目 录

[1 课题分析 1](#_Toc12104)

1.1课题分析

[1.1.1 课题背景 1](#_Toc21878)

[**1.2 功能需求分析**](#_Toc2570) 1

1.3 数据属性分析 4

1.3.1登陆界面信息属性 4

[2 系统设计 4](#_Toc27131)

[**2.1数据存储设计**](#_Toc23056)........................... 5

[**2.2 函数设计与说明** 6](#_Toc32671)

**2.3菜单设计** [8](#_Toc32069)

2.3.1登录界面 [8](#_Toc32069)

2.3.2管理员删除界面 [8](#_Toc32069)

2.3.3新建联系人界面 [8](#_Toc32069)

2.3.4主界面 [9](#_Toc32069)

2.3.5查询界面 [9](#_Toc32069)

2.3.6联系人详情信息 [1](#_Toc32069)0

2.3.7个人信息界面 [1](#_Toc32069)0

2.3.8新建联系人界面 [1](#_Toc32069)1

2.3.9忘记密码界面 [1](#_Toc32069)1

[2.4 文件清单与说明 1](#_Toc32069)2

[3系统实现 1](#_Toc27131)3

**3.1 新建用户的实现................................................13**

**3.2 修改眼睛图标，显示密码信息....................................13**

**3.3 忘记密码的实现................................................14**

**3.4 退出键的实现..................................................15**

3.4.1退出注册界面的实现.......................................16

3.4.2退出忘记密码界面的实现............................ ......16

3.4.3退出查找界面的实现.......................................16

3.3.4退出个人信息界面的实现...................................17

3.4.5退出新建联系人界面的实现.................................17

3.4.6退出登录界面的实现.......................................17

3.4.7退出详情页界面的实现 ....................................18

3.4.8 退出删除界面的实现......................................18

3.4.9 退出主界面的实现........................................18

[4 测试数据与测试过程记录 19](#_Toc10958)

[4.1.1 创建用户界面 19](#_Toc31191)

[4.1.2 忘记密码界面 19](#_Toc10991)

4.1.3 转换眼睛图标，显示密码 19

4.1.4退出键的实现 20

[5 课程设计遇到的问题及解决方法 21](#_Toc2818)

[6 课程设计总结 22](#_Toc19667)

[参考文献 22](#_Toc6771)

[课设报告评定 23](#_Toc29895)

# 1 课题分析

## 1.1.1 课题背景

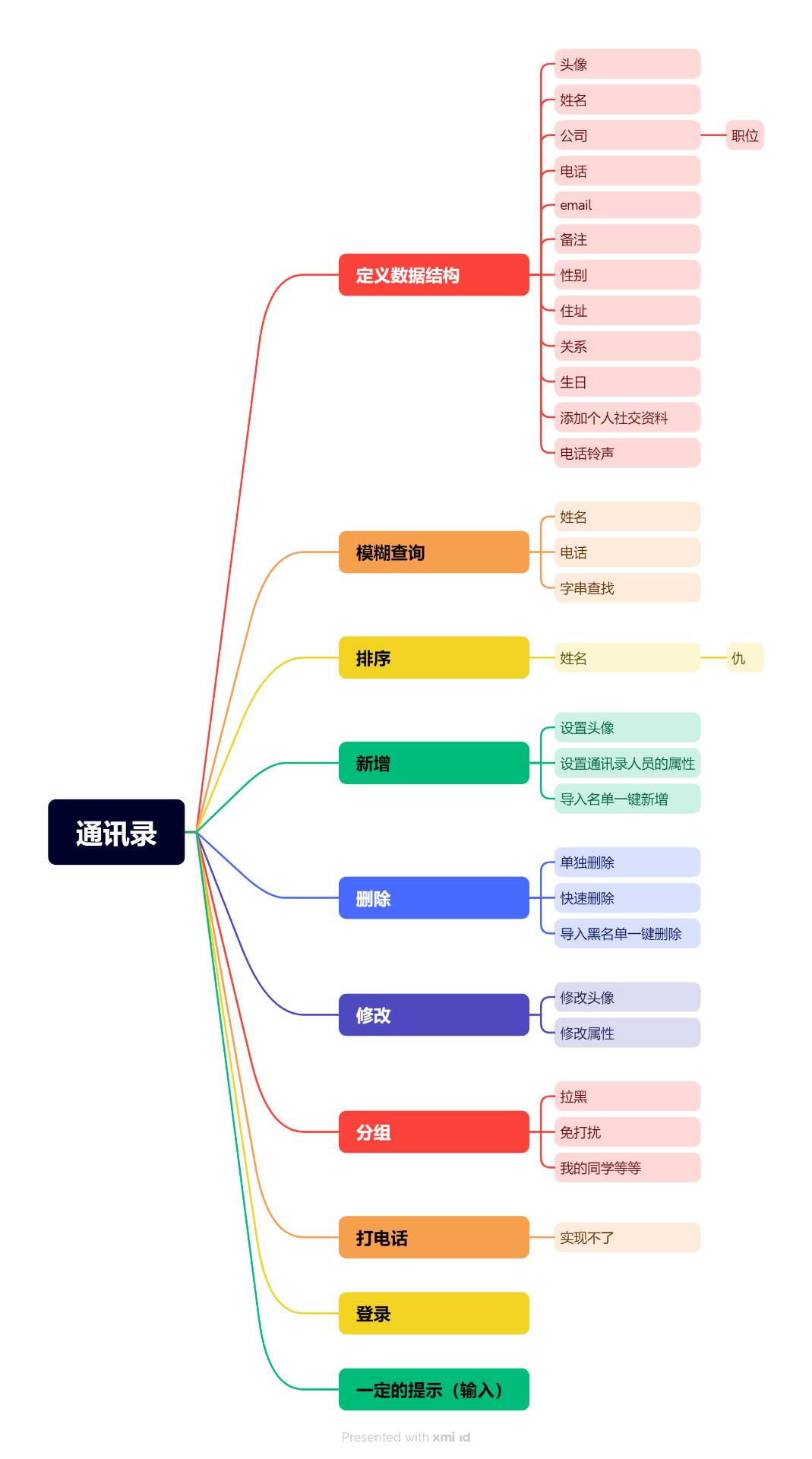
本次课程设计我们小组选择设计通讯录，一方面是认为其贴近我们的生活，在生活中时常会使用到，另一方面也是因为我们对这个课题感兴趣，也想尽力设计出一个相对功能完善的通讯录。我们通过使用c、c++、qt来设计，为了实现登录、新建用户、新增联系人、删除、分组等功能，从而设计出一个完善的通讯录。

