常州工学院

CHANGZHOU INSTITUTE OF TECHNOLOGY

# Graduation design specification

**Title**: Research and Design of "Please, help me" Mini Program Based on WeChat Mini Program

二级学院：计算机信息工程学院

专　　业：软件工程　　班级：16 软件（国际）

学生姓名：卢波　　学号：16210115

指导教师：严伟中　　职称：讲　师

评阅教师：　　职称：

05.2020

# RESEARCH AND DESIGN OF “PLEASE, HELP ME” MINI-PROGRAM BASED ON WECHAT MINI PROGRAM

## Abstract

Every day, people find themselves in various situations, and sometimes it is difficult to deal with them alone. There are rescue services, ambulances, firefighters, but not just rely on them. So difficult time for people, when all of us fights against epidemic, all people must unite and support each other.

For this purpose, the mini-program "Please help me" was created. This application will allow you to write help requests and send them online. You can limit the circle of people who send help requests to, or you can send it to all online users.

The mini program uses the MVC pattern to develop from three aspects: Model, View, and Control. It is implemented in a three-layer model: the data layer, the business logic layer, and the view layer. The data level includes operations on the database, the business logic level acts as an intermediate level for logical processing of user input, and then maps to the corresponding operation of the data level, and the presentation level includes the user interface, including the user authorization interface, interface sending and receiving zone of request message, etc.

The mini-program "Please help me" is based on Java language, there is no web pages, program use WeChat developer tools for view layer development, background development tool IntelliJ Idea, the database uses MySQL.

**Keywords:** WeChat mini program, "please, help me" applet, no web pages, Java, MVC model

# TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION ..... 3
1.1 Research background ..... 3
1.2 The significance of subject research ..... 3
1.3 Thesis structure ..... 3
1.4 Chapter summary ..... 4
CHAPTER 2: RELATED TECHNOLOGY INTRODUCTION ..... 5
2.1 Introduction to software development methods and related technologies ..... 5
2.2 WeChat Mini-Program ..... 5
2.3 Mini program's framework ..... 5
2.4 Mini Program's API ..... 6
2.5 MVC Model ..... 7
2.6 Chapter summary ..... 7
CHAPTER 3: PROGRAM ANALYSIS ..... 8
3.1 Introduction ..... 8
3.2 Feasibility Analysis ..... 8
3.2.1 Economic feasibility ..... 8
3.2.2 Technical feasibility ..... 8
3.2.3 Operational feasibility ..... 8
3.3 Demand analysis ..... 8
3.3.1 Functional requirements analysis ..... 9
3.4 Overall program design ..... 9
3.4.1 Goals of the program ..... 9
3.4.2 The physical design of the program ..... 10
3.5 Chapter summary ..... 11
CHAPTER 4: DETAILED PROGRAM DESIGN ..... 12
4.1 Functional Design ..... 12
4.1.1 Functional Design of Mini Program ..... 12
4.2 Program performance analysis ..... 12
4.3 Program Flowchart ..... 13
4.3.1 Flowchart of user help request sending ..... 13
4.4 Database Design ..... 14
4.4.1 E-R diagram design ..... 14
4.5 Chapter summary ..... 16
CHAPTER 5: CODING AND IMPLEMENTATION ..... 17
5.1 Introduction ..... 17
5.2 Mini Program Interface Design ..... 17
5.2.1 Login Interface Design ..... 17
5.2.2 Registration Interface Design ..... 18
5.2.3 Password Modification Interface Design ..... 19
5.2.4 Main Page Interface ..... 21

5.2.5 'My' Page Interface Design ..... 24
5.2.6 Friend Interface Design ..... 25
5.2.7 Find Friends Interface Design ..... 26
5.4 Chapter summary ..... 26
CHAPTER 6: PROGRAM TEST ..... 27
6.1 Testing purposes ..... 27
6.2 Test procedure ..... 27
6.3 Mini program test case ..... 27
6.4 Chapter summary ..... 29
CHAPTER7: SUMMARY AND OUTLOOK ..... 30
7.1 Summary ..... 30
7.2 Outlook ..... 30
THANKS ..... 31
REFERENCES ..... 32

# Chapter 1: Introduction

## 1.1 Research background

My life in China was full of exciting moments, but I also had difficult times. Sometimes, when I had dizzy, I couldn't find someone for help. Now when I think about it, I understand that people like me with poor health have a lot of difficulties on moments like this.

We also need to help old people with their heavy bags, with some housework etc.

Now, the situation in the world is also very tense, because the world suffers from the COVID-19 epidemic. I think that in so difficult period, people should be able to seek help.

The "Please, help me" mini-program main theme is sending help requests and responses. In the requested text, user can tell what kind of help he needs.

## 1.2 The significance of subject research

This topic mainly considers that the number of monthly active users on WeChat is close to 1.1 billion. Whether it is social or office, everyone is now using WeChat, so we decided to use WeChat mini program development, and WeChat mini program has four major advantages:

(1) Convenient and fast, ready to use;

(2) It is fast and does not occupy memory;

(3) Safe and stable, strong confidentiality;

(4) Low development cost and simple maintenance;

This makes the program easy to use. Through this platform, users can send help requests or respond to requests from other users. It is sufficient to enter the application only in your free time. The more users, someone may answer your request. Due to the huge number of WeChat users, this will promote the rapid growth of the number of users of the application, and the user-friendly and intuitive interface will satisfy users.

## 1.3 Thesis structure

The first chapter – introduction - introduces the significance of the subject in the past and present.

The second chapter - related technology introduction - introduces the "Please, help me" mini-program development technology and development direction, introduces mini-program advantages, framework, WeChat mini-program API and MVC model.

The third chapter - program analysis - introduces:

(1) Analyze the feasibility of the entire “please, help me” mini program from three aspects (economical, technical and operational feasibilities);

(2) Demand analysis;

(3) Define the goals of program development and design the system architecture of the program;

The fourth chapter - the detailed design of the program - describes the specific functions of the function module of the mini-program in detail, clarifies the request process of the mini-program user, and writes the design of the data table in detail.

Chapter five - Coding and Implementation - introduces the main functions, running interface and main parts of the code in the mini-program.

Chapter six - Program Testing - introduces the testing of the "Please, help me" mini-program.

Chapter seven is the summary of the article.

## 1.4 Chapter summary

The main part of this chapter: Introduces the research background of WeChat Mini Program "Please, help me", describes the advantages of WeChat Mini Program and the practical significance of this program, and introduces the structure of the article

# Chapter 2: Related technology introduction

## 2.1 Introduction to software development methods and related technologies

From the technical level, the software is based on WeChat mini-program, the client view is developed using WeChat development tools and the server is developed in IntelliJ IDEA and Tomcat.

## 2.2 WeChat Mini-Program

The WeChat Mini Program is an application that can be used without downloading and installing. It realizes the dream of "at your fingertips". Users can scan or search to open the application.

After the application is fully opened, developers with the main type of enterprise, government, media, other organizations or individuals can apply for the registration mini program. WeChat mini program, WeChat subscription number, WeChat service number, and WeChat enterprise number are parallel systems.

After nearly two years of development, a new WeChat mini program development environment and developer ecosystem have been constructed. WeChat Mini Program is also an innovation that can affect ordinary programmers in the Chinese IT industry for so many years. More than 1.5 million developers have joined the development of WeChat Mini Program, and we work together to promote WeChat Mini Program With the development, the number of WeChat mini program applications has exceeded one million, covering more than 200 subdivided industries, and daily users have reached 200 million. WeChat mini programs have also achieved support for subway and bus services in many cities.

