به نام خدا

پروژه سیستم مدیریت رستوران

نام درس : برنامه سازی پیشرفته

نام دانشجو : فرزاد وجودزاده

نام استاد : استاد ثبوتی

ترم : بهار 1405

معماری کلی برنامه

برنامه از 4 بخش اصلی تشکیل شده :

بخش اول : لایه دیتابیس

- کلاس Database مسیول اتصال به فایل SQLite است
- متد های open(), close(), executeQuery(), createtables() دارد
- جزییات SQLite را از بقیه حفظ میکند

بخش دوم : لایه DAO

- کلاس های userDAO, restaurantDAO, menuitemDAO, orderDAO, loyaltyDAO دارد
- مسیول ارتباط با دیتابیس
- هر DAO یک ارجاع به شی از کلاس database دارد (ترکیب)
- متد های insert, update, delete, findbyid, ... دارند

یخش دوم : لایه مبدل

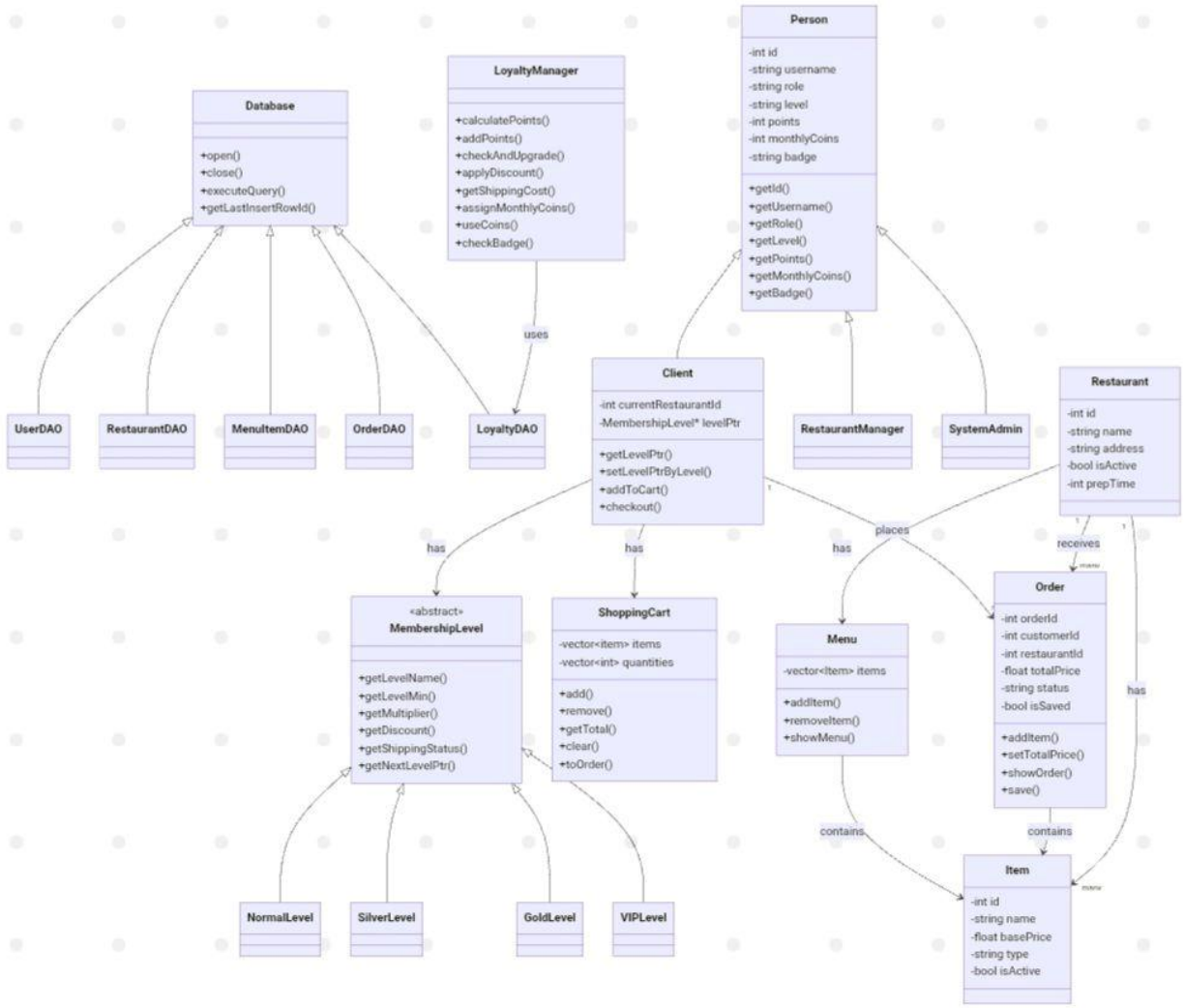
- کلاس های person, restaurant, item, oreder, membershiplevel, vipmember, normalmember,

