấu trúc chung



Hình 9: Cấu trúc model của reception service

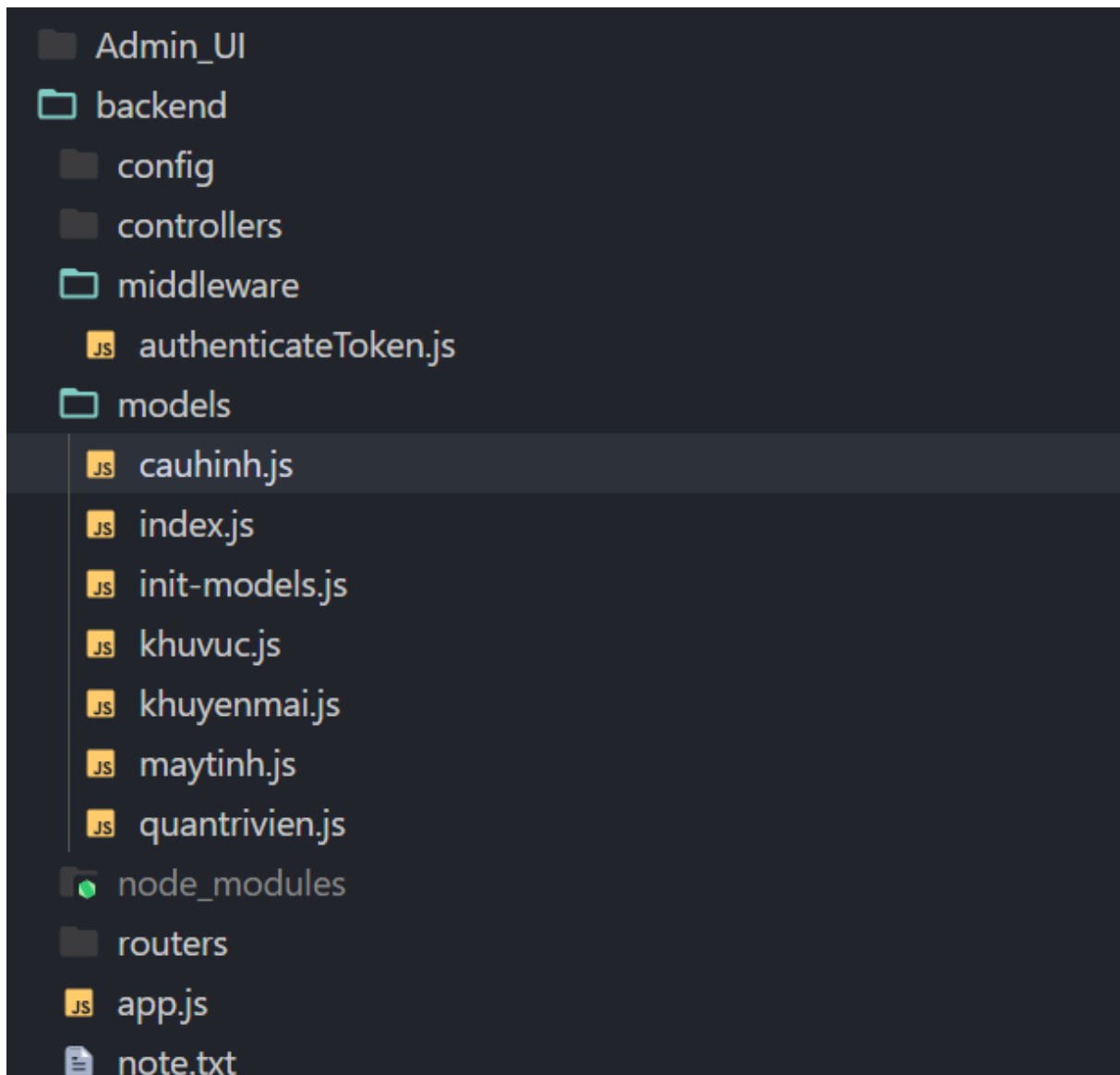


Hình 10: Kết nối DBMS reception service

#### 2.2.4 Admin service

Đây là service quản lý chung của chủ phòng net với phòng net của mình bao gồm các chức năng cấu hình phòng net. nhóm thiết kế ứng dụng cho service này theo hướng ứng dụng web theo mô hình kiến trúc layer. Chạy theo mô hình client server. Server web được xây dựng trên thư viện NodeJS và giao diện người dùng được xây dựng trên nền tảng HTML, CSS và JS. Có sử dụng Bootstrap là framework hỗ trợ thiết kế giao diện và thư viện JQuery hỗ trợ viết lệnh Js nhanh và tiện lợi hơn.

Server sẽ trực tiếp giao tiếp giao tiếp với DBMS để lấy dữ liệu từ database và sau đó gửi thông tin cho client thông qua các API. Với mô hình này, mọi giao tiếp giữa server với DBMS sẽ thông qua các module model (đặt trong thư mục model). Để tạo kết nối với DBMS, nhóm sử dụng thư viện sequelize.



Hình 11: Kiến trúc admin service

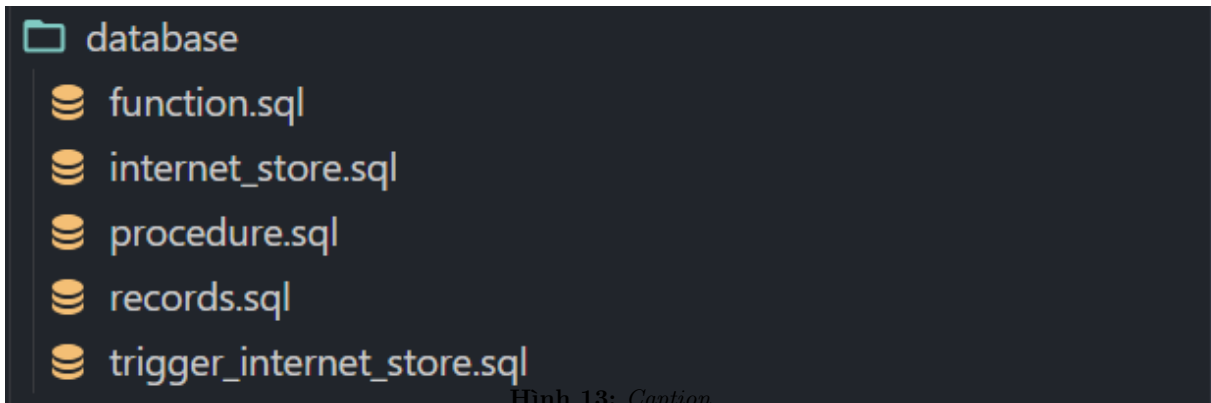
```
1  var DataTypes = require("sequelize").DataTypes;
2
3  var _quantrivien = require("./quantrivien");
4  var _khuvuc = require("./khuvuc");
5  var _maytinh = require("./maytinh");
6  var _cauhinh = require("./cauhinh");
7  var _khuyenmai = require("./khuyenmai");
8
9  function initModels(sequelize) {
10   var quantrivien = _quantrivien(sequelize, DataTypes);
11   var maytinh = _maytinh(sequelize, DataTypes);
12   var khuvc = _khuvuc(sequelize, DataTypes);
13   var cauhinh = _cauhinh(sequelize, DataTypes);
14   var khuyenmai = _khuyenmai(sequelize, DataTypes);
15
16   maytinh.belongsTo(cauhinh, {
17     foreignKey: "id cau hinh",
18   });
19
20   maytinh.belongsTo(khuvc, {
21     through: khuvc,
22     foreignKey: "phan loai khu vuc",
23   });
24   khuvc.hasMany(maytinh, {
25     foreignKey: "loai khu vuc",
26   });
27
28   return {
29     maytinh,
30     khuvc,
31     cauhinh,
32     quantrivien,
33   };
34 }
35 module.exports = initModels;
36 module.exports.initModels = initModels;
37 module.exports.default = initModels;
```

Hình 12: Tạo kết nối với DBMS admin service

## 3 Đặc tả cơ sở dữ liệu

### 3.1 Cấu trúc file

Nhóm xây dựng cơ sở dữ liệu trên 5 file script được viết bằng SQL khác nhau.



