

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

BÁO CÁO BÀI TẬP LỚN

**HỌC PHẦN THIẾT KẾ CƠ SỞ DỮ LIỆU
NĂM 2020**

ĐỀ TÀI: XÂY DỰNG CƠ SỞ DỮ LIỆU QUẢN LÝ QUÁN TRÀ SỮA

Người hướng dẫn: ThS NGUYỄN KIM SAO

Sinh viên thực hiện

NGUYỄN VĂN AN

Lớp: CNTT1-K59 Khoa: Công nghệ thông tin

LÊ QUANG DUY

Lớp: CNTT1-K59 Khoa: Công nghệ thông tin

MỤC LỤC

I. Lời nói đầu:.....	4
II. Khảo sát.....	5
1. Bài toán đặt ra:.....	5
2. Mô tả cơ sở thực tế:	5
III. Phân tích và thiết kế	10
1. Phân tích:	10
2. Thiết kế:.....	10
IV. Xây dựng CSDL trên SQL Server	13
1. Tạo CSDL:.....	13
2. Tạo các view:.....	15
3. Tạo các thủ tục:.....	16
4. Tạo trigger	17
5. Tạo hàm	18
V. Xây dựng chương trình kết nối	20

DANH MỤC HÌNH VẼ, SƠ ĐỒ:

Video khảo sát 1	10
Sơ đồ thực thể liên kết.....	11
Sơ đồ Diagram.....	12

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI

THÔNG TIN KẾT QUẢ BÀI TẬP LỚN

1. Thông tin chung:

- Tên đề tài: Thiết kế cơ sở dữ liệu quản lý quán trà sữa
- Sinh viên thực hiện: Nguyễn Văn An, Lê Quang Duy
- Người hướng dẫn: ThS. NGUYỄN KIM SAO

2. Mục tiêu đề tài:

- Xây dựng cơ sở dữ liệu đáp ứng những nhu cầu cơ bản và quan trọng của một quán trà sữa hoạt động.

3. Kết quả:

- Cơ sở dữ liệu có đầy đủ thông tin cần thiết để quản lý một quán trà sữa cơ bản, thao tác truy suất dữ liệu thuận tiện, đạt chuẩn 3NF.

Ngày tháng năm
Sinh viên chịu trách nhiệm thực hiện đề tài

(ký, họ và tên)

Nhận xét của người hướng dẫn *(phần này do người hướng dẫn ghi):*

Ngày tháng năm
Người hướng dẫn
(ký, họ và tên)

I. Lời nói đầu:

Trong thời đại công nghệ thông hiện nay, vấn đề đang thức thời đang thịnh hành là chuyển đổi số, tích hợp công nghệ và kỹ thuật số vào hoạt động kinh doanh của tổ chức, nhằm gia tăng hiệu quả vận hành, nâng cao trải nghiệm và làm hài lòng khách hàng hơn nữa tạo được lợi thế cạnh tranh trên thị trường.

Một mô hình doanh sẽ đạt được hiệu quả cao hơn nếu áp dụng công nghệ vào quản lý, công dụng chính để giảm thiểu chi phí, cải thiện chiến lược khách hàng, hệ thống vận hành, phân tích và bảo mật tốt hơn, tập trung hơn vào khách hàng tiềm năng, tăng sự đổi mới trong nhân sự.

Vì thế, áp dụng công nghệ để quản lý kinh doanh đã gần như là điều kiện tất yếu khi bắt đầu một chiến lược kinh doanh cụ thể.

Quản lý dữ liệu sẽ trở nên phức tạp và khó thực hiện nếu không lên kế hoạch, định trước các mô hình của bài toán.

Với những thực tế như vậy, để nâng cao khả năng quản lý, thiết kế cơ sở dữ liệu, chúng em quyết định chọn đề tài Quản lý quán trà sữa, là một đề tài dễ gần với sinh viên, mang tính thực tế, khả dụng cao.

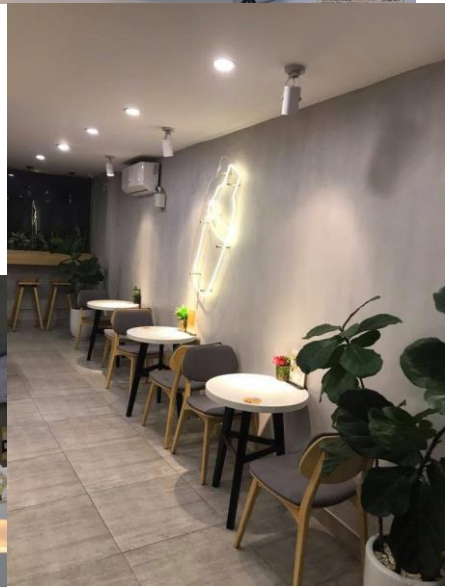
II. Khảo sát

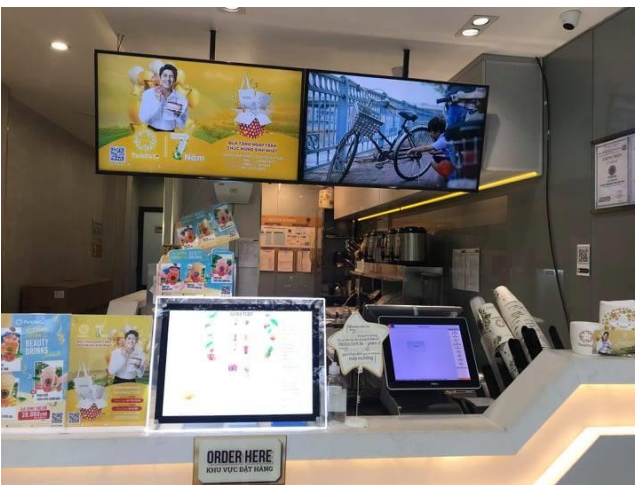
1. Bài toán đặt ra:

Để quản lý kinh doanh một quán trà sữa ngoài quản lý về hóa đơn bán, còn phải quản lý nguồn nhiên liệu còn dư, quản lý nhân viên, quản lý bàn còn trống hiện tại.