❖Silvermember, goldmember, normalmember, دارند
داده ها را به طور موقت نگه میدارند❖
هستند (کپسوله سازی) getter , setter دارای ❖

بخش چهارم : لایه منوی کنسولی

❖ توابع showclientmenu, showadminmenu,
showmanagermenu, singinmenu, signupmenu,
دارند firstmenu
مسیول نمایش منو ها و هدایت کاربر برای انجام کار❖
ها برای گرفتن و نگهداری داده استفاده مبکنند DAO ار ❖

دیاگرام کلاس ها : توضیحات در ادامه :



روابط اصلی :

ارث بری :

- ❖ کلاس های client, systemadmin, restaurantmanager, از ارث بری میکنند person
- ❖ کلاس های normalmember, silvermember, goldmember, ارث بری میکنن membershiplevel ا ز vipmember

ترکیب :

- ❖ کلاس client شامل shoppingcart, membershiplevel هست
- ❖ کلاس loyaltymanager شامل loyaltyDAO هست
- ❖ کلاس restaurant شامل menu است
- ❖ کلاس menu شامل item هست
- ❖ کلاس order شامل item هست

اصول شی گرایی و طراحی :

اصل کپسوله سازی :

getter , setter هستند و از طریق private تمام داده های کلاس ها ها به آنها دسترسی داریم

ارث بری :

از clinet, systemadmin, restaurantmanager کلاس های ارث بری میکنند person

normalmember, silvermember, goldmember, کلاس های ارث بری میکنند membershiplevel از vipmember

ترکیب :

- هست shoppingcart, membershiplevel شامل client کلاس ❖
- هست loyaltyDAO شامل loyaltymanager کلاس ❖
- است menu شامل restaurant کلاس ❖
- هست item شامل menu کلاس ❖
- هست item شامل order کلاس ❖

چند ریختی :

استفاده شده membershiplevel از توابع مجازی در کلاس رفتار خاص خود در تخفیف (normal, silver, gold, vip) هر سطح و هزینه ارسال و امتیاز دارد

الگوی state / sterategy :

برای تشخیص سطح کاربر از کلاس if-else به جای استفاده از
های جداگانه برای هر سطح استفاده شده
open / closed رعایت اصل

تصاویر از :

کپسوله سازی:

```
class restaurant{
    private:
        int id;
        string name;
        string phone;
        string description;
        int prepTime;
        bool isActive;
        string address;
        menu resmenu;
```

ترکیب:

```
class order{
    private:
        int orderid;
        int clientid;
        int restaurantid;
        vector<item> items;
        vector<int> numofitems;
```

ارث بری :

```
class vip : public membershiplevel{
    public:
        string getlevelname() const {return "vip";}
        int getlevelmin() const {return 700;}
        float getmulti() const {return 2.0;}
        float getdis() const {return 15.0;}
        string getshipstatus() const {return "free";}
        int getnextlevelmin() const {return -1;}
        membershiplevel* getnextlevelptr() {return nullptr;}
        float getshoppingcost() const {return 0.0;}
        ~vip() {}
};
```