On August 9th, 2019, WeChat released a new capability test and update announcement to developers. In the new version of WeChat PC version, it supports opening Wechat mini programs shared in chat。

## 2.3 Mini program's framework

The development of WeChat mini program adopts MVC development ideas, including logic layer, view layer, basic layer and so on. Each WeChat mini program includes three files: app.js, app.json and app.wxss files, they are the realization of mini program's public logic, public settings and public style sheet. These three files must be placed in the root directory. Each page composed of four files: js, wxml, wxss and json. These four files also

correspond to logic and structure, style and configuration.

The json file in the root directory is mainly for global configuration of the system, you can set the path of the page file, the behavior of the window and etc. The most important attributes: page attribute, which set the page path; window attribute,which set the window performance; tabbar attribute, which perform the bottom tab performance; networkTimeout, which configure the network timeout time; debug, which contains debug mode configurations.

The logic layer of this system is mainly written by JavaScript. This layer is mainly used to interact with the view layer(send and feeds back the processed data).

## 2.4 Mini Program's API

The development of this mini program requires many APIs provided by WeChat.

1. Interface

Use wx.showToast to notify the user. wx.showToast displays a message prompt box. Developer can customize the content of the prompt and the delay show prompt time

wx.showModal displays a modal dialog box. Developer can define the prompt title, the prompt content, the text of the cancel button and the text of the confirm button. The callback function for successful interface call has a res parameter: res.confirm is true if user pressed the OK button, res.cancel is true if user pressed the OK button

2. Network

wx.request is a API, which can process network requests.

wx.connectSocket is the API which creates a WebSocket connection. This API have only one required parameter 'url', which need to contain the wss API URL of developer server

wx.onSocketOpen listens on the event of enabling the WebSocket connection

wx.onSocketMessage listens on the event of receiving server messages by WebSocket.

3. Location

wx.getLocation gets current geographic location and speed. The API cannot be called when the user exits the Mini Program. Location simulation in the tool uses IP-based location, and there may have some error. The tool only supports GCJ-02 coordinates now.

wx.openLocation views location using the WeChat built-in map.

4. Open APIs

wx.getUserInfo gets user information

5. Routing

wx.switchTab redirects to the tabBar page and closes all other non-tabBar pages

wx.reLaunch closes all pages and opens to a page in the app

wx.redirectTo closes the current page and redirects to a page in the application, but it is not allowed to redirect to the tabbar page.

wx.navigateTo keeps the current page and redirects to a page in the applicatio, but it is not allowed to redirect to the tabbar page. Use wx.navigateBack to return to the original page. The page stack in the mini program is limited to ten layers.

## 2.5 MVC Model

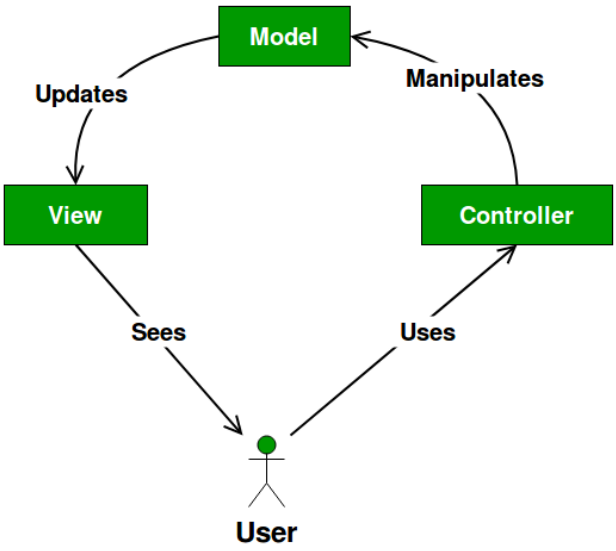



Fig. 2-1 MVC model

The model contains only pure application data, it does not have logic describing how to present data to the user.

A view presents model data to the user. The view knows how to access the model data, but does not know what the data means or what the user can do to manipulate it.

A controller exists between the view and the model. He listens to events caused by the view (or other external source), and performs a corresponding reaction to these events. In most cases, the reaction is to call the method on the model. Since the view and model are linked through the notification mechanism, the result of this action is automatically reflected in the view.

## 2.6 Chapter summary

The most important in this chapter is a detailed introduction to the related technologies needed for this program, including the WeChat mini program technology used in the front-end development and the MVC model technology used in the back-end development.

# Chapter 3: Program analysis

## 3.1 Introduction

Before designing and developing the program, it is necessary to carry out a feasibility study of the program, find possible problems and solutions. A deep analysis is needed to avoid unsolvable problems at the development stage. In this article, the Please Help Me program is analyzed from three perspectives: technology, economics and operations.

## 3.2 Feasibility Analysis

### 3.2.1 Economic feasibility

The necessary economic cost is a small fee required to purchase a server during the development process. In general, the economic cost of developing a program is low. These costs can be easily covered by adding a small amount of advertising or sponsorship investments.

Therefore, the program is economically feasible.

### 3.2.2 Technical feasibility

The technology required for the program development process has been described in Chapter 2 and will not be repeated here, and most of them have been studied. On the Internet have a lot of free public courses, which descript front-end and back-end development, and also have a lot of similar code for reference.

Therefore, the program has no technical difficulties and technically feasible。

### 3.2.3 Operational feasibility

The interface style of the application is simple and there are no complicated functions. The interface is intuitive, so at the beginning of using this application, users will have no problems. The program has the characteristics of simple and intuitive interface, simple operation, fast and easy to use, convenient and practical, strong interactivity, etc., and it can be developed during the operation process.

## 3.3 Demand analysis

The life cycle of the program mainly includes analysis, development, operation and maintenance, and the key to the analysis phase is the requirement analysis.

This stage is the medium between developers and users. You can fully understand the user's needs from program through visits or questionnaires, it’s will facilitate code

development, later upgrade and maintenance.

The operation of the program should be simple, easy to understand and intuitive. Since the applicable population of the program is not only IT specialist. Different people can use this mini program, so the operation of the program cannot be difficult to understand and need to be able to meet the needs of people with low computer skills.。

### 3.3.1 Functional requirements analysis

The program should provide management of basic user information, add and delete friends, log in, sing up, send a help request, and receive a response.

In order to meet the above requirements, the program is divided into six functional modules: user management, request management, response management, friend management, login and password modification.

1. User Management

User management includes entering, modification and deletion the user's basic information.

2. Request Management

Users can send their own requests, and when someone responds to the request, the program should change the status of the request. If the user cancels his request, the request is deleted.

3. Response management

When someone responds to request, need to save this user's information and send to requested user.

4. Friends management

The user can send a request for adding another user to friend, confirm or reject the new friend, and delete user from friend.

5. Login procedure

The login procedure is the first step to enter the program. This procedure is based on the C/S architecture and requires a specific client

6. Change password

If user forgot the password, he can change the password. Just need to remember answer for your question, which user need to write on registration form.

## 3.4 Overall program design

### 3.4.1 Goals of the program

1.Uniform style

2.Easy and flexible operation

3.Simple and friendly interface.

