

TRƯỜNG CAO ĐẲNG CÔNG NGHỆ THỦ ĐỨC

KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC LẬP TRÌNH DI ĐỘNG 2

TÊN ĐỀ TÀI: QUẢN LÝ THỰC ĐƠN

Giảng Viên HD: Trương Bá Thái

NHÓM 06

Họ tên SV:

MSSV:

1. Nguyễn Chí Tân	17211TT1494
2. Phan Văn Tùng	17211TT1542
3. Trần Đăng Khoa	17211TT1738
4. Ngô Hữu Thế	17211TT4446

TP. Hồ Chí Minh, 29 Tháng 05 Năm 2019

Contents

Chương 1: Giới thiệu về đề tài	4
Chương 2: Phân tích thiết kế hệ thống	5
2.1 Giao diện ứng dụng	5
2.1.1 Mô tả giao diện	5
2.1.2 Mô tả chức năng	5
2.1.3 Code	6
2.2 Giao diện Login	7
2.2.1 Mô tả giao diện:	7
2.2.2 Mô tả chức năng:	8
2.2.3 Code	8
2.3 Giao diện Quản lí	10
2.3.1 Mô tả giao diện:	10
2.3.2 Mô tả chức năng	10
2.3.3 Code	11
2.4 Giao diện thêm, xóa, sửa nước uống	12
2.4.1 Mô Tả Giao diện:	13
2.4.2 Mô tả chức năng:	13
2.4.3 Code	13
2.5 Giao diện thêm, xóa, sửa thức ăn	16
2.5.1 Mô tả giao diện	16
2.5.2 Mô tả chức năng	17
2.5.3 Code	17
2.6 Giao diện chọn bàn của Nhân Viên	20
2.6.1 Mô Tả Giao diện:	20
2.6.2 Mô tả chức năng:	20
2.6.3 Code	21
2.7 Giao diện Menu	22
2.7.1 Mô tả giao diện:	22
2.7.2 Mô tả chức năng	23

2.7.3 Code	23
2.8 Giao diện xem chi tiết sản phẩm để đặt món.....	27
2.8.1 Phân tích mô tả:	27
2.8.2 Mô tả chức năng:	28
2.8.3 Code	28
2.9 Giao diện giỏ hàng:	30
2.9.1 Phân tích mô tả:	30
2.9.2 Mô tả chức năng:	31
2.9.3 Code	31
2.10 Giao diện hóa đơn	33
2.10.1 Mô tả giao diện:	33
2.10.2 Mô tả chức năng:	33
2.10.3 Code	34
Chương 3: Kết luận & kiến nghị	35

Danh sách thành viên:

STT	Họ & Tên	Đóng Góp	Điểm
1	Nguyễn Chí Tân	25%	
2	Trần Đăng Khoa	25%	
3	Phan Văn Tùng	25%	
4	Ngô Hữu Thế	25%	
5	Nguyễn Lê Thành Công	0%	

Chương 1: Giới thiệu về đề tài

- Với hệ thống phần mềm quản lý truyền thống, sẽ tốn kém thời gian cho việc ghi order - nhập vào hệ thống và sau đó được mã hóa vào bếp. Việc này đôi lúc gây trì trệ. Phần mềm sẽ khắc phục hết tất cả các nhược điểm trên. Thao tác gọi món nhanh chóng sẽ chất lượng hóa dịch vụ và tối thiểu hóa rủi ro tính tiền. Nhân viên ghi nhận món ăn trực tiếp trên máy tính bảng và order sẽ gửi thẳng vào bếp. Màn hình tự động chuyển đến khách hàng kế tiếp mà không cần chạy qua thu ngân hay nhà bếp.

Phục vụ chuyên nghiệp

- Đem đến một trải nghiệm tuyệt vời cho cả thực khách và nhân viên lễ tân. Thời gian phục vụ được rút gọn 50% nhờ tính năng Order, gọi món trực tiếp tại bàn bằng điện thoại/máy tính bảng

Chương 2: Phân tích thiết kế hệ thống

2.1 Giao diện ứng dụng



2.1.1 Mô tả giao diện

- Màn hình có 1 ImageView
- Màn hình có 1 ImageButton
- Màn hình có 1 Textview Đăng Nhập
- Màn hình có 1 ActionBar Menu: Login và Exit

2.1.2 Mô tả chức năng

- Nếu chọn ImageButton hoặc Textview Đăng nhập sẽ chuyển đến giao diện Đăng Nhập
- ActionBar menu:
 - Nếu chọn Login thì chuyển đến giao diện đăng nhập
 - Nếu chọn Exit thoát khỏi ứng dụng

2.1.3 Code

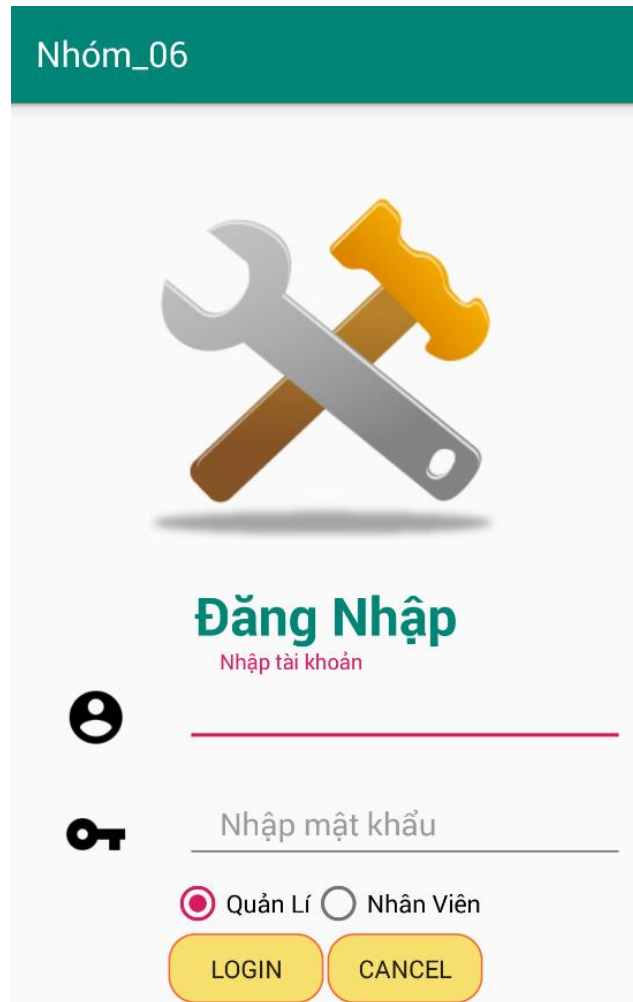
Nút đăng nhập

```
login.setOnClickListener((view) → {  
    Intent intent = new Intent( packageContext: MainActivity.this, Login.class);  
    startActivity(intent);  
});  
logintv.setOnClickListener((view) → {  
    Intent intent = new Intent( packageContext: MainActivity.this, Login.class);  
    startActivity(intent);  
});
```