چند ریختی :

```
class membershiplevel{
    public:
        virtual string getlevelname() const = 0;
        virtual int getlevelmin() const = 0;
        virtual float getmulti() const = 0;
        virtual float getdis() const = 0;
        virtual string getshipstatus() const = 0;
        virtual int getnextlevelmin() const = 0;
        virtual membershiplevel* getnextlevelptr() = 0;
        virtual float getshoppingcost() const = 0;
        virtual ~membershiplevel() {}
};
```

پایگاه داده :

استفاده میکند sqlite3 این برنامه از

جدول users

```
const string createuser = R"(
    CREATE TABLE IF NOT EXISTS users (
     id INTEGER PRIMARY KEY AUTOINCREMENT,
     username TEXT UNIQUE NOT NULL,
     password TEXT NOT NULL,
     role TEXT CHECK(role IN ('client', 'restaurant_manager', 'system_admin'))
NOT NULL,
     first_name TEXT NOT NULL,
     last_name TEXT NOT NULL,
     phone TEXT NOT NULL,
     level TEXT DEFAULT 'normal',
     points INTEGER DEFAULT 0,
     monthly_coins INTEGER DEFAULT 0,
     badge TEXT DEFAULT '',
     restaurant_id INTEGER DEFAULT -1
    );
)";
```

جدول restaurants

```
const string createmenuitems = R"(
    CREATE TABLE IF NOT EXISTS menu_items(
     id INTEGER PRIMARY KEY AUTOINCREMENT,
     restaurant_id INTEGER NOT NULL,
     name TEXT NOT NULL,
     description TEXT NOT NULL,
     base_price REAL NOT NULL,
     type TEXT CHECK(type IN('drink', 'food', 'other')) NOT NULL,
     is_active INTEGER DEFAULT 1,
     volume REAL DEFAULT 0,
     PREPTIME REAL DEFAULT 0,
     FOREIGN KEY(restaurant_id) REFERENCES restaurants(id) ON DELETE CASCADE
    );
```

جدول orders

```
const string createorders = R"(
    CREATE TABLE IF NOT EXISTS orders(
     id INTEGER PRIMARY KEY AUTOINCREMENT,
     client_id INTEGER NOT NULL,
     restaurant_id INTEGER NOT NULL,
     order_date DATETIME DEFAULT CURRENT_TIMESTAMP,
     total_price REAL NOT NULL,
     status TEXT CHECK(status IN('pending', 'preparing', 'ready_for_delivery',
'delivered')) DEFAULT 'pending',
     FOREIGN KEY(client_id) REFERENCES users(id),
     FOREIGN KEY(restaurant_id) REFERENCES restaurants(id)
    );
)";
```

جدول order_items

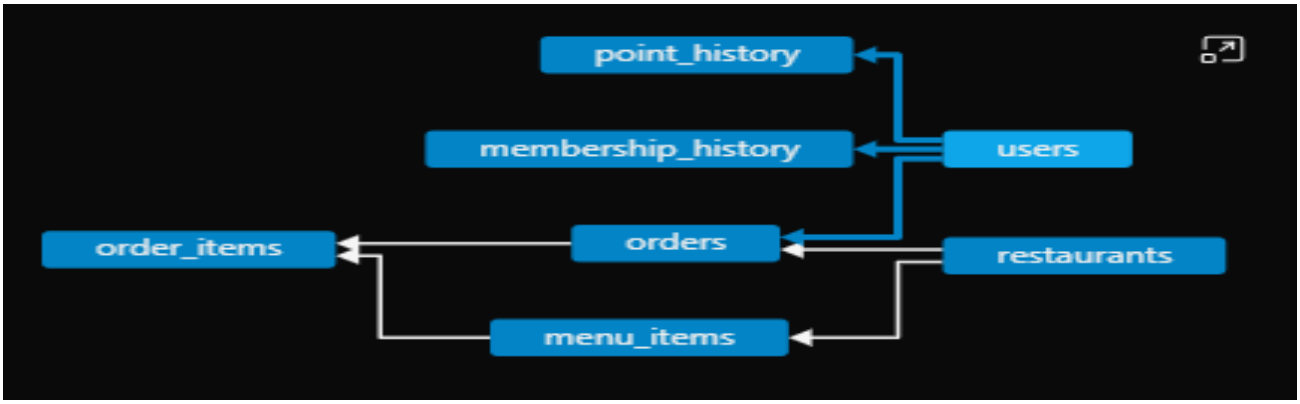
```
const string createorderitems = R"(
    CREATE TABLE IF NOT EXISTS order_items(
     id INTEGER PRIMARY KEY AUTOINCREMENT,
     order_id INTEGER NOT NULL,
     menu_item_id INTEGER NOT NULL,
     quantity INTEGER NOT NULL,
     total_price REAL NOT NULL,
     FOREIGN KEY(menu_item_id) REFERENCES menu_items(id),
     FOREIGN KEY(order_id) REFERENCES orders(id) ON DELETE CASCADE
    );
)";
```