4. Perfect information security control (The user's sensitive information, user's WeChat account and password cannot be seen, which guarantees the security of user data)

### 3.4.2 The physical design of the program

The physical design of the program includes presentation layer, network layer, function layer, technology layer and data layer.。

| | | |
|---|---|---|
| presentation | Mini Program | |
| network | mobile network, wi-fi | |
| function | main | send request |
| | | receive a response |
| | | send responsse |
| | | receive a request |
| | | cancel request |
| | | find friends info |
| | | confirm or reject a friend request |
| | my | change user info |
| | | get statistics |
| | map | location on map |
| | find | find another user |
| | | send friend request |
| technology | WeCaht Mini Program | |
| | MVC model | |
| data | MySQL | |

Fig. 3-1 Physical design of the program

Below we use a table to explain in detail each layer:

| Layer | Hierarchical implementation method |
|---|---|
| Presentation level | The front-end device uses mobile phones and other mobile devices for function presentation. The communication between the presentation layer and the network layer is performed through the HTTP network protocol.。 |
| Network layer | The client's access to the program server is through the mobile network or Wi-Fi |
| Functional layer | It is a most important layer and is the embodiment of the main ideas of the entire program. According to the overall design idea of the program, the functional layer should be |

| | |
|---|---|
| | refined and modularized. Therefore, the functions of the client can be divided into the following modules: homepage, friends, individuals, maps and user management. The homepage module includes the information to send the request, the information to get the request, the information to send the response, and the cancellation of your request. The friend module includes requests to find new friends, to see friends' information, to confirm or reject requests to make friends; the personal module includes to modify your own information, to obtain statistical information about your requests and responses, etc |
| Technology layer | implement refers to the technical. The front-end implementation uses WeChat mini program development tools, and the back-end implementation uses IntelliJ Idea |
| Data Layer | Basically, includes the method of using MySQL to store data |

## 3.5 Chapter summary

The main purpose of this chapter is to analyze the feasibility and needs of implementation from different perspectives, and consider the mini program design and the basic needs of users.

# Chapter 4: Detailed program design

## 4.1 Functional Design

The functional design of this program should meet the user needs analysis described in Chapter 3. Among them, mini program users' needs: users can send requests and respond to other users' requests, can search other users and add or delete them from friends, can change own information and obtain statistic.

### 4.1.1 Functional Design of Mini Program

1. Functional requirements of front-end

The functions of the front-end of the program include: sending and receiving requests, sending responses to requests, searching for users, adding and deleting friends and changing personal information. The specific requirements for these functions are as follows:

(1) Send and receive requests, send responses to requests

The user can click this button on the homepage to send a request for help. If the user opens the home page, other users' requests will also appear. If the user confirms the response to the request, the data will be sent to the original sender.

(2) Search users (partial or full name or login), add and delete as friends

On the search page, the user only needs to enter another user's name or login to find him. On this page, user also can send friend requests. On the friends' page, user can confirm or decline add to friend requests. On this page, he can also delete another user from his friends.

(3) Change personal information

On the 'my' page, user can change name, gender and information about himself.

(4) Change the password

When user forgot password, he can change password

## 4.2 Program performance analysis

A good program should not only realize the functions required by the requirements, but also must consider the performance of the program. Now analyze the performance requirements that the program should consider.

1. The interface need to be easy to use, beautiful, intuitive and interactive. It is best to use a flat style in the interface design to facilitate user positioning and save time. The interface design should have no button positions and can be directly controlled.

2. The function need to be complete and accurate. Meet the actual needs of users, interoperable

3. The operation need to be compliant. Comply with applicable laws and do not violate industry standards or disclose information.

4. The system need to be safe. Safety is not only a hardware requirement, but also a safe operating environment and construction mechanism. This system requires a password mechanism when the user logs in: a method that can be entered after entering and verifying the password.。

## 4.3 Program Flowchart

### 4.3.1 Flowchart of user help request sending

Send help request is the most important part on this mini program.

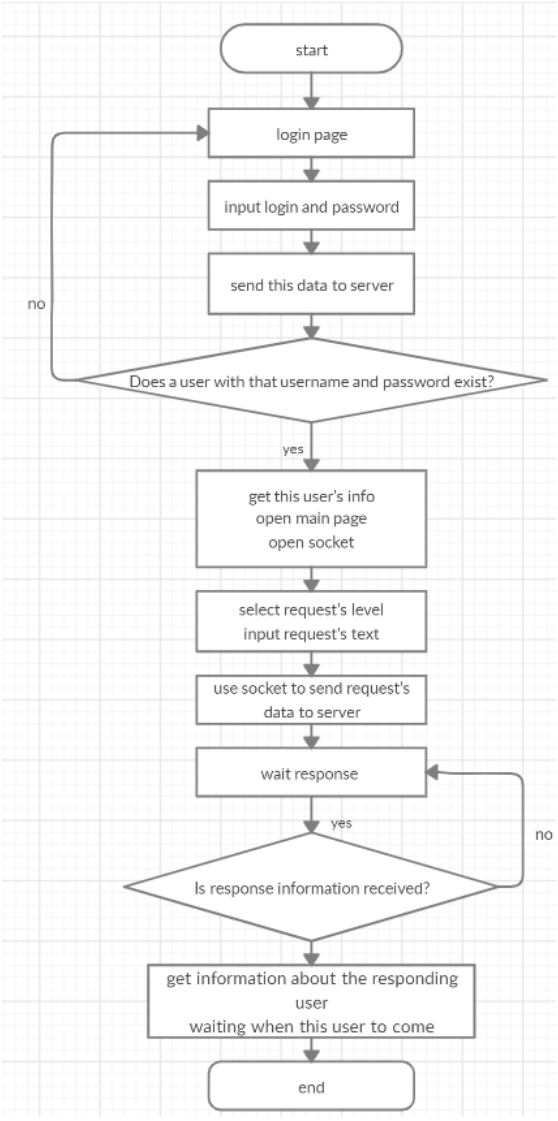



Fig. 4-1 ER diagram of help request's sending

## 4.4 Database Design

The database is the core of the program processing program. Designing a database with a reasonable structure can improve development efficiency. The ER diagram method can be used to create a conceptual database model. The model reflects the information structure of the entire application, the mutual restrictions between key information and various information. After merging and summarizing specific functions, the relationship between the structures is displayed in the form of an ER diagram for the convenience of users. The conceptual model should be abstract and should not include specific details.

The ER diagram can be divided into three stages:

1. All the objects and attributes of each part of the program, and the constraint relationship between the information to be displayed,

2. Integrate the E-R structure diagrams of each structural part into a global E-R mode

3. Optimize the global ER model to obtain the final ER diagram.

### 4.4.1 E-R diagram design

From the previous functional analysis and use case diagram analysis, the entities involved in this program are: user, response, request, friend.

User entity attributes: user ID, login, name, gender (1 male and 2 female), date of birth, question, answer, password, about, photo, peer (customized attribute of opening the socket session)

Request entity attributes: request ID, user ID, severity level (3 is very serious, 2 is serious, 1 is not serious), content, longitude, latitude, status (0 no response, 1 must have response), time.

Response entity attributes: response ID, user ID, request ID.

Friend entity attributes: friend ID, user ID, user ID, status (0 has not been determined yet 1 is determined).

(1) A user can send multiple requests, a request can be sent by only one user, there is a one-to-many relationship between the user and the request. E-R diagram between user and request:

(2) A user can make multiple friends, and each friend should be a user, so friends represent a many-to-many relationship between users and users.

