

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



HỆ CƠ SỞ DỮ LIỆU THÍ NGHIỆM (CO2014)

Hệ thống chăm sóc khách hàng cho chuỗi siêu thị

GVHD: Võ Thị Ngọc Châu

SV thực hiện:	Nguyễn Phúc Vinh	1915940 - Nhóm trưởng
	Vương Thanh Duyên	1912942
	Lê Xuân Nghĩa	1914310

Tp. Hồ Chí Minh, Tháng 10/2021

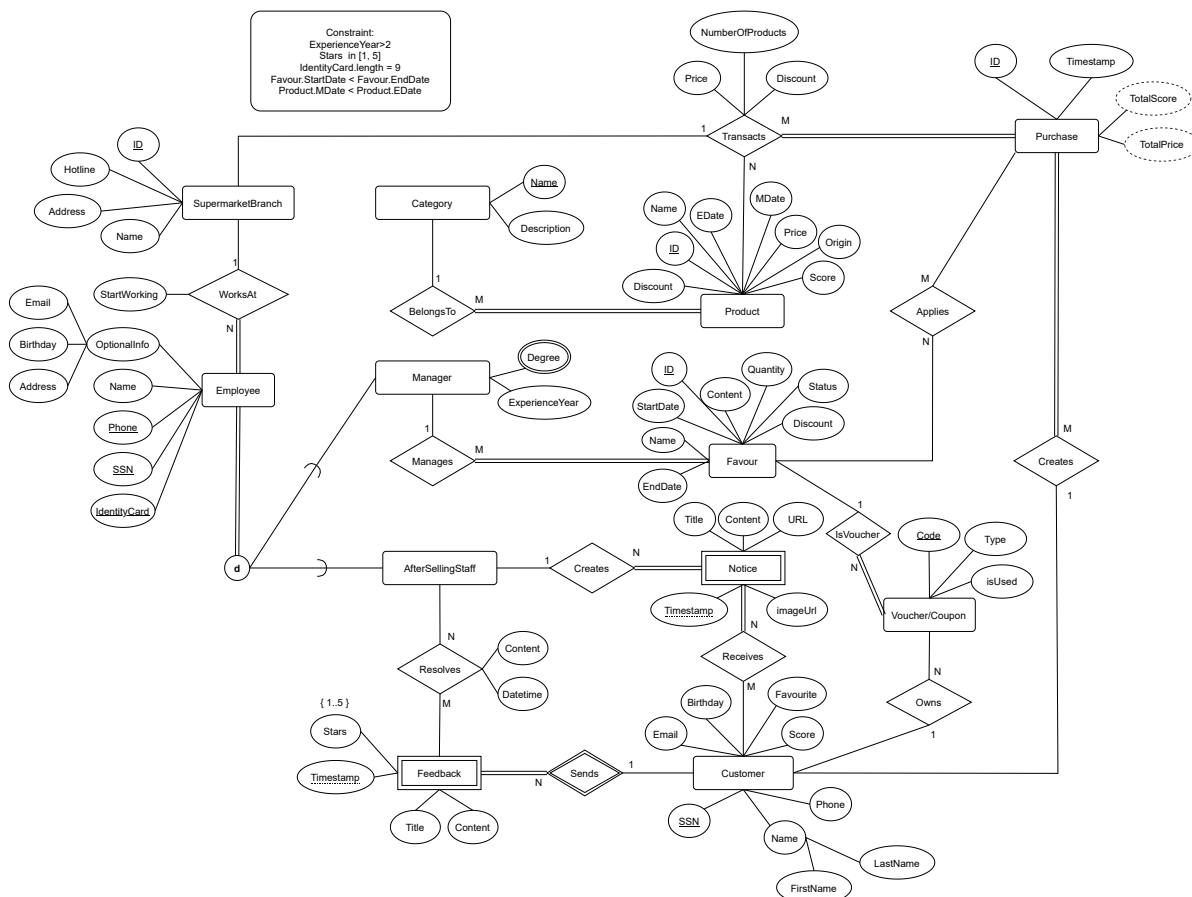
Mục lục

1	Thiết kế cơ sở dữ liệu ý niệm	2
2	Kiến trúc hệ thống	2
2.1	Sơ lược về kiến trúc phân lớp	3
2.2	Package Diagram	4
3	Triển khai cơ sở dữ liệu	5
3.1	Cài đặt các câu lệnh DDL	5
3.2	Sample Data	9
3.3	Cài đặt các câu lệnh DML	13
3.3.1	Trang thông tin cá nhân của khách hàng	13
3.3.2	Trang khách hàng gửi phản hồi	14
3.3.3	Trang xem lại lịch sử mua sắm của khách hàng	15
3.3.4	Kho lưu trữ ưu đãi cá nhân của khách hàng	17
3.3.5	Khách hàng xem trang thông báo	18
3.3.6	Nhân viên truy xuất thông tin của khách hàng	19
3.3.7	Nhân viên xem thông tin thống kê khách hàng	20
3.3.8	Nhân viên xem danh sách khách hàng	22
3.3.9	Nhân viên trả lời phản hồi của khách hàng	24
3.3.10	Nhân viên gửi thông báo cho khách hàng	26
3.3.11	Quản lý xem ưu đãi đang được áp dụng, đã quá hạn hoặc dự kiến áp dụng	27
3.3.12	Quản lý chi nhánh tạo ưu đãi mới	28
3.3.13	Quản lý chi nhánh cập nhật hoặc xóa ưu đãi	29
3.3.14	Quản lý chi nhánh xem danh sách sản phẩm	30
3.4	Cài đặt trigger	31
3.5	Kiểm tra chuẩn hóa cho cơ sở dữ liệu kết quả	32
3.5.1	Kiểu thực thể mạnh	32
3.5.2	Kiểu thực thể yếu	33
3.5.3	Mối quan hệ	34
4	Triển khai thiết kế và hiện thực tổng thể hệ thống cho các yêu cầu phi chức năng	35
4.1	Yêu cầu về I/O time < 2s của các câu truy vấn	35
4.1.1	Tạo index cho bảng CUSTOMER	35
4.1.2	Tạo index cho bảng PRODUCT	35
4.2	Bảo mật thông tin của khách hàng, nhân viên và quản lý	35
4.2.1	Cấp quyền cho nhóm đối tượng khách hàng	36
4.2.2	Cấp quyền cho nhóm đối tượng Nhân viên	36
4.2.3	Cấp quyền cho nhóm đối tượng Quản lý	37
4.2.4	Cập nhật cài đặt	37
5	Tổng kết	38
5.1	Phân chia khối lượng công việc	38
5.2	Tự đánh giá	38
5.2.1	Điểm đạt được	38
5.2.2	Điểm chưa đạt	38
	Repository	38

1 Thiết kế cơ sở dữ liệu ý niệm

Sau khi triển khai hệ thống, nhóm nhận thấy thiết kế lược đồ thực thể - mối liên kết (ERD) trong BTL 1 có một số điểm không còn phù hợp và hiệu quả.

Vì vậy nhóm hiệu chỉnh lại lược đồ ERD như hình 1 (phóng to để xem chi tiết)



Hình 1: Lược đồ ERD

Cách tính các thuộc tính dẫn xuất:

- Customer.Type: Được tính dựa trên Customer.Score.
 - Customer.Score < 5000 \Rightarrow Customer.Type = Bronze
 - $5000 \leq$ Customer.Score < 10000 \Rightarrow Customer.Type = Silver
 - Customer.Score \geq 10000 \Rightarrow Customer.Type = Gold
- Purchase.TotalScore: được tính bằng tổng Product.Score của các Product tham gia vào các mối quan hệ Transacts có cùng Purchase.ID
- Purchase.TotalPrice: được tính bằng tổng Product.Price của các Product tham gia vào các mối quan hệ Transacts có cùng Purchase.ID

2 Kiến trúc hệ thống

Nhóm lựa chọn mô hình kiến trúc phân lớp (layered architecture) cho hệ thống CSKH của chuỗi siêu thị. Kiến trúc gồm 3 lớp: Presentation Layer, Business Logic Layer và Data Layer và các lớp khác gồm Service Layer và

Cross-Cutting Layer.

2.1 Sơ lược về kiến trúc phân lớp

Trong một hệ thống phân lớp, một lớp có các đặc điểm sau:

- Phụ thuộc vào các lớp bên dưới
- Không can thiệp và không phụ thuộc vào các lớp trên

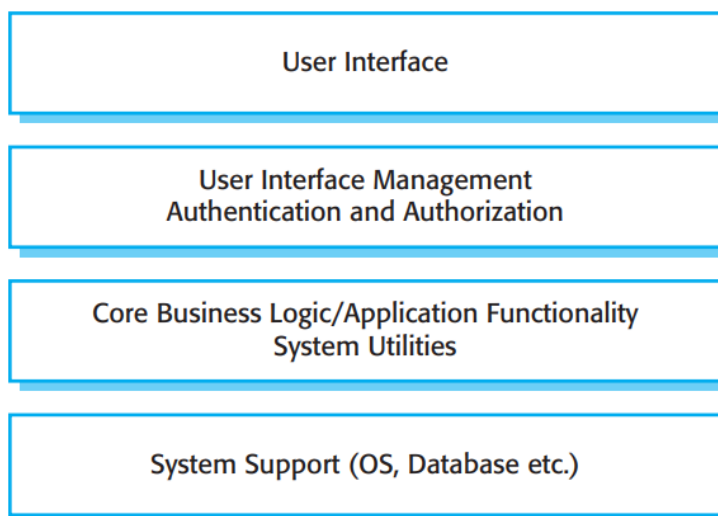
Các lớp được thiết kế theo hai kiểu:

- Một lớp chỉ có thể hiểu và sử dụng lớp ngay dưới nó
- Một lớp có thể sử dụng các layer song song (cross-cutting) và tất cả các lớp bên dưới. Kiểu thiết kế này có tên gọi là Anti-pattern: Lasagna Architecture, được sử dụng nhiều hơn kiểu thứ nhất, tránh trường hợp sử dụng lớp trung gian chỉ để truyền message dẫn đến giảm hiệu suất hệ thống.

Kiến trúc phân lớp gồm 3 lớp chính (hình 2):

- User Interface: Giao diện người dùng, nơi thao tác với ứng dụng
- Core Business Logic/Application Functionality/System Utilities: Nơi chứa các module xử lý logic nghiệp vụ của hệ thống, chuyên tiếp nhận các request từ client, xử lý và lưu trữ data thông qua Data source layer
- System Support (OS, Database etc.)

