**C PROJECT REPORT: MOTINUE**

# Abstract

Scientific research shows that obesity can cause many diseases. Obesity is not just a matter of weight, but has something to do with our health. Therefore, fitness has become the first choice for many people to improve and exercise ourselves. By keeping fit, we can not only reduce the troubles caused by obesity, but also allow us to meet a better version of ourselves. So far, there are many apps to help people get fit. The goal of our group is to design a fitness software that meets our requirements which are based on these apps. Many functions are added to the software to live up to different standard of people. We also hope to learn a lot about fitness by designing this software.

# Introduction

Fitness software has become very popular in recent years. In China, popular software is with short videos like KEEP, which teaches people how to exercise. Such software has strong practicality, but does not pay much attention to the user's diet and other aspects. Other apps built specifically for eating but at the same time are lack of fitness advice. This leaves the market short of a software that can do both. It can be said that fitness will become a very important part of middle age life now and for the next decade or more. So we will build a relatively comprehensive software -- MOTINUE.

In MOTINUE, we will achieve the goals of account management system and project recommendation. For this purpose, we use QT and MYSQL , these two soft wares. QT is a cross-platform application development framework for C++ users. QT Creator, a lightweight cross-platform integration environment, was used in our development. This compiler is very easy to operate. MYSQL is a relational database management system. It can store data in different tables rather than in a large warehouse. This approach increases speed and flexibility. In short, using these two development software makes our development process more efficient and perfect.

MOTINUE can achieve accounting login, registration, password modification and error reporting, information query, fitness recommendation, diet recommendation, planning and other functions. The implementation of these functions will be explained one by one in the following sections.

# Group division

|  |  |  |  |
| --- | --- | --- | --- |
| Name | student ID | College | Work percentage |
| Huang Mingwang |  | Biology | 25% |
| Bai Zixian | 12180904 | Chemistry | 25% |
| Zhang Chengyu |  | Chemistry | 25% |
| Zou Jiayun |  | Chemistry | 25% |

##### Member1 Huang Mingwang

Designing login interface, loading buttons.

Write functions about the login interface.

Link to the database.

Password encrypted.

Check for errors.

#### Member2 Bai Zixian

Design the database.

Account management and information inquiry system.

Login system and registration system.

Check for errors.

#### Member3 Zhang Chengyu

Look for calorie data on fruits and vegetables.

Look for the number of calories burned by various activities.

Make exercise plans.

Write functions related to motion.

Check for errors.

#### Member4 Zou Jiayun

Design the database.

Account management and information inquiry system.

Login system and registration system.

Check for errors.

# Analysis

The purpose of this software is to achieve information gathering and planning recommendations.

In the login screen, we want to implement the following functions. The first is the login function. If the account already exists, the user will go directly to the database to query his data. We didn't develop an administrator function so we could only query one data at a time. If the account does not exist, the user is asked to create a new account. The new account name cannot be repeated with the existing account name, and the password cannot be too short, otherwise the registration will not be completed. If the password is entered incorrectly while logging in, the system will report an error. Passwords are encrypted after they are stored. So we complete the login interface function like this.

One of the most important indicators we have is body fat. When gender, age, and waist circumference are entered, a function automatically generates the user's body fat rate data. After the data statistics are completed, the user completes the first phase of operation. Then we can choose to go into exercise plan design. Users can enter expected weight and expected completion time (reasonably). After all of this, users can choose preferred sports, such as running and swimming, which are more efficient, while basketball and football are less effective in the short term. In this way, the system will automatically design an exercise plan for the user, telling the user how to exercise and how long to exercise every day. The system also recommends dishes such as fruits, vegetables and nuts. After a period of exercise, users enter the amount of exercise, and the system will judge whether the amount is up to standard. If met, continue to complete the program; if not, the system will change the schedule appropriately for the user to complete and complete the program; if not, the system will change the schedule appropriately for the user to complete.

# Design

# 

Step1

Start

entering the initial information

input user name

whether the password reaches six digits and whether the complexity user name has been registered

retrieve information from the database

Step2

Read user information

input personal information (height, weight, waist circumference, sex, age)

calculate the body fat rate to get the expected weight, fat loss time

get the daily fitness plan

input daily steps, the actual weight is up to the expected standard

Step3

Daily menu selection from the main menu

Select from the sub-menu to get the daily recommendation menu with the expected calorie