(3) A user can have multiple responses; one response represents the relationship between a user and a request

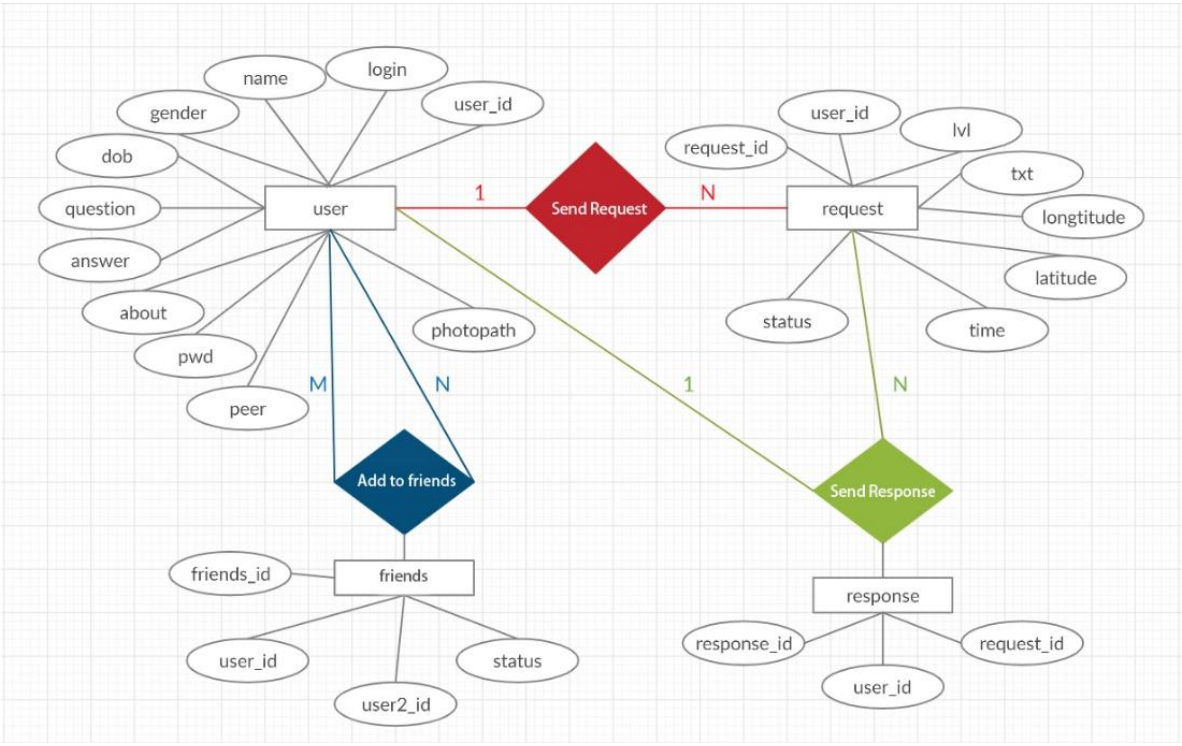



Fig. 4-2 “Please, help me” mini-program’s database E-R diagram

## 4.4.2 Database Tables Design

The "Please, help me" mini program database tables are: user table, friend table, request table, response table. The following describes the details of several tables in the database. The design structure of each table is as follows:

（1）user table（user）：user ID, login, name, gender（1 Male 2 Female）、date of birth, question, answer, password, about, photo, peer.

| Field name | Type of data | Length | Primary key |
|---|---|---|---|
| user_id | int | 11 | 是 |
| login | varchar | 50 | 否 |
| name | varchar | 30 | 否 |
| gender | int | 11 | 否 |
| dob | date | 0 | 否 |
| question | varchar | 50 | 否 |
| answer | varchar | 50 | 否 |
| about | varchar | 50 | 否 |
| pwd | varchar | 50 | 否 |
| photopath | varchar | 1000 | 否 |
| peer | varchar | 100 | 否 |

（2）friends table（friends）：friends ID，user ID，user ID，status（0 if second

user still not confirm that you are friends 1 if already confirmed ）。

| Field name | Type of data | Length | Primary key |
|---|---|---|---|
| friends_id | int | 11 | 是 |
| user1_id | int | 11 | 否 |
| user2_id | int | 11 | 否 |
| status | int | 11 | 否 |

（3）request table（request）:request ID, user ID, level（3 very seriously problem 2 seriously problem 1 unserious problem）, text, longitude, latitude, status (0 if have no response, 1 if have response), time.

| Field name | Type of data | Length | Primary key |
|---|---|---|---|
| request_id | int | 11 | 是 |
| user_id | int | 11 | 否 |
| lvl | int | 11 | 否 |
| txt | varchar | 50 | 否 |
| longitude | varchar | 100 | 否 |
| latitude | varchar | 100 | 否 |
| status | int | 11 | 否 |
| time | mediumtext | | 否 |

（4）response table（response）：response ID, user ID, request ID。

| 字段名 | 数据类型 | 长度 | 是否主键 |
|---|---|---|---|
| response_id | int | 11 | 是 |
| user_id | int | 11 | 否 |
| request_id | int | 11 | 否 |

## 4.5 Chapter summary

This chapter mainly starts from the overall design principles of the program. The functional design of the program is described in detail. Finally, all aspects of database design are described to design the "Please, help me" mini program.

# Chapter 5: Coding and implementation

## 5.1 Introduction

According to software engineering design ideas, this part is a detailed design. Because of the particularity of this program, the detailed design and program practice are combined. According to the analysis of the summary design, this chapter details two aspects including interface design and data design.

## 5.2 Mini Program Interface Design

WeChat users can scan QR code or find this mini program on WeChat. And after successful authorization, they can enter the mini program login interface.

### 5.2.1 Login Interface Design

The login interface has two input boxes and three buttons. The first input box is to enter the user name, the second input box is to enter the password, the sign in button is to log in, the sign up button opens the registration interface, and the password is forgotten button open the password modification interface。

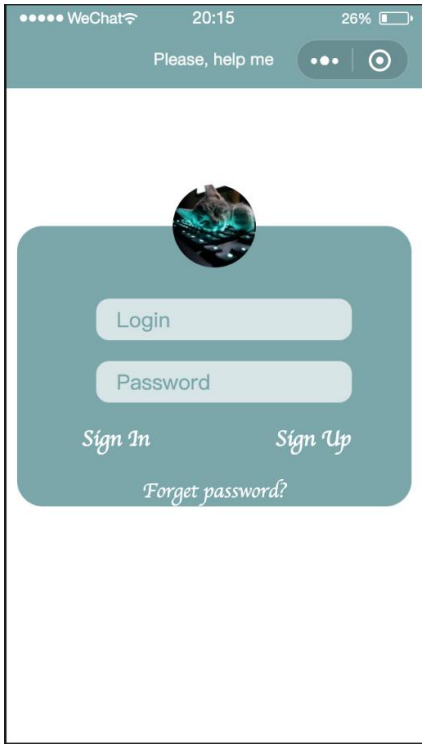



Fig. 5-1 Login interface

The main function of the login interface is to send the input data to the server through the request. Servlet uses User Service to check whether the user using the entered user name and password exists. If it exists, the server sends the user information back to the mini program in JSON format. We save this information in the Global Data of the app. In

this way, the login is successful, and then we use the wx.connectSocket method, this part is very important, because without Socket, the user can not send another online user his request.

```
wx.request({
  url: 'http://' + app.globalData.server_url + '/Login',
  method: "GET",
  header: {
    'content-type': 'application/json'
  },
  data: {
    login: self.data.user_login,
    pwd: self.data.user_pwd,
    photopath: app.globalData.userInfo.avatarUrl
  },
  success(res) {
    console.log(res.data)
    if (res.data == 'User not found') {
      wx.showToast({
        title: 'User not found. Please check login and password',
        duration: 4000,
        icon: 'none'
      })
    } else {
      wx.connectSocket({
        url: 'ws://' + app.globalData.server_url + '/socket',
      })
      app.globalData.myUserInfo = res.data
      app.globalData.user_id = app.globalData.myUserInfo.user_id
      wx.switchTab({
        url: '../main/main'
      })
    }
```

Fig. 5-1.1 Code of login request and socket's start

## 5.2.2 Registration Interface Design

The registration interface has seven input boxes, radio buttons, date selector and button. The user name input box, name input box, question input box, answer input box, and two password input boxes cannot be empty.