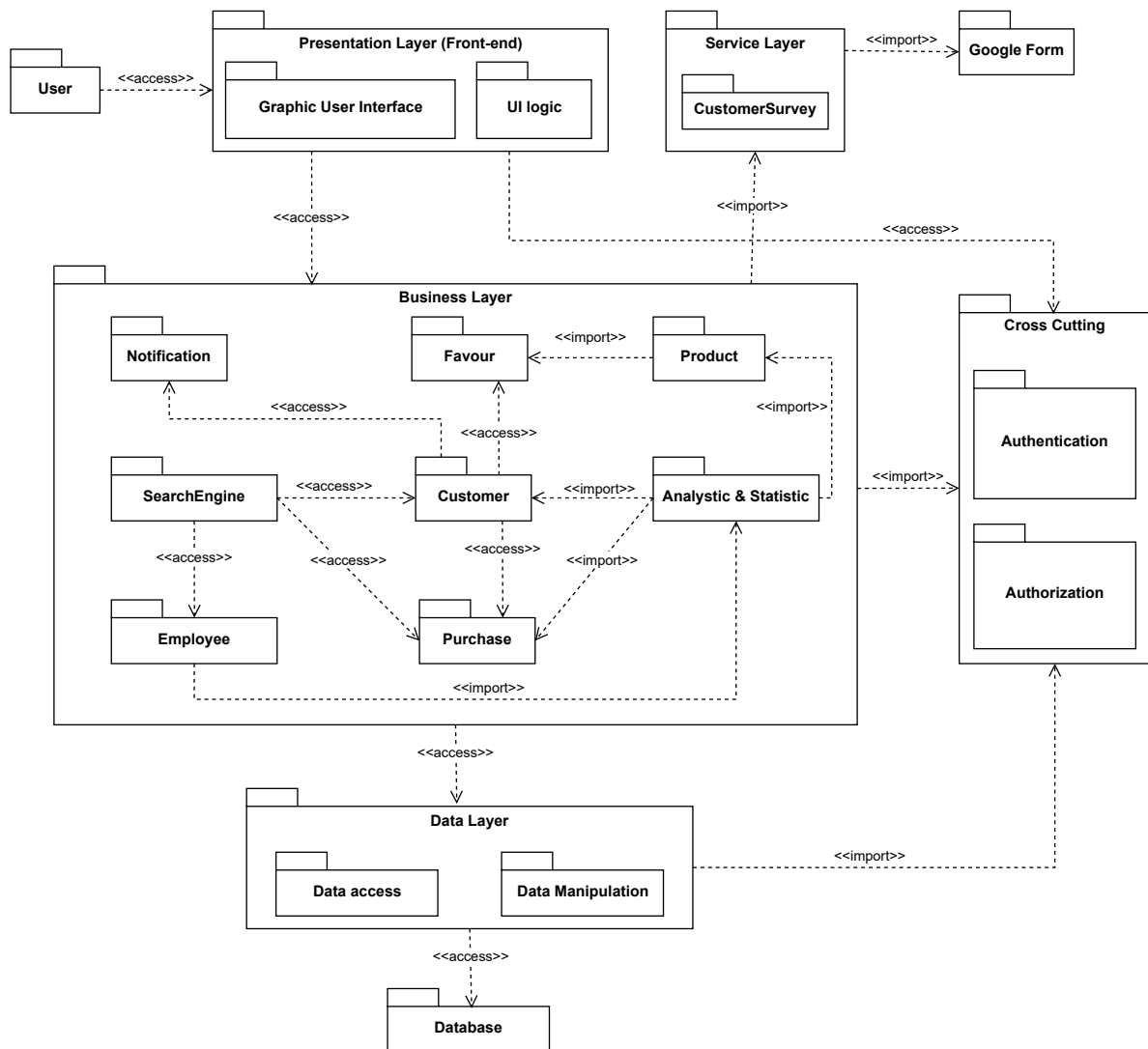
Kiến trúc phân lớp trong các ứng dụng có thể khác nhau, tùy vào thiết kế của nhà phát triển sao cho phù hợp hệ thống.



Hình 2: Kiến trúc phân lớp

2.2 Package Diagram

Chi tiết kiến trúc hệ thống của nhóm được thể hiện trong lược đồ gói (package diagram) như hình 3.



Hình 3: Kiến trúc hệ thống

User sẽ truy cập và điều khiển ứng dụng thông qua tầng Presentation Layer (chứa UI và các logic liên quan). Tầng Business Layer chứa các package xử lý nghiệp vụ hệ thống. Tầng Data Layer điều khiển các câu truy vấn, thao tác dữ liệu (CRUD). Các tầng khác gồm tầng Cross-Cutting Layer nhận nhiệm vụ xác thực và điều khiển truy cập tài khoản, tầng Service Layer sử dụng các dịch vụ/chức năng cung cấp từ bên thứ 3 để bổ sung chức năng cho app.

3 Triển khai cơ sở dữ liệu

Về hệ quản trị CSDL, nhóm đã chọn MySQL vì tính gọn nhẹ, phù hợp và tương thích với dòng máy sinh viên.
Về source code, nhóm sử dụng Repository [1] trên Github để lưu trữ và quản lý.

3.1 Cài đặt các câu lệnh DDL

Dựa trên kết quả của giai đoạn thiết kế cơ sở dữ liệu luận lý với hệ quản trị cơ sở dữ liệu đã được chọn, nhóm cài đặt các câu lệnh DDL sau:

```
1 DROP DATABASE IF EXISTS SUPERMARKET_CRM;  
2 CREATE DATABASE IF NOT EXISTS SUPERMARKET_CRM;  
3 USE SUPERMARKET_CRM;  
4  
5 CREATE TABLE IF NOT EXISTS SUPERMARKET_BRANCH (  
6     ID INT PRIMARY KEY AUTO_INCREMENT,  
7     hotline VARCHAR(20),  
8     `address` VARCHAR(100),  
9     `name` VARCHAR(100)  
10 );  
11  
12 CREATE TABLE IF NOT EXISTS EMPLOYEE (  
13     ssn INT PRIMARY KEY AUTO_INCREMENT,  
14     identityCard VARCHAR(9) NOT NULL UNIQUE,  
15     phone VARCHAR(20) NOT NULL UNIQUE,  
16     `name` VARCHAR(100),  
17     `address` VARCHAR(100),  
18     birthday DATE,  
19     email VARCHAR(100) UNIQUE,  
20     CHECK (LENGTH(identityCard) = 9)  
21 );  
22  
23 ALTER TABLE EMPLOYEE ADD (  
24     SBranchID INT NOT NULL,  
25     startWorkingDate DATE,  
26     FOREIGN KEY (SBranchID)  
27         REFERENCES SUPERMARKET_BRANCH(ID)  
28         ON DELETE CASCADE ON UPDATE CASCADE  
29 );  
30  
31 CREATE TABLE IF NOT EXISTS AFTER_SELLING_STAFF (  
32     ssn INT PRIMARY KEY,  
33     FOREIGN KEY (ssn)  
34         REFERENCES EMPLOYEE(ssn)  
35         ON DELETE CASCADE ON UPDATE CASCADE  
36 );  
37  
38 CREATE TABLE IF NOT EXISTS MANAGER (  
39     ssn INT PRIMARY KEY,  
40     experienceYear INT,  
41     FOREIGN KEY (ssn)  
42         REFERENCES EMPLOYEE (ssn)  
43         ON DELETE CASCADE ON UPDATE CASCADE,  
44     CHECK (experienceYear > 2)  
45 );  
46
```

```
47 CREATE TABLE IF NOT EXISTS CATEGORY (  
48     `name` VARCHAR(50) PRIMARY KEY,  
49     `description` TEXT  
50 );  
51  
52 CREATE TABLE IF NOT EXISTS PRODUCT (  
53     ID INT PRIMARY KEY AUTO_INCREMENT,  
54     `name` VARCHAR(100),  
55     mdate DATE,  
56     edate DATE,  
57     price FLOAT,  
58     origin VARCHAR(100),  
59     score INT,  
60     discount VARCHAR(20),  
61     CHECK (mdate < edate)  
62 );  
63  
64 ALTER TABLE PRODUCT ADD (  
65     categoryName VARCHAR(50),  
66     FOREIGN KEY (categoryName) REFERENCES CATEGORY(`name`)  
67     ON DELETE SET NULL ON UPDATE CASCADE  
68 );  
69  
70 CREATE TABLE IF NOT EXISTS FAVOUR(  
71     ID INT PRIMARY KEY AUTO_INCREMENT,  
72     `name` VARCHAR(100),  
73     content VARCHAR(100),  
74     startDate DATE,  
75     endDate DATE,  
76     quantity INT,  
77     `status` ENUM('terminated', 'applying', 'planning'),  
78     CHECK (startDate < endDate)  
79 );  
80  
81 ALTER TABLE FAVOUR ADD (  
82     mssn INT,  
83     FOREIGN KEY (mssn)  
84     REFERENCES MANAGER(ssn)  
85     ON DELETE SET NULL ON UPDATE CASCADE  
86 );  
87  
88 CREATE TABLE IF NOT EXISTS VOUCHER_COUPON (  
89     code INT PRIMARY KEY AUTO_INCREMENT,  
90     `type` ENUM('voucher', 'coupon'),  
91     isUsed ENUM('n', 'y')  
92 );  
93  
94 ALTER TABLE VOUCHER_COUPON ADD (  
95     favourID INT NOT NULL,  
96     FOREIGN KEY (favourID)  
97     REFERENCES FAVOUR(ID)  
98     ON DELETE CASCADE ON UPDATE CASCADE  
99 );  
100  
101 CREATE TABLE IF NOT EXISTS CUSTOMER (  
102     ssn INT PRIMARY KEY AUTO_INCREMENT,  
103     fname VARCHAR(100),
```

```
104     lname VARCHAR(100),
105     phone VARCHAR(20),
106     email VARCHAR(100),
107     birthday DATE,
108     score INT,
109     favorite TEXT(2000),
110     imageUrl TEXT(1000)
111 );
112
113 CREATE TABLE IF NOT EXISTS PURCHASE (
114     ID INT PRIMARY KEY AUTO_INCREMENT,
115     `time` DATETIME NOT NULL,
116     totalPrice FLOAT,
117     totalScore INT
118 );
119
120 ALTER TABLE PURCHASE ADD (
121     cssn INT NOT NULL,
122     FOREIGN KEY (cssn)
123     REFERENCES CUSTOMER(ssn)
124     ON DELETE CASCADE ON UPDATE CASCADE
125 );
126
127 -- WEAK ENTITIES
128
129 CREATE TABLE IF NOT EXISTS FEEDBACK (
130     cssn INT,
131     `time` DATETIME NOT NULL,
132     `stars` INT,
133     title VARCHAR(100),
134     content TEXT(2000),
135     CHECK (stars BETWEEN 1 AND 5),
136     PRIMARY KEY(cssn, `time`),
137     FOREIGN KEY (cssn) REFERENCES CUSTOMER(ssn) ON DELETE CASCADE ON UPDATE
138     CASCADE
139 );
140
141 CREATE TABLE IF NOT EXISTS NOTICES (
142     ass_ssn INT,
143     `time` DATETIME NOT NULL,
144     content TEXT NOT NULL,
145     title VARCHAR(200) NOT NULL,
146     `url` TEXT(500),
147     imageUrl TEXT(500),
148     PRIMARY KEY (ass_ssn, `time`),
149     FOREIGN KEY (ass_ssn)
150     REFERENCES AFTER_SELLING_STAFF(ssn)
151     ON DELETE CASCADE ON UPDATE CASCADE
152 );
153
154 -- RELATIONSHIPS
155
156 CREATE TABLE IF NOT EXISTS RESOLVES (
157     ass_ssn INT,
158     cssn INT NOT NULL,
159     feedbackTime DATETIME NOT NULL,
160     resolveTime DATETIME NOT NULL,
```



```
160     content TEXT(2000),
161     PRIMARY KEY (ass_ssn , cssn, feedbackTime),
162     FOREIGN KEY (ass_ssn)
163         REFERENCES AFTER_SELLING_STAFF(ssn)
164         ON DELETE CASCADE ON UPDATE CASCADE,
165     FOREIGN KEY (cssn , feedbackTime)
166         REFERENCES FEEDBACK(cssn , `time`)
167         ON DELETE CASCADE ON UPDATE CASCADE
168 );
169
170 CREATE TABLE IF NOT EXISTS APPLY_FOR_PURCHASE (
171     favourID INT,
172     purchaseID INT,
173     discount VARCHAR(20),
174     PRIMARY KEY (favourID , purchaseID),
175     FOREIGN KEY (favourID)
176         REFERENCES FAVOUR(ID)
177         ON DELETE CASCADE ON UPDATE CASCADE,
178     FOREIGN KEY (purchaseID)
179         REFERENCES PURCHASE(ID)
180         ON DELETE CASCADE ON UPDATE CASCADE
181 );
182
183 CREATE TABLE IF NOT EXISTS TRANSACTS (
184     productID INT,
185     purchaseID INT,
186     SBranchID INT NOT NULL,
187     numberOfProducts INT,
188     price FLOAT,
189     score INT,
190     discount VARCHAR(20),
191     PRIMARY KEY (productID, purchaseID),
192     FOREIGN KEY (productID)
193         REFERENCES PRODUCT(ID)
194         ON DELETE CASCADE ON UPDATE CASCADE,
195     FOREIGN KEY (purchaseID)
196         REFERENCES PURCHASE(ID)
197         ON DELETE CASCADE ON UPDATE CASCADE,
198     FOREIGN KEY (SBranchID)
199         REFERENCES SUPERMARKET_BRANCH(ID)
200         ON DELETE CASCADE ON UPDATE CASCADE
201 );
202
203 CREATE TABLE IF NOT EXISTS RECEIVES (
204     ass_ssn INT,
205     `time` DATETIME,
206     cssn INT,
207     PRIMARY KEY (ass_ssn, `time`, cssn),
208     FOREIGN KEY (ass_ssn, `time`)
209         REFERENCES NOTICES(ass_ssn, `time`)
210         ON DELETE CASCADE ON UPDATE CASCADE,
211     FOREIGN KEY (cssn)
212         REFERENCES CUSTOMER(ssn)
213         ON DELETE CASCADE ON UPDATE CASCADE
214 );
215
216 CREATE TABLE IF NOT EXISTS OWNS (
```

```
217     vcode INT PRIMARY KEY,
218     cssn INT NOT NULL,
219     FOREIGN KEY (vcode)
220         REFERENCES VOUCHER_COUPON (code)
221         ON DELETE CASCADE ON UPDATE CASCADE,
222     FOREIGN KEY (cssn)
223         REFERENCES CUSTOMER (ssn)
224         ON DELETE CASCADE ON UPDATE CASCADE
225 );
226
227 -- MULTIVALUED ATTRIBUTE
228
229 CREATE TABLE IF NOT EXISTS DEGREE (
230     mssn INT,
231     degreeName VARCHAR(250) NOT NULL,
232     PRIMARY KEY (mssn, degreeName),
233     FOREIGN KEY (mssn) REFERENCES MANAGER (ssn)
234 );
```