Hình 13: Caption

### 3.2 Bảng dữ liệu (table)

Cơ sở dữ liệu gồm 22 bảng, được tạo ra từ file internet\_store.sql

Tạo database:

```
DROP SCHEMA IF EXISTS `internet_store`;  
CREATE SCHEMA IF NOT EXISTS `internet_store`;  
USE `internet_store`;
```

Tạo bảng cấu hình:

```
CREATE TABLE IF NOT EXISTS `Cau Hinh` (  
ID INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
`kich thuoc man hinh` FLOAT NOT NULL,  
`cpu` CHAR(255) NOT NULL,  
`card do hoa` CHAR(255) NOT NULL,  
ram INT NOT NULL,  
`tan so man hinh` INT NOT NULL,  
gia INT NOT NULL,  
check (gia > 0 and ram>4 and `kich thuoc man hinh`>20 and `tan so man hinh`>=60)  
);
```

Tạo bảng khu vực:

```
CREATE TABLE IF NOT EXISTS `Khu Vuc` (  
`loai khu vuc` CHAR(255) NOT NULL PRIMARY KEY,  
`phu thu` INT NOT NULL  
);
```

Tạo bảng máy tính:

```
CREATE TABLE IF NOT EXISTS `May Tinh` (  
ID INT NOT NULL auto_increment PRIMARY KEY,  
hang CHAR(255) NOT NULL,  
`ngay mua` DATE NOT NULL,  
`phan loai khu vuc` CHAR(255) NOT NULL,
```

```
`id cau hinh` INT NOT NULL,  
FOREIGN KEY (`phan loai khu vuc`)  
    REFERENCES `Khu Vuc` (`loai khu vuc`),  
FOREIGN KEY (`id cau hinh`)  
    REFERENCES `Cau Hinh` (ID)  
);
```

Tạo bảng hội viên:

```
CREATE TABLE IF NOT EXISTS `Hoi Vien` (  
    `account` CHAR(255) NOT NULL PRIMARY KEY,  
    `password` CHAR(255) NOT NULL,  
    `ten` CHAR(255) NOT NULL,  
    `sdt` CHAR(10) NOT NULL UNIQUE,  
    email CHAR(255),  
    `so du` INT NOT NULL DEFAULT 0,  
    `level` INT NOT NULL DEFAULT 0,  
    CHECK (`level` >= 0 AND `so du` >= 0)  
);
```

Tạo bảng Session:

```
CREATE TABLE IF NOT EXISTS `Session` (  
    `Session ID` INT NOT NULL PRIMARY KEY auto_increment,  
    `gio bat dau` DATETIME NOT NULL,  
    `gio ket thuc` DATETIME,  
    `id may tinh` INT NOT NULL,  
    `tai khoan hv` CHAR(255) NOT NULL,  
    FOREIGN KEY (`id may tinh`)  
        REFERENCES `May Tinh` (id),  
    FOREIGN KEY (`tai khoan hv`)  
        REFERENCES `Hoi Vien` (`account`)  
);
```

Tạo bảng nhân viên:

```
CREATE TABLE IF NOT EXISTS `Nhan Vien` (  
    ID INT NOT NULL PRIMARY KEY AUTO_INCREMENT,  
    `ho ten` CHAR(255) NOT NULL,  
    `gioi tinh` INT NOT NULL,  
    `ngay sinh` DATE NOT NULL,  
    `ngay ki ket` DATE NOT NULL,  
    `ngay het han` DATE NOT NULL,  
    CHECK (`gioi tinh` >= 0 AND `gioi tinh` <= 3)  
);
```

Tạo bảng Kỹ thuật:

```
CREATE TABLE IF NOT EXISTS `Ky Thuat` (  
    ID INT NOT NULL PRIMARY KEY,  
    `loai hinh lam viec` CHAR(255) NOT NULL,  
    FOREIGN KEY (ID)  
        REFERENCES `Nhan Vien` (ID)  
);
```

Tạo bảng chuyên môn:

```
CREATE TABLE IF NOT EXISTS `Chuyen Mon` (  
    ID INT NOT NULL,  
    `chuyen mon` CHAR(255) NOT NULL PRIMARY KEY,  
    FOREIGN KEY (ID)  
        REFERENCES `Ky Thuat` (ID)  
);
```

Tạo bảng tập vụ:

```
CREATE TABLE IF NOT EXISTS `Tap Vu` (  
    ID INT NOT NULL PRIMARY KEY,  
    `nhiem vu` CHAR(255) NOT NULL,  
    FOREIGN KEY (ID)  
        REFERENCES `Nhan Vien` (ID)  
);
```

Tạo bảng lễ tân:

```
CREATE TABLE IF NOT EXISTS `Le Tan` (  
    ID INT NOT NULL PRIMARY KEY,  
    `account` CHAR(255) NOT NULL UNIQUE,  
    `password` CHAR(255) NOT NULL,  
    `lich truc` CHAR(255),  
    FOREIGN KEY (ID)  
        REFERENCES `Nhan Vien` (ID)  
);
```

Tạo bảng hóa đơn:

```
CREATE TABLE IF NOT EXISTS `Hoa Don` (  
    `ma hoa don` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    `ngay thuc hien` DATETIME NOT NULL,  
    `tong tien` INT NOT NULL DEFAULT(0),  
    `tai khoan hv` CHAR(255) NOT NULL,  
    `id le tan` INT NOT NULL,  
    FOREIGN KEY (`tai khoan hv`)  
        REFERENCES `Hoi Vien` (`account`),  
    FOREIGN KEY (`id le tan`)  
        REFERENCES `Le Tan` (ID),  
    CHECK (`tong tien` >= 0)  
);
```

Tạo bảng sản phẩm:

```
CREATE TABLE IF NOT EXISTS `San Pham` (  
    `ma san pham` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,  
    `ten san pham` CHAR(255) NOT NULL,  
    `gia niem yet` INT NOT NULL,  
    CHECK (`gia niem yet` >= 0)  
);
```

Tạo bảng chứa:



```
CREATE TABLE IF NOT EXISTS `Chua` (  
    `ma hoa don` INT NOT NULL,  
    `ma san pham` INT NOT NULL,  
    `so luong` INT NOT NULL,  
    FOREIGN KEY (`ma hoa don`)  
        REFERENCES `Hoa Don` (`ma hoa don`),  
    FOREIGN KEY (`ma san pham`)  
        REFERENCES `San Pham` (`ma san pham`),  
    PRIMARY KEY (`ma hoa don` , `ma san pham`)  
);
```

Tạo bảng giỏ chơi:

```
CREATE TABLE IF NOT EXISTS `Gio Choi` (  
    `ma san pham` INT NOT NULL PRIMARY KEY,  
    FOREIGN KEY (`ma san pham`)  
        REFERENCES `San Pham` (`ma san pham`)  
);
```

Tạo bảng nhà cung cấp:

```
CREATE TABLE IF NOT EXISTS `Nha Cung Cap` (  
    id INT NOT NULL PRIMARY KEY,  
    sdt CHAR(255) NOT NULL UNIQUE,  
    ten CHAR(255) NOT NULL UNIQUE,  
    address CHAR(255) NOT NULL  
);
```