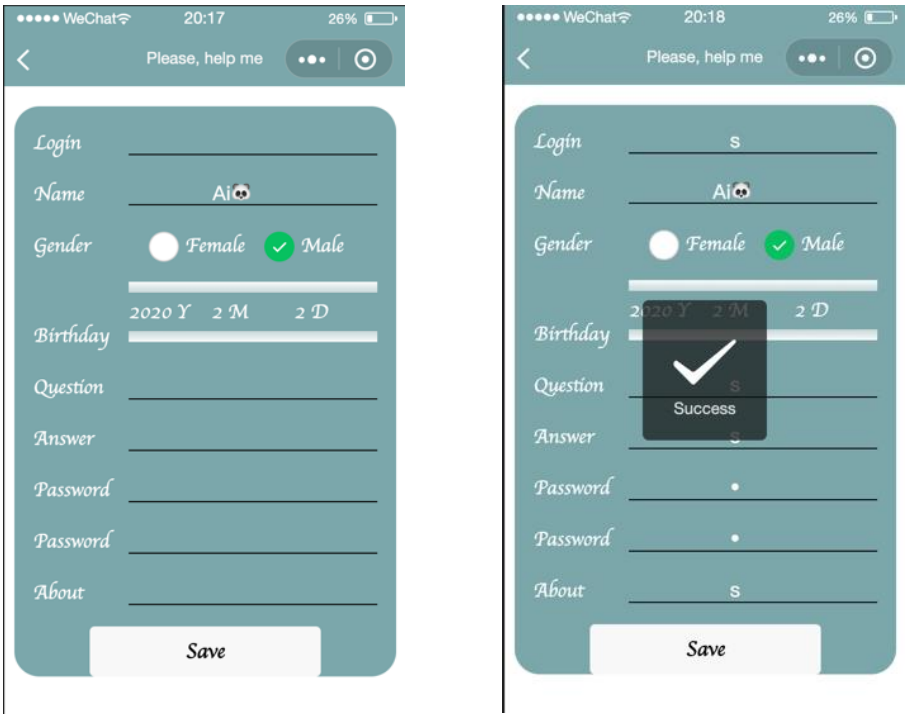



Fig. 5-2 Registration interface

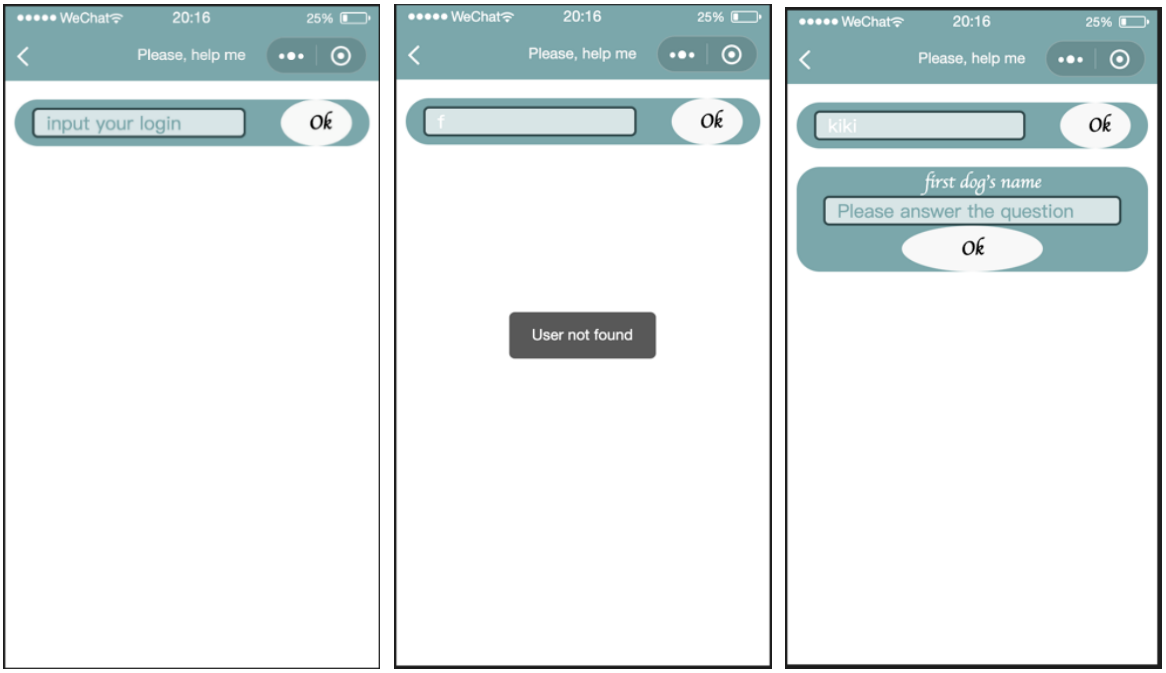
```java
@WebServlet("/Registration")
public class Registration extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        UserService us = new UserService();
        System.out.println("Registration..GET");
        if(us.checktoexist(req.getParameter( s: "login"))){
            resp.getWriter().print("This login is already exists");
        }else{
            User user = new User();
            user.setUser_login(req.getParameter( s: "login"));
            user.setUser_name(req.getParameter( s: "name"));
            user.setUser_gender(Integer.parseInt(req.getParameter( s: "gender")));
            DateUtil du = new DateUtil();
            user.setUser_dob(du.FromStringToDate(req.getParameter( s: "dob")));
            user.setUser_question(req.getParameter( s: "question"));
            user.setUser_answer(req.getParameter( s: "answer"));
            user.setUser_password(req.getParameter( s: "pwd"));
            user.setUser_about(req.getParameter( s: "about"));
            user.setUser_photopath(req.getParameter( s: "photo"));
            if(us.checktosaving(user)){
                resp.getWriter().print("Success");
            }else{
                resp.getWriter().print("Error");
            }
        }
    }
}
```

Fig. 5-2.1 Registration Servlet's code

### 5.2.3 Password Modification Interface Design

When it first opened, there was only one input box and one button. Enter the user name in the input box, and then press the button.

The user name entered by the user is sent to the server when the button is pressed. If a user with this login is exist, will open the next part of this interface. In the new input box, you need to enter the answer to the question saved during registration. If the answer is correct, will open the next part of this interface. Enter the same new password in the new two input boxes. This is the way how user can change the password. Program will open the login interface after successfully changing the password。