3.2 Sample Data

Nhóm sử dụng các câu lệnh INSERT sau để tạo dữ liệu mẫu:

```
1  -- ===== SUPERMARKET BRANCH ===== --
2  INSERT INTO SUPERMARKET_BRANCH (hotline, `address`, `name`) VALUES
3  ('0373359726', '4D, Trần Thị Vũng, p.An Bình, Dĩ An, Bình Dương', 'Chi
nhánh Bình Dương'),
4  ('0383553052', '268 Lý Thường Kiệt, phường 14, Quận 10, Tp.HCM', 'Chi nhánh
Q10'),
5  ('0383507284', 'Khu Đại học Quốc Gia Tp.HCM, Dĩ An, Bình Dương', 'Chi nhánh
Bình Dương');
6
7  -- ===== EMPLOYEE ===== --
8  INSERT INTO EMPLOYEE VALUES
9  (NULL, '233309782', '0373395726', 'Nguyễn Phúc Vinh', '4D, Trần Thị Vũng,
p.An Bình, Dĩ An, Bình Dương',
10 '2001-07-05', 'vinh.nguyen05072001@hcmut.edu.vn', 1, '2021-09-17');
11
12 INSERT INTO EMPLOYEE (identityCard, phone, `name`, SBranchID,
startWorkingDate) VALUES
13 ('987654321', '0357896145', 'Vương Thanh Duyên', 1, '2021-09-17'),
14 ('145697682', '0357896145', 'Lê Nghĩa', 1, '2021-09-17'),
15 ('147852369', '0147852369', 'Nguyễn Văn A', 1, '2021-09-17'),
16 ('321456987', '0147852369', 'Bùi Văn B', 1, '2021-09-17');
17
18 INSERT INTO MANAGER
19 VALUES (1, 3); -- Nguyễn Phúc Vinh
20
21 INSERT INTO AFTER_SELLING_STAFF
22 VALUES (3), -- Lê Nghĩa
23 (5); -- Bùi Văn B
24
25 -- ===== CATEGORY ===== --
26 INSERT INTO CATEGORY VALUES
27 ('Thực phẩm đông lạnh', 'Sử dụng trong vòng 30 ngày'),
28 ('Đồ gia dụng', 'Vật dụng sử dụng trong gia đình'),
```

```
29 ('Đồ điện tử', 'Máy tính, TV, các thiết bị / linh kiện điện tử ...'),
30 ('Gia vị', 'Gôm dầu ăn, nước mắm, nước tương, muối, mì chính ...'),
31 ('Thực phẩm khô', 'Gạo, bột mì, mì gói, lương khô ...');
32
33 -- ===== PRODUCT ===== --
34 INSERT INTO PRODUCT VALUES
35 (NULL, 'Thịt bò đông lạnh 1kg', '2021-11-08', '2021-11-30', 200000, 'Thịt
bò Úc nhập khẩu', 20, NULL, 'Thực phẩm đông lạnh'),
36 (NULL, 'Thớt', NULL, NULL, 40000, 'Việt Nam', 10, '10%', 'Đồ gia dụng'),
37 (NULL, 'TV Samsung 4K 85 inch', '2020-11-11', NULL, 12999000, 'Hàn Quốc',
100, '500000', 'Đồ điện tử'),
38 (NULL, 'Muối bột canh tôm', '2020-09-11', NULL, 5000, 'Hàn Quốc', 5, NULL,
'Gia vị'),
39 (NULL, 'Gạo 5kg', '2020-08-11', NULL, 80000, 'Hàn Quốc', 10, NULL, 'Thực
phẩm khô');
40
41 -- ===== FAVOUR ===== --
42 INSERT INTO FAVOUR (ID, `name`, `content`, startDate, endDate, mssn) VALUES
43 (NULL, 'Giảm giá 20/11', 'Giảm giá hàng năm nhân ngày lễ 20/11',
'2021-11-20', '2021-11-24', 1),
44 (NULL, 'Giảm giá lễ Noel', NULL, '2021-12-25', '2021-12-27', 1),
45 (NULL, 'ăn sale 11/11/2021', NULL, '2021-11-11', NULL, 1);
46
47 -- ===== VOUCHER_COUPON ===== --
48 INSERT INTO VOUCHER_COUPON VALUES
49 (NULL, 'voucher', 'y', 1),
50 (NULL, 'voucher', 'n', 1),
51 (NULL, 'coupon', 'n', 1),
52 (NULL, 'coupon', 'y', 1);
53
54 -- ===== CUSTOMER ===== --
55 INSERT INTO CUSTOMER VALUES
56 (NULL, 'Jordan', 'Peterson', '0123456789', 'jd.peterson@gmail.com', NULL,
5800, 'Book, gym facilities'),
57 (NULL, 'Nhi', 'Vũ Tuệ', '0321654789', 'tue.nhivu@gmail.com', '1996-11-13',
3200, 'Thực phẩm sạch'),
58 (NULL, 'Đặng', 'Nguyễn Hải', '0159874263', 'dang.nguyen@gmail.com',
'2001-07-05', 1250, 'Đồ điện tử'),
59 (NULL, 'Long', 'Nguyễn Hoàng', '0963258741', 'hoanglong@gmail.com',
'1993-10-13', 4875, 'Laptop Lenovo Thinkpad');
60
61 -- ===== PURCHASE ===== --
62 INSERT INTO PURCHASE VALUES
63 (NULL, DATE("2017-06-15 09:34:21"), 1),
64 (NULL, DATE("2017-06-16 09:34:21"), 2),
65 (NULL, DATE("2017-06-17 09:34:21"), 3),
66 (NULL, DATE("2017-06-18 09:34:21"), 4),
67 (NULL, DATE("2017-06-19 09:34:21"), 1),
68 (NULL, DATE("2017-06-20 09:34:21"), 2),
69 (NULL, DATE("2017-06-21 09:34:21"), 3);
70
71 -- ===== FEEDBACK ===== --
72 INSERT INTO FEEDBACK VALUES
73 (1, '2017-06-21 09:34:21', 3, 'Phản hồi đơn hàng #8', 'Nhân viên thiếu
nhiệt tình'),
74 (2, '2017-06-22 09:34:21', 4, 'Phản hồi đơn hàng #9', 'Nhân viên tư vấn
đúng ý khách hàng'),
```

```
75 (3, '2017-06-23 09:34:21', 5, 'Phản hồi đơn hàng #10', 'Giá cả phải chăng,
chất lượng sản phẩm tốt'),
76 (4, '2017-06-24 09:34:21', 1, 'Phản hồi đơn hàng #11', 'Sàn nhà trơn, đi té
dập mặt, phải mua thêm băng cá nhân'),
77 (1, '2017-06-25 09:34:21', 4, 'Chất lượng sản phẩm', 'Iphone chất lượng
cao, bền như Nokia, rơi xuống sàn mà sàn bể còn màn hình không sao');
78
79 -- ===== RESOLVES ===== --
80 INSERT INTO RESOLVES VALUES
81 (3, 1, '2017-06-21 09:34:21', NOW(), 'Xin lỗi quý khách vì điều này. Chúng
tôi sẽ cải thiện thái độ đối với khách hàng'),
82 (5, 2, '2017-06-22 09:34:21', NOW(), 'Cảm ơn quý khách đã sử dụng dịch vụ
của chúng tôi'),
83 (3, 3, '2017-06-23 09:34:21', NOW(), 'Cảm ơn quý khách đã tin tưởng chất
lượng sản phẩm của chúng tôi'),
84 (5, 4, '2017-06-24 09:34:21', NOW(), 'Thành thật xin lỗi quý khách vì đã
xảy ra sự cố ngoài ý muốn này'),
85 (3, 1, '2017-06-25 09:34:21', NOW(), 'Cảm ơn quý khách đã tin tưởng chất
lượng sản phẩm của chúng tôi');
86
87 -- ===== APPLY_FOR_PURCHASE ===== --
88 INSERT INTO APPLY_FOR_PURCHASE VALUES
89 (1, 3, '10%'),
90 (2, 4, '5000'),
91 (2, 5, '10000'),
92 (3, 6, '1000'),
93 (3, 7, '10%');
94
95 -- ===== TRANSACTS ===== --
96 INSERT INTO TRANSACTS VALUES
97 (2, 1, 1, 1),
98 (1, 1, 1, 2),
99 (3, 3, 1, 1),
100 (4, 1, 1, 2),
101 (2, 2, 1, 1),
102 (5, 2, 1, 2),
103 (5, 4, 1, 2),
104 (5, 5, 1, 2),
105 (4, 6, 1, 2),
106 (5, 7, 1, 2);
107
108 -- ===== NOTICES ===== --
109 INSERT INTO NOTICES VALUES
110 (3, '2021-11-09 15:26:02', 'Xin chúc mừng bạn đã trở thành khách hàng may
mắn đặc biệt của chúng tôi\n
111 Bạn được giảm giá đặc biệt 90% cho lần mua hàng có giá trị dưới 1 triệu
đồng', 'Giảm giá đặc biệt 90%'),
112 (5, '2021-11-09 15:26:03', 'Xin chúc mừng! Bạn đã trở thành khách hàng BẠC
!', 'Thăng bậc khách hàng'),
113 (5, '2021-11-09 15:26:04', 'Xin chúc mừng! Bạn đã trở thành khách hàng VÀNG
!', 'Thăng bậc khách hàng'),
114 (3, '2021-11-09 15:26:05', 'Sắp tới sự kiện chào mừng ngày nhà giáo 20/11,
sẵn sale thôi nào ! Hàng ngàn ưu đãi hấp dẫn v.v', 'Sự kiện 20/11');
115
116 -- ===== RECEIVES ===== --
117 INSERT INTO RECEIVES VALUES
118 (3, '2021-11-09 15:26:02', 1),
```



```
119 (3, '2021-11-09 15:26:02', 2),
120 (3, '2021-11-09 15:26:02', 3),
121 (3, '2021-11-09 15:26:02', 4),
122 (5, '2021-11-09 15:26:03', 2),
123 (5, '2021-11-09 15:26:03', 3),
124 (5, '2021-11-09 15:26:04', 3),
125 (5, '2021-11-09 15:26:04', 1),
126 (3, '2021-11-09 15:26:05', 3),
127 (3, '2021-11-09 15:26:05', 4);
128
129 -- ===== OWNS ===== --
130 INSERT INTO OWNS VALUES
131 (1, 1),
132 (3, 1),
133 (2, 2),
134 (4, 3);
135
136 -- ===== DEGREE ===== --
137 INSERT INTO DEGREE VALUES
138 (1, 'Bachelor Degree of Computer Science in Ho Chi Minh University of
    Technology'),
139 (1, 'Master Degree of Computer Science in Ho Chi Minh University of
    Technology');
```

3.3 Cài đặt các câu lệnh DML

Nhóm phân tích sơ bộ về UI để cung cấp góc nhìn trực quan từ phía người dùng và thể hiện rõ từng yêu cầu chức năng. Với mỗi chức năng trong từng trang UI, các câu lệnh DML tương ứng sẽ được cài đặt để phục vụ cho chức năng đó.