开发环境为Qt5.12.5，Qt Creator 12.0.2 (Community)，Visual Studio 2017，MinGW--64bit

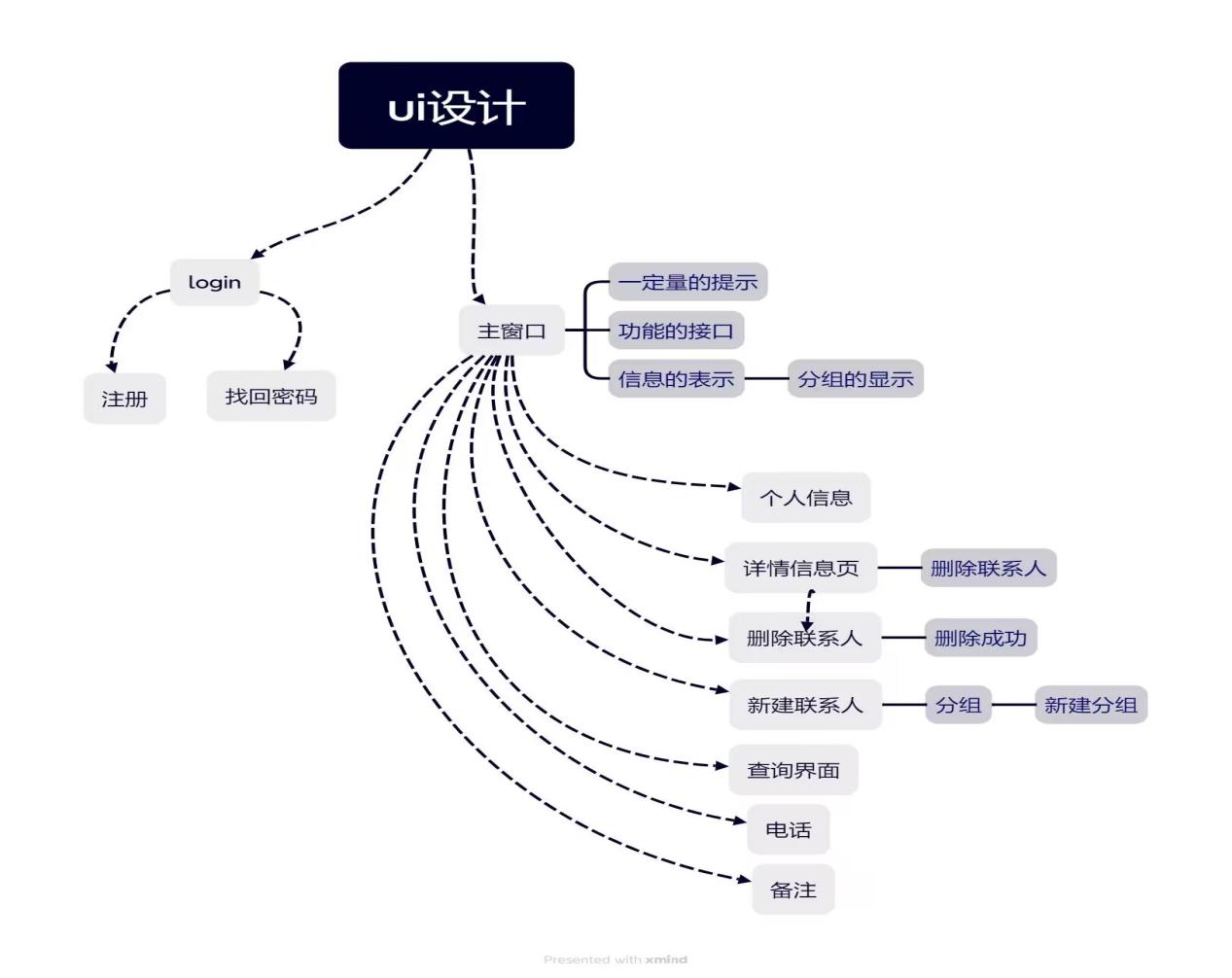
## 1.2 功能需求分析

如下为两个详细设计框架图。

1. 图1为系统功能模块图，我们经过参考以往通讯录的基本功能，有新增一个联系人，删除联系人，更改相关信息，对已有联系人进行排序，以及查询功能，并在此基础上增加了一些个性化的功能，如生日、头像、电话铃声等，为了便于用户自定义设置，为通讯录系统增添色彩。
2. 图2为UI窗口设计图，首先我们设计出了用户登录界面，使程序更具有隐私性，同时也是为了方便管理员对于用户的管理。然后是通讯录主界面窗口及一系列分支功能窗口。