جدول point_history

```
const string createpointhistory = R"(
    CREATE TABLE IF NOT EXISTS point_history(
     id INTEGER PRIMARY KEY AUTOINCREMENT,
     user_id INTEGER NOT NULL,
     points_change INTEGER NOT NULL,
     reason TEXT DEFAULT 'order',
     type TEXT CHECK(type IN('point', 'coin')) DEFAULT 'point',
     created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
     FOREIGN KEY(user_id) REFERENCES users(id) ON DELETE CASCADE
    );
)";
```

جدول membership_history

```
const string creatmemberhistory = R"(
    CREATE TABLE IF NOT EXISTS membership_history(
     id INTEGER PRIMARY KEY AUTOINCREMENT,
     user_id INTEGER NOT NULL,
     old_level TEXT CHECK(old_level IN('normal', 'silver', 'gold', 'vip'))
DEFAULT 'normal',
     new_level TEXT CHECK(new_level IN('normal', 'silver', 'gold', 'vip'))
DEFAULT 'normal',
     changed_at DATETIME DEFAULT CURRENT_TIMESTAMP,
     changed_by TEXT CHECK(changed_by IN('admin', 'system')) DEFAULT 'system',
     FOREIGN KEY(user_id) REFERENCES users(id) ON DELETE CASCADE
    );
)";
```



DAO لایه

این لایه پل ارتباطی بین برنامه و عملیات های دیتابیس است

جدول مربوطه	عملیات	کلاس DAP
User	Insert, delete, update, findbyid, findbyusername, findall	userDAO
Restaurants	Insert, delete, update, findbyid, findactive, findall	restaurantDAO
menu_items	Insert, update, delete, findbyid, findbyrestaurant	menuitemDAO
Orders, order_items	Insert, updatestatuse, findbyclient, findbyres	orderDAO
Users, point_history, membership_history	Getuserlevel, getuserpoint, getusercoins, addpointhistory, addmembershiphistory, ...	loyaltyDAO

هر کلاس DAO به ارجاع به کلاس database دارد و از کالبک ها برای استفاده میکند Select

```
class restaurantDAO
{
private:
    database &db;

    static int callbackfindbyid(void *data, int argc, char **argv, char
**azcolname);
    static int callbackfindall(void *data, int argc, char **argv, char
**azcolname);
    static int callbackfindactive(void *data, int argc, char **argv, char
**azcolname);
    static int callbacklastresid(void *data, int argc, char **argv, char
**azcolname);
```

DAO مزایای استفاده از لایه

جداسازی منطق کد های دستابیس از برنامه

تغییر در دیتابیس فقط در یک مکان نگهداری میشوند

امنیت استفاده از پایگاه داده را زیاد میکند

سیستم وفاداری :

هر 20000 نومان خرید یک امتیاز هست.