Actionbar lựa chọn:

```
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.menu, menu);  
    return true;  
}  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.login:  
            Intent intent = new Intent( packageContext: MainActivity.this, Login.class);  
            startActivity(intent);  
            break;  
  
        case R.id.exit:  
  
            android.app.AlertDialog.Builder builder = new AlertDialog.Builder( context: MainActivity.this, android.R.style.Theme_DeviceDefault_Light_Dialog) ;  
            builder.setTitle("Thông Báo");  
            //      gắn icon  
            builder.setMessage("Bạn Có Muốn Thoát Khỏi Chương Trình Không?");  
            builder.setIcon(R.drawable.thongbao);  
            builder.setPositiveButton( text: "Không", (dialogInterface, i) → {  
                // để trống vì khi chọn nó sẽ k xảy ra skien gì cả  
            });  
            builder.setNegativeButton( text: "Có", (dialogInterface, i) → {  
                Intent startMain = new Intent(Intent.ACTION_MAIN);  
                startMain.addCategory(Intent.CATEGORY_HOME);  
                startActivity(startMain);  
                finish();  
            });  
            builder.show();  
            break;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

2.2 Giao diện Login



The login interface features a teal header with the text "Nhóm_06". Below the header is a large graphic of crossed tools: a grey wrench and a yellow hammer. The title "Đăng Nhập" is displayed in teal, with the subtitle "Nhập tài khoản" in red below it. There are two input fields: the first is preceded by a person icon and the second by a key icon, both with the placeholder text "Nhập mật khẩu". Below the input fields are two radio buttons; the first is selected and labeled "Quản Lý", and the second is labeled "Nhân Viên". At the bottom are two yellow buttons labeled "LOGIN" and "CANCEL".

2.2.1 Mô tả giao diện:

- Có 2 EditText dùng để người dùng nhập Tài Khoản và Mật Khẩu.
- Có 2 Radio Button dùng để chọn quyền đăng nhập(quyền quản lý hoặc quyền nhân viên).
- 1 Nút đăng nhập dùng để chuyển trang.
- 1 Nút Hủy bỏ dùng để quay lại màn hình trước(màn hình welcome).

2.2.2 Mô tả chức năng:

- Nếu muốn đăng nhập vào phần mềm thì người dùng cần nhập đúng tài khoản và mật khẩu sau đó click vào nút đăng nhập.
 - ✓ Nếu tài khoản mật khẩu đúng thì chương trình sẽ chuyển vào màn hình tiếp theo.
 - ✓ Khi nhập sai tài khoản hoặc mật khẩu thì chương trình sẽ thông báo lỗi và không chuyển màn hình.
 - ✓ Nếu không nhập tài khoản mật khẩu thì chương trình sẽ báo lỗi
- Khi chọn nút hủy bỏ thì ứng dụng sẽ quay lại màn hình trước(Màn hình Welcome).

2.2.3 Code

Button đăng nhập

```
btnLogin.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        if(edtname.getText().length() != 0 && edtpass.getText().length() != 0) {  
            if (quanli.isChecked()) {  
                quatrangquanli();  
            } else {  
                quatrangnhanvien();  
            }  
        }  
        else  
        {  
            Toast.makeText(context: Login.this, text: "Bạn chưa nhập đầy đủ thông tin !!!", Toast.LENGTH_SHORT).show();  
        }  
    }  
});
```

Button quay lại

```
quaylai.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent intent = new Intent(packageContext: Login.this, MainActivity.class);  
        startActivity(intent);  
    }  
});
```


Khởi tạo mảng tài khoản và mật khẩu nhân viên, quản lí

```
public void KhoiTao()
{
    data.add(new QuanLy( sname: "ql1", spass: "12345" ));
    data.add(new QuanLy( sname: "ql2", spass: "12345" ));
}

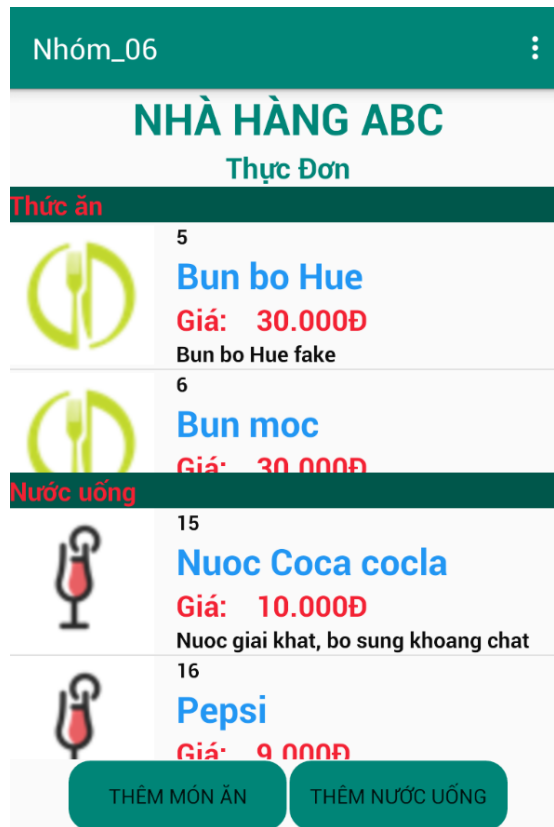
public void KhoiTao1()
{
    data1.add(new NhanVien( name: "nv1", pass: "12345" ));
    data1.add(new NhanVien( name: "nguyenvanb", pass: "12345" ));
}
```

Hàm kiểm tra tài khoản nhân viên và quản lí để chuyển trang

```
public void quatrangquanli()
{
    for (int i = 0; i < data.size(); i++){
        if (edtname.getText().toString().trim().equals(data.get(i).sname) && edtpass.getText().toString().trim().equals(data.get(i).spass) )
        {
            Intent intent = new Intent( packageContext: Login.this, Admin.class);
            startActivity(intent);
            break;
        }
        else {
            Toast.makeText( context: this, text: "Bạn đã nhập sai tài khoản hoặc mật khẩu!!! Vui lòng nhập lại", Toast.LENGTH_SHORT).show();
        }
    }
}

public void quatrangnhanvien() {
    for (int i = 0; i < data.size(); i++){
        if (edtname.getText().toString().trim().equals(data1.get(i).name) && edtpass.getText().toString().trim().equals(data1.get(i).pass) )
        {
            Intent intent = new Intent( packageContext: Login.this, ChonBan.class);
            startActivity(intent);
            break;
        }
        else {
            Toast.makeText( context: this, text: "Bạn đã nhập sai tài khoản hoặc mật khẩu!!! Vui lòng nhập lại", Toast.LENGTH_SHORT).show();
        }
    }
}
```

2.3 Giao diện Quản lí



2.3.1 Mô tả giao diện:

- Màn hình có 2 List View hiển thị danh sách món ăn và nước uống
- Màn hình có 2 Button Thêm món ăn và Thêm Nước Uống
- Màn hình có 1 ActionBar gồm 2 lựa chọn: Logout và Exit

2.3.2 Mô tả chức năng

- Màn hình có 1 ActionBar gồm 2 lựa chọn: Logout và Exit
 - ✓ Nếu chọn Logout thì thoát khỏi giao diện quản lí quay về màn hình Đăng nhập
 - ✓ Nếu chọn Exit thì thoát khỏi ứng dụng
- Màn hình có 2 list view hiển thị các món ăn và các loại nước từ database
- Button thêm món ăn: khi chọn vào sẽ chuyển sang giao diện thêm món ăn
- Button thêm nước uống : khi chọn vào sẽ chuyển sang giao diện thêm nước uống

2.3.3 Code

Hàm Load thức ăn, nước uống từ database

```
public void loadFood() {
    DatabaseFood dbfood = new DatabaseFood( context: this);
    data.clear();
    dbfood.getfood(data);
    // adapterfood.notifyDataSetChanged();
}
public void loadDrink() {
    DatabaseDrink db = new DatabaseDrink( context: this);
    datadrink.clear();
    db.getDrink(datadrink);
    // adapterdrink.notifyDataSetChanged();
}
```

Đổ dữ liệu vào Listview