end

**Initial form of code**

For female

a = yaowei\*0.74,

b = weight\*0.082 + 34.89,

rate = (a-b)/weight \*100%

For male

a = yaowei\*0.74,

b = weight\*0.082 + 44.74,

rate = (a-b)/weight \*100%

Sample

Q = Q\*weight / 56

dishes

beef noodle 119

tomato with egg 87

toasted bread 278

cooked cabbage 46

cooked eggplant 74

egg flower with rice 154

cooked red fish 216

salty shrimp 92

cooked bean 300

fried pancake 259

spicy potato slice 122

braised pork 470

braised egg 148

Mapo tofu 133

tomato soup 14

sour fish 98

sushi 141

cooked cabbage 48

luosi noodle 98

zongzi 195

vegetable salad 40

onion with eggplant 147

braised loin 248

coarse pancake 243

pork floss bread 326

sweet buns 319

soy sause cucumber 51

braised hailtail 175

shrimp with melon 45

braised chicken 224

stewed chicken claw 236

egg tart 375

beef buns 216

cooked noodle 194

puff pastry 269

cooked pancake 151

cooked carrot pieces 62

braised tofu 98

braised eggplant 75

cucumber with egg 83

garlic chicken 102

pork with brocolli 97

stewed egg 152

shrimp steamed noodle 125

Beijing roast duck 282

cooked peanut 588

fish flavored pork 182

tremella soup 10

mixed kelp 42

roasted chicken wing 204

low sugar soymilk 13

roasted river fish 195

rate = (m / 66) \* 100%

heavily overwheight

breakfast 50g fruit + 150g main food + 150g drink 423 < E < 473

lunch 220g main food + 300g dishes + 200g drink + 50g bean&nuts 639 < E < 760

supper 100g fruit + 250g milk produck + 200g main food + 240g dishes + 150g drink 588 < E < 694

overweight

breakfast 50g fruit + 140g main food + 150g drink 412 < E < 473

lunch 220g main food + 280g dishes + 200g drink + 50g bean&nuts 628 < E < 760

supper 90g fruit + 250g milk produck + 210g main food + 240g dishes + 150g drink 576 < E < 694

slight overweight

breakfast 50g fruit + 140g main food + 150g drink 412 < E < 473

lunch 250g main food + 310g dishes + 200g drink + 40g bean&nuts 644 < E < 760

supper 100g fruit + 240g milk produck + 220g main food + 240g dishes + 150g drink 601 < E < 694

healthy

breakfast 50g fruit + 170g main food + 150g drink 455 < E < 473

lunch 240g main food + 320g dishes + 220g drink + 50g bean&nuts 644 < E < 760

supper 100g fruit + 250g milk produck + 230g main food + 250g dishes + 160g drink 603 < E < 694

sexy&slim

breakfast 50g fruit + 150g main food + 150g drink 470 < E < 483

lunch 250g main food + 340g dishes + 240g drink + 60g bean&nuts 688 < E < 760

supper 100g fruit + 260g milk produck + 230g main food + 260g dishes + 150g drink 622 < E < 694

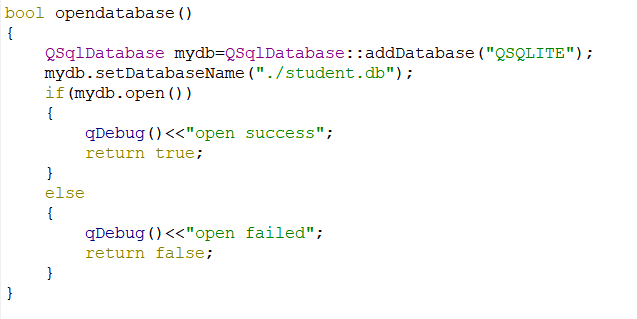
We control the exercise cycle to be seven days a time, whose frecuency ranges from thirty minutes a day to three hours a day. Also counts our daily step number. The final code made several adjustments, which is originated from an program “calorie clock”.

Our dish suggest menu is simplified from the original menu, which can be found not as flexible as imagined. Hope we can do it beter next time.

Ⅰ.The store of the information

As we all know, there are lots of information of the members to store and they should be remembered by the computer. So, how to realize this function? We finally choose the database to store these information, so that we can store as much information as we can. By this way, we can see information of one member easily.

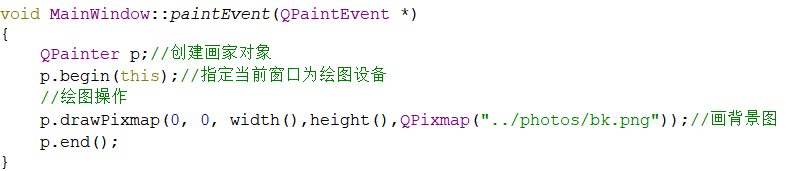
First, we use the sql database which is already in the QT to create a database which names student. If this database already exists in this computer, this database will be opened, if not, the system will create a new database Automatically.

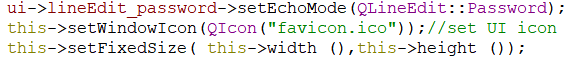


Besides, we need to create a window which can realize the functions of login in and register. In order to connect this program with the database, we call some data from the database and use them to realize the basic functions. To give the members more comfortable experience when use our program, we also use the QMessage windows to give users the information in return.



The common windows are not beautiful, so we use the QPainter to draw a picture in the background. What’s more ? We also give the windows a logo in the upper left corner ,make the windows have a fixed size and give the password have encryption.