کوین در ماه	ضریب امتیاز	ارسال	تخفیف	امتیاز لازم	سطح
0	1.0	کامل	0%	0	Nomal
1	1.2	نیم بها	5%	100	Silver
3	1.5	نیم بها	10%	300	Gold
5	2.0	رایگان	15%	700	VIP

نشان های تشویقی :

Frequent Buyer حداقل 10 سفارش :

Night Customerسفارش بین ساعت 12 شب تا 4 صبح :

========== برخی ویژگی ها==========

- در این سیستم هر کوین برابر با 10000 تومان تخفیف است
- در این سیستم در منوی سفارشات در منوی مدیر رستوران هر چه فردی لول بالا تری داشته باشد سفارش او در اول تر لیست خواهد بود
- در اول هر ماه توکن داده میشود ادمین هم میتواند خودش در روز دلخواه به همه کوپن بده

```
===========factor=============
order id : 15
level of user: gold
date : 2026-07-09 19:51:18
status : pending
the total prices (without VAT) is: 221000
total price you must pay (with VAT (10%)) : 243100
==============================

===========factor=============
order id : 1
level of user: silver
date : 2026-07-08 14:34:55
status : pending
the total prices (without VAT) is: 250000
total price you must pay (with VAT (10%)) : 275000
==============================
```

-

سناریو های تست :

شماره	سناریو	نتیجه مورد انتظار
1	ثبت رستوران جدید	your restaurant is saved successfully!!
2	ثبت سفارش و ارتقای خودکار سطح	your order is registered successfully!! your level upgraded. your new level is: silver
3	تنزل سطح یه کاربر توسط ادمین	the level of client is changed successfully
4	اعمال تخفیف و ارسال رایگان برای VIP	your total price with discount (without coins) is : 125000

مدیریت خطا :

در جاهایی از برنامه که انتظار میرفت کاربر داده غلط وارد کند به ویژه در بحث استفاده شده است while جداول دیتابیس از

بودن پوینتر ها چک شده تا برنامه nullptr همینطور در خیلی از جاهای برنامه ها هم قبل عملیات چک میشود vector کرش نکند . همینطور خالی بودن

```
while (true)
    {
        if (pr->getpassword() == pass)
        {
            return pr;
        }
        else
        {
            cout << "the password is not match!! enter correct pass\n";
            cin >> pass;
        }
    }
```

```
vector<person> prs = dao.findbyroll("client");

if (prs.empty()){
      cout << "there is no client yet\n";
      system("pause");
      system("cls");
      continue;
}
```

```
person *p = dao.findbyid(cliid);

if ((p == nullptr) || !(p->getrole() == "client")){
      cout << "the client with this id isn't found please try again\n";
      system("pause");
      system("cls");
      continue;
}
```

مدیریت حافظه:

ها خیلی استفاده شده است و بیشتر آنها با pointer در این برنامه از delete ساخته شده اند و برای جلوگیری از نشت حافظه حتما new با حافظه را آزاد کردم

```
person* userDAO::findbyid(int id){
    person* user = new person;

    string query = "SELECT * FROM users WHERE id = " + std::to_string(id) + ";";

    char* errmsg = nullptr;
    int result;

    result = sqlite3_exec(db.getconnection(), query.c_str(), callbackfindbyid,
user, &errmsg);

    if(result != SQLITE_OK){
        std::cerr << "error in find user by id : " << errmsg << std::endl;
        sqlite3_free(errmsg);
        delete user;
        return nullptr;
    }

    if(user->getid() == -1){
        delete user;
        return nullptr;
    }

    return user;

}
```

تصاویری از محیط برنامه :

منوی مشتری :

```
**************the client menu**************

name: raha helaei | your badge is: (  )
your point is: 122 | your level is: silver | number of your coins is : 0 | points to next level: 178

enter the number of option you want(for exit inter 0)
1.show all active restaurants
2.choose restaurant
3.show current restaurant menu
4.add item to shopping cart
5.show shopping cart
6.check out order
7.show history of orders
```

سبد خرید مشتری :

```
-----------the shopping cart------------
kebab x3 the price for per item: 120000
the total base price is : 360000

Discount (5%) : -18000
shopping const : +5000
============================
the total prices that you must pay is : 347000

Press any key to continue . . .
```