**图1 系统功能模块图**

****

**图2 UI窗口设计图**

（3）下表为本次设计中使用的函数及功能

**表1 函数列表**

|  |  |  |
| --- | --- | --- |
| 序号 | 函 数 原 型 | 功 能 说 明 |
| 1 | on\_pushButton\_clicked+checkusername | 获取创建用户的用户信息 |
| 2 | on\_toolButton\_clicked | 实现密码显示时，眼睛图标的转换 |
| 3 | on\_pushButton\_4\_clicked | 忘记密码按钮实现 |
| 4 | Signup::on\_pushButton\_2\_clicked() | 通过点击退出按钮，实现注册界面的退出 |
| 5 | Retrieverpassword::on\_pushButton\_2\_clicked() | 通过点击退出按钮，实现忘记密码界面的退出 |
| 6 | Queryinterface::on\_pushButton\_3\_clicked() | 通过点击退出按钮，实现查找界面的退出 |
| 7 | Owninformation::on\_pushButton\_2\_clicked() | 通过点击退出按钮，实现个人信息界面的退出 |
| 8 | Newperson::on\_pushButton\_2\_clicked() | 通过点击退出按钮，实现新建联系人界面的退出 |
| 9 | Login::on\_pushButton\_2\_clicked() | 通过点击退出按钮，实现登陆界面的退出 |
| 10 | Detailpage::on\_pushButton\_clicked() | 通过点击返回按钮，实现详情页的退出和返回 |
| 11 | Delete::on\_pushButton\_2\_clicked() | 通过点击退出按钮，实现删除界面的退出 |
| 12 | MainWindow::on\_pushButton\_4\_clicked() | 退出按钮的实现，实例化Owninformation类，并退出 |

## 1.3 数据属性分析

**1.3.1 登陆界面信息属性**

插过插入“eyeclose.png”“eyesopen.png”眼睛图标的图片，来控制密码的显示：

表示在该状态下，密码不可见。

点击图片后变为“eyeopen”，

表示在此状态下，密码可见。

# 2 系统设计

# 2.1 数据存储设计

QString Name=ui->lineEdit->text();

QString Email=ui->lineEdit\_2->text();

QString Password=ui->lineEdit\_3->text();

QString Confirmpassword=ui->lineEdit\_4->text();

string name=Name.toStdString();

string email=Email.toStdString();

string password=Password.toStdString();

string confirmpassword=Confirmpassword.toStdString();

char name\_c[100];

strcpy(name\_c,name.c\_str());

//cout<<name\_c;

if(checkusername(name))

{

QMessageBox::information(this,"提示","用户名已存在");

//this->close();

}

else

{

if(password!=""&&name!=""&&email!="")

{

if(password==confirmpassword)//若相同，记录users的信息

{

const char \*name\_f=name.c\_str();

const char \*password\_f=password.c\_str();

const char \*email\_f=email.c\_str();

FILE \*file;

file=fopen("User.bat","ab+");

fputs(name\_f,file);

fprintf(file,"\n");

fputs(password\_f,file);

fprintf(file,"\n");

fputs(email\_f,file);

fprintf(file,"\n");

fclose(file);

QMessageBox::information(this,"提示","创建成功");

this->close();

}

else

{

QMessageBox::warning(this,"警告","两次输入密码不正确");

}

}

else

{

QMessageBox::warning(this,"警告","输入信息不能为空");

}

}

}

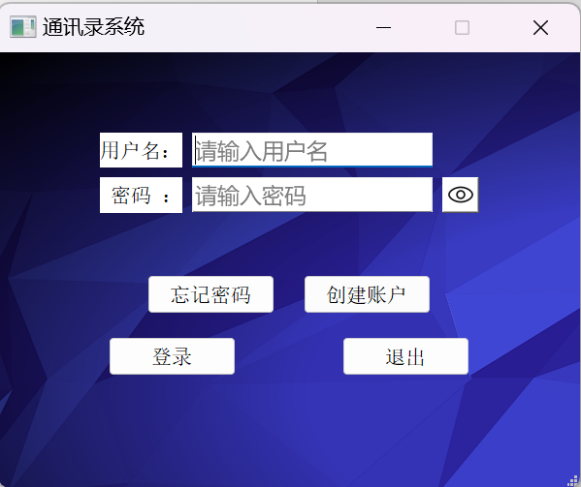
## 创建一个新的文档，向下转型写入数据，来获取创建用户的用户信息，并且储存用户的所有信息，比如用户名，邮箱和密码。然后通过比较，来判断文档里是否存在相同的信息，如果没有则显示创建成功，如果有则显示用户名已存在，并返回。

## 2.2 函数设计与说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **函数原型** | **参数** | **返回值** | **功能描述** |
| **相关函数** | | | | |
| 1 | on\_pushButton\_clicked+checkusername | 无 | 无 | 用来获取创建用户的用户信息 |
| 2 | on\_toolButton\_clicked | 无 | 无 | 实现密码显示时，眼睛图标的转换 |
| 3 | on\_pushButton\_4\_clicked | 无 | 无 | 忘记密码按钮实现 |
| **退出键相关函数** | | | | |
| 1 | Signup::on\_pushButton\_2\_clicked() | 无 | 无 | 通过点击退出按钮，实现注册界面的退出 |
| 2 | Retrieverpassword::on\_pushButton\_2\_clicked() | 无 | 无 | 通过点击退出按钮，实现忘记密码界面的退出 |
| 3 | Queryinterface::on\_pushButton\_3\_clicked() | 无 | 无 | 通过点击退出按钮，实现查找界面的退出 |
| 4 | Owninformation::on\_pushButton\_2\_clicked() | 无 | 无 | 通过点击退出按钮，实现个人信息界面的退出 |
| 5 | Newperson::on\_pushButton\_2\_clicked() | 无 | 无 | 通过点击退出按钮，实现新建联系人界面的退出 |
| 6 | Login::on\_pushButton\_2\_clicked() | 无 | 无 | 通过点击退出按钮，实现登陆界面的退出 |
| 7 | Detailpage::on\_pushButton\_clicked() | 无 | 无 | 通过点击返回按钮，实现详情页的退出和返回 |
| 8 | Delete::on\_pushButton\_2\_clicked() | 无 | 无 | 通过点击退出按钮，实现删除界面的退出 |
| 9 | MainWindow::on\_pushButton\_4\_clicked() | 无 | 无 | 退出按钮的实现，实例化Owninformation类，并退出 |