Tạo bảng dịch vụ thêm:

```
CREATE TABLE IF NOT EXISTS `Dich Vu Them` (  
    `ma san pham` INT NOT NULL PRIMARY KEY,  
    `id nha cung cap` INT NOT NULL,  
    `so luong` INT NOT NULL,  
    `loai san pham` CHAR(255) NOT NULL,  
    `hinh anh` CHAR(255),  
    `mo ta` CHAR(255),  
    `loai the` CHAR(255),  
    `nha phat hanh` CHAR(255),  
    check (`so luong` >=0),  
    FOREIGN KEY (`ma san pham`)  
        REFERENCES `San Pham` (`ma san pham`),  
    FOREIGN KEY (`id nha cung cap`)  
        REFERENCES `Nha Cung Cap` (id)  
);
```

Tạo bảng hóa đơn chi:

```
CREATE TABLE IF NOT EXISTS `Hoa Don Chi` (  
    `ma hoa don` INT NOT NULL PRIMARY KEY,  
    `ngay thuc hien` DATETIME NOT NULL,  
    `id le tan` INT NOT NULL,  
    FOREIGN KEY (`id le tan`)  
        REFERENCES `Le Tan` (id)  
);
```

Tạo bảng ghi hóa đơn chi:

```
CREATE TABLE IF NOT EXISTS `Ghi Hoa Don Chi` (  
    `ma hoa don chi` INT NOT NULL,  
    `ma san pham` INT NOT NULL,  
    `so luong` INT NOT NULL,  
    `gia nhap` INT NOT NULL,  
    CHECK (`so luong` > 0 AND `gia nhap` > 0),  
    FOREIGN KEY (`ma san pham`)  
        REFERENCES `Dich Vu Them` (`ma san pham`),  
    FOREIGN KEY (`ma hoa don chi`)  
        REFERENCES `Hoa Don Chi` (`ma hoa don`),  
    PRIMARY KEY (`ma hoa don chi`, `ma san pham`)  
);
```

Tạo bảng biên bản:

```
CREATE TABLE IF NOT EXISTS `Bien Ban` (  
    `ma bien ban` INT NOT NULL PRIMARY KEY,  
    `phuong an xu ly` CHAR(255) NOT NULL,  
    `tinh trang truoc bao tri` CHAR(255) NOT NULL,  
    `chi phi sua chua` INT NOT NULL,  
    `ngay thuc hien` DATE NOT NULL,  
    `tinh trang sau bao tri` CHAR(255) NOT NULL,  
    `id may tinh` INT NOT NULL,  
    `ma hoa don chi` INT NOT NULL,  
    CHECK (`chi phi sua chua` >= 0),  
    FOREIGN KEY (`id may tinh`)  
        REFERENCES `May Tinh` (id),  
    FOREIGN KEY (`ma hoa don chi`)  
        REFERENCES `Hoa Don Chi` (`ma hoa don`)  
);
```

Tạo bảng khuyến mãi:

```
CREATE TABLE IF NOT EXISTS `Khuyen Mai` (  
    `ma khuyen mai` INT NOT NULL PRIMARY KEY,  
    `ten chuong trinh` CHAR(255) NOT NULL,  
    `mo ta` CHAR(255),  
    `ngay bat dau` DATETIME NOT NULL,  
    `ngay ket thuc` DATETIME NOT NULL,  
    `loai` CHAR(255) NOT NULL,  
    `dieu kien` INT NOT NULL,  
    `muc giam` FLOAT NOT NULL,  
    CHECK (`ngay ket thuc` > `ngay bat dau`  
        AND `muc giam` > 0  
        AND `muc giam` < 1  
        AND `dieu kien` > 0)  
);
```

Tạo bảng áp dụng sản phẩm:

```
CREATE TABLE IF NOT EXISTS `Ap Dung San Pham` (  
    `ma san pham` INT NOT NULL,
```

```
`ma khuyen mai` INT NOT NULL,
FOREIGN KEY (`ma khuyen mai`)
REFERENCES `Khuyen Mai` (`ma khuyen mai`),
FOREIGN KEY (`ma san pham`)
REFERENCES `San Pham` (`ma san pham`),
PRIMARY KEY (`ma san pham` , `ma khuyen mai`)
);
```

Tạo bảng áp dụng hóa đơn:

```
CREATE TABLE IF NOT EXISTS `Ap Dung Hoa Don` (
`ma hoa don` INT NOT NULL PRIMARY KEY,
`ma khuyen mai` INT NOT NULL,
FOREIGN KEY (`ma hoa don`)
REFERENCES `Hoa Don` (`ma hoa don`),
FOREIGN KEY (`ma khuyen mai`)
REFERENCES `Khuyen Mai` (`ma khuyen mai`)
);
```

Sau khi thực hiện thêm các bảng, cơ sở dữ liệu thu được sẽ gồm:

Name	Engine	Version	Row Format	Rows	Avg Row Length	Data Length	Max Data Length	Index Length	Data Free	Auto Incre...
ap dung hoa don	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
ap dung san pham	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
bien ban	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0
cau hinh	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
chua	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
chuyen mon	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
dich vu them	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
ghi hoa don chi	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
gio choi	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
hoa don	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0
hoa don chi	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
hoi vien	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
khu vuc	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
khuyen mai	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
ky thuat	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
le tan	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	16.0 KiB	0.0 bytes	0
may tinh	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0
nha cung cap	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0
nhân viên	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
san pham	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0
session	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	32.0 KiB	0.0 bytes	0
tap vu	InnoDB	10	Dynamic	0	0	16.0 KiB	0.0 bytes	0.0 bytes	0.0 bytes	0

Hình 14: Kết quả sau khi thực hiện thêm bảng

### 3.3 Các trigger

Trong hệ thống, với các thao tác kiểm tra điều kiện phức tạp không thể kiểm tra đơn thuần thông qua Check hay để có sự thay đổi tự động liên quan đến các bảng khác nhau. Nhóm sẽ trình bày 1 số trigger phức tạp mà nhóm thực hiện. Tất cả các trigger nhóm thực hiện được lưu trữ trong file trigger\_internet\_store.sql.

#### 3.3.1 Trigger thay đổi số dư tài khoản hội viên ( update\_account\_balance)

**Mô tả chung:** Thực hiện tự động thay đổi số dư của tài khoản hội viên được chứa trong bảng `hoi vien` khi có sản phẩm là sản phẩm giờ chơi (sản phẩm có mã sản phẩm 1) được thêm vào bảng chứa với mã hóa đơn là hóa đơn của hội viên.

**Tiền điều kiện (Precondition):**

- Thực hiện thêm dữ liệu trên bảng `Chua`

- Dữ liệu được thêm vô có `ma san pham` là 1

#### Hậu điều kiện (Postcondtion):

- Thay đổi giá trị trường `so du` trong bảng `hoi vien` có `account` ứng với khóa ngoại của bảng `hoa don` được trường `ma hoa don` tham khảo tới.

#### Hiện thực:

```
DELIMITER //
CREATE TRIGGER update_account_balance
AFTER INSERT
ON `Chua`
FOR EACH ROW
BEGIN
    DECLARE tai_khoan_hv VARCHAR(255);
    DECLARE new_so_du INT;

    IF NEW.`ma san pham` = 1 THEN
        SELECT `tai khoan hv` INTO tai_khoan_hv FROM `Hoa Don` WHERE `ma hoa don`
            = NEW.`ma hoa don`;
        SELECT `so du` INTO new_so_du FROM `Hoi Vien` WHERE `account` =
            tai_khoan_hv;
        SET new_so_du = new_so_du + NEW.`so luong` * 1000;
        UPDATE `Hoi Vien` SET `so du`=new_so_du WHERE `account`= tai_khoan_hv;
    END IF;
END;
//
DELIMITER ;
```

### 3.3.2 Trigger đăng xuất tài khoản hội viên (sign\_out)

**Mô tả chung:** Khi người dùng thực hiện đăng xuất khỏi máy tính, tài khoản hội viên tương ứng sẽ được trừ tiền theo tài giá theo giờ chơi ứng với máy đang chơi.