Nhóm lựa chọn ngôn ngữ javascript để hiện thực ứng dụng. Trong javascript, kí hiệu '\$*x*' chỉ giá trị biến *x* dưới dạng string. Nhóm sử dụng ký hiệu trên để chỉ rõ input cho những câu truy vấn đơn, tránh sử dụng stored procedure không cần thiết.

3.3.1 Trang thông tin cá nhân của khách hàng

11/22/21, 8:11 PM React App

SupermarketCRM Cá nhân Thông báo Phản hồi

Thông tin cá nhân

Lịch sử mua sắm

Ưu đãi/Voucher

Hồ sơ khách hàng

Mã khách hàng

Họ

Tên

Email address

Số điện thoại

Ngày sinh

Điểm tích lũy

Quan tâm, sở thích

Quần áo

Reset

Save

1azw6zxc

Nguyễn

Linh

nguyennngoclinh@gmail.com

0123456789

11/11/2021

5336

Choose File

No file chosen

localhost:3000/customer

1/1

Yêu cầu chức năng:

- Khách hàng xem toàn bộ thông tin cá nhân của mình
- Khách hàng chỉnh sửa mọi thông tin cá nhân, trừ mã ssn và điểm tích lũy

```
1 -- Khách hàng truy xuất thông tin cá nhân dựa trên SSN của khách hàng đó --
2 SELECT * FROM CUSTOMER WHERE SSN = '${CustomerSSN}';
3 -- Khách hàng cập nhật thông tin cá nhân --
4 UPDATE CUSTOMER
5 SET fName = '${fName}',
6     lName = '${lName}',
7     phone = '${phone}',
8     email = '${email}',
9     birthday = '${birthday}',
10    favorite = '${favorite}'
11 WHERE SSN = '${CustomerSSN}';
```

Ví dụ:

Trả về thông tin khách hàng có mã SSN là 111122223:

ssn	fName	lName	phone	email	birthday	score	favorite	imageUrl
111122223	Jordan	Peterson	0123456789	jd.peterson@gmail.com	NULL	5800	Book, gym facilities	imageUrl
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL



Cập nhật ngày sinh cho khách hàng có mã SSN là 111122223:

	ssn	fname	lname	phone	email	birthday	score	favorite	imageUrl
▶	111122223	Jordan	Peterson	0123456789	jd.peterson@gmail.com	2001-01-01	5800	Book, gym facilities	imageUrl
	222233334	Nhi	Vũ Tuệ	0321654789	tue.nhivu@gmail.com	1996-11-13	3200	Thực phẩm sạch	imageUrl
	333344445	Đặng	Nguyễn Hải	0159874263	dang.nguyen@gmail.com	2001-07-05	1250	Đồ điện tử	imageUrl
	444455556	Long	Nguyễn Hoàng	0963258741	hoanglong@gmail.com	1993-10-13	4875	Laptop Lenovo Thinkpad	imageUrl
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.3.2 Trang khách hàng gửi phản hồi

11/22/21, 8:18 PM React App

SupermarketCRM Cá nhân Thống báo Phản hồi

Tiêu đề:

Tiêu đề

Nội dung phản hồi:

Mọi thắc mắc cần được giải đáp, xin hãy liên hệ với chúng tôi.

★★★★★

Gửi

localhost:3000/customer/feedback

1/1

Yêu cầu chức năng: Khách hàng gửi feedback, gồm tiêu đề, nội dung feedback, đánh giá từ 1-5 sao (nếu có)

```
INSERT INTO FEEDBACK VALUES ('${CustomerSSN}', '${datetime.NOW()}',  
'${title}', '${star}', '${content}');
```

Ví dụ:

Khách hàng 111122223 gửi phản hồi:

	cssn	time	stars	title	content
	111122223	2021-10-12 12:00:00	3	Phản hồi đơn hàng #1	Nhân viên thiếu nhiệt tình
	111122223	2021-10-13 16:03:01	4	Phản hồi đơn hàng #2	Nhân viên tư vấn đúng ý khách hàng
▶	111122223	2021-11-23 16:04:17	5	Đánh giá đơn hàng #1	Sản phẩm tốt
	222233334	2021-10-14 17:04:02	5	Phản hồi đơn hàng #3	Giá cả phải chăng, chất lượng sản phẩm tốt
	222233334	2021-10-15 18:05:03	1	Phản hồi đơn hàng #4	Sàn nhà trơn, đi té đập mặt, phải mua thêm bả...
	333344445	2021-10-16 19:06:04	4	Chất lượng sản phẩm	Iphone chất lượng cao, bền như Nokia, rơi xuống...
*	NULL	NULL	NULL	NULL	NULL



3.3.3 Trang xem lại lịch sử mua sắm của khách hàng

11/22/21, 8:17 PM

React App

SupermarketCRM

Cá nhân Thống báo Phản hồi

Thông tin cá nhân

Lịch sử mua sắm

Ưu đãi/Voucher

Lịch sử mua sắm

Search for purchase

Thời gian

Tăng

Mã đơn hàng	Thời gian	Chi nhánh	Tổng tiền	Tích lũy
#za5x2q	11/22/2021, 8:08:01 PM	Chi nhánh Thủ Đức	500 000đ	100
#a2xw6a	11/22/2021, 8:08:01 PM	Chi nhánh Bình Dương	1000 000đ	200
#56ae9z	11/22/2021, 8:08:01 PM	Chi nhánh Bình Dương	200 000đ	50
#swd5s6	11/22/2021, 8:08:01 PM	Chi nhánh Bình Dương	1000 000đ	100

< 1 >

localhost:3000/customer 1/1

11/22/21, 8:17 PM

React App

Chi tiết đơn hàng axq56a

Danh sách sản phẩm

Mã sản phẩm	Tên	Nguồn gốc	NSX	HSD	Đơn giá	Giảm giá	Số lượng	Tích lũy
1424	Kệ đặt dép	Việt Nam	11/22/2021		60 000đ	5%	2	20
1424	Kệ đặt dép	Việt Nam	11/22/2021		60 000đ	5%	2	20
1424	Kệ đặt dép	Việt Nam	11/22/2021		60 000đ	5%	2	20

Nơi mua hàng

Chi nhánh Thủ Đức
4D, Trần Thị Vững, p.An Bình, Dĩ An, Bình Dương
0373 395 726

Thời gian

11/22/2021, 8:17:46 PM

Tổng giá trị: 500 000đ

Tổng điểm tích lũy: 50

localhost:3000/customer 1/1

Yêu cầu chức năng:

- Khách hàng lấy danh sách lịch sử mua sắm. Danh sách lịch sử mua sắm gồm mã giao dịch, thời gian, các thông tin của sản phẩm trong giao dịch đó gồm mã sản phẩm, điểm thưởng sản phẩm, giá sản phẩm, số lượng sản phẩm từng loại. Danh sách lịch sử mua sắm sắp xếp theo thời gian gần nhất.

- Khách hàng xem lịch sử mua sắm
- Với một tập dữ liệu lớn, không thể load 1 lần hết tất cả các record, mà phải sử dụng pagination


```
3  -- Kiểu dữ liệu cho phần PRODUCT.discount là một VARCHAR. Tùy vào giá trị
   của PRODUCT.discount, nếu PRODUCT.discount là NULL không có sự giảm giá, nếu
   chứa kí tự '%' sẽ quy đổi ra giá trị phần trăm và nhân với PRODUCT.price,
   nếu chỉ chứa kí tự số sẽ trừ thẳng vào PRODUCT.price. Vì vậy, việc xử lí
   logic tính toán nên để ứng dụng thực hiện thay cho hệ quản trị CSDL.
4
5  DROP PROCEDURE IF EXISTS GET_PURCHASE_LIST;
6  delimiter //
7  CREATE PROCEDURE GET_PURCHASE_LIST(IN cssn INT, IN beginIndex INT, IN
   numItemsPerPage INT)
8      BEGIN
9          SELECT
10             PURCHASE.ID as purchaseID, PURCHASE.`time`,
11             PRODUCT.ID as productID, PRODUCT.`name` as productName,
12             SUPERMARKET_BRANCH.`name` as branchName,
13             SUPERMARKET_BRANCH.`address` as branchAddr,
14             SUPERMARKET_BRANCH.hotline as branchHotline,
15             totalScore, totalPrice
16         FROM (
17             PURCHASE JOIN TRANSACTS
18                 ON PURCHASE.ID = TRANSACTS.purchaseID
19             JOIN PRODUCT
20                 ON TRANSACTS.productID = PRODUCT.ID)
21         WHERE PURCHASE.cssn = cssn
22         ORDER BY `time` DESC
23         GROUP BY PURCHASE.ID
24         LIMIT numItemsPerPage OFFSET beginIndex;
25     END //
26 delimiter ;
27
28 -- Khách hàng xem thông tin chi tiết của một sản phẩm
29 SELECT * FROM PRODUCT WHERE ID = '${productID}';
```

LƯU Ý: Trong thực tế, KHÔNG THỂ sử dụng thông tin của sản phẩm để hiển thị đơn hàng, vì các sản phẩm có thể thay đổi giá. Ví dụ, tại ngày 05/11 giá của bánh mì là 12.000đ, sang ngày 9/11 bánh mì tăng giá lên 15.000đ. Do đó nếu sử dụng câu procedure GET_PURCHASE_LIST ở trên, kết quả trả về sẽ bị sai lệch nếu có sự thay đổi giá.

Cách giải quyết là copy các thông tin biến động của sản phẩm liên quan tại thời điểm giao dịch sang relation TRANSACTS (gồm đơn giá, mức giảm giá, điểm tích lũy).

Ngoài ra, trong lược đồ ERD [1] có hai derived attribute là totalScore và totalPrice. Với mỗi lần truy xuất thông tin từ PURCHASE, ta phải thực hiện một FUNCTION/PROCEDURE cho việc tính toán hai attribute đó. Việc tính toán gây tải lên DBMS không cần thiết, khi mà lập trình viên có thể tính 1 lần duy nhất và lưu lại trong bảng PURCHASE. Khi đó có hai điều cần đảm bảo:

- Dữ liệu sau khi tính không được chỉnh sửa/xoá. Ta có thể sử dụng điều khiển truy cập để thu hồi quyền UPDATE và DELETE trên bảng PURCHASE
- Việc tính toán dữ liệu phải nằm bên server. Điều này có thể đảm bảo khi hiện thực ứng dụng.

Nhóm đã chỉnh sửa trong lược đồ ERD phù hợp với yêu cầu.