```
loadFood();
loadDrink();
FoodAdapter adapter = new FoodAdapter( context: this, R.layout.listview_food_item, data);
listView1.setAdapter(adapter);
DrinkAdapter adapter1 = new DrinkAdapter( context: this, R.layout.listview_drink_item, datadrink);
listView2.setAdapter(adapter1);
```

Button thêm thức ăn hoặc nước uống

```
btnadfood.setOnClickListener((view) → {
    Intent intent = new Intent( packageContext: Admin.this, Addfood.class);
    startActivity(intent);
});
btnaddDrink.setOnClickListener((view) → {
    Intent intent = new Intent( packageContext: Admin.this, AddDrink.class);
    startActivity(intent);
});
```




ActionBar lựa chọn

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.logout:
            Intent intent = new Intent( packageContext: Admin.this, Login.class);
            startActivity(intent);
            break;

        case R.id.exit:

            android.app.AlertDialog.Builder builder = new AlertDialog.Builder( context: Admin.this, android.R.style.Theme_DeviceDefault_Light_Dialog) ;
            builder.setTitle("Thông Báo");
            //      gắn icon
            builder.setMessage("Bạn Có Muốn Thoát Khỏi Chương Trình Không?");
            builder.setIcon(R.drawable.thongbao);
            builder.setPositiveButton( text: "Không", (dialogInterface, i) → {
                // để trống vì khi chọn nó sẽ k xảy ra skien gì cả
            });
            builder.setNegativeButton( text: "Có", (dialogInterface, i) → {
                Intent startMain = new Intent(Intent.ACTION_MAIN);
                startMain.addCategory(Intent.CATEGORY_HOME);
                startActivity(startMain);
                finish();
            });
            builder.show();
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

2.4 Giao diện thêm, xóa, sửa nước uống

Nhóm_06	
THÊM XÓA SỬA NƯỚC UỐNG	
ID:	
Tên nước uống:	
Giới thiệu:	
Giá:	
<div>THÊM CẬP NHẬT HỦY</div>	
	15 Nuoc Coca cocla Giá: 10.000Đ Nuc giai khat, bo sung khoang chat
	16 Pepsi Giá: 9.000Đ Nuc giai khat Viet Nam
	22 Sting

2.4.1 Mô Tả Giao diện:

- Màn hình có 1 TextView để hiển thị tiêu đề của trang.
- Màn hình có 1 TextView để hiển thị id của nước uống.
- Có 3 EditText để người dùng nhập 3 thông tin của sản phẩm(tên, chi tiết và giá của sản phẩm).
- 3 nút Button: Thêm, Cập nhật, Hủy bỏ
- 1 ListView hiển thị danh sách nước uống.

2.4.2 Mô tả chức năng:

- Muốn thêm nước uống, người dùng phải nhập đầy đủ 3 thông tin là: tên, mô tả và giá tiền của sản phẩm, nếu bỏ trống ứng dụng sẽ hiển thị thông báo lỗi. Sau khi nhập đầy đủ thông tin thì người dùng chọn nút Thêm để thêm dữ liệu mới vào database và sau đó load lên listview.
- Nút Cập nhật mặc định nó sẽ bị ẩn đi, nếu muốn cập nhật người dùng cần chọn vào 1 sản phẩm ở trong Listview. Sau khi người dùng chọn vào 1 sản phẩm thì các thông tin của sản phẩm sẽ được hiển thị trên các ô EditText để người dùng sửa và lưu lại.(Khi đang thực hiện cập nhật thì nút Thêm sẽ bị ẩn đi).
- Nếu người dùng muốn quay lại trang trước thì chọn nút Hủy Bỏ, nút này sẽ đưa ứng dụng trở về màn hình Admin.
- Nếu muốn xóa 1 sản phẩm thì người dùng chỉ cần giữ lâu vào 1 sản phẩm và nó sẽ hiện thông báo và 2 lựa chọn: Nếu chọn Có thì sẽ xóa khỏi database món ăn tại vị trí click vào, nếu chọn không thì không xóa.

2.4.3 Code

Thêm dữ liệu vào database

```
bntAddDrink.setOnClickListener((view) -> {  
    if(edtName.getText().length() !=0 && edtprice.getText().length() !=0 && edtintroduce.getText().length() !=0)  
    {  
        saveDB();  
        loadDB();  
    }  
    else  
    {  
        Toast.makeText( context: AddDrink.this, text: "Bạn chưa nhập đầy đủ thông tin !!!", Toast.LENGTH_SHORT).show();  
    }  
});  
cancel.setOnClickListener((view) -> {  
    Intent intent = new Intent( packageContext: AddDrink.this, Admin.class);  
    startActivity(intent);  
});
```

Hàm save và load database

```
public void saveDB()
{
    DatabaseDrink dbDrink = new DatabaseDrink( context: this);
    Drink drink = new Drink();
    drink.setTitle1(editName.getText().toString() + "");
    drink.setMota(editintrocude.getText().toString() + "");
    drink.setPrice(Integer.parseInt( S: edtprice.getText() + " ));
    dbDrink.saveDrink(drink);
    editName.setText("");
    editintrocude.setText("");
    edtprice.setText("");
    txtiddrink.setText("");
}

public void loadDB() {
    DatabaseDrink db = new DatabaseDrink( context: this);
    data.clear();
    db.getDrink(data);
    adapter.notifyDataSetChanged();
}
```

Cập nhật dữ liệu vào database

```
btnUpdateDrink.setOnClickListener((view) -> {
    if(editName.getText().length() !=0 && edtprice.getText().length() !=0 && editintrocude.getText().length() !=0)
    {
        btnUpdateDrink.setEnabled(false);
        bntAddDrink.setEnabled(true);
        Drink drink =new Drink();
        drink.setId(Integer.parseInt(String.valueOf(txtiddrink.getText())));
        drink.setTitle1(editName.getText() + "");
        drink.setMota(editintrocude.getText() + "");
        drink.setPrice(Integer.parseInt( S: edtprice.getText() + "));
        int result = dbdrink.updateDrink(drink);
        if(result > 0)
        {
            data.clear();
            dbdrink.getDrink(data);
            adapter.notifyDataSetChanged();

            txtiddrink.setText("");
            editName.setText("");
            editintrocude.setText("");
            edtprice.setText("");
        }
        else
        {
            Toast.makeText( context: AddDrink.this, text: "Bạn chưa nhập đầy đủ thông tin !!!", Toast.LENGTH_SHORT).show();
        }
    }
});
```

Xóa dữ liệu Database

```
listViewDrink.setOnItemLongClickListener((adapterView, view, i, l) -> {
    Drink drink = data.get(i);
    txtiddrink.setText(String.valueOf(drink.getId()));
    android.app.AlertDialog.Builder builder = new AlertDialog.Builder( context: AddDrink.this, android.R.style.Theme_DeviceDefault_Light_Dialog) ;
    builder.setTitle("Thông Báo");
    builder.setMessage("Bạn có muốn xóa không?");
    builder.setIcon(R.drawable.thongbao);
    builder.setPositiveButton( text: "Có", (dialogInterface, i) -> {
        deleteAccount();
    });
    builder.setNegativeButton( text: "Không", (dialogInterface, i) -> {
        txtiddrink.setText("");
        edtName.setText("");
        edtintrocude.setText("");
        edtprice.setText("");
    });
    builder.show();
    return false;
});
```

Hàm xóa databse