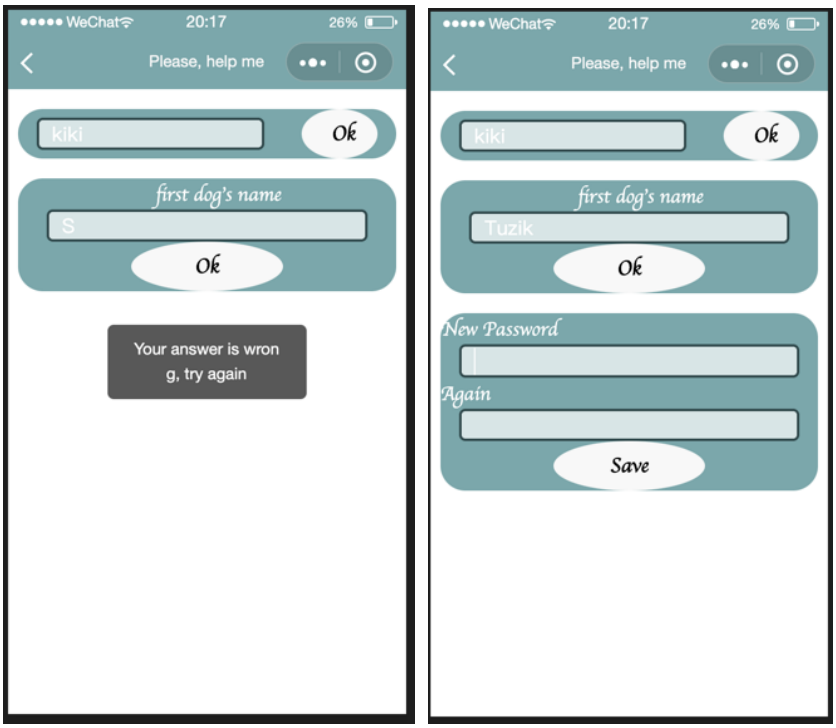



Fig. 5-3 Password modification interface

Front-end code important part：

```
findUser:function(){
  const self = this;
  wx.request({
    url: 'http://' + app.globalData.server_url + '/FindUser',
    method: "GET",
    data: {
      user_login: self.data.user_login
    },
    header: {
      'content-type': 'application/json'
    },
    success(res) {
      console.log(res.data)
      let x = res.data
      if (x==''){
        wx.showToast({
          title: 'User not found',
          icon: 'none',
          duration: 4000
        })
        self.setData({
          isQuestionTime: false
        })
      }else{
        self.setData({
          user:res.data,
          isQuestionTime: true
        })
      }
    }
  })
},
```

Fig. 5-3.1 Find friends request's code

```
<view class="container">
<view style="display:flex; align-items: center;justify-content: center; ">
  <input placeholder-class="phcolor" placeholder="input your login"  type="text" bindinput="loginInput"/>
  <button style="width:20%; border-radius: 50%;" bindtap="findUser">Ok</button>
</view>
</view>

<block wx:if="{{isQuestionTime}}">
<view class="container">
<view style="display:flex; align-items: center;justify-content: center; flex-direction:column; ">
  <text style="margin-left:6%">{{user.user_question}}</text>
  <input placeholder-class="phcolor" placeholder="Please answer the question"  type="text" bindinput="answerInput" style="width:80%"/>
  <button style="width:40%; border-radius: 50%;" bindtap="checkAnswer">Ok</button>
</view>
</view>
</block>

<block wx:if="{{answerIsRight}}">
<view class="container">
<text>New Password</text>
<input password type="text" bindinput="userPassword1Input"></input>
<text>Again</text>
<input password type="text" bindinput="userPassword2Input" ></input>
  <button style="width:40%; border-radius: 50%;" bindtap="savePwd">Save</button>
</view>
</block>
```

Fig. 5-3.2 Password modification page's wxml code

## 5.2.4 Main Page Interface

At the top of the main page, we have sent and cancel request button. On main page we also have the selection of importance and the input box for the requested content.

On the main interface, users can send requests, take requests, and send responses.：

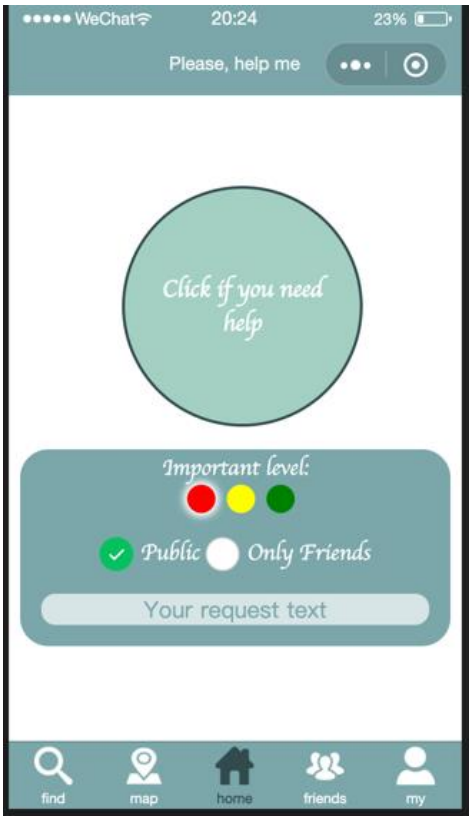



Fig. 5-4 Main Interface

The main interface is the most important interface of this program. All its functions are provided by Socket, because information must be sent and received in real time.。

The first openSocket request is sent when the main interface is onLoad.

```
98          case("\"openSocket\""):
99              System.out.println("Socket is open");
100             //保存peer在数据库
101             userService.SaveUserPeer(peer.toString(), Integer.parseInt(json.get("user_id").toString()));
102             //можно ничего не отправлять?
103             JsonObject json1 = new JsonObject();
104             json1.addProperty( property: "param", value: "opened");
105             String req1 = gson.toJson(json1);
106             return req1;
107
```

Fig. 5-4.1 Code of openSocket request

```
40      @OnMessage
41      public String onMessage(Session peer, String message) throws SQLException {
42          Gson gson = new Gson(); //新建gson     GSON实现java对象和json数据之间的相互转换
43
44          JsonParser parser = new JsonParser();   //JavaParser 可以把string转换到JsonObject
45          JsonObject json = (JsonObject) parser.parse(message);
46
47          //我们用一个 "param"变量 为了保存信息的内容的类型
48          //we use "param" for distinguish the message type
49          switch (json.get("param").toString()){
50              case ("\"req\""):
51                  System.out.println("It's request   这是请求");
52                  //因为我们知道这是请求， 我们把message转换到request对象
53                  //Convert message to Request object
54                  Request request = gson.fromJson(message,Request.class);
55                  //把request保存到数据库
56                  //Save request to DataBase
57                  if(requestService.SaveNewRequest(request)) {
58                      //find this request by userId, cause we need to find request_id
59                      String req = gson.toJson(requestService.FindRequestByUserId(request.getUser_id()));
60                      //find user's name for sending
61                      String user_name = userService.findNameByID(request.getUser_id());
62                      //convert request String to JsonObject
63                      JsonObject json1 = (JsonObject) parser.parse(req);
64                      //Add two property - param for distinguish the type for client and user_name for sending
65                      json1.addProperty( property: "param",  value: "request");
66                      json1.addProperty( property: "user_name", user_name);
67                      String req1 = gson.toJson(json1); //Convert JsonObject to String
68                      System.out.println(json.get("private"));
69                       //Is this request private or not?
70                      if (Integer.parseInt(json.get("private").toString()) == 1) {
71                          //If request is private and only for friends
72                          //We need to find all this user's friends
73                          for (FriendInfo u : friendService.FindMyFriendsInfoByID(request.getUser_id())) {
74                              // we need to check online this friend or not
75                              if (!u.getPeer().isEmpty()) {
76                                  for (Session peer1 : peers) {
77                                      //if this friend online, we send him(her) this request
78                                      if (peer1.toString().equals(u.getPeer())) {
79                                          return req1;
80                                      } else {
81                                          return null;
82                                      }
83                                  }
84                              }
85                          }
86                      } else if (Integer.parseInt(json.get("private").toString()) == 0) {
87                          //If request is public, we send request for all online users except request's author
88                          for (Session peer1 : peers) {
89                              if (peer1.toString().equals(peer.toString())) {
90                                  return null;
91                              } else {
92                                  return req1;
93                              }
94                          }
95                      }
96                      //return req1;  USE FOR CHECK
97                  }
```

Fig. 5-4.2 Send Request Socket's code

For the convenience of the user, at the front end program calculate the distance between take the request's user and them.

```
haversine_km: function (lat1, long1, lat2, long2) {
  console.log("lat1:" + lat1 + " long1:" + long1 + " lat2:" + lat2 + " long2:" + long2)
  const d2r = Math.PI / 180
  const dlong = (long2 - long1) * d2r
  const dlat = (lat2 - lat1) * d2r

  let a = Math.pow(Math.sin(dlat / 2.0), 2) + Math.cos(lat1 * d2r) * Math.cos(lat2 * d2r) * Math.pow(Math.sin(dlong / 2.0), 2)
  let c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a))
  let d = 6367 * c
  return (d/1000).toFixed(2)
},
```