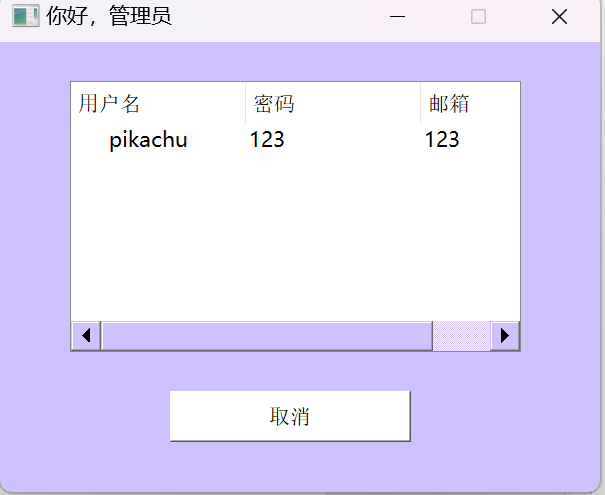
## 2.3 菜单设计

2.3.1登录界面()



登录界面，程序一开始展现给用户的界面，提供了忘记密码，创建用户登录和退出以及眼睛（用来转变密码格式）的几个按钮，以及两个输入文本框用以接受用户输入的数据

2.3.2管理员删除界面（实现）



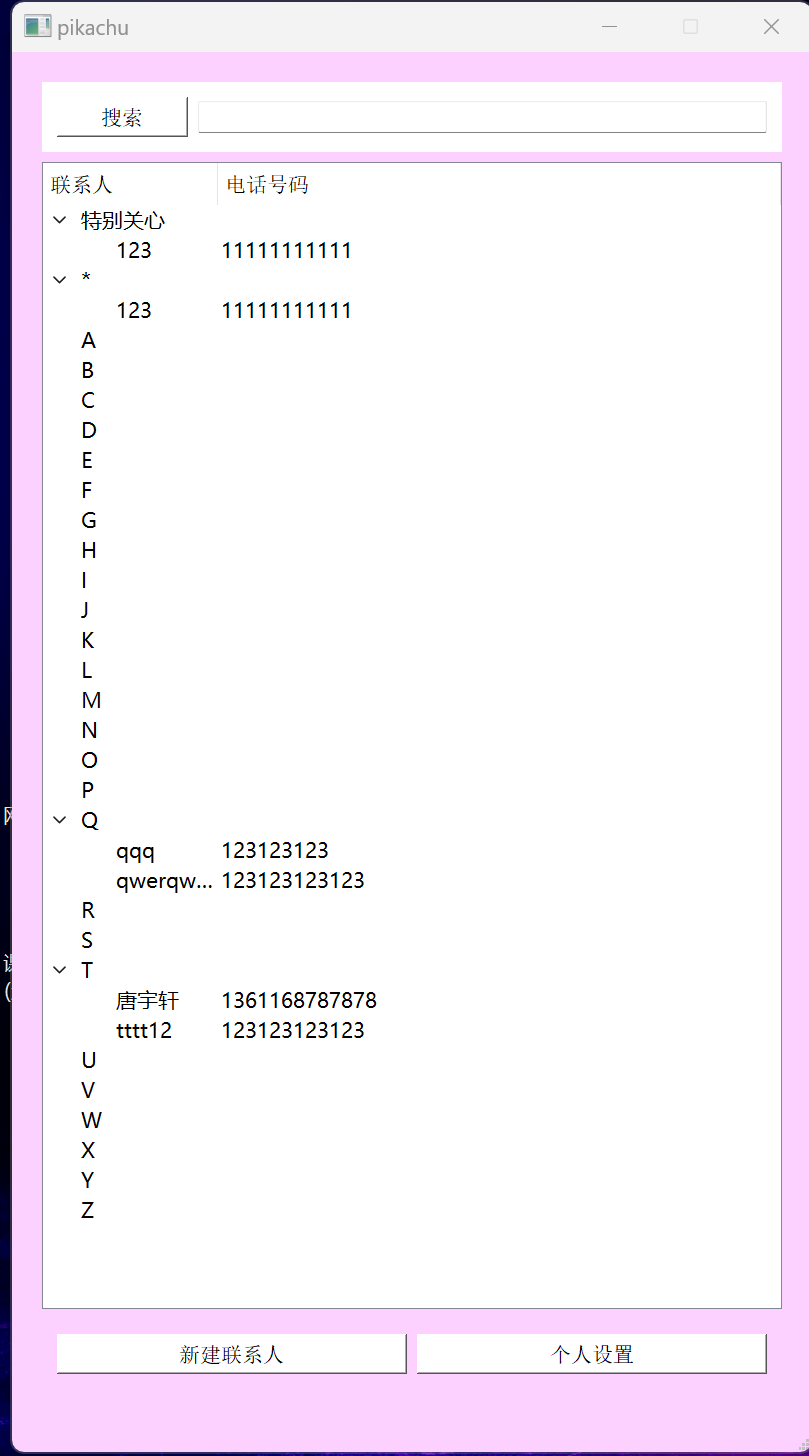
管理员界面，当文本框接收到来自管理员的身份验证的时候，跳转至管理员界面。管理员界面是给管理员进行用户的管理的界面，可以用以查找当前用户，也可以删除当前用户