#### Precondition:

- Thực hiện cập nhật `gio ket thuc` trên bảng `Session`

#### Postcondtion:

- Tài khoản hội viên được thay đổi trường `so du`

#### Hiện thực:

```
DELIMITER //
create trigger sign_out before update on `session`
for each row
begin
    declare price int;
    declare phuthu int;
    declare cost float;
    declare period float;
    declare sodu int;
```

```
if old.`gio ket thuc` is null then
    set period = new.`gio ket thuc` - new.`gio bat dau`;
set period = period / 3600;
SELECT gia INTO price FROM `may tinh`
    JOIN `cau hinh` ON `may tinh`.`id cau hinh` = `cau hinh`.ID
    WHERE `may tinh`.ID = new.`id may tinh`;

SELECT `phu thu` INTO phuthu FROM `may tinh`
    JOIN `khu vuc` ON `may tinh`.`phan loai khu vuc` = `khu vuc`.`loai khu vuc`
    WHERE `may tinh`.ID = new.`id may tinh`;
set cost = period*(price+phuthu);
select `so du` into sodu from `hoi vien` where `account` = new.`tai
    khoan hv`;
set sodu = sodu - cost;
update `hoi vien` set `so du` = sodu where `account` = new.`tai khoan hv`
    `;
end if;
end;
//
DELIMITER ;
```

### 3.3.3 Trigger kiểm tra thêm sản phẩm được áp dụng khuyến mãi (add\_product\_for\_discount)

**Mô tả chung:** Kiểm tra điều kiện khi thêm một sản phẩm vào chương trình khuyến mãi.

**Precondition:**

- Mã sản phẩm, mã chương trình khuyến mãi.

**Postcondition:**

- Thêm sản phẩm được áp dụng khuyến mãi nếu khuyến mãi có loại sản phẩm.
- Không thêm sản phẩm được áp dụng khuyến mãi nếu khuyến mãi không có loại sản phẩm.

**Hiện thực:**

```
DELIMITER //
CREATE TRIGGER add_product_for_discount BEFORE INSERT ON `ap dung san pham`
FOR EACH ROW
BEGIN
    declare loai_km varchar(255);
    select loai into loai_km from `khuyen mai` where `ma khuyen mai`=new.`ma
        khuyen mai`;
    if loai_km != 'product' then
        SIGNAL SQLSTATE '45000';
    end if;
END;
//
DELIMITER ;
```

### 3.4 Hàm và thủ tục

Tất cả các dữ liệu được truy vấn từ tầng ứng dụng lên cơ sở dữ liệu đều truy vấn thông qua các thủ tục (procedure) và các hàm (function). Tất cả các theo tác CRUD (create, read, update, delete) đều thực hiện thông qua các procedure.

#### 3.4.1 Các thủ tục (procedure)

Các procedure được lưu trong file procedure.sql bao gồm các procedure để truy vấn dữ liệu hay thêm xóa sửa dữ liệu của một bảng. Tuy vậy, trong mục này, nhóm sẽ trình bày chính về các procedure phức tạp trong phần này.

##### 3.4.1.a Đăng nhập (signIn)

**Mô tả chung:** Thực hiện chức năng đăng nhập vào máy với tài khoản hội viên. Có kiểm tra các điều kiện khi đăng nhập.

**Input:**

- hoivien varchar(255): `account` của `hội viên`
- matkhau varchar(255): `password` của `hội viên`
- matkhau int: ID của `máy tính`

**Output:**

- `status` boolean: Trạng thái đăng nhập, True cho thành công, False cho thất bại.
- session\_id int: Mã session được cấp nếu thành công.
- Bảng `Session` thêm một dòng nếu đăng nhập thành công.

**Hiện thực:**

```
DELIMITER //
CREATE PROCEDURE signIn(
    IN hoivien varchar(255),
    IN matkhau varchar(255),
    IN maytinh int,
    OUT `status` boolean,
    OUT session_id int
)
BEGIN
    declare tai_khoan varchar(255);
    declare id_may_tinh int;
    declare latest_end_time_account datetime;
    declare start_time datetime;
    declare latest_end_time_pc datetime;
    select `account` into tai_khoan from `Hoi Vien` where `account` = hoivien
        and `password` = matkhau;
    select id into id_may_tinh from `máy tính` where id = maytinh;
    if tai_khoan is not null and id_may_tinh is not null then
        select `gio ket thuc`, `gio bat dau` into
            latest_end_time_account, start_time
```

```
from `session`
where `tai khoan hv` = tai_khoan
ORDER BY `gio bat dau` DESC
LIMIT 1;
select `gio ket thuc` into latest_end_time_pc
from `session`
where `id may tinh` = id_may_tinh
ORDER BY `gio bat dau` DESC
LIMIT 1;
if start_time is null then
    insert into `session` (`gio bat dau`,`id may tinh`,`tai
        khoan hv`) values (now(),maytinh,hoivien);
    set `status`=true;
    select `Session ID` into session_id from `session` where `id may
        tinh`=id_may_tinh and `gio ket thuc` is null;
else
    if latest_end_time_account is null or latest_end_time_pc
        is null then
        set `status` = FALSE;
    else
        if latest_end_time_account > now() or
            latest_end_time_pc>now() then
        set `status`=False;
        else
            insert into `session` (`gio bat dau`,`id
                may tinh`,`tai khoan hv`) values (
                    now(),maytinh,hoivien);
            set `status`=true;
            select `Session ID` into session_id from `session` where `id
                may tinh`=id_may_tinh and `gio ket thuc` is null;
            end if;
        end if;
    end if;
ELSE
    set `status` = FALSE;
END IF;
END;
//
DELIMITER ;
```

### 3.4.1.b Thêm sản phẩm vào hóa đơn (addProductToBill)

**Mô tả chung:** Thực hiện chức năng thêm một sản phẩm vào hóa đơn. Tự động áp dụng mã giảm giá theo sản phẩm nếu thỏa điều kiện.

**input:**

- p\_product\_id INT: Mã sản phẩm.
- p\_so\_luong int: Số lượng sản phẩm.
- p\_bill\_id int: Mã hóa đơn.

### Output:

- Một sản phẩm được thêm vào bảng chưa.
- Nếu thêm thành công, cập nhật tổng tiền của đơn hàng.