```
public void deleteAccount() {
    DatabaseDrink db= new DatabaseDrink( context: this);
    boolean result = db.deleteDrink(txtiddrink.getText().toString());
    if (result) {
        data.clear();
        db.getDrink(data);
        adapter.notifyDataSetChanged();
        txtiddrink.setText("");
        edtName.setText("");
        edtintrocude.setText("");
        edtprice.setText("");
        Toast.makeText( context: this, text: "Delete drink thành công ^^ ", Toast.LENGTH_SHORT).show();
    } else
        Toast.makeText( context: this, text: "Delete fail", Toast.LENGTH_SHORT).show();
}
```

2.5 Giao diện thêm, xóa, sửa thức ăn

Nhóm_06

THÊM XÓA SỬA MÓN ĂN

ID

Tên thức ăn::


Giới thiệu:

Giá:


THÊM

CẬP NHẬT


HỦY BỎ



5
Bun bo Hue
Giá: 30.000Đ
Bun bo Hue fake



6
Bun moc
Giá: 30.000Đ
Bun moc gia truyền, đặc biệt thơm ngon



7
Bun Cha Ha Noi

2.5.1 Mô tả giao diện

- Màn hình có 1 TextView để hiển thị tiêu đề của trang.
- Màn hình có 1 TextView để hiển thị id của món ăn.
- Có 3 EditText để người dùng nhập 3 thông tin của sản phẩm(tên, chi tiết và giá của sản phẩm).
- 3 nút Button: Thêm, Cập nhật, Hủy bỏ
- 1 ListView hiển thị danh sách món ăn.

2.5.2 Mô tả chức năng

- Muốn thêm thức ăn, người dùng phải nhập đầy đủ 3 thông tin là: tên, mô tả và giá tiền của sản phẩm, nếu bỏ trống ứng dụng sẽ hiển thị thông báo lỗi. Sau khi nhập đầy đủ thông tin thì người dùng chọn nút Thêm để thêm dữ liệu mới vào database và sau đó load lên listview.
- Nút Cập nhật mặc định nó sẽ bị ẩn đi, nếu muốn cập nhật người dùng cần chọn vào 1 sản phẩm ở trong Listview. Sau khi người dùng chọn vào 1 sản phẩm thì các thông tin của sản phẩm sẽ được hiển thị trên các ô EditText để người dùng sửa và lưu lại. (Khi đang thực hiện cập nhật thì nút Thêm sẽ bị ẩn đi).
- Nếu người dùng muốn quay lại trang trước thì chọn nút Hủy Bỏ, nút này sẽ đưa ứng dụng trở về màn hình Admin.
- Nếu muốn xóa 1 sản phẩm thì người dùng chỉ cần giữ lâu vào 1 sản phẩm và nó sẽ hiện thông báo và 2 lựa chọn: Nếu chọn Có thì sẽ xóa khỏi database món ăn tại vị trí click vào, nếu chọn không thì không xóa.

2.5.3 Code

Thêm dữ liệu vào database

```
btnThemFood.setOnClickListener((view) -> {  
  
    if(addname.getText().length() != 0 && addgioithieu.getText().length() != 0 && addprice.getText().length() != 0 )  
    {  
        saveDB();  
        loadDB();  
    }  
    else  
    {  
        Toast.makeText( context: Addfood.this, text: "Bạn chưa nhập đầy đủ thông tin", Toast.LENGTH_SHORT).show();  
    }  
  
});
```

Hàm save và load từ database

```
public void saveDB() {
    DatabaseFood dbfood = new DatabaseFood( context: this);
    Food food = new Food();
    food.setTitle(addname.getText().toString() + "");
    food.setMota(addgioithieu.getText().toString() + "");
    food.setPrice(Integer.parseInt( $: addprice.getText().toString() + ""));
    dbfood.saveFood(food);
    txtid.setText("");
    addname.setText("");
    addgioithieu.setText("");
    addprice.setText("");
}

public void loadDB() {
    DatabaseFood db = new DatabaseFood( context: this);
    data.clear();
    db.getfood(data);
    adapter.notifyDataSetChanged();
}
```

Cập nhật vào database

```
capnhatFood.setOnClickListener((view) → {
    if(addname.getText().length() != 0 && addgioithieu.getText().length() != 0 && addprice.getText().length() != 0 )
    {
        capnhatFood.setEnabled(false);
        btnThemFood.setEnabled(true);
        Food food = new Food();
        food.setId(Integer.parseInt(String.valueOf(txtid.getText())));
        food.setTitle(addname.getText() + "");
        food.setMota(addgioithieu.getText() + "");
        food.setPrice(Integer.parseInt( $: addprice.getText() + ""));
        int result = dbfood.updateFood(food);
        if (result > 0) {
            data.clear();
            dbfood.getfood(data);
            adapter.notifyDataSetChanged();
        }
        txtid.setText("");
        addname.setText("");
        addgioithieu.setText("");
        addprice.setText("");
    }
    else
    {
        Toast.makeText( context: Addfood.this, text: "Bạn chưa nhập đầy đủ thông tin !!!", Toast.LENGTH_SHORT).show();
    }
});
```

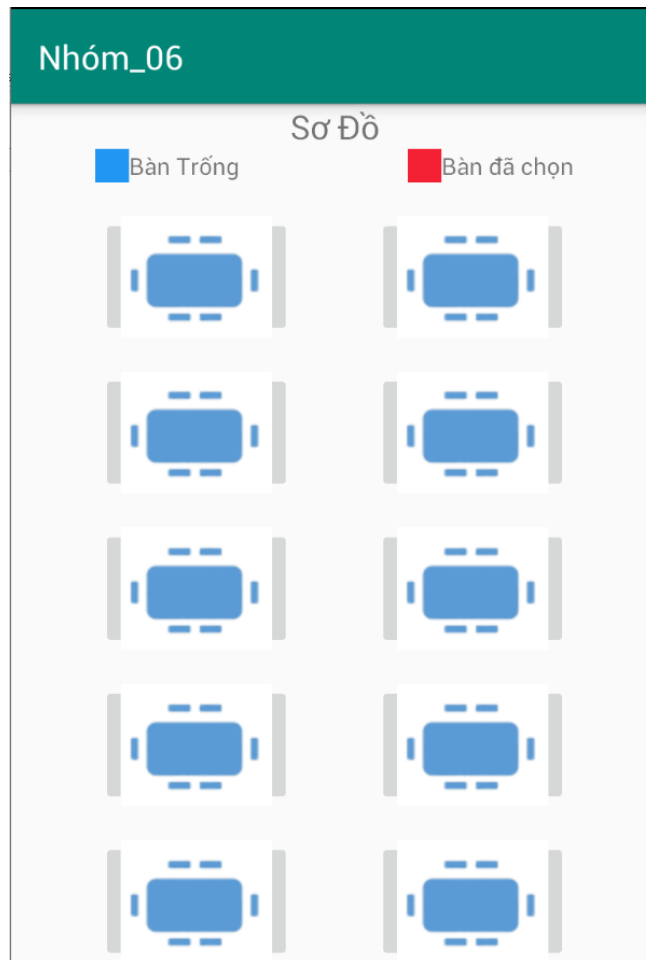
Xóa dữ liệu từ database

```
listViewFood.setOnItemClickListener((adapterView, view, i, l) → {
    Food food = data.get(i);
    txtid.setText(String.valueOf(food.getId()));
    android.app.AlertDialog.Builder builder = new android.app.AlertDialog.Builder( context: Addfood.this, android.R.style.Theme_DeviceDefault_Light_Dialog) ;
    builder.setTitle("Thông Báo");
    builder.setMessage("Bạn có muốn xóa không?");
    builder.setIcon(R.drawable.thongbao);
    builder.setPositiveButton( text: "Có", (dialogInterface, i) → {
        deleteAccount();
    });
    builder.setNegativeButton( text: "Không", (dialogInterface, i) → {
        txtid.setText("");
        addname.setText("");
        addgioithieu.setText("");
        addprice.setText("");
    });
    builder.show();
    return false;
});
```

Hàm xóa database