Fig. 5-4.3 Distance calculation code

```
wx.onSocketMessage(function (res) {
  console.log("ONSOCKETMESSAGE  " + res.data)
  console.log("ONSOCKETMESSAGE  " + JSON.parse(res.data).param)
  self.getLoc();
  if (JSON.parse(res.data).param == "request") {
    console.log("It is req")
    //re to another function
    const mylatitude1 = JSON.parse(res.data).mylatitude
    const mylongitude1 = JSON.parse(res.data).mylongitude
    const req_id = JSON.parse(res.data).request_id
    wx.showModal({
      title: 'Your Friend ' + JSON.parse(res.data).user_name + ' need a help',
      content: JSON.parse(res.data).txt + " " + self.haversine_km(mylatitude1, mylongitude1, self.data.mylatitude, self.data.mylongitude)
+ "km from you",
      cancelText: "ignore",
      confirmText: "location",
      success(res) {
        if (res.confirm) {
          console.log('"OK" is tapped')
          wx.openLocation({
            latitude: Number(mylatitude1),
            longitude: Number(mylongitude1),
            scale: 20
          })
          wx.showModal({
            title: 'Can you help?',
            content: 'Press Yes if you can help or No if can not',
            cancelText: "no",
            confirmText: "yes",
            success(res) {
              if (res.confirm) {
                wx.sendSocketMessage({
                  data: '{param:res, user_id:' + app.globalData.myUserInfo.user_id + ', request_id:' + req_id + '}',
                }),
                app.globalData.res_lat = mylatitude1
                app.globalData.res_long = mylongitude1
              } else if (res.cancel) {
              }
            }
          })
        } else if (res.cancel) {
          console.log('"Cancel" is tapped')
        }
      }
    })
  } else if (JSON.parse(res.data).param == "response") {
    console.log("It is res")
    wx.showToast({
      title: 'Your Friend ' + JSON.parse(res.data).user_name + 'went to help you',
      icon: 'none',
      duration: 5000
    })
  }
})
},
```

Fig. 5-4.3 Get help request and send response code

```
case("\"res\""):
    System.out.println("It's response");
    //因为我们知道这是响应， 我们把message转换到response对象
    //Convert message to Response object
    Response response = gson.fromJson(message, Response.class);
    //把response保存到数据库
    //Save this response information to DataBase
    if(responseService.SaveNewResponse(response.getUser_id(),response.getRequest_id())){
        //改变 request的status     为1
        // Change request's status    to 1[it's mean this request already have response]
        requestService.UpdateRequestStatus(response.getRequest_id());
    }
    //Send response user name to request's author
    User user = userService.findUserById(response.getUser_id());
    JsonObject jsonObject = new JsonObject();
    jsonObject.addProperty( property: "param", value: "response");
    jsonObject.addProperty( property: "user_name", user.getUser_name());

    //return gson.toJson(jsonObject); USE FOE CHECK

    for (Session peer1: peers) {
        //System.out.println("RESPONSE PEER "+peer1+" AND "+userService.FindPeerByRequest(response.getRequest_id()));
        if(peer1.toString().equals(userService.FindPeerByRequest(response.getRequest_id()))){
            return gson.toJson(jsonObject);
        }else{
            return null;
        }
    }
```

Fig. 5-4.3 Send response code (socket)

## 5.2.5 'My' Page Interface Design

The personal interface has two buttons, three input boxes and six text boxes. The interface is divided into two logical parts (my information and my statistics）

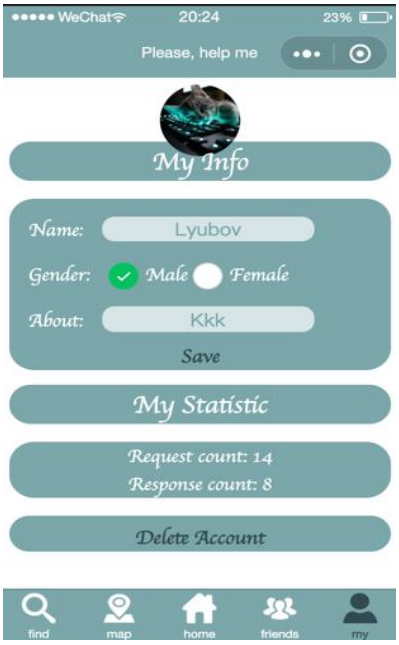



Fig. 5-5 My page Interface

When user delete his account, database can send an error, if database settings is not like on fig.5-5.2

```
public void DeleteUser(int user_id) throws SQLException {
    System.out.println("DeleteUser      UserDao");
    String sql ="DELETE FROM diploma.user WHERE user_id="+user_id;
    DBUtil db=new DBUtil();
    Connection conn =db.getConnection();
    Statement stmt = conn.createStatement();
    int rows = stmt.executeUpdate(sql);
    System.out.println("Success    DeleteUser");
}
```

Fig. 5-5.1 Delete user's code

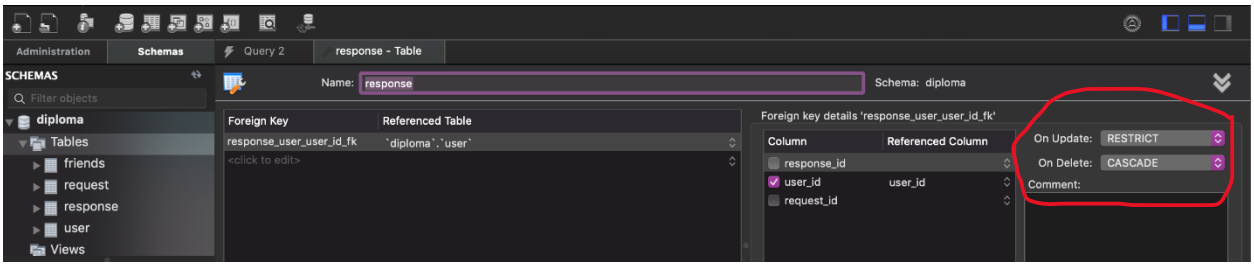



Fig 5-5.2 Database settings

## 5.2.6 Friend Interface Design

The interface is divided into two parts (my friend and new friend request). The user can confirm or reject the friend in the part requested by the new friend, and delete the friend in the part of my friend.


Fig. 5-6 Friend interface

```
deleteFriend: function(e){
  console.log(e.target.dataset.user_id)
   wx.showModal({
     title: 'Delete friend confirm',
     content: 'Do you really want to delete' + e.target.dataset.user_name + 'from friends',
     cancelText: 'No',
     confirmText:'Yes',
     success(res){
       if(res.confirm){
       wx.request({
         url: 'http://' + app.globalData.server_url + '/DeleteFriend',
         method: "GET",
         header: {
           'content-type': 'application/json'
         },
         data: {
           user1_id: e.target.dataset.user_id,
           user2_id: app.globalData.myUserInfo.user_id
         },
       })
      }
    }
  })
```

Fig. 5-6.1 Delete friend request's code

### 5.2.7 Find Friends Interface Design

There is an input box on the search interface. When the information in the input box is changed, the qualified user information is displayed, and then a friend request can be sent.