2. Mô tả cơ sở thực tế:

Sau khi đặt ra bài toán, nhóm quyết định tiến hành khảo sát một cơ sở thực tế, xin phép được dấu tên như yêu cầu của cơ sở khảo sát.







Dưới đây là video mô tả lại ứng dụng trà sữa hiện đang được dùng ở quán:



Figure 1 Video khảo sát 1

III. Phân tích và thiết kế

1. Phân tích:

Với những hình ảnh khảo sát ở trên và bài toán ban đầu, trước tiên nhóm xác định các thực thể liên kết cần thiết để quản lý quán trà sữa là:

- + Hóa đơn: hiển thị các mặt hàng đã mua, số lượng, tổng tiền sau khi giảm giá.
- + Chi tiết hóa đơn: hiển thị từng mặt hàng đã mua, vì trà sữa thường đi kèm topping nên khi mua trà sữa có thể mua topping hoặc không mua topping, mỗi một cốc trà sữa có thể có nhiều topping.
- + Trà sữa: thông tin chi tiết về một loại trà sữa, như giá bán, thành phần
- + Topping: tương tự như trà sữa, bao gồm giá bán cho từng loại topping.
- + Nhân viên: lưu thông tin của nhân viên để xác thực đăng nhập, giao dịch với cơ sở dữ liệu sau này.
- + Người dùng: lưu thông tin chung của người dùng, để tiến hành thống kê, áp dụng điều kiện khuyến mại sau này.
- + Kho: quản lý nguyên liệu làm nên trà sữa và topping.
- + Bàn: quản lý bàn còn trống, bàn đang có người theo thời gian thực

2. Thiết kế:

Với những thực thể ban đầu như trên, nhóm đã thảo luận, thống nhất mối quan hệ giữa các thực thể để đưa ra được sơ đồ thực thể liên kết cuối cùng như sau:

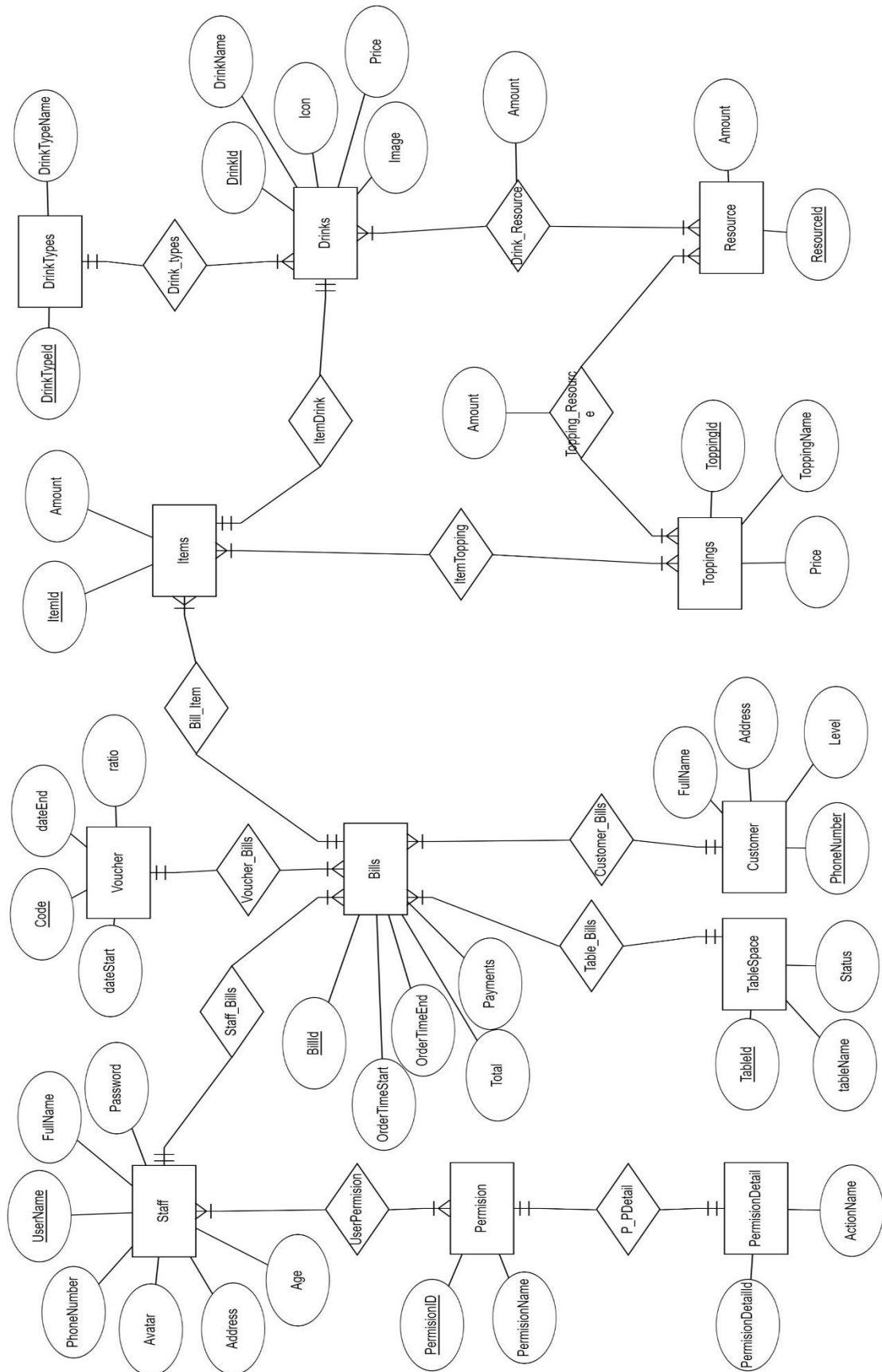


Figure 2 Sơ đồ thực thể liên kết

Còn đây là biểu đồ Diagram sau khi chuẩn hóa dữ liệu:



Figure 3 Sơ đồ Diagram

IV. Xây dựng CSDL trên SQL Server

Dựa trên các yêu cầu khảo sát được, dưới đây là mã SQL để tạo cơ sở dữ liệu như trên trong SQL Server:

1. Tạo CSDL:

```

1. CREATE DATABASE tea01
2.
3. use tea01
4.
5. CREATE TABLE DrinkTypes
6. (
7.     DrinkTypeId VARCHAR(10) NOT NULL,
8.     DrinkTypeName VARCHAR(100) DEFAULT NULL,
9.     PRIMARY KEY (DrinkTypeId)
10.);
11.
12. CREATE TABLE Toppings
13. (
14.     ToppingId VARCHAR(10) NOT NULL,
15.     ToppingName VARCHAR(10) DEFAULT NULL,
16.     Price INT DEFAULT NULL,
17.     PRIMARY KEY (ToppingId)
18.);
19.
20. CREATE TABLE Resources
21. (
22.     ResourceId VARCHAR(10) NOT NULL,
23.     ResourceName VARCHAR(100) DEFAULT NULL,
24.     Amount INT DEFAULT NULL,
25.     PRIMARY KEY (ResourceId)
26.);
27.
28. CREATE TABLE ToppingResource
29. (
30.     ResourceId VARCHAR(10) NOT NULL,
31.     ToppingId VARCHAR(10) NOT NULL,
32.     FOREIGN KEY (ResourceId) REFERENCES Resources(ResourceId),
33.     FOREIGN KEY (ToppingId) REFERENCES Toppings(ToppingId)
34.);
35.
36. CREATE TABLE Staff
37. (
38.     UserName VARCHAR(50) NOT NULL,
39.     Password VARCHAR(100) DEFAULT NULL,
40.     FullName VARCHAR(100) DEFAULT NULL,
41.     Age INT DEFAULT NULL,
42.     PhoneNumber VARCHAR(10) DEFAULT NULL,
43.     Address VARCHAR(100) DEFAULT NULL,
44.     Avatar VARCHAR(100) DEFAULT NULL,
45.     PRIMARY KEY (UserName)
46.);
47.
48. CREATE TABLE Permission
49. (
50.     PermissionID VARCHAR(10) NOT NULL,
51.     PermissionName VARCHAR(100) DEFAULT NULL,
52.     PRIMARY KEY (PermissionID)

```



```

53. );
54.
55. CREATE TABLE UserPermission
56. (
57.     UserName VARCHAR(50) NOT NULL,
58.     PermissionID VARCHAR(10) NOT NULL,
59.     FOREIGN KEY (UserName) REFERENCES Staff(UserName),
60.     FOREIGN KEY (PermissionID) REFERENCES Permission(PermissionID)
61. );
62.
63. CREATE TABLE PermissionDetail
64. (
65.     PermissionDetailId VARCHAR(10) NOT NULL,
66.     ActionName VARCHAR DEFAULT NULL,
67.     PermissionID VARCHAR(10) NOT NULL,
68.     PRIMARY KEY (PermissionDetailId),
69.     FOREIGN KEY (PermissionID) REFERENCES Permission(PermissionID)
70. );
71.
72. CREATE TABLE Customer
73. (
74.     PhoneNumber VARCHAR(10) NOT NULL,
75.     FullName VARCHAR(100) DEFAULT NULL,
76.     Address VARCHAR(100) DEFAULT NULL,
77.     Level INT DEFAULT NULL,
78.     PRIMARY KEY (PhoneNumber)
79. );
80.
81. CREATE TABLE TableSpace
82. (
83.     TableId VARCHAR(10) NOT NULL,
84.     TableName VARCHAR(100) DEFAULT NULL,
85.     Status INT DEFAULT NULL,
86.     PRIMARY KEY (TableId)
87. );
88.
89. CREATE TABLE Bills
90. (
91.     BillId VARCHAR(10) NOT NULL,
92.     OrderTimeStart DATE DEFAULT NULL,
93.     Payments VARCHAR(10) DEFAULT NULL,
94.     TableId VARCHAR(10) DEFAULT NULL,
95.     Total INT DEFAULT NULL,
96.     OrderTimeEnd DATE DEFAULT NULL,
97.     UserName VARCHAR(50) DEFAULT NULL,
98.     PhoneNumber VARCHAR(10) DEFAULT NULL,
99.     PRIMARY KEY (BillId),
100.     FOREIGN KEY (UserName) REFERENCES Staff(UserName),
101.     FOREIGN KEY (PhoneNumber) REFERENCES Customer(PhoneNumber),
102.     FOREIGN KEY (TableId) REFERENCES TableSpace(TableId)
103. );
104.
105. CREATE TABLE Drinks
106. (
107.     DrinkId VARCHAR(10) NOT NULL,
108.     DrinkName VARCHAR(100) DEFAULT NULL,
109.     Price INT DEFAULT NULL,
110.     Image VARCHAR(100) DEFAULT NULL,
111.     Icon VARCHAR(100) DEFAULT NULL,
112.     DrinkTypeId VARCHAR(10) DEFAULT NULL,
113.     PRIMARY KEY (DrinkId),
114.     FOREIGN KEY (DrinkTypeId) REFERENCES DrinkTypes(DrinkTypeId)
115. );