```
public void deleteAccount() {
    DatabaseFood db = new DatabaseFood( context: this);
    boolean result = db.deleteFood(txtid.getText().toString());
    if (result) {
        data.clear();
        db.getfood(data);
        adapter.notifyDataSetChanged();
        txtid.setText("");
        addname.setText("");
        addgioithieu.setText("");
        addprice.setText("");
        Toast.makeText( context: this, text: "Delete food thành công ^_^ ", Toast.LENGTH_SHORT).show();
    } else
        Toast.makeText( context: this, text: "Delete fail", Toast.LENGTH_SHORT).show();
}
```

2.6 Giao diện chọn bàn của Nhân Viên



2.6.1 Mô Tả Giao diện:

- Màn hình có 10 ImageButton hiển thị danh sách bàn
- Màn hình có 1 ActionBar hiển thị 2 lựa chọn: Logout và Exit

2.6.2 Mô tả chức năng:

Khi nhân viên chọn bàn khách ngồi để khách ngồi sẽ xuất hiện thông báo “Hãy nhập lựa chọn”

+ Nếu chọn ‘Chọn bàn’ sẽ chuyển đến menu để chọn thức ăn, bàn sẽ được chuyển sang trạng thái bàn đã chọn

+ Nếu chọn “Hủy bàn” sẽ delete đơn hàng của bàn và chuyển trạng thái bàn còn trống

+ Màn hình có 1 ActionBar gồm 2 lựa chọn: Logout và Exit

- ✓ Nếu chọn Logout thì thoát khỏi giao diện quản lí quay về màn hình Đăng nhập
- ✓ Nếu chọn Exit thì thoát khỏi ứng dụng

2.6.3 Code

Sự kiện khi chọn bàn

```
ban1.setOnClickListener((view) -> {
    android.app.AlertDialog.Builder builder = new android.app.AlertDialog.Builder( context: ChonBan.this, android.R.style.Theme_DeviceDefault_Light_Dialog) ;
    builder.setTitle("Thông Báo");
    //      gắn icon

    builder.setMessage("Hãy lựa chọn bên dưới");
    builder.setIcon(R.drawable.thongbao);
    builder.setPositiveButton( text: "Hủy Bàn", (dialogInterface, i) -> {
        ban1.setImageResource(R.drawable.controngme);
    });
    builder.setNegativeButton( text: "Chọn Bàn", (dialogInterface, i) -> {
        Intent intent = new Intent( packageContext: ChonBan.this, Listmenu.class);
        intent.putExtra( name: "ban1", value: 01);
        startActivity(intent);
        ban1.setImageResource(R.drawable.datbane);
    });
    builder.show();
});
```

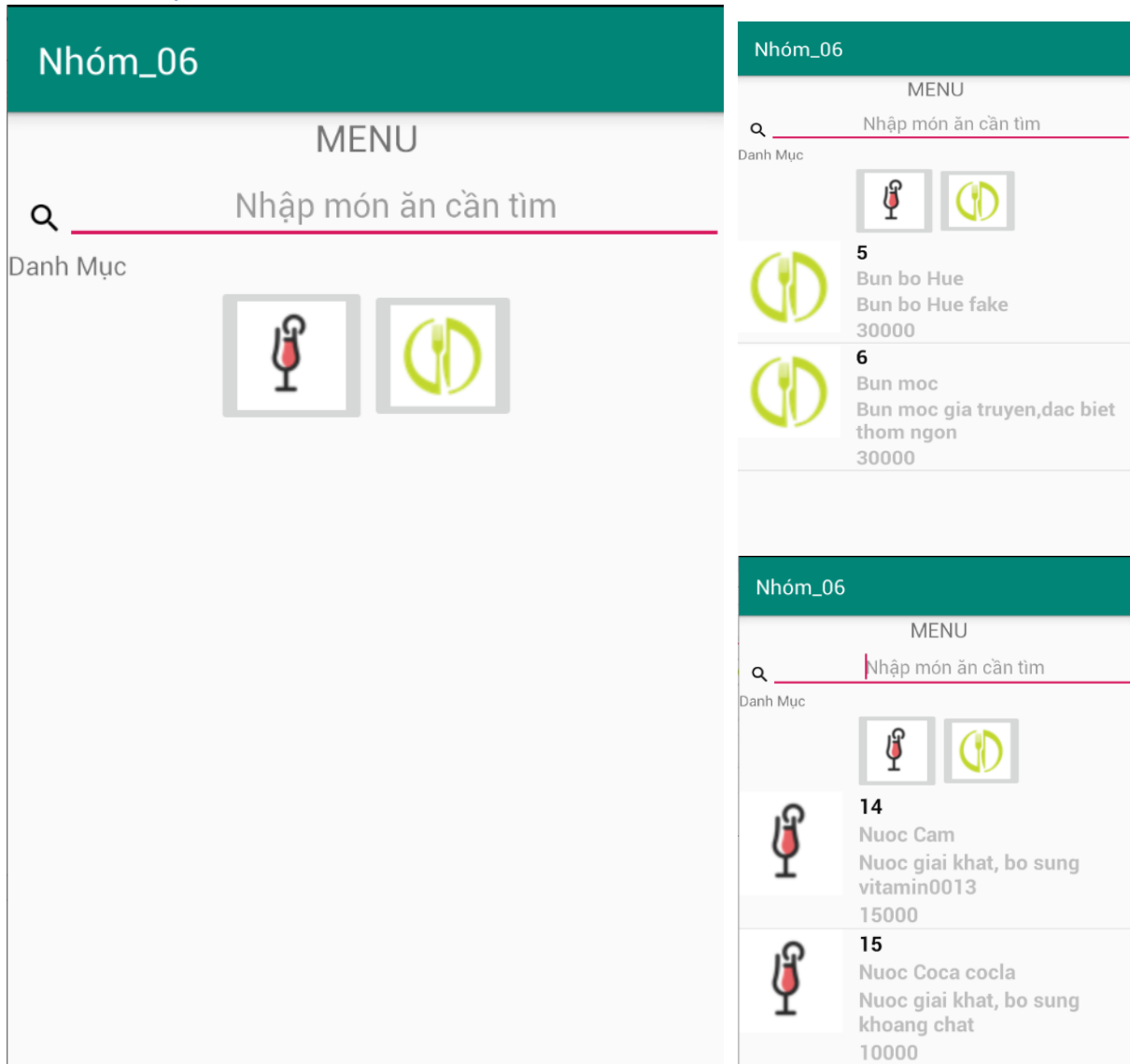
ActionBar lựa chọn

```
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.logout:
            Intent intent = new Intent( packageContext: Admin.this, Login.class);
            startActivity(intent);
            break;

        case R.id.exit:

            android.app.AlertDialog.Builder builder = new AlertDialog.Builder( context: Admin.this, android.R.style.Theme_DeviceDefault_Light_Dialog) ;
            builder.setTitle("Thông Báo");
            //      gắn icon
            builder.setMessage("Bạn Có Muốn Thoát Khỏi Chương Trình Không?");
            builder.setIcon(R.drawable.thongbao);
            builder.setPositiveButton( text: "Không", (dialogInterface, i) -> {
                // để trống vì khi chọn nó sẽ k xảy ra skien gì cả
            });
            builder.setNegativeButton( text: "Có", (dialogInterface, i) -> {
                Intent startMain = new Intent(Intent.ACTION_MAIN);
                startMain.addCategory(Intent.CATEGORY_HOME);
                startActivity(startMain);
                finish();
            });
            builder.show();
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