2.3.3新建联系人界面



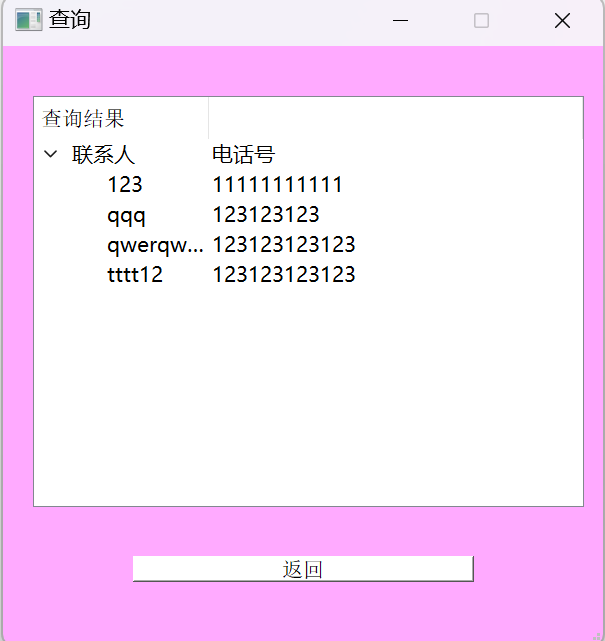
创建用户界面，当用户在登陆界面按下创建用户界面的时候，跳转至此界面，此界面可以完成用户的新建操作，两次输入密码保证了用户密码的正确性，确定创建用户后，在文件中写入用户的信息

2.3.4主界面



当用户在登陆界面输入相应的信息后，跳转至此界面。此界面提供了搜索，个人设置，新建联系人，双击联系人可以跳转至联系人详情页的几个按钮，a-z index直观的展示每个联系人的拼音或者英文首字母位置，为用户检索提供了便捷，我们也实现了中英文混合排序，更加方便了用户的检索

2.3.5查询界面

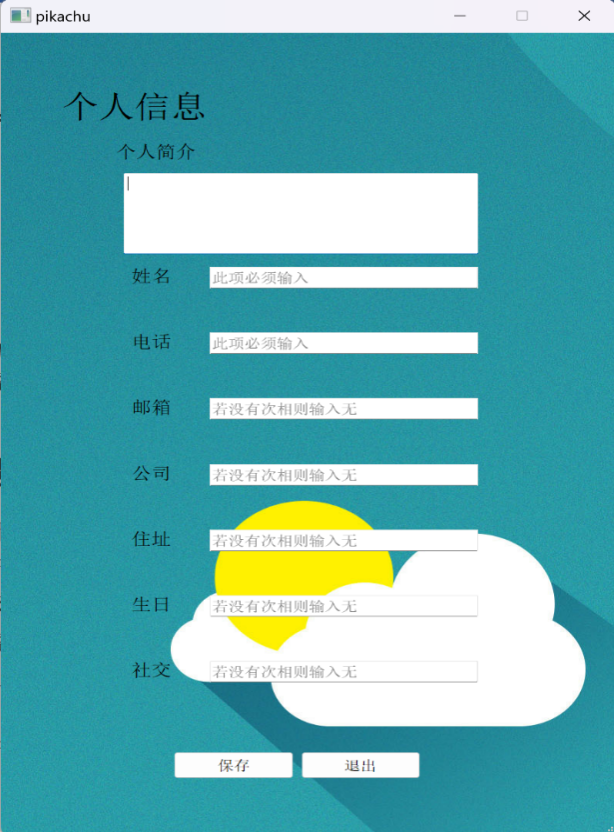
查询界面，当用户在搜索框中输入想要查询的字段，并且按下搜索按钮之后，会跳转至查询界面，查询的结果主要是展现联系人以及电话号两个信息，方便用户挑选用户需要的联系人，双击联系人信息，依然可以跳转至联系人详情页查看联系人，或者对联系人进行修改

2.3.6联系人详情信息



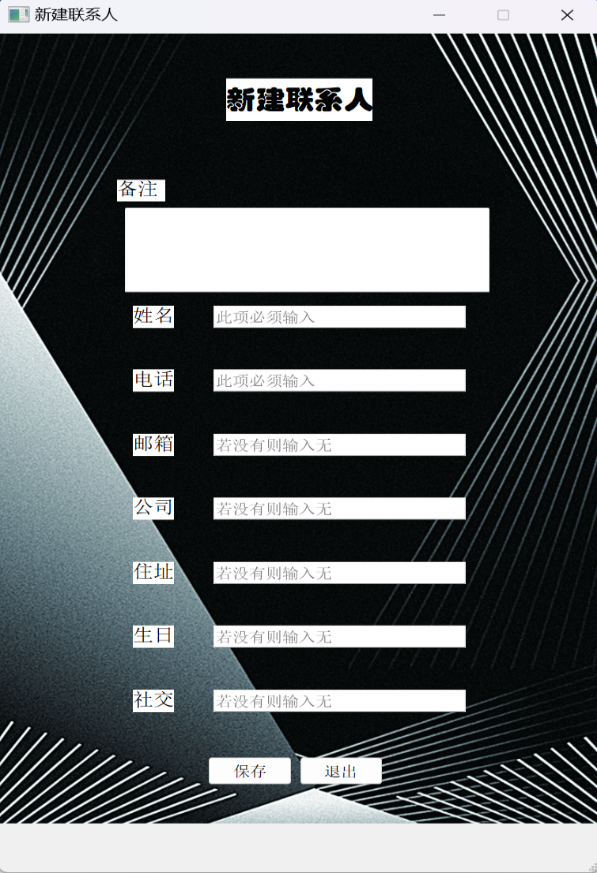
联系人详情页界面，提供了几个文本框用于展示联系人的详情信息，提供了收藏按钮用于用户对联系人进行分组，分享联系人可以导出联系人的信息至文件中，之后用户可以发送给其他人，提供了删除按钮用于删除联系人，最后是保存修改按钮，用于保存用户对于联系人信息的修改