```

```

116.
117.     CREATE TABLE DrinkTopping
118.     (
119.         DrinkId VARCHAR(10) NOT NULL,
120.         ToppingId VARCHAR(10) NOT NULL,
121.         FOREIGN KEY (DrinkId) REFERENCES Drinks(DrinkId),
122.         FOREIGN KEY (ToppingId) REFERENCES Toppings(ToppingId)
123.     );
124.
125.
126.     CREATE TABLE DrinkResource
127.     (
128.         ResourceId VARCHAR(10) NOT NULL,
129.         DrinkId VARCHAR(10) NOT NULL,
130.         FOREIGN KEY (ResourceId) REFERENCES Resources(ResourceId),
131.         FOREIGN KEY (DrinkId) REFERENCES Drinks(DrinkId)
132.     );
133.
134.     CREATE TABLE Items
135.     (
136.         Amount INT DEFAULT NULL,
137.         BillId VARCHAR(10) NOT NULL,
138.         DrinkId VARCHAR(10) NOT NULL,
139.         FOREIGN KEY (BillId) REFERENCES Bills(BillId),
140.         FOREIGN KEY (DrinkId) REFERENCES Drinks(DrinkId)
141.     );

```

2. Tạo các view:

2.1. Tạo view các hóa đơn bán trong ngày hôm nay:

```

1. create view View01_HoaDonBanTrongNgay as
2. select BillId, UserName, PhoneNumber, CodeVoucher, Total, Payments, TableId, OrderTime
   Start, OrderTimeEnd from Bills
3. where DAY(OrderTimeStart) = Day(getDate()) and YEAR(OrderTimeStart) = YEAR(GETDATE()) a
   nd MONTH(OrderTimeStart) = MONTH(GETDATE())

```

2.2. Tạo view top 3 nhân viên có số lượng hóa đơn cao nhất tháng vừa qua (nếu số lượng hóa đơn bằng nhau thì sẽ so sánh bằng tổng tiền của các hóa đơn)

```

1. create view View02_Top3NhanVien as
2. select top(3) Bills.UserName, COUNT(BillId) as SoHoaDon, Sum(Total) as Tong from Staff
   inner join Bills on Bills.UserName = Staff.UserName
3. group by Bills.UserName
4. order by SoHoaDon, Tong desc

```

2.3. Tạo view quán sẽ tặng voucher cho 5 khách hàng mua nhiều nhất trong tháng trước (tổng mua >=500000), xuất ra thông tin của khách hàng này

```

1. create view View03_Top5KhachHang as
2. select Top(5) Bills.PhoneNumber, FullName, SUM(Total) as Tong from Customer inner join
   Bills on Bills.PhoneNumber = Customer.PhoneNumber
3. group by Bills.PhoneNumber,FullName
4. having SUM(Total) >= 500000
5. order by Tong desc

```

2.4. Thống kê tổng tiền các tháng trong năm hiện tại

```

1. create view View04_ThongKeTongTienHangThang as
2. SELECT 'TongThu' AS TongThuTheoThang,
3. [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]
4. FROM
5. (SELECT MONTH(OrderTimeStart) as Thang, Total
6. FROM Bills) AS BangNguon
7. PIVOT
8. (
9. SUM(Total)
10. FOR Thang IN ([1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12])
11. ) AS BangChuyen;

```

3. Tạo các thủ tục:

3.1. Tạo thủ tục thống kê số lượng bán trà, topping theo ngày hôm nay thời gian thực, không cần truyền tham số

```

1. create proc proc01_ThongKe as
2. begin
3.     select ISNULL(Table1.DrinkId, '') as DrinkID, ISNULL(DrinkName, '') as DrinkName, ISN
4.     ULL(SLBanD, '') as SLBanDrink, ISNULL(Table2.ToppingId, '') as ToppingId, ISNULL(ToppingN
5.     ame, '') as ToppingName, ISNULL(SLBanT, '') as SLBanTopping
6.     from (select ROW_NUMBER() OVER (ORDER BY d1.DrinkId) AS [STT], d1.DrinkId, DrinkNam
7.     e, count(d1.DrinkId) as SLBanD from (Bills b1 inner join Items i1 on b1.BillId = i1.Bil
8.     lId) full join Drinks d1 on d1.DrinkId = i1.DrinkId WHERE DAY(OrderTimeStart) = DAY(GE
9.     TDATE()) and MONTH(OrderTimeStart) = MONTH(GETDATE()) and YEAR(OrderTimeStart) = YEAR(G
10.    ETDATE()) group by d1.DrinkId, DrinkName) Table1
11.    full join
12.    ( select ROW_NUMBER() OVER (ORDER BY t2.ToppingId) AS [STT], t2.ToppingId, ToppingN
13.    ame, count(t2.ToppingId) as SLBanT from ((Bills b2 inner join Items i2 on b2.BillId =
14.    i2.BillId) inner join ItemTopping it2 on it2.ItemId = i2.ItemId) full join Toppings t2
15.    on t2.ToppingId = it2.ToppingId WHERE DAY(OrderTimeStart) = DAY(GETDATE()) and MONTH(O
16.    rderTimeStart) = MONTH(GETDATE()) and YEAR(OrderTimeStart) = YEAR(GETDATE()) group by t
17.    2.ToppingId, ToppingName ) Table2 on Table1.STT = Table2.STT
18. end

```

3.2. Tạo thủ tục hiển thị các nhân viên đã xuất hóa đơn ngày hôm nay

```

1. CREATE PROC staffDay AS
2. BEGIN
3.     SELECT * FROM Bills INNER JOIN Staff ON Bills.UserName = Staff.UserName WHERE DAY(B
4.     ills.OrderTimeStart) = DAY(GETDATE()) AND MONTH(Bills.OrderTimeStart) = MONTH(GETDATE()
5.     ) AND YEAR(Bills.OrderTimeStart) = YEAR(GETDATE())
6. END
7. EXEC staffDay

```

3.3. Tạo thủ tục nhập vào id trà in ra danh sách nguyên liệu, số lượng cần để làm thành trà

```

1. CREATE PROC teaResource @idTea VARCHAR(10) AS
2. BEGIN
3.     SELECT Resources.ResourceName, Resources.ResourceId FROM Drinks INNER JOIN Drink_Re
       source ON Drinks.DrinkId = Drink_Resource.DrinkId INNER JOIN Resources ON Resources.Res
       ourceId = Drink_Resource.ResourceId WHERE Drinks.DrinkId = @idTea
4. END
5.
6. DECLARE @idTea VARCHAR(10)
7. SET @idTea = '01'
8. EXEC dbo.teaResource @idTea