### Hiện thực:

```
DELIMITER //
```

```
CREATE PROCEDURE addProductToBill(  
    IN p_product_id INT,  
    IN p_so_luong INT,  
    IN p_bill_id INT  
)  
BEGIN  
    DECLARE date_apply DATETIME;  
    DECLARE muc_giam FLOAT;  
    DECLARE ma_khuyen_mai INT;  
    DECLARE thanh_tien INT;  
    DECLARE bill_cost INT;  
    DECLARE available_quantity INT;  
  
    IF p_product_id IS NULL OR p_so_luong IS NULL OR p_bill_id IS NULL THEN  
        SIGNAL SQLSTATE '45000'  
        SET MESSAGE_TEXT = 'Tham so dau vao khong duoc bo trong';  
    END IF;  
  
    IF p_product_id <> 1 THEN  
        SELECT `so luong` INTO available_quantity  
        FROM `Dich Vu Them`  
        WHERE `ma san pham` = p_product_id;  
  
        IF available_quantity = 0 THEN  
            SIGNAL SQLSTATE '45000'  
            SET MESSAGE_TEXT = 'So luong cua san pham nay trong Dich Vu Them  
                dang la 0';  
        END IF;  
  
        IF available_quantity < p_so_luong THEN  
            SET p_so_luong = available_quantity;  
        END IF;  
    END IF;  
  
    SELECT `ngay thuc hien` INTO date_apply FROM `Hoa Don` WHERE `ma hoa don` =  
        p_bill_id;  
    CALL getDiscountForProduct(p_product_id, p_so_luong, date_apply,  
        ma_khuyen_mai, muc_giam);  
  
    SELECT `gia niem yet` INTO thanh_tien FROM `San Pham` WHERE `ma san pham` =  
        p_product_id;
```



```
SELECT `tong tien` INTO bill_cost FROM `Hoa Don` WHERE `ma hoa don` =  
    p_bill_id;  
  
IF muc_giam IS NOT NULL THEN  
    SET thanh_tien = (1 - muc_giam) * thanh_tien * p_so_luong;  
ELSE  
    SET thanh_tien = thanh_tien * p_so_luong;  
END IF;  
  
SET bill_cost = bill_cost + thanh_tien;  
UPDATE `Hoa Don` SET `tong tien` = bill_cost WHERE `ma hoa don` = p_bill_id;  
  
INSERT INTO `Chua` (`ma hoa don`, `ma san pham`, `so luong`) VALUES (  
    p_bill_id, p_product_id, p_so_luong);  
  
IF p_product_id <> 1 THEN  
    UPDATE `Dich Vu Them`  
    SET `so luong` = available_quantity - p_so_luong  
    WHERE `ma san pham` = p_product_id;  
END IF;  
END //  
DELIMITER ;
```

#### 3.4.1.c Thông kê khuyến mãi (thongKeKhuyenMai)

**Mô tả chung:** Thực hiện thống kê các sản phẩm và số lượng sản phẩm bán được theo một chương trình khuyến mãi.

**Input:**

- Mã khuyến mãi

**Output:**

- Mã sản phẩm và số lượng sản phẩm bán được

**Hiện thực:**

```
DELIMITER //  
CREATE PROCEDURE thongKeKhuyenMai(  
    in ma_khuyen_mai int  
)  
BEGIN  
    Select ct.`ma san pham`, ct.`so luong`, ct.`ma hoa don`  
    from `hoa don` as hd natural join  
    (SELECT *  
    FROM chua as c natural join  
    (SELECT km.`ngay bat dau`, km.`ngay ket thuc`, adsp.`ma san pham`  
    from `khuyen mai` AS km  
    natural join `ap dung san pham` as adsp  
    where km.`ma khuyen mai`= ma_khuyen_mai and km.`ma khuyen mai` = adsp.`ma  
    khuyen mai`) as spkm
```

```
        where spkm.`ma san pham` = c.`ma san pham`) AS ct
    where hd.`ma hoa don` = ct.`ma hoa don`
    and hd.`ngay thuc hien` >= ct.`ngay bat dau`
    and hd.`ngay thuc hien` <= ct.`ngay ket thuc`;
END;
//
DELIMITER ;
```

### 3.4.2 Các hàm (function)

Các hàm được lưu trong file function.sql

#### 3.4.2.a Kiểm tra mã máy tính (checkPCID)

```
DELIMITER //
create function checkPCID(pc_id int)
RETURNS int
deterministic
BEGIN
    declare id_check int;
    select ID into id_check from `may tinh` where ID = pc_id;
    if id_check is not null then
        return 1;
    else
        return 0;
    end if ;
END;
//
DELIMITER ;
```

#### 3.4.2.b Lấy giá máy tính (getPCPrice)

```
DELIMITER //
CREATE FUNCTION getPCPrice(pc_id INT)
RETURNS INT
DETERMINISTIC
BEGIN
    DECLARE price INT;
    DECLARE phuthu INT;
    SELECT gia INTO price FROM `may tinh`
    JOIN `cau hinh` ON `may tinh`.`id cau hinh` = `cau hinh`.ID
    WHERE `may tinh`.ID = pc_id;

    SELECT `phu thu` INTO phuthu FROM `may tinh`
    JOIN `khu vuc` ON `may tinh`.`phan loai khu vuc` = `khu vuc`.`loai khu vuc`
    WHERE `may tinh`.ID = pc_id;

    SET price = price + phuthu;
    RETURN price;
```



```
END ;  
//  
DELIMITER ;
```

### 3.5 Dữ liệu mẫu

Dữ liệu mẫu nhóm chuẩn bị được đặt trong file records.sql

## 4 Hiện thực và các tính năng

### 4.1 Session Management Service (dành cho hội viên)

#### 4.1.1 Đăng nhập

##### 4.1.1.a Mô tả

Hội viên thực hiện đăng nhập vào máy.



Hình 15: Giao diện đăng nhập

	Session ID	gio bat dau	gio ket thuc	id may tinh	tai khoan hv
▶	1	2023-12-10 16:08:59	2023-12-10 16:09:03	1	account1

Hình 16: Trước khi đăng nhập

	Session ID	gio bat dau	gio ket thuc	id may tinh	tai khoan hv
▶	1	2023-12-10 16:08:59	2023-12-10 16:09:03	1	account1
	2	2023-12-10 16:09:45	NULL	1	account1