2.3.7个人信息界面



个人信息界面，用于用户存储自己的个人信息，在之后如果用户想要添加联系人，可以展示给其他人看，方便用户添加联系人

2.3.8新建联系人界面

新建联系人界面，在用户按下新建联系人按钮的时候跳转至新建联系人界面，几个有关联系人信息的文本框用户获取联系人的信息，并且在按下保存按钮之后查看电话号码的格式是否正确，再决定是否保存数据

2.3.9忘记密码界面



忘记密码界面，当按下忘记密码按钮的时候跳转至此界面，增加了两个数学式子用于验证是否是用户要修改自己的密码，两个确认密码用于确认用户输入的新密码

**2.4 文件清单与说明**

**表2 文件列表**

|  |  |  |
| --- | --- | --- |
| 序号 | 文件名称 | 文件说明 |
| 1 | User.bat | 用于存储用户信息的文件，为保证其安全性，保存为bat文件。 |
| 2 | Usernew.bat | 在改变用户信息时，用于覆盖久文件的文件，为保证安全性，保存为bat文件。 |
| 3 | username+"connecters.bat" | 用户名加上connecters作为文件名的bat文件，用于存储该用户的联系人信息，为后续展示修改做准备，为保证其安全性，保存为bat文件。 |
| 4 | username+"connectersnew.bat" | 在改变相应用户的联系人的信息时，用于覆盖旧文件的文件，为保证其安全性，保存为bat文件 |
| 5 | username+"bestconnexters.bat" | 用户名加上bestconnexters作为文件名的bat文件，保存了用户特别关心分组的文件，为后续展示修改做准备，为保证其安全性，保存为bat文件。 |
| 6 | username+"owninformation.bat" | 用户名加上owninformation作为文件名的bat文件，用于存储该用户的个人信息，为后续展示修改做准备，为保证其安全性，保存为bat文件。 |
| 7 | username+"owninformationnew.bat" | 在改变相应用户的信息时，用于覆盖旧文件的文件，为保证其安全性，保存为bat文件 |
| 8 | connectsname.txt | 以联系人作为名字的txt文件，在分享联系人创建，为了方便用户读取以及分享，储存为txt文件。 |

3 系统实现

**3.1 新建用户的实现**

QString Name=ui->lineEdit->text();

QString Email=ui->lineEdit\_2->text();

QString Password=ui->lineEdit\_3->text();

QString Confirmpassword=ui->lineEdit\_4->text();

string name=Name.toStdString();

string email=Email.toStdString();

string password=Password.toStdString();

string confirmpassword=Confirmpassword.toStdString();

char name\_c[100];

strcpy(name\_c,name.c\_str());

//cout<<name\_c;

if(checkusername(name))

{

QMessageBox::information(this,"提示","用户名已存在");

//this->close();

}

else

{

if(password!=""&&name!=""&&email!="")

{

if(password==confirmpassword)//若相同，记录users的信息

{

const char \*name\_f=name.c\_str();

const char \*password\_f=password.c\_str();

const char \*email\_f=email.c\_str();

FILE \*file;

file=fopen("User.bat","ab+");

fputs(name\_f,file);

fprintf(file,"\n");

fputs(password\_f,file);

fprintf(file,"\n");

fputs(email\_f,file);

fprintf(file,"\n");

fclose(file);

QMessageBox::information(this,"提示","创建成功");

this->close();

}

else

{

QMessageBox::warning(this,"警告","两次输入密码不正确");}}

}

else

{

QMessageBox::warning(this,"警告","输入信息不能为空");

}

}

}

## 创建一个新的文档，向下转型写入数据，来获取创建用户的用户信息，并且储存用户的所有信息，比如用户名，邮箱和密码。然后通过比较，来判断文档里是否存在相同的信息，如果没有则显示创建成功，如果有则显示用户名已存在，并返回。

**3.2修改眼睛图标，显示密码信息**

void Login::on\_toolButton\_clicked()//实现密码显示时，眼睛图标的转换

{

if(!flag)//如果判定flag为假，显示眼睛开始的状态

{

ui->toolButton->setIcon(QIcon(":\eyesopen.png"));

ui->lineEdit->setEchoMode(QLineEdit::Password);

flag=!flag;//flag标签反转

}

else//反之，显示眼睛关闭的状态

{

ui->toolButton->setIcon(QIcon(":\eyesclose.png"));

ui->lineEdit->setEchoMode(QLineEdit::Normal);//转换密码形式为可见，用户查看密码

flag=!flag;}}

}

}

通过flag来判断来实现眼睛图标的转换，flag在一般情况下判定为真，无法查看具体密码。点击后，改变flag值，判定flag为假，显示眼睛开始的状态，转换密码形式，并显示密码。

**3.3忘记密码的实现**

void Login::on\_pushButton\_4\_clicked()//忘记密码按钮实现

{

srand((unsigned)time(NULL));

int randnum1 = rand() % 10+1;

int randnum2 = rand() % 10+1;

int randchar = rand() % (3-1);

char x;

x=characterx[randchar];

string str=std::to\_string(randnum1)+x+std::to\_string(randnum2);

Retrieverpassword \*retrie2=new Retrieverpassword();

retrie2->ui->lineEdit\_5->setText(QString::fromStdString(str));

int randnum3 = rand() % 10+1;

int randnum4 = rand() % 10+1;

int randchar2 = rand() % (3-1);

x=characterx[randchar2];

str=std::to\_string(randnum3)+x+std::to\_string(randnum4);

retrie2->ui->lineEdit\_8->setText(QString::fromStdString(str));

retrie2->show();}

通过随机数来设置两道十以内的运算题识别用户，避免多余用户占文档内存。指针指向Retrieverpassword()实例，保证忘记密码界面一直存在并显示，再通过向下转型来改写新的密码，并确认新的密码，显示界面，302最后将新的用户名密码覆盖掉原来的，并将原来的用户名密码删除来释放内存。