2.7 Giao diện Menu



2.7.1 Mô tả giao diện:

Giới thiệu:

- Màn hình để tìm kiếm, xem và chọn thực đơn

Đối tượng

- Dành cho nhân viên

Điều kiện trước:

- Nhân viên phải đăng nhập
- Nhân viên phải chọn bàn

Điều kiện sau:

- Nhân viên có thể tìm kiếm
- Nhân viên có thể xem thực đơn theo danh mục
- Chuyển đến màn hình chi tiết

Mô tả:

- Một EditText nhập từ khóa tìm kiếm
- 2 ImageButton Danh mục hiển thị danh sách
- 1 ListView danh sách
- 2 ImageButton tìm kiếm
- Một ActionBar giỏ hàng
- Một ActionBar lựa chọn

2.7.2 Mô tả chức năng

- Sẽ có 1 Edit text để nhập từ khóa tìm kiếm. Khi nhập xong từ khóa bạn chọn tìm kiếm theo thức ăn hoặc nước uống, kết quả tìm được sẽ hiển thị trên Listview.
- Sẽ có danh mục lựa chọn của menu, khi người dùng chọn vào danh mục thức ăn hoặc nước uống sẽ load dữ liệu food hoặc drink lên Listview.
- Khi người dùng chọn vào một sản phẩm sẽ chuyển sang giao diện xem chi tiết để đặt sản phẩm
- Một ActionBar giỏ hàng chuyển đến giao diện giỏ hàng xem các món đã chọn
- Một ActionBar với 2 lựa chọn Back và Exit
 - Nếu chọn back sẽ quay về giao diện chọn bàn
 - Nếu chọn Exit sẽ thoát khỏi chương trình

2.7.3 Code

Hàm load database Food

```
public void loadFood() {  
    DatabaseFood dbfood = new DatabaseFood( context: this);  
    datafood.clear();  
    dbfood.getfood(datafood);  
}
```

Hàm load database Drink

```
public void loadDrink() {  
    DatabaseDrink db = new DatabaseDrink( context: this);  
    datadrink.clear();  
    db.getDrink(datadrink);  
    // |.notifyDataSetChanged();  
}
```

Sự kiện click danh mục Food hoặc Drink và click vào item trên ListView truyền dữ liệu, chuyển sang giao diện xem chi tiết

```
//su kien click Imagebuttonn Drink  
drink.setOnClickListener((view) → {  
    loadDrink();  
    listViewWater.setAdapter(adapter1);  
    listViewWater.setOnItemClickListener((adapterView, view, i, l) → {  
        Intent intent = new Intent(getApplicationContext(), Information_product.class);  
        intent.putExtra( name: "thongtindrunk", datadrink.get(i));  
        startActivity(intent);  
    });  
});
```

```
//su kien click Imagebuttonn Drink  
food.setOnClickListener((view) → {  
    loadFood();  
    listViewFood.setAdapter(adapter);  
    listViewFood.setOnItemClickListener((adapterView, view, i, l) → {  
        Intent intent = new Intent(getApplicationContext(), Information_food.class);  
        intent.putExtra( name: "thongtinfood", datafood.get(i));  
        startActivity(intent);  
    });  
});
```


Chức năng tìm kiếm

Hàm tìm kiếm thức ăn

```
public void timKiemFood(String namefood)
{
    final FoodAdapter adapter = new FoodAdapter( context: this, R.layout.listview_food_item, datafood);
    DatabaseFood dbfood = new DatabaseFood( context: this);
    datafood.clear();
    dbfood.findFood(namefood,datafood);
    listViewFood.setAdapter(adapter);
    //adapterdrink.notifyDataSetChanged();
}
```

Sự kiện click tìm kiếm

```
txtfood.setOnClickListener((view) → {
    if(timkiem.getText().length() != 0)
    {
        listViewFood.setAdapter(adapter);
        String name= timkiem.getText().toString();
        timKiemFood(name);
    }
    else{
        Toast.makeText( context: Listmenu.this, text: "Bạn chưa nhập thông tin tìm kiếm", Toast.LENGTH_SHORT).show();
    }
});
btntimkiemfood.setOnClickListener((view) → {
    if(timkiem.getText().length() != 0)
    {
        listViewFood.setAdapter(adapter);
        String name= timkiem.getText().toString();
        timKiemFood(name);
    }
    else{
        Toast.makeText( context: Listmenu.this, text: "Bạn chưa nhập thông tin tìm kiếm", Toast.LENGTH_SHORT).show();
    }
});
txtdrink.setOnClickListener((view) → {
```

Hàm tìm kiếm nước uống

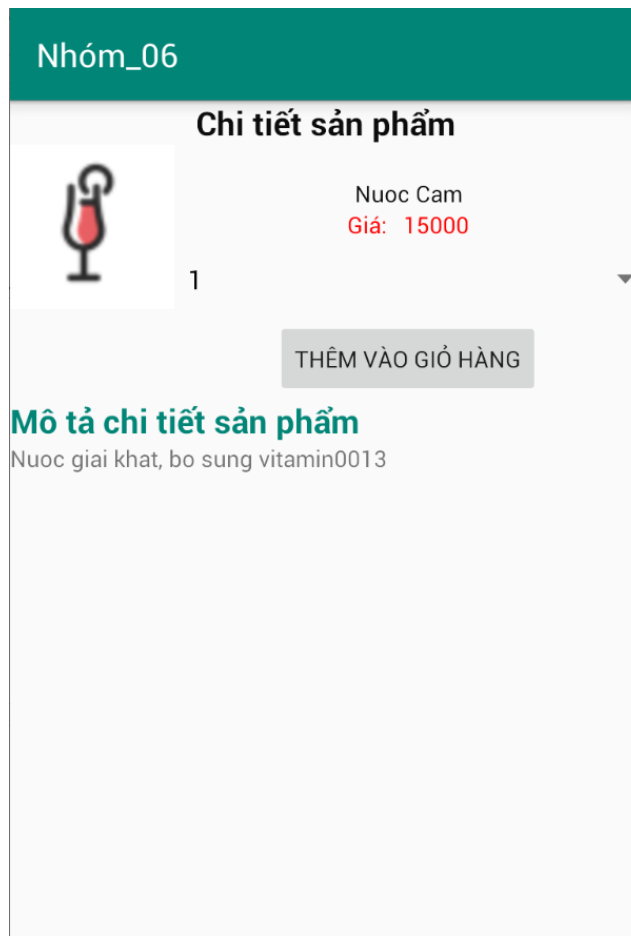
```
public void timKiemDrink(String namedrink)
{
    final DrinkAdapter adapter1 = new DrinkAdapter( context: this, R.layout.listview_drink_item, datadrink);
    DatabaseDrink db = new DatabaseDrink( context: this);
    datadrink.clear();
    db.findDrink(namedrink, datadrink);
    listViewWater.setAdapter(adapter1);
}
```

Sự kiện click tìm kiếm

```
drink.setOnClickListener((view) → {
    loadDrink();
    listViewWater.setAdapter(adapter1);
    listViewWater.setOnItemClickListener((adapterView, view, i, l) → {
        Intent intent = new Intent(getApplicationContext(), Information_drink.class);
        intent.putExtra( name: "thongtindrunk", datadrink.get(i));
        startActivity(intent);
    });
});

food.setOnClickListener((view) → {
    loadFood();
    listViewFood.setAdapter(adapter);
    listViewFood.setOnItemClickListener((adapterView, view, i, l) → {
        Intent intent = new Intent(getApplicationContext(), Information_food.class);
        intent.putExtra( name: "thongtinfood", datafood.get(i));
        startActivity(intent);
    });
});
```