Hình 17: Sau khi đăng nhập

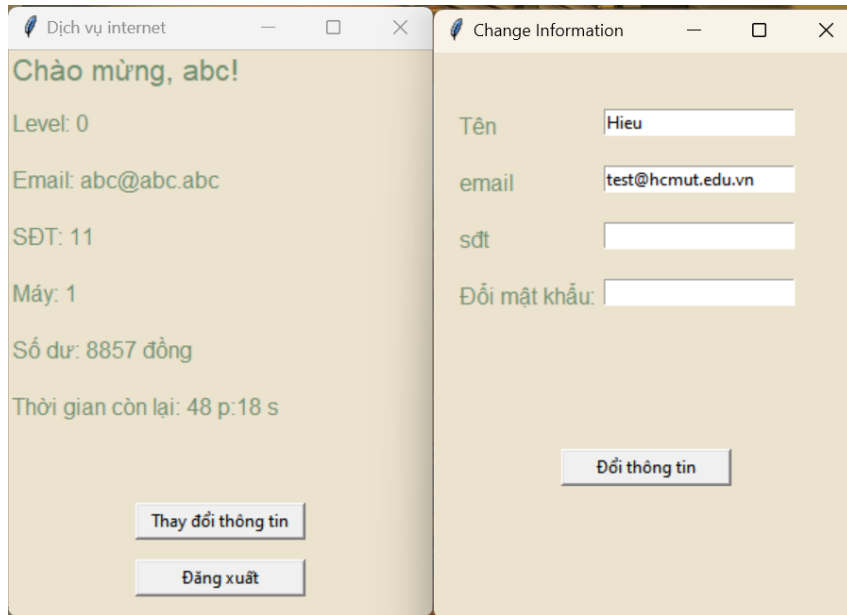
##### 4.1.1.b Thủ tục

Thực hiện thủ tục trình bày trong phần 3.4.1.a

### 4.1.2 Thay đổi thông tin

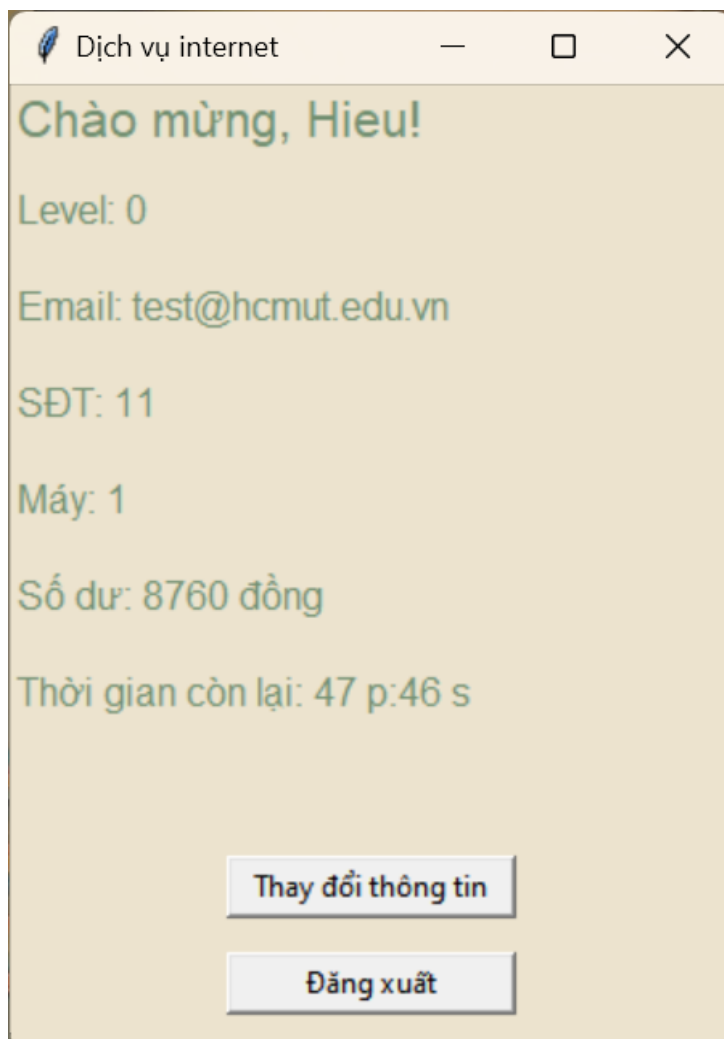
#### 4.1.2.a Mô tả

Người dùng thực hiện thay đổi thông tin tài khoản



The image shows two browser windows side-by-side. The left window, titled 'Dịch vụ internet', displays a user profile for 'abc!'. It lists: Level: 0, Email: abc@abc.abc, SĐT: 11, Máy: 1, Số dư: 8857 đồng, and Thời gian còn lại: 48 p:18 s. At the bottom are buttons for 'Thay đổi thông tin' and 'Đăng xuất'. The right window, titled 'Change Information', contains input fields for 'Tên' (filled with 'Hieu'), 'email' (filled with 'test@hcmut.edu.vn'), 'sdt', and 'Đổi mật khẩu'. A 'Đổi thông tin' button is at the bottom.

Hình 18: Thực hiện thay đổi thông tin



The image shows a single browser window titled 'Dịch vụ internet'. The user profile now displays: 'Chào mừng, Hieu!', Level: 0, Email: test@hcmut.edu.vn, SĐT: 11, Máy: 1, Số dư: 8760 đồng, and Thời gian còn lại: 47 p:46 s. The 'Thay đổi thông tin' and 'Đăng xuất' buttons remain at the bottom.

Hình 19: sau khi thay đổi thông tin

#### 4.1.2.b Thủ tục

```
DELIMITER //
create procedure signOutBySessionID(
    in session_id int
)
BEGIN
    update `session` set `gio ket thuc` = now() where `session id` =
        session_id;
END ;
//
DELIMITER ;
```

#### 4.1.3 Đăng xuất

##### 4.1.3.a Mô tả

Cho phép người dùng đăng xuất khỏi hệ thống.

	Session ID	gio bat dau	gio ket thuc	id may tinh	tai khoan hv
▶	1	2023-12-10 16:08:59	2023-12-10 16:09:03	1	account1
	2	2023-12-10 16:09:45	2023-12-10 16:19:02	1	account1

Hình 20: Sau khi đăng xuất

#### 4.1.3.b Thủ tục

```
DELIMITER //
create procedure signOutBySessionID(
    in session_id int
)
BEGIN
    update `session` set `gio ket thuc` = now() where `session id` =
        session_id;
END ;
//
DELIMITER ;
```

## 4.2 Reception Service

### 4.2.1 Xem danh sách hội viên

#### 4.2.1.a Mô tả

Lễ tân có thể xem danh sách hội viên

Quản lý

Hội viên

Khuyến mãi

Hoá đơn

Dịch vụ khác cho lễ tân

Đăng xuất

Quản lý hội viên

Tên	Email	Level	SĐT	Số dư
abc1	abc@abc.abc	0	11	10000
abc10	abc@abc.abc	0	110	0
abc2	abc@abc.abc	0	12	2000
abc3	abc@abc.abc	0	13	4000
abc4	abc@abc.abc	0	14	1000
abc5	abc@abc.abc	0	15	4000
abc6	abc@abc.abc	0	16	3000
abc7	abc@abc.abc	0	17	10000
abc8	abc@abc.abc	0	18	0
abc9	abc@abc.abc	0	19	0

Hình 21: Màn hình xem danh sách hội viên

#### 4.2.1.b Thủ tục

```
DELIMITER //
CREATE PROCEDURE SelectHoiVienInfo()
BEGIN
    SELECT
        `ten` AS `name`,
        `sdt` AS `phoneNumber`,
        `email`,
        `so du` AS `balance`,
        `level`
    FROM
        `Hoi Vien`;
END //
DELIMITER ;
```

#### 4.2.2 Thêm hội viên

##### 4.2.2.a Mô tả

Lễ tân thực hiện thêm một tài khoản hội viên mới.

[Quản lý](#) [Hội viên](#) [Khuyến mãi](#) [Hoá đơn](#) [Dịch vụ khác cho lễ tân](#) [Đăng xuất](#)

### Thêm hội viên

Tài khoản

Password

Tên

Email

Số điện thoại

Thêm hội viên

Hình 22: Thêm hội viên

[Quản lý](#) [Hội viên](#) [Khuyến mãi](#) [Hoá đơn](#) [Dịch vụ khác cho lễ tân](#) [Đăng xuất](#)

### Thêm hội viên

Thêm hội viên mới thành công!

Tài khoản

ahahaha

Password

.....

Tên

The Hieu

Email

hieu.pham14022003@hcmut.edu.vn

Số điện thoại

0961919761

Thêm hội viên

Hình 23: Thêm hội viên thành công

#### 4.2.2.b Thủ tục

```
CREATE PROCEDURE AddHoiVien(  
    IN p_account CHAR(255),  
    IN p_password CHAR(255),  
    IN p_ten CHAR(255),  
    IN p_sdt CHAR(10),  
    IN p_email CHAR(255)  
)  
BEGIN  
    IF p_account IS NULL OR p_account = '' THEN  
        SIGNAL SQLSTATE '45001'  
        SET MESSAGE_TEXT = 'Khong duoc bo trong tai khoan hoi vien';  
    END IF;
```



```
IF p_password IS NULL OR p_password = '' THEN
    SIGNAL SQLSTATE '45002'
    SET MESSAGE_TEXT = 'Khong duoc bo trong mat khau hoi vien';
END IF;

IF p_ten IS NULL OR p_ten = '' THEN
    SIGNAL SQLSTATE '45003'
    SET MESSAGE_TEXT = 'Khong duoc bo trong ten hoi vien';
END IF;

IF p_sdt IS NULL OR p_sdt = '' THEN
    SIGNAL SQLSTATE '45004'
    SET MESSAGE_TEXT = 'Khong duoc bo trong so dien thoai hoi vien';
END IF;

IF p_email IS NULL OR p_email = '' THEN
    SIGNAL SQLSTATE '45005'
    SET MESSAGE_TEXT = 'Khong duoc bo trong email hoi vien';
END IF;

IF NOT REGEXP_LIKE(p_sdt, '^[0-9]+$') THEN
    SIGNAL SQLSTATE '45006'
    SET MESSAGE_TEXT = 'So dien thoai hoi vien chi duoc chua ki tu
    so, khong duoc co ki tu chu cai hay dac biet';
END IF;

IF EXISTS (SELECT 1 FROM `Hoi Vien` WHERE email = p_email) THEN
    SIGNAL SQLSTATE '45007'
    SET MESSAGE_TEXT = 'Email nay da duoc dang ky boi nguoi khac';
END IF;

IF EXISTS (SELECT 1 FROM `Hoi Vien` WHERE sdt = p_sdt) THEN
    SIGNAL SQLSTATE '45008'
    SET MESSAGE_TEXT = 'So dien thoai nay da duoc dang ky boi nguoi khac';
END IF;

INSERT INTO `Hoi Vien` (
    `account`,
    `password`,
    `ten`,
    `sdt`,
    `email`
) VALUES (
    p_account,
    p_password,
    p_ten,
    p_sdt,
    p_email
);
```

```
);  
END;  
  
//  
  
DELIMITER ;
```