Fig. 5-7 Find friends interface

```
    wx.request({
      url: 'http://' + app.globalData.server_url + '/FindFriend',
      method: "GET",
      header: {
        'content-type': 'application/json'
      },
      data: {
        loginORname: self.data.nameORlogin,
        user_id: app.globalData.myUserInfo.user_id
      },
      success(res) {
        console.log(res.data)
        self.setData({
          users: res.data
        })
      }
    })
  },
  addFriend: function (e) {
    console.log(e.target.dataset.user_id)
    wx.request({
      url: 'http://' + app.globalData.server_url + '/AddFriend',
      method: "GET",
      header: {
        'content-type': 'application/json'
      },
      data: {
        user1_id: app.globalData.myUserInfo.user_id,
        user2_id: e.target.dataset.user_id
      },
    })
  }
```

Fig. 5-7.1 Find friend request's and add friend request's code

## 5.4 Chapter summary

This chapter mainly describes how to implement the mini program interface and business functions of the program through the MVC model, as well as the highlights of the web interface design, focusing on the user, in line with user habits.

# Chapter 6: Program Test

## 6.1 Testing purposes

We need to check how realized the functions of the "please, help me" mini program. To test the mini program interface and program function. At the same time, the we test is the program functions is used according to the specification, is program can correctly receive input data to obtain correct output information, and at the same time ensure the integrity of external information or not.

## 6.2 Test procedure

(1) Write test technology and design test cases;

(2) Run the program to test whether the operation between each interface is successful;

(3) Perform a test to see if there is a problem with the program.。

## 6.3 Mini program test case

| Test of mini program's functions | | | |
|---|---|---|---|
| Testing process | Testing case | Expected results | Test Results |
| Log In | 1. login：kiki<br>password：123 | Prompt of password or username error | Operation pass |
| | 2. login：kiki<br>password：empty | Prompt that password and user name cannot be empty | Operation pass |
| | 3. login：kiki<br>password：1234 | login successful | Operation pass |
| Registration | 1. login：empty<br>name：empty<br>gender：Male<br>birth：2020/2/2<br>question：empty<br>answer：empty<br>password：empty<br>password：empty<br>about：empty | Prompt that user name, name, question, answer, password cannot be empty | Operation pass |
| | 2. login：kiki<br>name：X<br>gender：Male<br>birth：2020/2/2<br>question：X<br>answer：X<br>password：12<br>password：12 | Prompt that a user with such a username already exists | Operation pass |

| | | | |
|---|---|---|---|
| | about：empty | | |
| | 3. login：c<br>name：a<br>gender：Male<br>birth：2005/2/2<br>question：X<br>answer：X<br>password：12<br>password：12<br>关于：empty | registration success | Operation pass |
| Forgot password | 1. login：a<br>answer：x | Prompt that answer is incorrect | Operation pass |
| | 2. login：y | Prompt that a user with such a username does not exist | Operation pass |
| | 3. login：kiki<br>answer：Tuzik<br>password：123<br>password：12 | Prompt that password error (need to write the same password for confirming) | Operation pass |
| | 4. login：kiki<br>answer：Tuzik<br>password：123<br>password：123 | password has been successfully updated, Open the login interface | Operation pass |
| Find user | 1. part of name or login：a | Display the information of all users who satisfy the conditions | Operation pass |
| Add to friend | 1. After finding a user, you can send a friend request to this user.<br>you just must press the "+" button | Prompt that friends request was send Update the search interface information | Operation pass |
| Update personal information | 1. name:<br>gender：change to Male<br>about: | Prompt that the update is successful | Operation pass |
| Help request | 1. Select the request severity level: red<br>Requested content: A | Prompt request has been sent | Operation pass |
| | 2. Select the request severity level: red<br>Requested content: A | Prompt request has been sent | Operation pass |
| Help response | 1. 1. Get the requested information and press "location" | See the requested user's location on the map | Operation pass |
| | 2. Get the requested information, press "Ignore" | Ignore request | Operation pass |
| | 3. Send the response | The requested user's address can be viewed on the map interface | Operation pass |
| Delete friend | 1. Confirm deleting | Update interface information | Operation pass |

## 6.4 Chapter summary

This section mainly tests the different functions of user registration, login, request and response, and friends of the mini program. The test cases are shown in tables. Passing this test made me realize that the details of this program still need to be improved to make users the experience raises a level.

# Chapter7: Summary and outlook

## 7.1 Summary

This article is based on my life experience and the professional knowledge gained in this major. In the process of developing the program, the data type and structure of the program table are fully considered. This mini program has no web pages and other types of implementation. The main idea of the plan is that despite the difficult times in the world, people continue to show the best human qualities and help those who really need help.

In this project, I not only mastered the knowledge learned from the textbooks, realized my practical ability, but also acquired a lot of extracurricular knowledge. During the implementation of the graduation project, there were many problems. I first considered why this error occurred, and then I searched online to see how others solved it, or went to other people's posts to leave a message. In the future study and life, I will continue to enrich, acquire new knowledge and skills with a humble and persistent attitude, and improve my professional skills and progress.

## 7.2 Outlook

We have a lot of way, how to improve this program. For example, by adding different languages and icons that will make the interface easier to understand. Also, we can improve the process of subtracting user statistics. For example, made one month of statistics, that will make possible to delete all the data of the request and response to the request once a month, which will greatly simplify the work of the database. You can also add functionality to cancel the response to the request.

# Thanks

I remember the small moment on my childhood when I first time saw the computer. It was left so deep impression on me that I decided to forever connect my life with computer. Now, I don’t even remember that I don’t have any technique in my hand, smartphones and computers always surround me. In order to realize my dream of becoming a programmer, when I was 17 years old, I left 9,000 kilometers from home, and come to China. At that time, I would not say anything except "你好".

If I talk about gratitude, firstly I must thank my parents, because they believed in me and my dreams. I must thank China, for nice and warm welcome. Thanks to all our teachers for their hard work, the knowledge they gave me, and their patience for my "听不懂". They used the simplest words and examples to explain the content of the topic. I must thank Mr. Jin and Ms. Xing, because these four year they, like parents, taking care of all our international students.

This is the last chapter I wrote as student. I am happy and sad. I’m really sad because my student life is coming to an end, but I am also happy, because now will start new chapter of my life. A lot of interesting things are waiting for me.

Finally, I would to thanks Mr. Yan for agreeing to be my graduate work instructor. He often helps me, answers to my questions, and help me find a lot of useful information.

I will continue to learn, because technology is constantly evolving, I will work hard, and I hope that after many years I will become such kind of person that the out institute will be proud.

# References

[1] 张继军，董卫编.《JavaWeb 应用开发技术与案例教程》[J].机械工业出版社 2013(09).

[2] 吴彦文,李诗,秦颖.基于微软云平台的实验学习程序的设计与实现[J].计算机工程与设计. 2013(04).

[3] 张胜利,王鹏.JAVA 和 C++实现面向对象方法的分析[J].电子技术与软件工程. 2017(23).

[4] 王善发,吴道荣.Java Web 编程中中文信息处理出现乱码的研究[J].保山学院学报. 2010(05).

[5] 张丽.基于 Java Web 在线考试程序[J].江西科学. 2016(04).

[6] 詹慧静.软件测试技术及实践[M].清华大学出版社.2016.

[7] 耿祥义.《JSP 基础教程》.北京清华大学出版社 2007

[8] 林上杰林康司.《JSP2.0 技术手册》[M]北京:电子工业出版社,2004 年 4 月

[9] 孙卫琴,李洪成《Tomcat 与 JSP Web 开发技术详解》.电子工业出版社,2003 年 6 月

[10] 林健,吴才健.基于微信小程序的校友社区网络管理平台的设计总结[J].电脑知识与技术,2019,15(05):76-77

[11] https://www.google.com

[12] http://www.baidu.com

[13] https://developers.weixin.qq.com