3.3.4 Kho lưu trữ ưu đãi cá nhân của khách hàng

11/22/21, 8:17 PM Read App

SupermarketCRM Cá nhân Thống báo Phản hồi

Thông tin cá nhân

Lịch sử mua sắm

Ưu đãi/Voucher

Kho lưu trữ ưu đãi

Code	Mô tả	Ngày áp dụng	Ngày hết hạn	Giảm	
za5x2q	Giảm giá 20% cho tất cả đơn hàng	11/22/2021, 8:08:01 PM	11/22/2021, 8:08:01 PM	20%	🗑
qw2a6w	Giảm giá 500k cho các đồ điện tử có giá trị > 10 triệu	11/22/2021, 8:08:01 PM	11/22/2021, 8:08:01 PM	500 000đ	🗑

localhost:3000/customer

1/1

Yêu cầu chức năng: Khách hàng xem kho lưu trữ các voucher/coupon cá nhân. Thông tin cần truy xuất bao gồm voucher/coupon code, loại voucher hoặc coupon, mọi thông tin về chương trình khuyến mãi/sự kiện liên quan đến voucher/coupon đó, với điều kiện voucher/coupon chưa được sử dụng.

```
1  -- Hiển thị danh sách
2  SELECT
3      VOUCHER_COUPON.code as code,
4      VOUCHER_COUPON.`type` as `type`,
5      FAVOUR.*
6  FROM VOUCHER_COUPON JOIN FAVOUR ON VOUCHER_COUPON.favourID = FAVOUR.ID
7  WHERE VOUCHER_COUPON.code IN (
8      SELECT OWNS.vcode FROM OWNS
9      WHERE OWNS.cssn = '${CustomerSSN}'
10 ) AND VOUCHER_COUPON.isUsed = 'n'
11 AND FAVOUR.`status` = 'applying';
12
13 -- Xóa ưu đãi kho sở hữu cá nhân
14 DELETE FROM OWNS WHERE vcode='${vcode}';
```

Ví dụ:

Kho lưu trữ ưu đãi của khách hàng 222233334:

code	type	ID	name	content	startDate	endDate	quantity	status	mssn
2	voucher	1	Giảm giá 20/11	Giảm giá hàng năm nhân ngày lễ 20/11	2021-11-20	2021-11-24	100	applying	111222333

3.3.5 Khách hàng xem trang thông báo


11/22/21, 8:18 PM

Read App

SupermarketCRM

Cá nhân Thông báo Phản hồi

Thông báo




Ưu đãi cuối năm

Thời gian: Từ ngày **Mon Nov 22 2021** đến ngày **Mon Nov 22 2021**

Đối tượng áp dụng: Tất cả khách hàng

Giảm giá 20% cho tất cả các đơn hàng có giá trị trên 1 triệu

[Chi tiết](#)



Ưu đãi 20/11

Thời gian: Từ ngày **Mon Nov 22 2021** đến ngày **Mon Nov 22 2021**

Đối tượng áp dụng: Khách hàng Bạc

Giảm giá 10% cho tất cả quà tặng 20/11

[Chi tiết](#)

localhost:3000/customer/notifications

1/1

Yêu cầu chức năng: Khách hàng nhận thông báo từ nhân viên chăm sóc sau bán. Thông tin cần nhận bao gồm tiêu đề, thời gian và nội dung.

```

1  -- Sử dụng paging để xử lý lượng dữ liệu lớn
2  DROP PROCEDURE IF EXISTS GET_NOTIFICATION_LIST;
3  delimiter //
4  CREATE PROCEDURE GET_NOTIFICATION_LIST(IN cssn INT, IN beginIndex INT, IN
numItemsPerPage INT)
5      BEGIN
6          SELECT `time`, title, content
7          FROM NOTICES
8          WHERE (ass_ssn, `time`) IN (
9              SELECT ass_ssn, `time` FROM RECEIVES
10             WHERE RECEIVES.cssn = cssn
11         )
12         ORDER BY `time` DESC
13         LIMIT numItemsPerPage OFFSET beginIndex;
14      END //
15  delimiter ;

```

Ví dụ:

Danh sách thông báo của khách hàng 111122223:

time	title	content
2021-11-09 15:26:02	Giảm giá đặc biệt 90%	Xin chúc mừng bạn đã trở thành khách hàng ma...



3.3.6 Nhân viên truy xuất thông tin của khách hàng

SupermarketCRM

Tư vấn tại chỗ Chăm sóc sau bán

Đăng xuất

Thông tin KH

Thông kê

Mã số khách hàng

Tên khách hàng

Email

Số điện thoại

Ngày sinh

Điểm tích lũy

Nhóm khách hàng

Sở thích khách hàng

Yêu cầu chức năng: Nhân viên có thể truy xuất thông tin của khách hàng từ ssn, phone hoặc email của khách hàng. Thông tin cần nhập: chuỗi string. Thông tin trả về: Họ tên khách hàng, MSKH, ngày sinh, số điện thoại, email, điểm tích lũy, loại khách hàng và sở thích do khách hàng lưu lại.

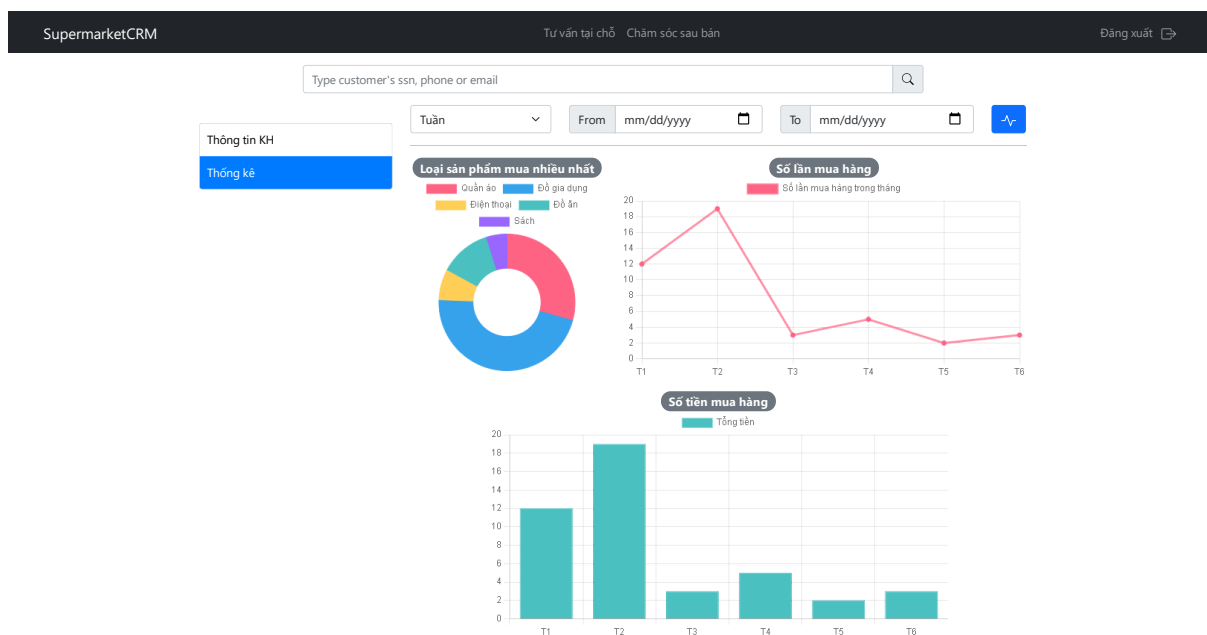
```
1 -- Loại khách hàng được ứng dụng thực hiện tính toán từ điểm tích lũy của
  khách hàng.
2 -- Trả về thông tin khách hàng dựa vào input
3 SELECT * FROM CUSTOMER WHERE ssn = '${input}' OR phone = '${input}' or email
  = '${input}';
```

Ví dụ:

Nhân viên truy xuất thông tin khách hàng 111122223:

	ssn	fname	lname	phone	email	birthday	score	favorite	imageUrl
▶	111122223	Jordan	Peterson	0123456789	jd.peterson@gmail.com	2001-01-01	5800	Book, gym facilities	imageUrl
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.3.7 Nhân viên xem thông tin thống kê khách hàng



Yêu cầu chức năng, nhân viên có thể xem phân tích thông kê về:

- Tần suất mua sắm
- Loại phẩm mua nhiều nhất (Mặc định là 5)
- Điểm tích lũy của khách hàng

```

1  -- Thống kê biểu đồ tròn về loại sản phẩm mua nhiều nhất --
2  SELECT COUNT(*) as `buyTimes`
3  FROM TRANSACTS JOIN PRODUCT ON TRANSACTS.productID = PRODUCT.ID
4  WHERE purchaseID IN (
5      SELECT ID FROM PURCHASE
6      WHERE PURCHASE.cssn = '${CustomerSSN}'
7  ) GROUP BY PRODUCT.categoryName
8  ORDER BY `buyTimes`
9  LIMIT 5;
10
11 -- Thống kê biểu đồ đường về số lần mua hàng --
12 -- Theo tuần
13 SELECT
14     COUNT(*) as 'buyTimes',
15     CONCAT(YEAR(`time`), '/', WEEK(`time`)) as `week`
16 FROM PURCHASE JOIN CUSTOMER ON PURCHASE.cssn = CUSTOMER.ssn
17 WHERE `time` > '${startTime}' AND `time` < '${endTime}';
18 GROUP BY `week`;
19
20 -- Theo tháng
21 SELECT
22     COUNT(*) as 'buyTimes',
23     CONCAT(YEAR(`time`), '/', MONTH(`time`)) as `month`
24 FROM PURCHASE JOIN CUSTOMER ON PURCHASE.cssn = CUSTOMER.ssn

```

```
25 WHERE `time` > '${startTime}' AND `time` < '${endTime}';
26 GROUP BY `month`;
27
28 -- Theo quý
29 SELECT
30     COUNT(*) as 'buyTimes',
31     CONCAT(YEAR(`time`), '/', MONTH(`time`) DIV 3) as `quarter`
32 FROM PURCHASE JOIN CUSTOMER ON PURCHASE.cssn = CUSTOMER.ssn
33 WHERE `time` > '${startTime}' AND `time` < '${endTime}';
34 GROUP BY `quarter`;
35
36 -- Theo năm
37 SELECT
38     COUNT(*) as 'buyTimes', YEAR(`time`) as `year`
39 FROM PURCHASE JOIN CUSTOMER ON PURCHASE.cssn = CUSTOMER.ssn
40 WHERE `time` > '${startTime}' AND `time` < '${endTime}';
41 GROUP BY `year`;
42
43 -- Thống kê biểu đồ cột về điểm tích lũy mua hàng --
44 -- Theo tuần
45 SELECT
46     SUM(totalScore) as `score`,
47     CONCAT(YEAR(`time`), '/', WEEK(`time`)) as `week`
48 FROM PURCHASE JOIN CUSTOMER ON PURCHASE.cssn = CUSTOMER.ssn
49 WHERE `time` > '${startTime}' AND `time` < '${endTime}';
50 GROUP BY `week`;
51 -- Theo tháng
52 SELECT
53     SUM(totalScore) as `score`,
54     CONCAT(YEAR(`time`), '/', MONTH(`time`)) as `month`
55 FROM PURCHASE JOIN CUSTOMER ON PURCHASE.cssn = CUSTOMER.ssn
56 WHERE `time` > '${startTime}' AND `time` < '${endTime}';
57 GROUP BY `month`;
58 -- Theo quý
59 SELECT
60     SUM(totalScore) as `score`,
61     CONCAT(YEAR(`time`), '/', MONTH(`time`) DIV 3) as `quarter`
62 FROM PURCHASE JOIN CUSTOMER ON PURCHASE.cssn = CUSTOMER.ssn
63 WHERE `time` > '${startTime}' AND `time` < '${endTime}';
64 GROUP BY `quarter`;
65 -- Theo năm
66 SELECT
67     SUM(totalScore) as `score`,
68     YEAR(`time`) as `year`
69 FROM PURCHASE JOIN CUSTOMER ON PURCHASE.cssn = CUSTOMER.ssn
70 WHERE `time` > '${startTime}' AND `time` < '${endTime}';
71 GROUP BY `year`;
```

3.3.8 Nhân viên xem danh sách khách hàng

11/22/21, 8:19 PM

React App

SupermarketCRM

Tư vấn tại chỗ · Chăm sóc sau bán

Danh sách KH

Tiếp nhận phản hồi

Gửi thông báo

Danh sách khách hàng

Search customer ...