### 4.2.3 Thêm hóa đơn

#### 4.2.3.a Mô tả:

Thực hiện chức năng thêm hóa đơn

ID	Tên sản phẩm	Giá	Số lượng trong kho	Số lượng cần mua
7	sting	10000	1000	<input type="text" value="1"/> <input type="button" value="Thêm sản phẩm"/>

**Sản phẩm đã thêm**

ID	Tên sản phẩm	Giá niêm yết	Số lượng cần mua	Thành tiền tạm tính (chưa áp dụng khuyến mãi)
1	gio chơi	1000	3	3000
7	sting	10000	2	20000

Hình 24: Thêm sản phẩm vào hóa đơn

**Chi tiết hoá đơn**  
 ID: 8  
 Hội viên: ahahaha  
 Lễ tân: 1  
 Ngày thực hiện: 2023-12-10T11:45:34.000Z  
 Tổng tiền (đã áp dụng các khuyến mãi): 23000

**Các sản phẩm đã mua**

ID	Tên sản phẩm	Giá niêm yết	Số lượng cần mua
1	gio chơi	1000	3
7	sting	10000	2

Hình 25: Thêm hóa đơn

#### 4.2.3.b Thủ tục

Thêm hóa đơn:

```
DELIMITER //  
  
CREATE PROCEDURE InsertHoaDon(IN tai_khoan_hv VARCHAR(255), IN id_le_tan INT)  
BEGIN  
    DECLARE existing_account INT;  
    DECLARE existing_le_tan INT;
```

```
IF tai_khoan_hv IS NULL OR tai_khoan_hv = '' THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Tai Khoan HV không được bỏ trống';
END IF;

IF id_le_tan IS NULL OR id_le_tan = '' THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'ID Le Tan không được bỏ trống';
END IF;

SELECT COUNT(*) INTO existing_account
FROM `Hoi Vien`
WHERE `account` = tai_khoan_hv;

IF existing_account = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Tai Khoan HV không tồn tại';
END IF;

SELECT COUNT(*) INTO existing_le_tan
FROM `Le Tan`
WHERE `ID` = id_le_tan;

IF existing_le_tan = 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'ID Le Tan không tồn tại';
END IF;

INSERT INTO `Hoa Don` (`ngay thuc hien`, `tong tien`, `tai khoan hv`, `id le
    tan`)
VALUES (NOW(), 0, tai_khoan_hv, id_le_tan);

SELECT LAST_INSERT_ID() AS `insertId`;
END //

DELIMITER ;
```

### Thêm sản phẩm vào hóa đơn

```
DELIMITER //

CREATE PROCEDURE addProductToBill(
    IN p_product_id INT,
    IN p_so_luong INT,
    IN p_bill_id INT
)
BEGIN
    DECLARE date_apply DATETIME;
    DECLARE muc_giam FLOAT;
    DECLARE ma_khuyen_mai INT;
    DECLARE thanh_tien INT;
```

```
DECLARE bill_cost INT;
DECLARE available_quantity INT;

IF p_product_id IS NULL OR p_so_luong IS NULL OR p_bill_id IS NULL THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Tham so dau vao khong duoc bo trong';
END IF;

IF p_product_id <> 1 THEN
    SELECT `so luong` INTO available_quantity
    FROM `Dich Vu Them`
    WHERE `ma san pham` = p_product_id;

    IF available_quantity = 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'So luong cua san pham nay trong Dich Vu Them
        dang la 0';
    END IF;

    IF available_quantity < p_so_luong THEN
        SET p_so_luong = available_quantity;
    END IF;
END IF;

SELECT `ngay thuc hien` INTO date_apply FROM `Hoa Don` WHERE `ma hoa don` =
p_bill_id;
CALL getDiscountForProduct(p_product_id, p_so_luong, date_apply,
ma_khuyen_mai, muc_giam);

SELECT `gia niem yet` INTO thanh_tien FROM `San Pham` WHERE `ma san pham` =
p_product_id;

SELECT `tong tien` INTO bill_cost FROM `Hoa Don` WHERE `ma hoa don` =
p_bill_id;

IF muc_giam IS NOT NULL THEN
    SET thanh_tien = (1 - muc_giam) * thanh_tien * p_so_luong;
ELSE
    SET thanh_tien = thanh_tien * p_so_luong;
END IF;

SET bill_cost = bill_cost + thanh_tien;
UPDATE `Hoa Don` SET `tong tien` = bill_cost WHERE `ma hoa don` = p_bill_id;

INSERT INTO `Chua` (`ma hoa don`, `ma san pham`, `so luong`) VALUES (
p_bill_id, p_product_id, p_so_luong);

IF p_product_id <> 1 THEN
    UPDATE `Dich Vu Them`
```

```
        SET `so luong` = available_quantity - p_so_luong
        WHERE `ma san pham` = p_product_id;
    END IF;
END //

DELIMITER ;
```

### Lấy thông tin sản phẩm

```
DELIMITER //

CREATE PROCEDURE GetProductInfo(IN p_ten_san_pham VARCHAR(255))
BEGIN
    -- Validate input
    IF p_ten_san_pham IS NULL OR p_ten_san_pham = '' THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Khong duoc bo trong Ten san pham';
    END IF;

    -- Check if there are any matching rows in San Pham
    IF EXISTS (
        SELECT 1
        FROM `San Pham`
        WHERE LOWER(`ten san pham`) LIKE LOWER(CONCAT('%', p_ten_san_pham, '%'))
    ) THEN
        -- If there are matching rows, select from San Pham with or without Dich Vu Them
        SELECT
            sp.`ma san pham` AS `productId`,
            sp.`ten san pham` AS `name`,
            sp.`gia niem yet` AS `price`,
            dv.`so luong` AS `instock`
        FROM `San Pham` sp
        LEFT JOIN `Dich Vu Them` dv ON sp.`ma san pham` = dv.`ma san pham`
        AND dv.`ma san pham` <> 1 -- Exclude the case where ma san pham is 1
        WHERE LOWER(sp.`ten san pham`) LIKE LOWER(CONCAT('%', p_ten_san_pham, '%'));
    ELSE
        -- If there are no matching rows, return an empty result set
        SELECT
            NULL AS `productId`,
            NULL AS `name`,
            NULL AS `price`,
            NULL AS `instock`
        WHERE 1 = 0; -- Always false condition to return an empty result set
    END IF;
END //

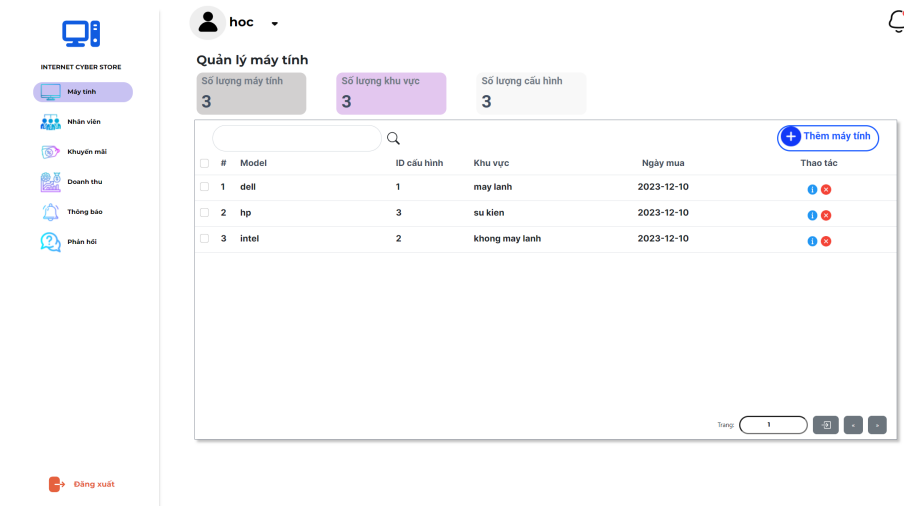
DELIMITER ;
```