**3.4.1退出注册界面的实现**

void Signup::on\_pushButton\_2\_clicked()//退出界面

{

Signup::close();

}

通过点击退出按钮，实现注册界面的退出，并关闭注册界面。

**3.4.2退出忘记密码界面的实现**

void Retrieverpassword::on\_pushButton\_2\_clicked()//退出按钮实现

{

this->close();

}

通过点击退出按钮，实现忘记密码界面的退出，并关闭界面。

**3.4.3退出查找界面的实现**

void Queryinterface::on\_pushButton\_3\_clicked()//退出按钮实现

{

this->close();

MainWindow \*main=new MainWindow(username);

main->ini();

main->show();

this->close();

}

通过点击退出按钮，实现查找界面的退出，指针指向MainWindow(username)实

例，保证主界面一直存在并显示，退出查找界面并跳转到主界面。

**3.4.4退出个人信息界面的实现**

void Owninformation::on\_pushButton\_2\_clicked()//退出按钮实现

{

this->close();

MainWindow \*main=new MainWindow(username);

main->ini();

main->show();

}

通过点击退出按钮，实现个人信息界面的退出，指针指向MainWindow(username)

实例，保证主界面一直存在并显示，退出个人信息界面并跳转到主界面

**3.4.5退出新建联系人界面的实现**

void Newperson::on\_pushButton\_2\_clicked()//退出实现

{

this->close();

MainWindow \*main=new MainWindow(currentname);

main->ini();

main->show();

this->close();}

}

通过点击退出按钮，实现新建联系人界面的退出，指针指向MainWindow(currentname)

# 实例，保证主界面一直存在并显示，退出新建联系人界面并跳转到主界面。

**3.4.6退出登录界面的实现**

void Login::on\_pushButton\_2\_clicked()//退出键的实现，按下按钮退出

{

Login::close();

}

# 通过点击退出按钮，实现登录界面的退出，并关闭界面。

**3.4.7退出详情页界面的实现**

void Detailpage::on\_pushButton\_clicked()//返回功能实现

{

MainWindow \*main=new MainWindow(username);

main->ini();

main->show();

this->close();

}

通过点击退出按钮，实现详情页界面的退出，指针指向MainWindow(username)

# 实例，保证主界面一直存在并显示，退出详情页界面并跳转到主界面。

**3.4.8退出删除界面的实现**

void Delete::on\_pushButton\_2\_clicked()//退出按钮实现

{

this->close();

}

# 通过点击退出按钮，实现删除界面的退出，并关闭界面。

**3.4.9退出主界面的实现**

void MainWindow::on\_pushButton\_4\_clicked()//个人信息

{

Owninformation \*own=new Owninformation(username);

own->show();

this->close();

}

通过点击退出按钮，实现主界面的退出，指针指向Owninformation(username);

# 实例，保证个人信息界面一直存在并显示，退出主界面并跳转到个人信息界面。

# 4 测试数据与测试过程记录

## 4.1.1 创建用户界面



图 创建用户ui界面 图 创建用户成功ui界面

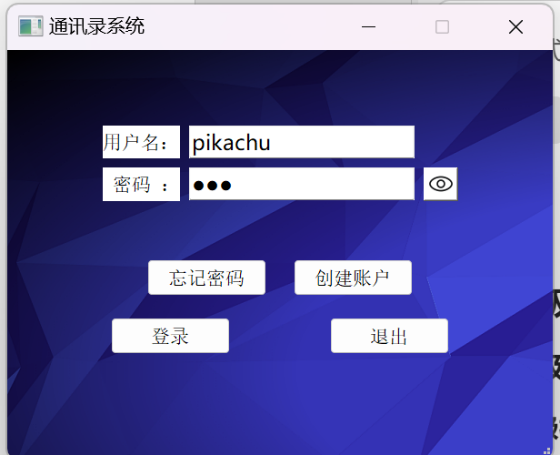
## 4.1.2 忘记密码界面

## 



图 忘记密码ui界面 图 更改密码成功ui界面

**4.1.3 转换眼睛图标，显示密码**



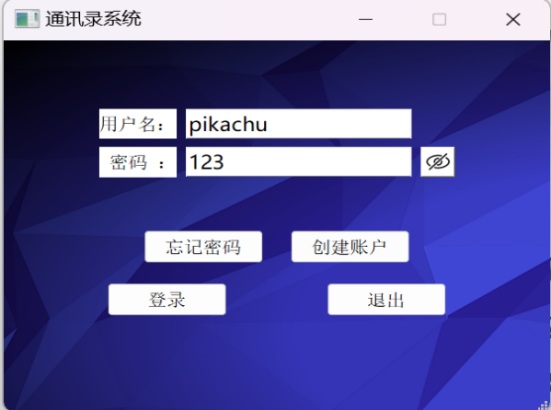
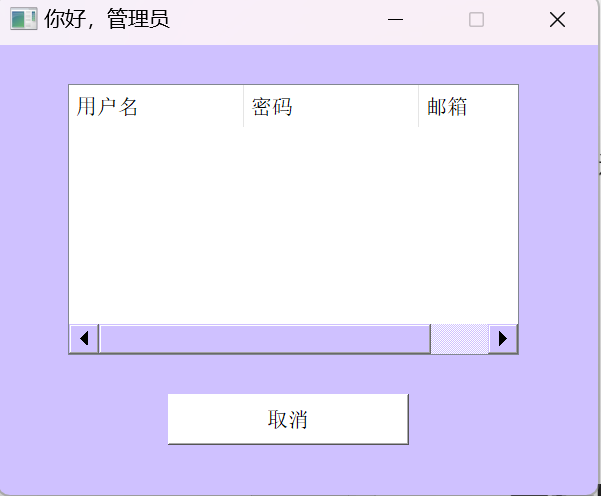