```

3.4. Tạo thủ tục in ra tất cả khả năng trà sữa kết hợp được với topping

```

1. CREATE PROC combiTeaTopping AS
2. BEGIN
3.     SELECT Drinks.DrinkName, Toppings.ToppingName FROM Drinks INNER JOIN Items ON Items.
       DrinkId = Drinks.DrinkId INNER JOIN ItemTopping ON ItemTopping.DrinkId = Drinks.DrinkId
       FULL OUTER JOIN Toppings ON ItemTopping.ToppingId = Toppings.ToppingId
4. END
5.
6. EXEC dbo.combiTeaTopping

```

4. Tạo trigger

4.1. Tạo trigger tự động cập nhật trạng thái bàn khi nhân viên tiến hành đặt hàng (chỉ được 4 người ngồi 1 bàn) :

```

1. CREATE TRIGGER updateTable ON Bills FOR INSERT,UPDATE,DELETE AS
2. BEGIN
3.     DECLARE @count int
4.     SELECT @count = ISNULL(COUNT(Bills.TableId),0) FROM Bills INNER JOIN TableSpace ON
       Bills.TableId = TableSpace.TableId WHERE Bills.OrderTimeEnd IS NULL GROUP BY TableSpace
       .TableId HAVING TableSpace.TableId = '1'
5.
6.     IF @count >= 4
7.         UPDATE TableSpace SET Status=0 WHERE TableId = (SELECT TableId FROM INSERTED)
8.     ELSE
9.         UPDATE TableSpace SET Status=1 WHERE TableId = (SELECT TableId FROM INSERTED)
10. END

```

4.2. Tạo trigger tự động cập nhật level cho khách mua hàng khi tiến hành thanh toán thành công (cứ 500000 thì tăng level):

```

1. ALTER TRIGGER updateLevel ON Bills FOR INSERT,UPDATE,DELETE AS
2. BEGIN
3.     DECLARE @sum int
4.     SELECT @sum = SUM(Total) FROM Bills GROUP BY PhoneNumber HAVING PhoneNumber = (SELE
       CT PhoneNumber FROM inserted)

```

```

5.
6.     UPDATE Customer SET Level = @sum/500000 WHERE PhoneNumber = (SELECT PhoneNumber FROM
M inserted)
7. END

```

4.3. Tạo trigger cập nhật lại kho khi hóa đơn thanh toán thành công (có sử dụng 2 hàm mô tả ở dưới):

```

1. ALTER TRIGGER updateResource ON Items FOR INSERT,UPDATE AS
2. BEGIN
3.     DECLARE @idBill nvarchar(10)
4.     SELECT @idBill = BillId FROM INSERTED
5.     UPDATE Resources SET Amount = Amount - ISNULL(dbo.getAmountResourceDrink(ResourceId
,@idBill),0)
6.     UPDATE Resources SET Amount = Amount - ISNULL(dbo.getAmountResourceTopping(Resource
Id,@idBill),0)
7. END

```

4.4. Tạo trigger thực hiện việc xóa một hóa đơn, thay vì xóa luôn hóa đơn đó, ta sẽ xóa các bảng kết nối với hóa đơn đó trước

```

1. CREATE TRIGGER autoDeleteBill ON Bills INSTEAD OF Delete AS
2. BEGIN
3.     DECLARE @billId varchar(10)
4.     SELECT @billId=BillId FROM DELETED
5.
6.     DELETE ItemTopping WHERE ItemTopping.BillId = @billId
7.     DELETE Items WHERE Items.BillId = @billId
8.     DELETE Bills WHERE BillId = @billId
9. END

```

5. Tạo hàm

5.1. Tạo các hàm trả về số lượng tài nguyên cần dùng để tạo của một mã trà sữa, hoặc topping, phục vụ cho trigger 4.3:

```

1. CREATE FUNCTION getAmountResourceTopping(@idResource varchar(10), @idBill varchar(10))
RETURNS INT AS
2. BEGIN
3.     DECLARE @res int
4.     SELECT @res = ISNULL(Topping_Resource.Amount,0) FROM Items INNER JOIN ItemTopping O
N ItemTopping.ItemId = Items.ItemId INNER JOIN Toppings ON Toppings.ToppingId = ItemTop
ping.ToppingId INNER JOIN Topping_Resource ON Topping_Resource.ToppingId = Toppings.Top

```



```

pingId INNER JOIN Resources ON Resources.ResourceId = Topping_Resource.ResourceId WHERE
Items.BillId = @idBill AND Resources.ResourceId = @idResource
5.     RETURN @res
6. END
7.
8. CREATE FUNCTION getAmountResourceDrink(@idResource varchar(10), @idBill varchar(10)) RE
TURNS INT AS
9. BEGIN
10.    DECLARE @res int
11.    SELECT @res = ISNULL(Drink_Resource.Amount,0) FROM Bills INNER JOIN Items ON Items.
BillId = Bills.BillId INNER JOIN Drink_Resource ON Drink_Resource.DrinkId = Items.Drink
Id WHERE Items.BillId = @idBill AND Drink_Resource.ResourceId = @idResource
12.    RETURN @res
13. END

```

5.2. Tạo hàm kiểm tra xem một nhân viên có quyền Thêm (I), Xóa (D), Sửa (E), Xem (V) trong cơ sở dữ liệu hay không:

```

1. CREATE FUNCTION checkPer(@per varchar(1), @userName varchar(10)) RETURNS BIT AS
2. BEGIN
3.     DECLARE @res BIT
4.     DECLARE @id varchar(10)
5.     SELECT @id = Staff.UserName FROM Staff INNER JOIN UserPermission ON Staff.UserName =
UserPermission.UserName INNER JOIN Permission ON Permission.PermissionID = UserPermission.P
ermissionID INNER JOIN PermissionDetail ON PermissionDetail.PermissionID = Permission.Permis
ionID WHERE Staff.UserName = @userName AND PermissionDetail.ActionName = @per
6.
7.     IF @id = @userName
8.         SET @res = 1
9.     ELSE
10.        SET @res = 0
11.
12.    RETURN @res
13. END
14. -- Ví dụ
15. print dbo.checkPer('C','prochicken007')