Đồng

MSKH	Tên KH	Ngày sinh	Nhóm KH	SĐT	Email
1915940	Nguyễn Phúc Vinh	05/07/2001	Vàng	0373 395 726	phucvinh57@gmail.com
1919191	Vương Thanh Duyên	10/10/2001	Bạc	0383 553 052	vtduyen@gmail.com
1915940	Lê Nghĩa	11/11/2001	Đồng	0963 145 368	nghia.le@gmail.com

<

1

>

localhost:3000/staff/ass

1/1

Yêu cầu chức năng:

- Nhân viên có thể tìm kiếm khách hàng bằng SĐT hoặc email của khách hàng
- Nhân viên có thể xem danh sách tất cả khách hàng
- Nhân viên có thể lọc danh sách khách hàng theo loại
- Thông tin khách hàng gồm: Tên, ngày sinh, nhóm KH, SĐT, email

```

1  -- Tìm kiếm khách hàng bằng phone hoặc email
2  SELECT * FROM CUSTOMER WHERE phone LIKE '%${input}%' OR email LIKE
   '%${input}%';
3
4  -- Trả về danh sách khách hàng
5  delimiter //
6  CREATE DEFINER=`root`@`localhost` PROCEDURE `GET_CUSTOMER_LIST`(IN
   beginIndex INT, IN numItemsPerPage INT, IN typeCustomer INT)
7  BEGIN
8      SELECT *
9      FROM CUSTOMER
10     WHERE (score<5000 AND typeCustomer='bronze')
11           OR (score>=5000 AND score<10000 AND typeCustomer='silver')
12           OR (score>10000 AND typeCustomer='gold')
13           OR (typeCustomer='all')
14     ORDER BY `fname`
15     LIMIT numItemsPerPage OFFSET beginIndex;
16 END //
17 delimiter ;
18
19 -- Áp dụng GET_CUSTOMER_LIST
20 call
   GET_CUSTOMER_LIST('${beginIndex}', '${numItemsPerPage}', '${typeCustomer}');
```



Ví dụ:

Trả về danh sách khách hàng có chuỗi '23' trong số điện thoại hoặc email:

	ssn	fname	lname	phone	email	birthday	score	favorite	imageUrl
▶	111122223	Jordan	Peterson	0123456789	jd.peterson@gmail.com	2001-01-01	5800	Book, gym facilities	imageUrl
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Trả về danh sách khách hàng đồng:

	ssn	fname	lname	phone	email	birthday	score	favorite	imageUrl
▶	222233334	Nhi	Vũ Tuệ	0321654789	tue.nhivu@gmail.com	1996-11-13	3200	Thực phẩm sạch	imageUrl
	333344445	Đăng	Nguyễn Hải	0159874263	dang.nguyen@gmail.com	2001-07-05	1250	Đồ điện tử	imageUrl



3.3.9 Nhân viên trả lời phản hồi của khách hàng

SupermarketCRM

Tư vấn tại chỗ Chăm sóc sau bán

Đăng xuất

Danh sách KH

Tiếp nhận phản hồi

Gửi thông báo

Tiếp nhận phản hồi

Search customer by phone or email

Đồng

Gần nhất

Đã trả lời

Tiêu đề	Thời gian	Tên khách hàng	Trạng thái	Rating
Máy tính	11/23/2021, 8:58:57 PM	Nguyễn Phúc Vinh	Đang chờ	5/5
Quần áo	11/23/2021, 8:58:57 PM	Vương Thanh Duyên	Đã trả lời	3/5
Thịt đông lạnh	11/23/2021, 8:58:57 PM	Lê Nghĩa	Đã trả lời	4/5

< 1 >

11/22/21, 8:33 PM

React App

SupermarketCRM

Tư vấn tại chỗ Chăm sóc sau bán

Tiếp nhận phản hồi

Danh sách KH

Search customer ...

Đồng

Gần nhất

Nội dung feedback [Quần áo](#)

X

Nội dung phản hồi

Trả lời phản hồi

Quần áo bị giãn chun sau 1 tháng mặc

Trả lời

localhost:3000/staff/ass

1/1

Yêu cầu chức năng:

- Nhân viên có thể tìm kiếm khách hàng bằng SĐT hoặc email
- Hiển thị danh sách phản hồi của khách hàng. Thông tin khách hàng gồm tên, loại khách hàng, số sao đánh giá, tiêu đề phản hồi/đánh giá, thông tin đánh giá/phản hồi của khách hàng, thời gian khách hàng đánh

giá/phản hồi đến hiện tại, trạng thái phản hồi/đã phản hồi

- Nhân viên trả lời khách hàng

```

1  -- Tìm kiếm phản hồi bằng keyword: input
2  SELECT fname,lname,`time`,stars,title,content
3  FROM FEEDBACK, CUSTOMER
4  WHERE FEEDBACK.cssn = CUSTOMER.ssn AND (phone LIKE '%${input}%' OR email
5  LIKE '%${input}%')
6
7  ORDER BY `time` DESC;
8
9  -- Trả về danh sách phản hồi của khách hàng
10 SELECT fname,lname,`time`,stars,title,content
11 FROM FEEDBACK, CUSTOMER
12 WHERE FEEDBACK.cssn = CUSTOMER.ssn
13 ORDER BY `time` DESC;
14
15 -- Nhân viên phản hồi đánh giá của khách hàng
16 INSERT INTO RESOLVES (ass_ssn,cssn,feedbackTime,resolveTime,content) VALUES
17 ('${ass_ssn}','${cssn}','${feedbackTime}','${datetime.NOW()}', '${content}');

```

Ví dụ:

Trả về danh sách phản hồi của khách hàng có chuỗi 'tue' trong số điện thoại hoặc email:

	fname	lname	time	stars	title	content
▶	Nhi	Vũ Tuệ	2021-10-15 18:05:03	1	Phản hồi đơn hàng #4	Sản nhà tron, đi té dập mặt, phải mua thêm bả...
	Nhi	Vũ Tuệ	2021-10-14 17:04:02	5	Phản hồi đơn hàng #3	Giá cả phải chăng, chất lượng sản phẩm tốt

Nhân viên 222333444 trả lời phản hồi của khách hàng 111122223:

	ass_ssn	cssn	feedbackTime	resolveTime	content
▶	222333444	111122223	2021-10-12 12:00:00	2021-10-12 13:00:00	Xin lỗi quý khách vì điều này. Chúng tôi sẽ cải th...
	222333444	111122223	2021-10-13 16:03:01	2021-10-13 17:00:01	Cảm ơn quý khách đã sử dụng dịch vụ của chún...
	222333444	111122223	2021-11-23 16:04:17	2021-11-23 17:04:17	Cảm ơn bạn đã đánh giá
	222333444	222233334	2021-10-14 17:04:02	2021-10-14 18:04:02	Cảm ơn quý khách đã tin tưởng chất lượng sản ...
	333444555	222233334	2021-10-15 18:05:03	2021-10-15 19:05:03	Thành thật xin lỗi quý khách vì đã xảy ra sự cố ...
	333444555	333344445	2021-10-16 19:06:04	2021-10-16 19:36:04	Cảm ơn quý khách đã tin tưởng chất lượng sản ...
•	NULL	NULL	NULL	NULL	NULL

3.3.10 Nhân viên gửi thông báo cho khách hàng

SupermarketCRM

Tư vấn tại chỗ Chăm sóc sau bán

Đăng xuất

Danh sách KH

Tiếp nhận phản hồi

Gửi thông báo

Tiêu đề:

Tiêu đề

Gửi đến:

Khách hàng đồng

Link đính kèm

Ảnh

Choose File

No file chosen

Nội dung thông báo

Gửi

Yêu cầu chức năng:

- Nhân viên gửi thông báo cho khách hàng. Thông báo gồm tiêu đề thông báo và nội dung thông báo
- Nhân viên có thể gửi thông báo cho khách hàng với những loại khách hàng khác nhau

```
1 -- Nhân viên tạo thông báo
2 INSERT INTO NOTICES (ass_ssn,`time`,content,title) VALUES
  ('${ass_ssn}','${datetime.NOW()}', '${content}', '${title}');
3
4 -- Trả về danh sách khách hàng được lọc (đã thực hiện)
5
6 -- Gửi thông tin cho khách hàng có cssn thuộc danh sách trên
7 INSERT INTO RECEIVES (ass_ssn,`time`,cssn) VALUES
  ('${ass_ssn}','${datetime.NOW()}', '${cssn}')
```

3.3.11 Quản lý xem ưu đãi đang được áp dụng, đã quá hạn hoặc dự kiến áp dụng

11/23/21, 5:08 PM React App

SupermarketCRM Sản phẩm Ưu đãi Đăng xuất

Quá hạn Đang áp dụng Dự kiến

Mức giảm giá Tăng

#ID	Tên	Loại	Ngày bắt đầu	Ngày kết thúc	Giảm giá	SL còn lại	
142436	Voucher Giáng sinh	Voucher	11/23/2021	11/23/2021	50%	125	
191594	Coupon 20/11	Coupon	11/23/2021	11/23/2021	100 000đ	1675	
159786	Giảm giá cuối năm	Sale	11/23/2021	11/23/2021	50%		

< 1 >

localhost:3000/manager/favours

1/1

Yêu cầu chức năng:

- Quản lý xem danh sách các ưu đãi
- Quản lý tìm kiếm ưu đãi bằng mã ưu đãi
- Có áp dụng phân trang

```

1  -- Quản lý chi nhánh tìm kiếm ưu đãi bằng mã ưu đãi
2  SELECT *
3  FROM FAVOUR
4  WHERE ID='{favourID}';
5
6  -- Quản lý chi nhánh quản lý các mã ưu đãi
7  -- statusValue = 1 => Đang áp dụng
8  -- statusValue = 2 => Dự kiến
9  -- statusValue = 3 => Quá hạn
10
11 DROP PROCEDURE `GET_FAVOUR_LIST`;
12 delimiter //
13 CREATE DEFINER=`root`@`localhost` PROCEDURE `GET_FAVOUR_LIST` (IN beginIndex
14 INT, IN numItemsPerPage INT, statusValue INT)
15 BEGIN
16     SELECT *
17     FROM FAVOUR
18     WHERE (`status`=statusValue)
19     ORDER BY ID
20     LIMIT numItemsPerPage OFFSET beginIndex;
21 END //
22 delimiter ;

```

Ví dụ:

Quản lý tìm kiếm ưu đãi có mã là 1:

ID	name	content	startDate	endDate	quantity	status	mssn
1	Giảm giá 20/11	Giảm giá hàng năm nhân ngày lễ 20/11	2021-11-20	2021-11-24	100	applying	111222333
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Quản lý xem danh sách ưu đãi đang được áp dụng:

ID	name	content	startDate	endDate	quantity	status	mssn
1	Giảm giá 20/11	Giảm giá hàng năm nhân ngày lễ 20/11	2021-11-20	2021-11-24	100	applying	111222333
2	Giảm giá lễ Noel	Giảm giá lễ Noel	2021-12-25	2021-12-27	100	applying	111222333
3	Săn sale	Săn sale 11/11/2021	2021-11-10	2021-11-12	100	applying	111222333

3.3.12 Quản lý chi nhánh tạo ưu đãi mới

11/23/21, 5:10 PM React App

SupermarketCRM Sản phẩm Ưu đãi Đăng xuất

Quá hạn Đang áp dụng Dự kiến

Mức giảm giá Tăng

#ID

142436

191594

159786

Thêm ưu đãi

Tên:

Voucher Giáng sinh

Trạng thái:

Quá hạn

Loại:

Voucher

Ngày bắt đầu:

mm/dd/yyyy

Ngày kết thúc:

mm/dd/yyyy

Giảm giá:

Số lượng:

Hủy bỏ

Thêm

localhost:3000/manager/favours 1/1

Yêu cầu chức năng:

- Quản lý chi nhánh tạo các ưu đãi mới
- Nội dung một ưu đãi gồm: Mã ưu đãi, tên ưu đãi, nội dung ưu đãi, ngày bắt đầu, ngày kết thúc, số lượng ưu đãi và mức giảm giá
- Quản lý chi nhánh áp dụng ưu đãi và hủy áp dụng ưu đãi

```

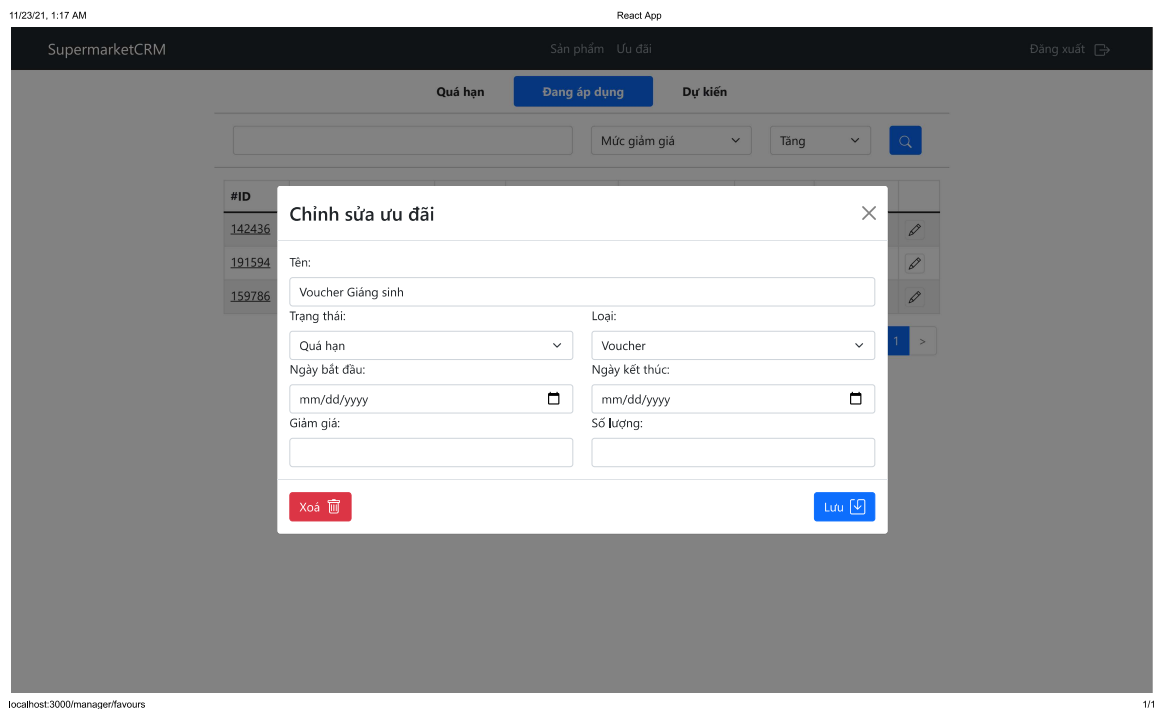
1 -- Quản lý tạo mới ưu đãi
2 INSERT INTO FAVOUR VALUES (NULL, '${name}', '${content}', '${startDate}',
  '${endDate}', '${quantity}', '${discount}', '${status}', '${mssn}');
```

Ví dụ:

Quản lý tạo mới ưu đãi:

ID	name	content	startDate	endDate	quantity	status	mssn
1	Giảm giá 20/11	Giảm giá hàng năm nhân ngày lễ 20/11	2021-11-20	2021-11-24	100	applying	111222333
2	Giảm giá lễ Noel	Giảm giá lễ Noel	2021-12-25	2021-12-27	100	applying	111222333
3	Săn sale	Săn sale 11/11/2021	2021-11-10	2021-11-12	100	applying	111222333
4	giảm gia	giảm 10k	2021-11-23	2021-11-24	100	applying	111222333
	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.3.13 Quản lý chi nhánh cập nhật hoặc xóa ưu đãi



Yêu cầu chức năng:

- Quản lý có thể chỉnh sửa hoặc xóa ưu đãi
- Nội dung có thể chỉnh sửa gồm Tên ưu đãi, trạng thái ưu đãi, ngày bắt đầu, ngày kết thúc, số lượng ưu đãi và mức giảm giá

```

1  -- Hiển thị chi tiết một ưu đãi
2  SELECT *
3  FROM FAVOUR
4  WHERE ID='${favourID}';
5
6  -- Quản lý chỉnh sửa ưu đãi
7  UPDATE FAVOUR
8  SET `name`='${name}',
9      `status`='${status}',
10     `discount`='${discount}',
11     `startDate`='${startDate}',
12     `endDate`='${endDate}',
13     `quantity`='${quantity}',
14     `content`='${content}'
15  WHERE ID='${favourID}';
16
17  -- Quản lý xóa ưu đãi
18  DELETE FROM FAVOUR WHERE ID='${favourID}';

```

Ví dụ:

Quản lý chỉnh sửa ưu đãi:

	ID	name	content	startDate	endDate	quantity	discount	status	msn
	1	Giảm giá 20/11	Giảm giá hàng năm nhân ngày lễ 20/11	2021-11-20	2021-11-24	100	10000	applying	111222333
	2	Giảm giá lễ Noel	Giảm giá lễ Noel	2021-12-25	2021-12-27	100	10%	applying	111222333
	3	Săn sale	Săn sale 11/11/2021	2021-11-10	2021-11-12	100	10000	applying	111222333
	4	Giảm giá	Giảm ngay 10000	2021-11-23	2021-11-24	100	10000	applying	111222333

3.3.14 Quản lý chi nhánh xem danh sách sản phẩm

11/23/21, 1:16 AM React App

SupermarketCRM Sản phẩm Ưu đãi Đăng xuất

Search by product name or ID ...

#ID	Tên	Loại SP	NSX	HSD	Xuất xứ	Giá	Ưu đãi
142436	Quần jean co dân	Quần áo	11/23/2021	11/23/2021	Việt Nam	500 000đ	
156478	Kệ đựng dép	Đồ gia dụng	11/23/2021	11/23/2021	Việt Nam	80 000đ	142436
624569	Laptop Lenovo Thinkpad	Đồ điện tử	11/23/2021	11/23/2021	Japan	18 000 000đ	

< 1 >

localhost:3000/manager/products

1/1

Yêu cầu chức năng:

- Quản lý xem danh sách các sản phẩm
- Quản lý có thể tìm kiếm sản phẩm bằng từ khóa

```

1  -- Tìm kiếm sản phẩm bằng từ khóa input
2  SELECT * FROM PRODUCT WHERE id LIKE '%${input}%' OR `name` LIKE
   '%${input}%';
3
4  -- Hiển thị sản phẩm
5  delimiter //
6  CREATE DEFINER=`root`@`localhost` PROCEDURE `GET_PRODUCT_LIST`(IN beginIndex
   INT, IN numItemsPerPage INT)
7  BEGIN
8      SELECT *
9      FROM PRODUCT
10     ORDER BY ID
11     LIMIT numItemsPerPage OFFSET beginIndex;
12 END //
13 delimiter ;
14
15 call GET_PRODUCT_LIST('${beginIndex}', '${numItemsPerPage}');
```

Ví dụ:

Quản lý tìm kiếm ưu đãi bằng từ khóa 'bo' :

	ID	name	mdate	edate	price	origin	score	discount	categoryName
▶	1	Thịt bò đông lạnh 1kg	2021-11-08	2021-11-30	200000	Thịt bò Úc nhập khẩu	20	NULL	Thực phẩm đông lạnh
	4	Muối bột canh tôm	2020-09-11	NULL	5000	Hàn Quốc	5	NULL	Gia vị
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Quản lý xem danh sách sản phẩm:

ID	name	mdate	edate	price	origin	score	discount	categoryName
1	Thịt bò đông lạnh 1kg	2021-11-08	2021-11-30	200000	Thịt bò Úc nhập khẩu	20	NULL	Thực phẩm đông lạnh
2	Thịt	NULL	NULL	40000	Việt Nam	10	10%	Đồ gia dụng
3	TV Samsung 4K 85 inch	2020-11-11	NULL	12999000	Hàn Quốc	100	500000	Đồ điện tử
4	Muối bột canh tôm	2020-09-11	NULL	5000	Hàn Quốc	5	NULL	Gia vị
5	Gạo 5kg	2020-08-11	NULL	80000	Hàn Quốc	10	NULL	Thực phẩm khô
6	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

3.4 Cài đặt trigger

Để đảm bảo total participation phía Purchase cho mỗi quan hệ Transacts 1-M-N giữa Supermarket_branch, Purchase, Product, cần cài đặt trigger sau mỗi lần tạo một Purchase mới.

```

1 DROP TRIGGER IF EXISTS NEW_PURCHASE;
2 delimiter //
3 CREATE TRIGGER NEW_PURCHASE AFTER INSERT ON PURCHASE
4 FOR EACH ROW
5 BEGIN
6     SET @cnt = (SELECT COUNT(*) FROM TRANSACTS WHERE purchaseID =
7     NEW.ID);
8     IF @cnt = 0 THEN
9         SET @msg = 'A PURCHASE MUST PARTICIPATE IN AT LEAST 1 TRANSACT';
10        SIGNAL SQLSTATE '45000' SET message_text = @msg;
11    END IF;
12 END //
13 delimiter ;

```

Một thông báo được tạo ra bởi nhân viên phải có đối tượng nhận (total participation phía Notices trong mỗi quan hệ Receives M-N giữa Notices và Customer)

```

1 DROP TRIGGER IF EXISTS NEW_NOTICE;
2 delimiter //
3 CREATE TRIGGER NEW_NOTICE AFTER INSERT ON NOTICES
4 FOR EACH ROW
5 BEGIN
6     SET @cnt = (SELECT COUNT(*) FROM RECEIVES WHERE ass_ssn =
7     NEW.ass_ssn AND `time` = NEW.`time`);
8     IF @cnt = 0 THEN
9         SET @msg = 'A NOTIFICATION MUST HAS SPECIFIC TARGET';
10        SIGNAL SQLSTATE '45000' SET message_text = @msg;
11    END IF;
12 END //
13 delimiter ;

```


3.5 Kiểm tra chuẩn hóa cho cơ sở dữ liệu kết quả

3.5.1 Kiểu thực thể mạnh

- **SupermarketBranch (ID, Hotline, Address, Name)**
 - PK : ID
 - Non-key attributes: Hotline, Address, Name
 - FD1: ID -> (Hotline, Address, Name)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Employee (SSN, IdentityCard, Phone, Name, Address, Birthday, Email, SBranchID, StartWorkingDate)**
 - PK : SSN
 - Non-key attributes: IdentityCard, Phone, Name, Address, Birthday, Email, SBranchID, StartWorkingDate
 - FD1: SSN -> (IdentityCard, Phone, Name, Address, Birthday, Email, SBranchID, StartWorkingDate)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Manager (SSN, ExperienceYear, SBranchID)**
 - PK : SSN
 - Non-key attributes: ExperienceYear, SBranchID
 - FD1: SSN -> (ExperienceYear, SBranchID)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Customer (SSN, FName, LName, Phone, Email, Birthday, Score, Favourite))**
 - PK : SSN
 - Non-key attributes: FName, LName, Phone, Email, Birthday, Score, Favourite
 - FD1: SSN -> (FName, LName, Phone, Email, Birthday, Score, Favourite)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Category (Name, Description)**
 - PK : Name
 - Non-key attributes: Description
 - FD1: Name -> Description
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.

- * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
- Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Product (ID, Name, EDate, Mdate, Price, Origin, Discount, CategoryName)**
 - PK : ID
 - Non-key attributes: Name, EDate, Mdate, Price, Origin, Discount, CategoryName
 - FD1: ID -> (Name, EDate, Mdate, Price, Origin, Discount, CategoryName)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Purchase(ID, Time, CSSN, totalPrice, totalScore)**
 - PK: ID
 - Non-key attributes: Time, CSSN, totalPrice, totalScore
 - FD1: ID -> (Time, CSSN, totalPrice, totalScore)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Favour (ID, Name, Content, Discount, StartDate, EndDate, MSSN, status)**
 - PK : ID
 - Non-key attributes: Name, Content, Discount, StartDate, EndDate, MSSN
 - FD1: ID -> (Name, Content, Discount, StartDate, EndDate, MSSN, Status)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Voucher/Coupon(Code, Type, IsUsed, FavourID)**
 - PK: Code
 - Non-key attributes: Type, IsUsed, FavourID
 - FD1: Code -> (Type, IsUsed, FavourID)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.

3.5.2 Kiểu thực thể yếu

- **Feedback (CSSN, Time, Star, Title, Content)**
 - PK : CSSN, Time
 - Non-key attributes: Star, Title, Content
 - FD1: (CSSN, Time) -> (Star, Title, Content)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.

- * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
- Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Notices (ASS_SSN, Time, Content, Title, linkUrl, imageUrl)**
 - PK : ASS_SSN, Time
 - Non-key attributes: Content, Title, linkUrl, imageUrl
 - FD1: (ASS_SSN, Time) -> (Content, Title)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.

3.5.3 Mối quan hệ

- **Resolves (ASS_SSN, CSSN, FeedbackTime, ResolvesTime, Content)**
 - PK : ASS_SSN, CSSN, FeedbackTime
 - Non-key attributes: ResolvesTime, Content
 - FD1: (ASS_SSN, CSSN, FeedbackTime) -> (ResolvesTime, Content)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **ApplyForPurchase (FavourID, PurchaseID)**
 - PK : FavourID, PurchaseID
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Transacts (ProductID, PurchaseID, SBranchID, NumberOfProducts, price, discount, score)**
 - PK : ProductID, PurchaseID
 - Non-key attributes: SBranchID, NumberOfProducts, price, discount
 - FD1: (ProductID, PurchaseID) -> (SBranchID, NumberOfProducts, price, discount, score)
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.
- **Owns(VCode, CSSN)**
 - PK: VCode
 - Non-key attributes: CSSN
 - FD1: VCode -> CSSN
 - Ta thấy:
 - * Thuộc tính không khóa không có phụ thuộc hàm riêng phần vào khóa.
 - * Thuộc tính không khóa không có phụ thuộc hàm bắc cầu vào khóa.
 - Kết luận: Vậy quan hệ này ở dạng chuẩn 3NF.

4 Triển khai thiết kế và hiện thực tổng thể hệ thống cho các yêu cầu phi chức năng

Các yêu cầu chức năng đã đặt ra trong BTL 1:

#	Yêu cầu phi chức năng	Độ ưu tiên
1	Khách hàng và nhân viên dễ dàng sử dụng hệ thống sau 1 hoặc 2 lần hướng dẫn	1
2	Hệ thống dựa trên nền tảng web giúp người dùng thuận tiện truy cập mà không cần tải app.	1
3	1 câu truy vấn yêu cầu I/O time < 2s	2
4	Bảo mật thông tin của khách hàng, nhân viên và quản lý	3

Dựa trên UI/UX của ứng dụng, các yêu cầu phi chức năng 1 và 2 đã hoàn thành. Sau đây là phần hiện thực các yêu cầu phi chức năng 3 và 4.

4.1 Yêu cầu về I/O time < 2s của các câu truy vấn

Bộ dữ liệu mẫu của nhóm không lớn, nên thời gian thực thi các câu lệnh SQL rất nhỏ (0.00001s). Nhóm không có tập dữ liệu benchmark cụ thể để đánh giá chuẩn yêu cầu đã đặt ra.

Tuy nhiên, dựa trên lý thuyết, một số câu truy vấn có thể tăng tốc bằng cách sử dụng chỉ mục (INDEX), điển hình là các truy vấn SELECT cho việc truy xuất/tìm kiếm.

Theo yêu cầu chức năng, nhóm dự đoán các bảng CUSTOMER và PRODUCT là hai bảng vẫn có số lượng yêu cầu truy vấn cao (do thao tác tìm kiếm từ khách hàng và nhân viên). Bảng FAVOUR cũng liên quan tới chức năng tìm kiếm, tuy nhiên chỉ có quản lý sử dụng, lượng dữ liệu phát triển không lớn theo thời gian, nên không cần sử dụng chỉ mục để tăng tốc độ truy vấn.

4.1.1 Tạo index cho bảng CUSTOMER

Việc tìm kiếm trên bảng CUSTOMER dựa trên 3 cột: ssn, phone, email (tất cả đều là thuộc tính UNIQUE). Có hai cách để tạo index:

- Lấy tổ hợp 3 attribute ssn, phone và email làm index
- Với mỗi attribute, tạo một index riêng cho attribute đó

Chuỗi input của nhóm là một trong 3 attribute trên, vì vậy nhóm chọn cách 2.

```
1 CREATE UNIQUE INDEX ssn_idx ON CUSTOMER(ssn);  
2 CREATE UNIQUE INDEX phone_idx ON CUSTOMER(phone);  
3 CREATE UNIQUE INDEX email_idx ON CUSTOMER(email);
```

4.1.2 Tạo index cho bảng PRODUCT

Tương tự CUSTOMER, bảng PRODUCT cũng tìm kiếm dựa trên 2 giá trị ID hoặc name.

```
1 CREATE UNIQUE INDEX id_idx ON PRODUCT(ID);  
2 CREATE UNIQUE INDEX name_idx ON CUSTOMER(`name`);
```

4.2 Bảo mật thông tin của khách hàng, nhân viên và quản lý

Bảo mật hệ thống có hai phần: Bảo mật ứng dụng và bảo mật cơ sở dữ liệu. Trong phạm vi bài tập lớn, nhóm chỉ hiện thực bảo mật cho CSDL, cụ thể là cài đặt điều khiển truy cập cho các user dựa trên từng chức năng.

4.2.1 Cấp quyền cho nhóm đối tượng khách hàng

```
1  -- ===== Cấp quyền cho Khách hàng =====
2  --
3  DROP USER IF EXISTS 'crm_customer'@'localhost';
4  CREATE USER 'crm_customer'@'localhost' IDENTIFIED BY '${password}';
5
6  -- Cho phép xem, cập nhật thông tin cá nhân
7  GRANT SELECT, UPDATE ON SUPERMARKET_CRM.CUSTOMER TO
8  'crm_customer'@'localhost';
9
10 -- Cho phép xem, tạo (gửi) phản hồi
11 GRANT SELECT, INSERT ON SUPERMARKET_CRM.FEEDBACK TO
12 'crm_customer'@'localhost';
13
14 -- Cho phép xem lịch sử mua sắm và các thông tin liên quan về sản phẩm, chi
15 nhánh thực hiện giao dịch
16 GRANT SELECT ON SUPERMARKET_CRM.PURCHASE TO 'crm_customer'@'localhost';
17 GRANT SELECT ON SUPERMARKET_CRM.TRANSACTIONS TO 'crm_customer'@'localhost';
18 GRANT SELECT ON SUPERMARKET_CRM.PRODUCT TO 'crm_customer'@'localhost';
19 GRANT SELECT ON SUPERMARKET_CRM.SUPERMARKET_BRANCH TO
20 'crm_customer'@'localhost';
21
22 -- Cho phép xem thông báo từ nhân viên
23 GRANT SELECT ON SUPERMARKET_CRM.NOTICES TO 'crm_customer'@'localhost';
24 GRANT SELECT ON SUPERMARKET_CRM.RECEIVES TO 'crm_customer'@'localhost';
25
26 -- Cho phép xem kho lưu trữ ưu đãi cá nhân và thông tin ưu đãi
27 GRANT SELECT ON SUPERMARKET_CRM.FAVOUR TO 'crm_customer'@'localhost';
28 GRANT SELECT ON SUPERMARKET_CRM.VOUCHER_COUPON TO
29 'crm_customer'@'localhost';
30 GRANT SELECT ON SUPERMARKET_CRM.OWNS TO 'crm_customer'@'localhost';
31
32 -- Cho phép xóa voucher/coupon khỏi kho lưu trữ ưu đãi cá nhân
33 GRANT DELETE ON SUPERMARKET_CRM.OWNS TO 'crm_customer'@'localhost';
```

4.2.2 Cấp quyền cho nhóm đối tượng Nhân viên

```
1  -- ===== Cấp quyền cho Nhân viên =====
2  DROP USER IF EXISTS 'crm_staff'@'localhost';
3  CREATE USER 'crm_staff'@'localhost' IDENTIFIED BY '${password}';
4
5  -- Xem danh sách khách hàng và thông tin khách hàng
6  GRANT SELECT ON SUPERMARKET_CRM.CUSTOMER TO 'crm_staff'@'localhost';
7
8  -- Truy cập PURCHASE, TRANSACTIONS, PRODUCT để đưa ra thống kê cho Nhân viên về
9  xu hướng khách hàng
10 GRANT SELECT ON SUPERMARKET_CRM.PURCHASE TO 'crm_staff'@'localhost';
11 GRANT SELECT ON SUPERMARKET_CRM.TRANSACTIONS TO 'crm_staff'@'localhost';
12 GRANT SELECT ON SUPERMARKET_CRM.PRODUCT TO 'crm_staff'@'localhost';
13
14 -- Xem feedback của khách hàng
15 GRANT SELECT ON SUPERMARKET_CRM.FEEDBACK TO 'crm_staff'@'localhost';
16
17 -- Trả lời phản hồi của khách hàng
18 GRANT SELECT, INSERT ON SUPERMARKET_CRM.RESOLVES TO 'crm_staff'@'localhost';
```

```
19 -- Gửi thông báo cho khách hàng
20 GRANT SELECT, INSERT ON SUPERMARKET_CRM.NOTICES TO 'crm_staff'@'localhost';
21 GRANT SELECT, INSERT ON SUPERMARKET_CRM.RECEIVES TO 'crm_staff'@'localhost';
```

4.2.3 Cấp quyền cho nhóm đối tượng Quản lý

```
1 -- ===== Cấp quyền cho Quản lý ===== --
2 DROP USER IF EXISTS 'crm_manager'@'localhost';
3 CREATE USER 'crm_manager'@'localhost' IDENTIFIED BY 'phucvinh';
4
5 -- Thêm, xoá, sửa, cập nhật ưu đãi
6 GRANT SELECT, INSERT, DELETE, UPDATE ON SUPERMARKET_CRM.FAVOUR TO
  'crm_staff'@'localhost';
7 GRANT SELECT, INSERT, DELETE, UPDATE ON SUPERMARKET_CRM.VOUCHER_COUPON TO
  'crm_staff'@'localhost';
8
9 -- Xem thông tin về sản phẩm
10 GRANT SELECT ON SUPERMARKET_CRM.PRODUCT TO 'crm_staff'@'localhost';
```

4.2.4 Cập nhật cài đặt

Sau mỗi lần thay đổi điều khiển hệ thống, chạy lệnh sau để cập nhật cài đặt:

```
1 -- Cập nhật cài đặt
2 FLUSH PRIVILEGES;
```

5 Tổng kết

5.1 Phân chia khối lượng công việc

#	Thành viên	Công việc thực hiện	%
1	Nguyễn Phúc Vinh	Quản lí source code và tiến độ chung của nhóm. Tham gia B.IV.1, B.IV.2 và B.IV.3. Hiện thực code backend.	40%
2	Lê Xuân Nghĩa	Tham gia B.IV.2 và B.IV.3 Hiện thực code frontend.	30%
3	Vương Thanh Duyên	Tham gia B.IV.2, B.IV.3 và B.IV.4. Viết writing report và presentation report	30%

5.2 Tự đánh giá

5.2.1 Điểm đạt được

- Về mặt nội dung:
 - Nhóm đã hoàn thành căn bản các mục B.VI.1, B.VI.2, B.VI.3, B.VI.4.
 - Các yêu cầu chức năng trong BTL1 đã được cụ thể hoá và tách biệt thành các trang UI, mỗi trang có nhiều chức năng khác nhau.
 - Một số chức năng có độ phức tạp truy vấn cao
- Về mặt quản lí:
 - Nhóm phân chia khối lượng công việc đồng đều giữa các thành viên
 - Cách thức phân chia đảm bảo được các thành viên tham gia xuyên suốt và nắm rõ quá trình làm việc.
 - Tổ chức theo các sprint để cập nhật tiến độ và chỉnh sửa kịp thời
 - Xây dựng ứng dụng song song với triển khai cơ sở dữ liệu

5.2.2 Điểm chưa đạt

- Nhóm quản lý nhân lực chưa tốt
- Ứng dụng chưa hoàn thiện đủ các chức năng đã đặt ra trong BTL 1
- Chưa có benchmark cụ thể để đánh giá hiệu suất hệ thống

Repository

[1] *SupermarketCRM*. URL: <https://github.com/phucvinh57/SupermarketCRM>.