منوی رستوران دار:

```
----------the manager menu--------------
enter the number your choosen option:
1.view my restaurant informations
2.edit restaurant information
3.manag the menu
4.view orders
5.chang the status of order(you must know your order id)
0.for exit program
```

منوی ادمین گزارش فروش :

```
========= salse report =============
total salse (all orders): 16200000 tomans
-------------------------------------
orders by status:
-pending: 10 orders
-preparing: 0 orders
-ready for delivery: 1 orders
-delivered: 1 orders
-------------------------------------
total orders: 12
total users: 5
total restaurants: 1
=====================================
Press any key to continue . . .
```

منوی ادمین دیدن تغییر سطح کلاینت ها

```
user id: 1 | gold -> vip | 2026-07-08 18:30:55 | system

user id: 1 | silver -> gold | 2026-07-08 18:26:29 | system

user id: 4 | normal -> silver | 2026-07-08 14:35:41 | system

user id: 1 | normal -> silver | 2026-07-08 14:31:30 | admin

Press any key to continue . . .
```

منوی اولیه ورود به برنامه

```
C:\Users\MAT\Desktop\MINIPROJECT>god.exe
are you signed up? if yes enter 1 and no enter 2 (for exit from program enter 0):
```

منوی کامل ادمین:

```
choose an option (enter the number of option)
1.show all restaurants
2.active/disactive restaurant(you must have id of restaurant)
3.view salse report
4.view all users
5.delete a user
6.change level of a client
7.assign coins to all clients
8.view membership history for all client
9.view membership history for a contain client
10.view point history for all
11.view point history for a contain client
(enter 0 for exit)
```

منوی ادمین دیدن تغییر امتیاز کلاینت ها

```
user id: 1 | change: 12 | reason: order | type: point | 2026-07-08 18:32:41

user id: 1 | change: 12 | reason: order | type: point | 2026-07-08 18:32:30

user id: 1 | change: 12 | reason: order | type: point | 2026-07-08 18:32:20

user id: 1 | change: 12 | reason: order | type: point | 2026-07-08 18:31:52

user id: 1 | change: 12 | reason: order | type: point | 2026-07-08 18:31:42

user id: 1 | change: 12 | reason: order | type: point | 2026-07-08 18:31:22

user id: 1 | change: 720 | reason: order | type: point | 2026-07-08 18:30:55

user id: 1 | change: 259 | reason: order | type: point | 2026-07-08 18:26:29

user id: 1 | change: -1 | reason: coins_used | type: coin | 2026-07-08 18:26:14

user id: 1 | change: 1 | reason: monthly_coins | type: coin | 2026-07-08 17:35:50
```

نتیجه گیری:

سیستم مدیریت سفارش آنلاین با موفقیت پیاده سازی شد

قابلیت های پیاده سازی شده

ثبت نام و ورود کاربران با سه نقش (مشتری . مدیر رستوران . ادمین)

مدیریت کامل رستوران ها و منو ها و سفارش ها برای مدیر رستوران

سبد خرید و ثبت سفارش

مشاهده تاریخچه سفارش ها

گزارش فروش کلی برای ادمین

سیستم وفاداری با 4 سطح عضویت (normal, sliver, gold, vip)

امتیاز دهی هوشمند و ارتقا خودکار سطح (همچنین ارتقا دستی سطح با ادمین)

اعمال تخفیف و هزینه ارسال بر اساس سطح

کوین های ماهیانه و استفاده از کوپن

نشان های تشویقی (Frequent Buyer, Night customer)

تاریخچه تغییرات سطح و امتیاز قابل مشاهده برای ادمین

استفاده از الگوی state/sterategy به جای if-else

لینک گیت هاب پروژه:

[GitHub - farzadvojoud963-stack/MINIPROJECT: a system restaurant management with C++ langueg and sqlite. · GitHub]

با تشکر از

حسن توجه شما