```

5.3. Tạo hàm đưa ra tổng doanh thu các hóa đơn bán từ ngày bắt đầu đến ngày kết thúc

```

1. CREATE FUNCTION billCustom(@start VARCHAR(10), @end VARCHAR(10)) RETURNS INT AS
2. BEGIN
3.     DECLARE @startDate DATETIME
4.     DECLARE @endDate DATETIME
5.     DECLARE @res INT
6.
7.     SET @startDate = CONVERT(DATETIME,@start,103)
8.     SET @endDate = CONVERT(DATETIME,@end,103)
9.
10.    SELECT @res = SUM(Bills.Total) FROM Bills WHERE Bills.OrderTimeStart>=@startDate AN
D Bills.OrderTimeStart<=@endDate
11.
12.    RETURN @res

```

```

13. END
14.
15. print dbo.billCustom('26/11/2020', '27/11/2020')

```

5.4. Tạo hàm kiểm tra xem một bàn có những ai đang ngồi

```

1. CREATE FUNCTION checkTable(@idTable varchar(10)) RETURNS TABLE AS
2. RETURN (SELECT Customer.FullName FROM TableSpace INNER JOIN Bills ON Bills.TableId = Ta
   bleSpace.TableId INNER JOIN Customer ON Customer.PhoneNumber = Bills.PhoneNumber WHERE
   Bills.OrderTimeEnd IS NULL AND TableSpace.TableId = @idTable
3. )
4.
5. SELECT * FROM dbo.checkTable('1')

```

V. Xây dựng chương trình kết nối

Để thuận tiện cho việc kiểm thử và mô phỏng, chúng em xây dựng chương trình kết nối bằng ngôn ngữ C# Winform.

Dưới đây là một số mô tả về phần mềm:

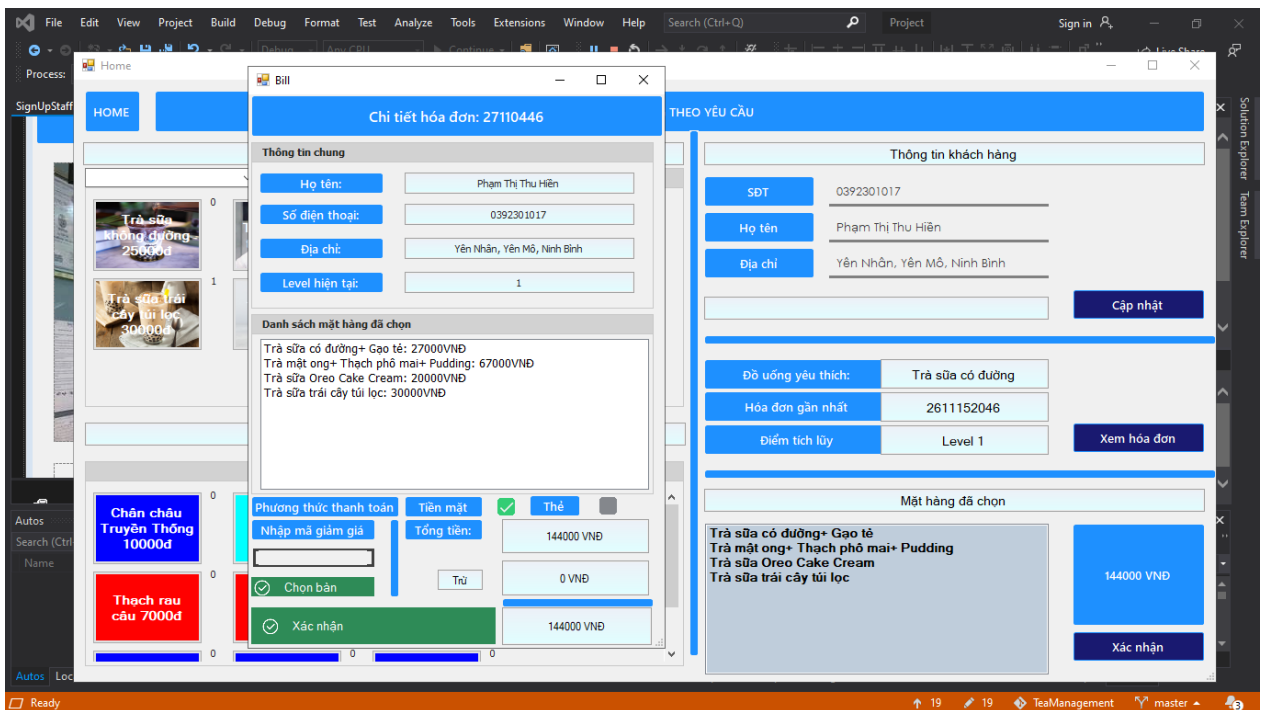


Figure 4 Giao diện chi tiết hóa đơn

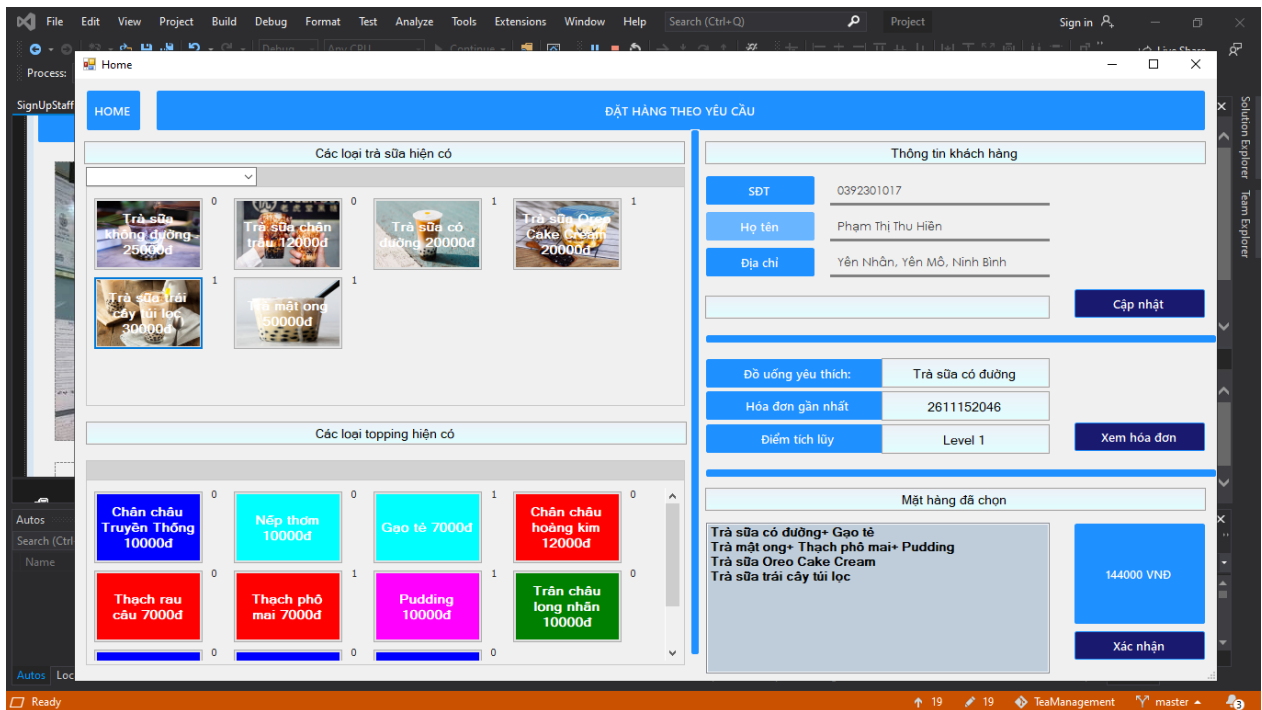


Figure 5 Giao diện chọn hóa đơn

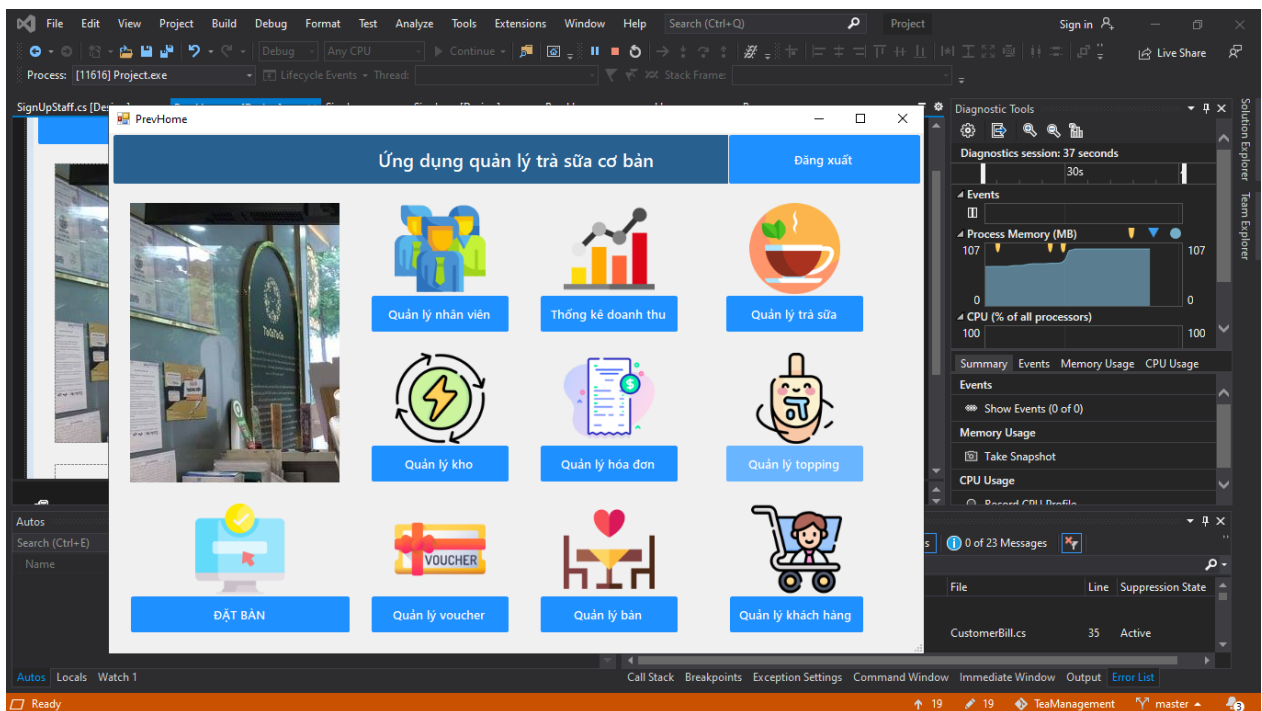


Figure 6 Giao diện khi đăng nhập vào ứng dụng

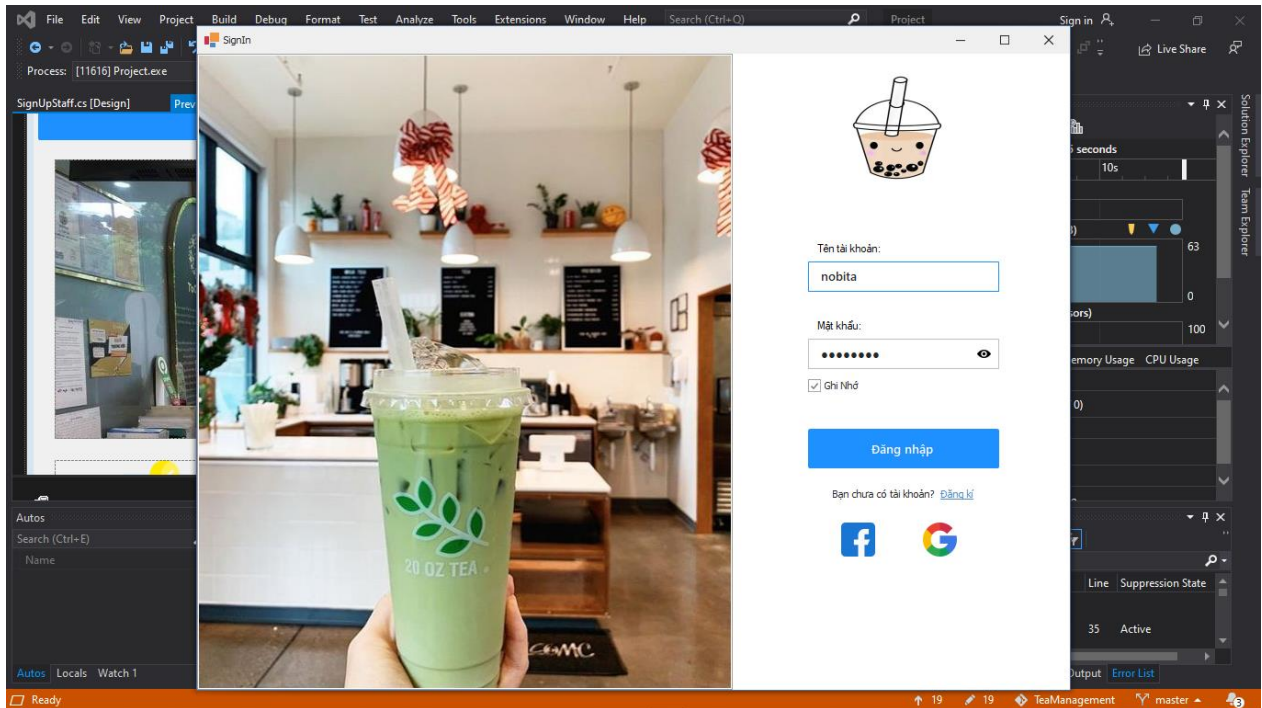