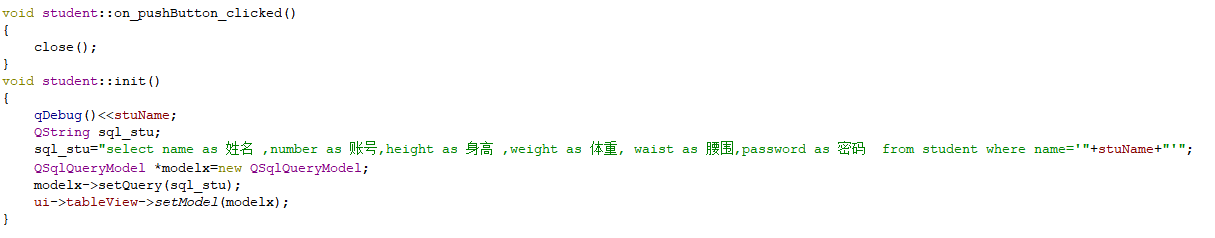
There is one case that the users may have a wrong information in this database or he just have an incomplete information. When he wants to modify it, he must find the wrong information and then deletes it or polishes it. So, we use the algorithm to Sorting.



After finding this information , how to delete it is the problem. This way should be as easy as possible. So, we try to delete the information by just one little information like “name”. Just delete it by name and the whole information will be deleted.



The users’ everyday information will be recorded in the final function, just show it in a windows and we can check how we exercise everyday.

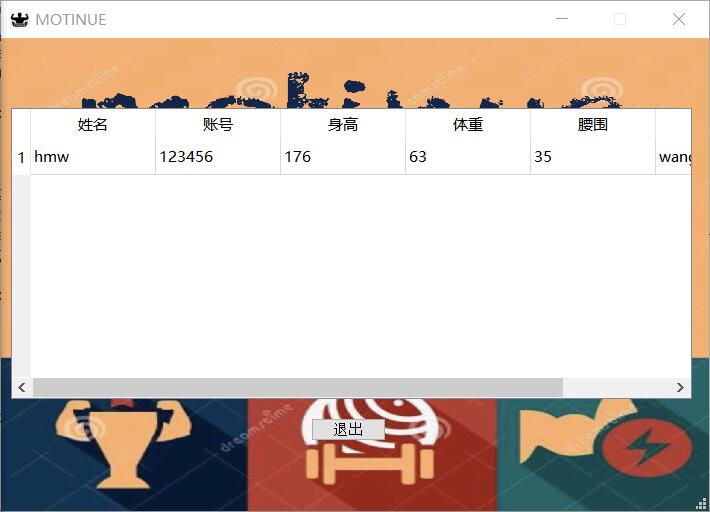


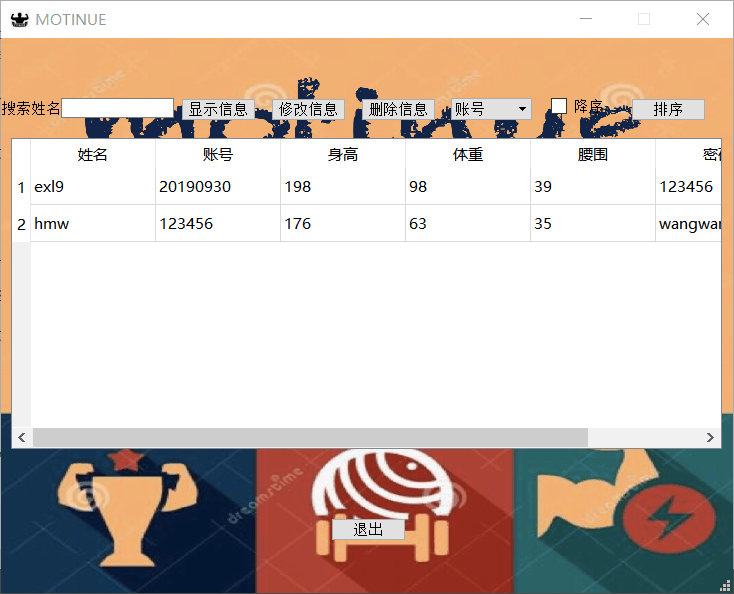
**Testing**

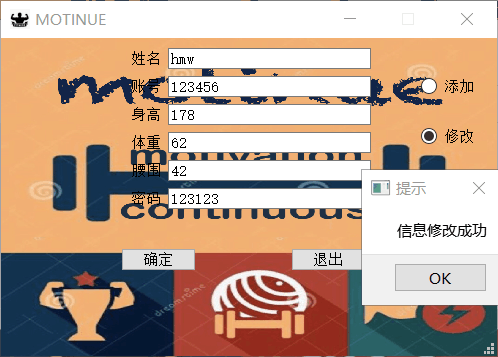


Login page. It consists of username input box, password input box, password-display selection box and login button, registration button and exit button.

Login page. It consists of username input box, password input box, password-display selection box and login button, registration button and exit button.

Users can click the View button to view their account information and health information.

Administrator can search all data of users and modify or delete someone's information .Besides admin user can sort all information according to an attribute.



Users can modify their actual health data including weight, waist circumference every day.