图 忘记密码ui界面

图 显示密码界面 图 隐藏密码界面

**4.1.4退出键的实现**



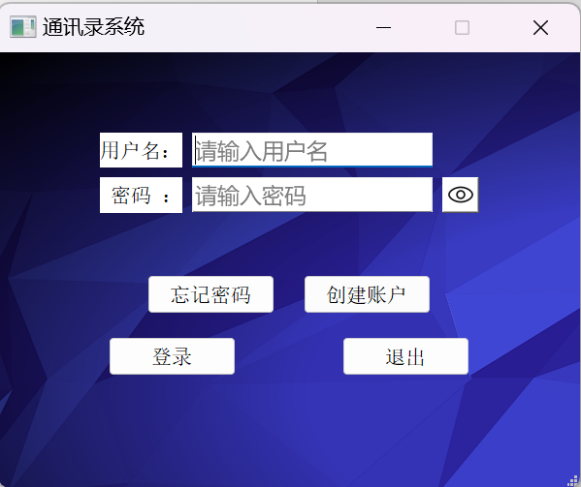


图 登录界面退出键 图 管理员界面退出键

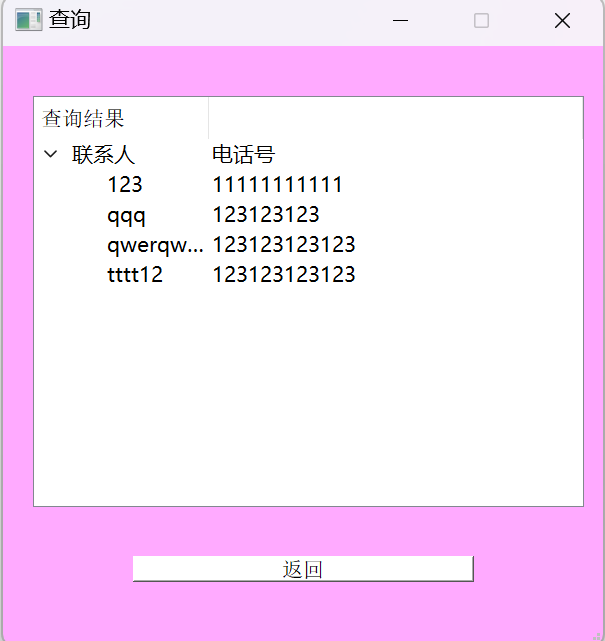




图 创建用户界面退出键 图 查询界面退出键

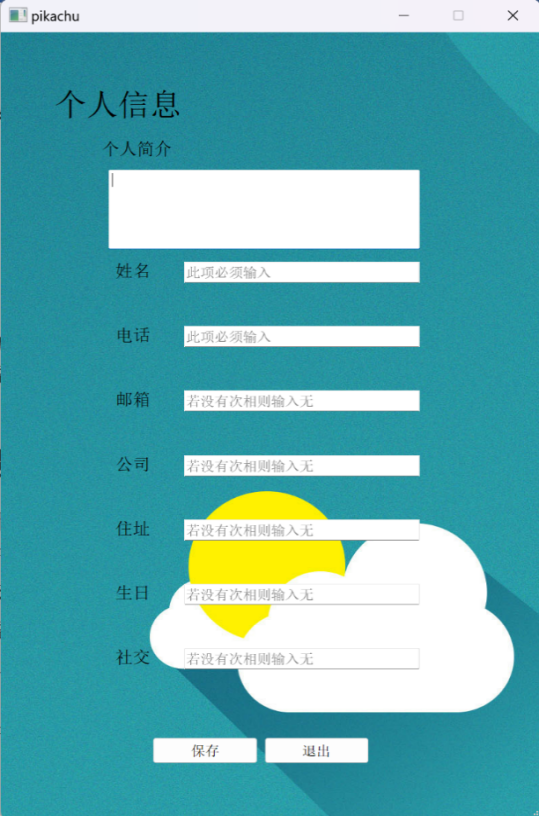




图 个人信息界面退出键 图 个人信息界面退出键





图 新建联系人界面退出键 图 忘记密码界面退出键

# 5 课程设计遇到的问题及解决方法

课程设计过程中随时记录遇到的问题，每个问题描述问题名称、问题内容、对问题的分析和解决方法，如下为示例。

（1）问题名称：眼睛图标ui美化设计

问题内容：在插入眼睛图标后，会被后面美化的图片遮挡住，显示出一块黑框，无法看到眼睛图标的变化

问题分析：没有添加背景图片和颜色，导致无法显示

解决方法：添加了新的白色背景以及更改了白色的背景颜色，使图标成功显示。

（2）问题名称：ui美化设计无法显示图片

问题内容：在插入图片做ui美化的时候，预设里可以看到图片已经被插入，但是实际运行则没有

问题分析：图片类型及保存的形式不对，导致无法显示

解决方法：重新下载并以“.png”结尾保存图片

（3）问题名称：不同电脑之间ui美化图片不兼容

问题内容：在一台电脑上做的美化在另一台主要演示电脑上无法展示

问题分析：可能是因为上传时出错，也可能因为每个电脑所配置的环境不同，导致美化图片无法显示

解决方法：集中在主要演示电脑上重新做美化，然后重新再上传。

# 参考文献

1. 算法竞赛入门经典/刘汝佳，北京：清华大学出版社2009.11
2. 数据结构：c语言/严蔚敏，吴伟民，北京：清华大学出版社1997.4
3. C++ Primer Plus中文版：第六版/（美）史蒂芬-普拉达；张海龙，袁国忠译北京：人民邮电出版社，2020.7
4. C程序设计教程与实验/吉顺如，北京：清华大学出版社，2022.1
5. Qt5.12实战 朱晨冰 / 李建英 北京：清华大学出版社