2.8 Giao diện xem chi tiết sản phẩm để đặt món



2.8.1 Phân tích mô tả:

- Màn hình Có 1 Image để hiển thị hình ảnh của sản phẩm
- Màn hình có 2 TextView hiển thị tiêu đề: “Chi tiết sản phẩm” và “Mô tả chi tiết sản phẩm”
- Màn hình 1 TextView hiển thị tên sản phẩm
- Màn hình có 1 TextView hiển thị giá
- Màn hình có 1 Combobox để chọn số lượng sản phẩm muốn gọi
- Màn hình có 1 TextView thể hiện mô tả sản phẩm
- Màn hình có ActionBar giỏ hàng
- Màn hình có 1 ActionBar với 2 lựa chọn: Back và Exit

2.8.2 Mô tả chức năng:

- Người dùng có thể xem chi tiết sản phẩm ,giới thiệu về sản phẩm
- Người dùng có thể thay đổi số lượng muốn gọi món thông qua combobox số lượng. Khi người dùng click vào combobox này, combobox sẽ hiển thị 1 dãy các con số để người dùng chọn, sau khi chọn số lượng xong người dùng sẽ ấn vào nút thêm giỏ hàng để gọi được món.
- Nếu chọn giỏ hàng thì sẽ chuyển đến giỏ hàng xem các món đã gọi
- Với ActionBar menu có 2 lựa chọn:
 - Nếu chọn Back sẽ quay về giao diện chọn bàn
 - Nếu chọn Exit sẽ thoát khỏi ứng dụng

2.8.3 Code

Thêm món ăn chọn vào giỏ hàng

```
btnAddDrink.setOnClickListener((view) -> {
    if (MainActivity.manggiohang.size() > 0) {
        int sl = Integer.parseInt(spinner.getSelectedItem().toString());
        boolean exits = false;
        for (int i = 0; i < MainActivity.manggiohang.size(); i++) {
            if (MainActivity.manggiohang.get(i).getTen().equals( TenChiTietDrink)) {
                MainActivity.manggiohang.get(i).setSoLuong(MainActivity.manggiohang.get(i).getSoLuong() + sl);
                if (MainActivity.manggiohang.get(i).getSoLuong() >= 10) {
                    MainActivity.manggiohang.get(i).setSoLuong(10);
                }
                MainActivity.manggiohang.get(i).setGia(GiaDrink * MainActivity.manggiohang.get(i).getSoLuong());
                exits = true;
            }
        }
        if (exits == false) {
            int soluong = Integer.parseInt(spinner.getSelectedItem().toString());
            int giamoi = soluong * GiaDrink;
            MainActivity.manggiohang.add(new HoaDon(idrink, TenChiTietDrink, giamoi, soluong));
        }
    } else {
        int soluong = Integer.parseInt(spinner.getSelectedItem().toString());
        int giamoi = soluong * GiaDrink;
        MainActivity.manggiohang.add(new HoaDon(idrink, TenChiTietDrink, giamoi, soluong));
    }

    Intent intent = new Intent( packageContext: Information_drink.this, DonHang.class);
    startActivity(intent);
});
```

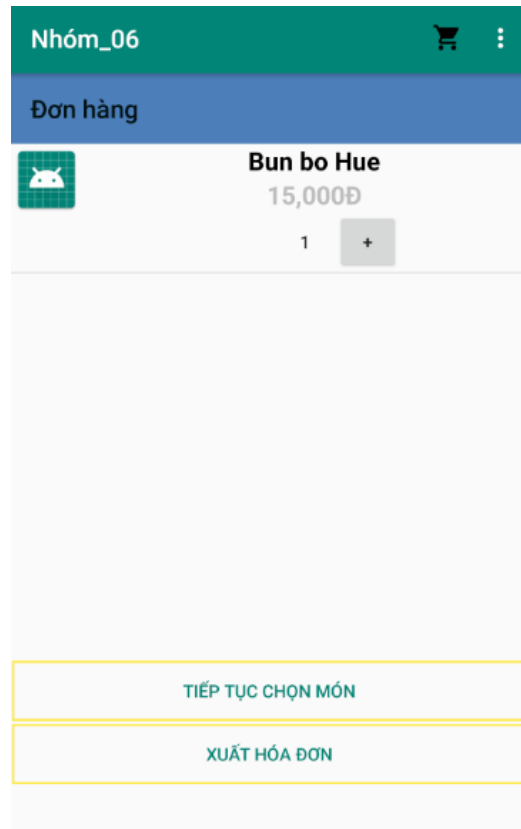
Hàm nhận dữ liệu từ màn hình Thực đơn

```
private void getSanPhamDrink() {  
  
    Drink drink = (Drink) getIntent().getSerializableExtra( name: "thongtindrink");  
    TenChiTietDrink = drink.getTitle1();  
    iddrink = drink.getId();  
    GiaDrink = drink.getPrice();  
    motaDrink = drink.getMota();  
    DecimalFormat decimalFormat = new DecimalFormat( pattern: "###,###,###");  
    priceprocut.setText(decimalFormat.format(GiaDrink) + "Đ");  
    introduceproduct.setText(motaDrink);  
    nameproduct.setText(TenChiTietDrink);  
}
```

Actionbar lựa chọn và giỏ hàng

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.menugiohang:  
            Intent intent = new Intent( packageContext: Information_drink.this, DonHang.class);  
            startActivity(intent);  
            break;  
  
        case R.id.back:  
            Intent intent1 = new Intent( packageContext: Information_drink.this, ChonBan.class);  
            startActivity(intent1);  
            break;  
  
        case R.id.exit:  
  
            android.app.AlertDialog.Builder builder = new AlertDialog.Builder( context: Information_drink.this, android.R.style.Theme_DeviceDefault_Light_Dialog);  
            builder.setTitle("Thông Báo");  
            //      gắn icon  
            builder.setMessage("Bạn Có Muốn Thoát Khỏi Chương Trình Không?");  
            builder.setIcon(R.drawable.thongbao);  
            builder.setPositiveButton( text: "Không", (dialogInterface, i) -> {  
                // để trống vì khi chọn nó sẽ k xảy ra skien gì cả  
            });  
            builder.setNegativeButton( text: "Có", (dialogInterface, i) -> {  
                Intent startMain = new Intent(Intent.ACTION_MAIN);  
                startMain.addCategory(Intent.CATEGORY_HOME);  
                startActivity(startMain);  
                finish();  
            });  
            builder.show();  
            break;  
    }  
    return super.onOptionsItemSelected(item);  
}
```

2.9 Giao diện giỏ hàng:



2.9.1 Phân tích mô tả:

- Có 2 Textview thể hiện tên của sản phẩm và giá của sản phẩm
- 1 Listview đơn hàng
- 4 nút Button:
 - Button +
 - Button -
 - Button Tiếp tục chọn món
 - Button Xuất hóa đơn

2.9.2 Mô tả chức năng:

- Khi ấn nút – số lượng sản phẩm sẽ bị giảm đi 1, sẽ cập nhật tổng giá sản phẩm. Chỉ giảm đến số lượng 1 sản phẩm
- Khi ấn nút + số lượng sản phẩm sẽ được cộng thêm 1 đơn vị, sẽ cập nhật tổng giá sản phẩm. Chỉ tăng tối đa 10 sản phẩm
- Khi chọn Button tiếp tục chọn món, sẽ quay lại giao diện thực đơn.
- Khi chọn xem hóa đơn nó sẽ chuyển giao diện hóa đơn
- Khi quay lại thực đơn thêm món ăn thì sẽ được cập nhật vào đơn hàng, nếu trùng tên món đã chọn sẽ tăng số lượng và cập nhật giá
- Listview hiển thị các món đã thêm vào đơn hàng

2.9.3 Code

Button tiếp tục và xuất hóa đơn