## 4.3 Admin Service

### 4.3.1 Giao diện quản lý máy tính

#### 4.3.1.a Mô tả

Tính năng cho phép admin lưu trữ quản lý máy tính



Hình 26: Quản lý máy tính

#### 4.3.1.b Thủ tục

```
drop procedure if exists getAllComputerInfo;
DELIMITER //
create procedure getAllComputerInfo()
Begin
    SELECT pc.*, ch.gia, kv.`phu thu`
    FROM `may tinh` as pc
    join (SELECT id, gia from `cau hinh`) as ch
    join (SELECT `loai khu vuc`, `phu thu` from `khu vuc`) as kv
    on pc.`id cau hinh` = ch.id and pc.`phan loai khu vuc` = kv.`loai khu vuc`
    ORDER BY pc.ID ASC;
END;
//
DELIMITER ;
```

### 4.3.2 Thông tin máy tính

#### 4.3.2.a Mô tả

Hiển thị thông tin của máy tính



Hình 27: Thông tin máy in

#### 4.3.2.b Thủ tục

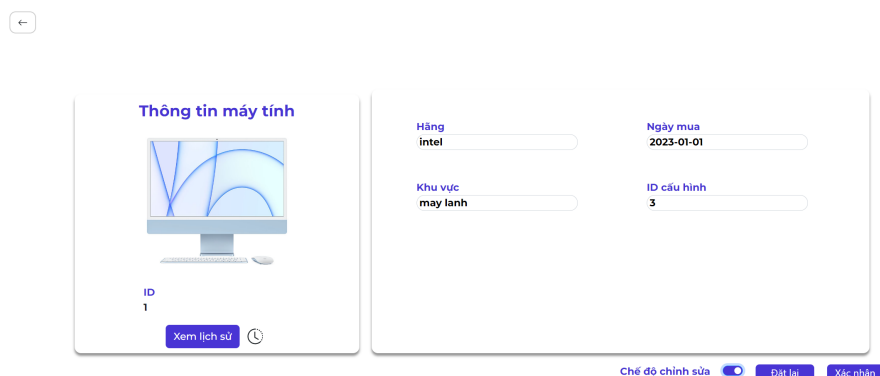
```
CREATE PROCEDURE GetMayTinhInfo(IN may_tinh_id INT)
BEGIN
    IF may_tinh_id IS NULL OR may_tinh_id = '' OR NOT may_tinh_id REGEXP '^[0-9]+$' OR may_tinh_id <= 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'Mã máy tính không được bỏ trống và phải là số nguyên dương';
    END IF;
    SELECT ch.`gia` AS `price`, kv.`phu thu` AS `additionalCharge`
    FROM `May Tinh` mt
    JOIN `Cau Hinh` ch ON mt.`id cau hinh` = ch.ID
    JOIN `Khu Vuc` kv ON mt.`phan loai khu vuc` = kv.`loai khu vuc`
    WHERE mt.ID = may_tinh_id;
END //

DELIMITER //
```

#### 4.3.3 Chỉnh sửa thông số máy tính

##### 4.3.3.a Mô tả

Cho phép chỉnh sửa thông tin máy tính



Hình 28: *Chỉnh sửa thông tin máy tính*

#### 4.3.3.b Thủ tục

```
drop procedure if exists updateComputerInfo;
DELIMITER //
CREATE PROCEDURE updateComputerInfo(
    IN computerID INT,
    IN newHang CHAR(255),
    IN newNgayMua DATE,
    IN newPhanLoaiKhuVuc CHAR(255),
    IN newIdCauHinh INT
)
BEGIN
    UPDATE `May Tinh`
    SET
        hang = newHang,
        `ngay mua` = newNgayMua,
        `phan loai khu vuc` = newPhanLoaiKhuVuc,
        `id cau hinh` = newIdCauHinh
    WHERE ID = computerID;
END //
DELIMITER ;
```


#### 4.3.4 Thêm máy tính

##### 4.3.4.a Mô tả

Tính năng cho phép admin thêm máy tính vào hệ thống



THÊM MỚI MÁY TÍNH



Hãng

Khu vực

Ngày mua

ID Cấu hình

Hủy Đặt lại Thêm máy tính

Hình 29: Thêm máy tính

#### 4.3.4.b Thủ tục

```

drop procedure if exists addComputer;
DELIMITER //
CREATE PROCEDURE addComputer(
    IN newHang CHAR(255),
    IN newNgayMua DATE,
    IN newPhanLoaiKhuVuc CHAR(255),
    IN newIdCauHinh INT
)
BEGIN
    insert into `May Tinh` (hang, `ngay mua`, `phan loai khu vuc`, `id cau hinh`)
    values (newHang, newNgayMua, newPhanLoaiKhuVuc, newIdCauHinh);
END //
DELIMITER ;

```

#### 4.3.5 Thống kê sản phẩm khuyến mãi

##### 4.3.5.a Mô tả

Cho phép thống kê sản phẩm theo mã khuyến mãi

**Thống kê sản phẩm đã bán theo khuyến mãi**

Chọn khuyến mãi để thống kê:

202330 - happy net

Mã sản phẩm	Tên sản phẩm	Số lượng đã áp dụng khuyến mãi
2	mi tom	1
3	card viettel 50k	1
3	card viettel 50k	10
2	mi tom	1
3	card viettel 50k	5
2	mi tom	10

Trang: 1

Hình 30: Thống kê sản phẩm theo mã khuyến mãi

#### 4.3.5.b Thủ tục

```
DELIMITER //
```

```
CREATE PROCEDURE thôngKeSPKhuyenMai(  
    in ma_khuyen_mai int  
)  
BEGIN  
    Select `ma san pham`, `ten san pham`, `so luong`  
    from `san pham` as sp natural join  
    (Select ct.`ma san pham`, ct.`so luong`  
    from `hoa don` as hd natural join  
    (SELECT *  
    FROM chua as c natural join  
    (SELECT km.`ngay bat dau`, km.`ngay ket thuc`,adsp.`ma san pham`  
    from `khuyen mai` AS km  
    natural join `ap dung san pham` as adsp  
    where km.`ma khuyen mai`= ma_khuyen_mai and km.`ma khuyen mai` = adsp.`ma  
    khuyen mai`) as spkm  
    where spkm.`ma san pham` = c.`ma san pham`) AS ct  
    where hd.`ma hoa don` = ct.`ma hoa don`  
    and hd.`ngay thuc hien` >= ct.`ngay bat dau`  
    and hd.`ngay thuc hien` <= ct.`ngay ket thuc`) as kq  
    where sp.`ma san pham` = kq.`ma san pham`;  
END;  
//  
DELIMITER ;
```