Figure 7 Giao diện đăng nhập ứng dụng

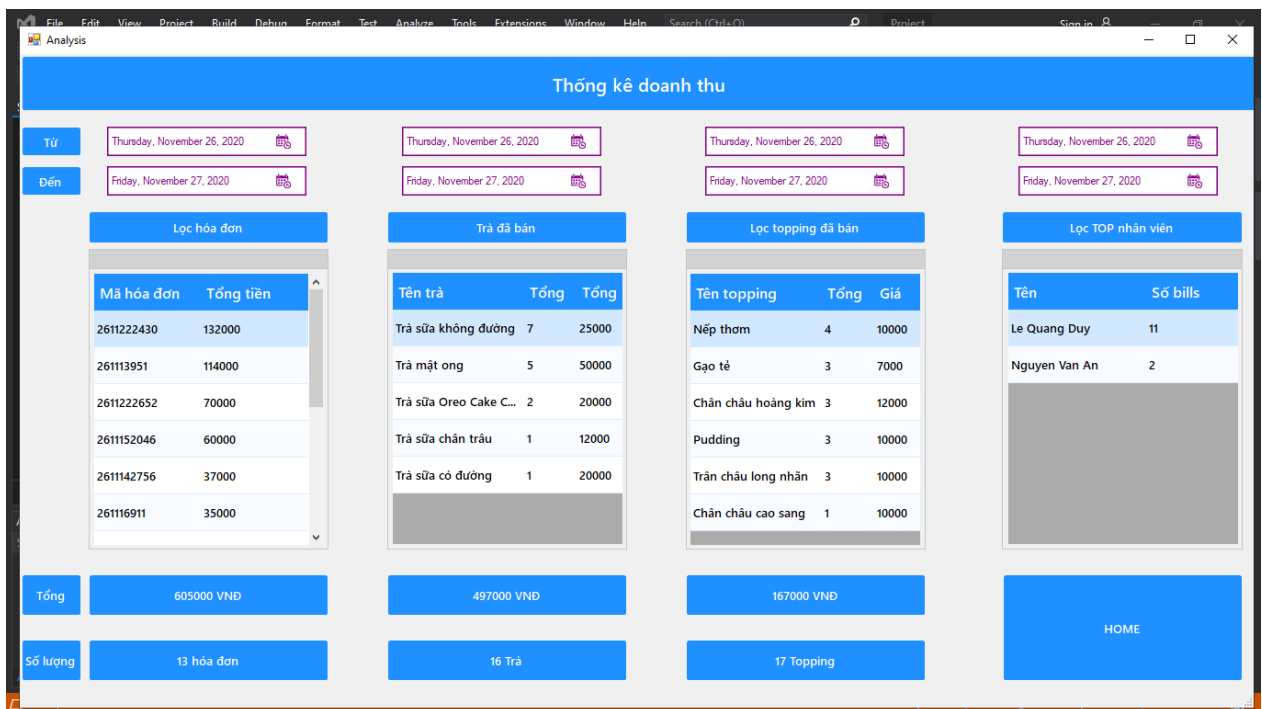


Figure 8 Thống kê doanh thu

VI. Tổng kết

Xây dựng cơ sở dữ liệu đáp ứng được nhu cầu truy xuất dữ liệu, quản lý tài nguyên, đã chuẩn hóa 3NF.

Xin chân thành cảm ơn