```
btnxuathD.setOnClickListener((view) → {  
    Intent intent = new Intent( packageContext: DonHang.this, XuatHoaDon.class);  
    startActivity(intent);  
});  
btntieptuc.setOnClickListener((view) → {  
    Intent intent = new Intent( packageContext: DonHang.this, Listmenu.class);  
    startActivity(intent);  
});
```

Hàm tính tổng tiền theo từng món

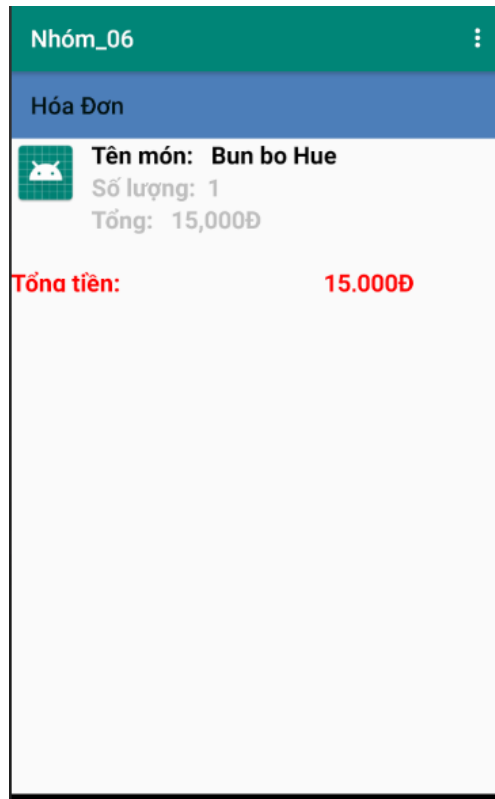
```
public static void TinhTien()  
{  
    int tongtien=0;  
    for(int i = 0; i < MainActivity.manggiohang.size(); i++)  
    {  
        tongtien += MainActivity.manggiohang.get(i).getGia();  
    }  
    DecimalFormat decimalFormat =new DecimalFormat( pattern: "###,###,###");  
}
```

Button tăng và giảm số lượng

```
hoaDonHolder.btntang.setOnClickListener((view) → {  
    int slmoinhat = Integer.parseInt(finalHoaDonHolder.btngiatrri.getText().toString()) + 1;  
    int slhientai = MainActivity.manggiohang.get(position).getSoLuong();  
    int giahientai = MainActivity.manggiohang.get(position).getGia();  
    MainActivity.manggiohang.get(position).setSoLuong(slmoinhat);  
    int giamoinhat = (giahientai * slmoinhat) / slhientai;  
    MainActivity.manggiohang.get(position).setGia(giamoinhat);  
    DecimalFormat decimalFormat1 = new DecimalFormat( pattern: "###,###,###");  
    finalHoaDonHolder1.txtGia.setText(decimalFormat1.format(giamoinhat) + "Đ");  
    DonHang.TinhTien();  
    if (slmoinhat > 9)  
    {  
        finalHoaDonHolder2.btntang.setVisibility(View.INVISIBLE);  
        finalHoaDonHolder2.btngiam.setVisibility(View.VISIBLE);  
        finalHoaDonHolder.btngiatrri.setText(String.valueOf(slmoinhat));  
    }  
    else {  
        finalHoaDonHolder2.btngiam.setVisibility(View.VISIBLE);  
        finalHoaDonHolder2.btntang.setVisibility(View.VISIBLE);  
        finalHoaDonHolder.btngiatrri.setText(String.valueOf(slmoinhat));  
    }  
});
```

```
hoaDonHolder.btngiam.setOnClickListener((view) → {  
    int slmoinhat = Integer.parseInt(finalHoaDonHolder.btngiatrri.getText().toString()) - 1;  
    int slhientai = MainActivity.manggiohang.get(position).getSoLuong();  
    int giahientai = MainActivity.manggiohang.get(position).getGia();  
    MainActivity.manggiohang.get(position).setSoLuong(slmoinhat);  
    int giamoinhat = (giahientai * slmoinhat) / slhientai;  
    MainActivity.manggiohang.get(position).setGia(giamoinhat);  
    DecimalFormat decimalFormat2 = new DecimalFormat( pattern: "###,###,###");  
    finalHoaDonHolder1.txtGia.setText(decimalFormat2.format(giamoinhat) + "Đ");  
    DonHang.TinhTien();  
    if (slmoinhat < 2 )  
    {  
        finalHoaDonHolder2.btngiam.setVisibility(View.INVISIBLE);  
        finalHoaDonHolder2.btntang.setVisibility(View.VISIBLE );  
        finalHoaDonHolder.btngiatrri.setText(String.valueOf(slmoinhat));  
    }  
    else {  
        finalHoaDonHolder2.btngiam.setVisibility(View.VISIBLE);  
        finalHoaDonHolder2.btntang.setVisibility(View.VISIBLE);  
        finalHoaDonHolder.btngiatrri.setText(String.valueOf(slmoinhat));  
    }  
});  
return convertView;
```


2.10 Giao diện hóa đơn



2.10.1 Mô tả giao diện:

- Màn hình có ListView hóa đơn
- Màn hình có 2 Textview tiêu đề “ Hóa đơn” và “Tổng tiền”
- Màn hình có 1 Textview tổng tiền
- Màn hình có 1 ActionBar với 2 lựa chọn: Back và Exit

2.10.2 Mô tả chức năng:

- Listview hiển thị các món đã thêm vào đơn hàng
- Textview tổng tiền tính tổng tiền tất cả món ăn
- ActionBar 2 lựa chọn:
 - Nếu chọn Back: quay về màn hình chọn bàn
 - Nếu chọn Exit thì thoát khỏi ứng dụng

2.10.3 Code

Tính tổng tiền

```
int tongtien=0;
for(int i = 0;i < MainActivity.manggiohang.size();i++)
{
    tongtien += MainActivity.manggiohang.get(i).getGia();
}
DecimalFormat decimalFormat =new DecimalFormat( pattern: "###,###,###");
tongtienhd.setText(decimalFormat.format(tongtien) + "Đ");
```

ActionBar lựa chọn

```
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.menuhoadon, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {

        case R.id.back:
            Intent intent1 = new Intent( packageContext: XuatHoaDon.this, ChonBan.class);
            startActivity(intent1);
            break;

        case R.id.exit:
            android.app.AlertDialog.Builder builder = new AlertDialog.Builder( context: XuatHoaDon.this,android.R.style.Theme_DeviceDefault_Light_Dialog) ;
            builder.setTitle("Thông Báo");
            //        gắn icon
            builder.setMessage("Bạn Có Muốn Thoát Khỏi Chương Trình Không?");
            builder.setIcon(R.drawable.thongbao);
            builder.setPositiveButton( text: "Không", (dialogInterface, i) → {
                // để trống vì khi chọn nó sẽ k xảy ra skien gì cả
            });
            builder.setNegativeButton( text: "Có", (dialogInterface, i) → {
                Intent startMain = new Intent(Intent.ACTION_MAIN);
                startMain.addCategory(Intent.CATEGORY_HOME);
                startActivity(startMain);
                finish();
            });
            builder.show();
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

Chương 3: Kết luận & kiến nghị

Khó khăn:

- Khó quản lí database của android studio
- Không xem được cấu trúc database SQLite
- Kiến thức chuyên ngành nhiều

Kinh nghiệm:

- Tìm hiểu thêm nhiều về Database
- Truyền dữ liệu giữa các layout