JAVA课程大作业程序设计说明书

1. 项目概述
2. 选题原因

由于项目管理系统是一个在学习生活中比较常见的系统，而且在相关开发中需要实现多种功能，还需要给两种身份提供不同的界面（老师和学生所需功能不同），非常具有实用性和挑战性，因此我选择实现的题目是：国际校区学生项目信息管理系统。

1. 题目分析

（1）这个系统的主要定位是便于国际校区的老师和学生对项目信息进行管理，因此项目信息和用户信息都需要进行存储，这里我统一使用txt文件进行存储，原因主要为txt文件体积小而且读取、修改速度快。

（2）每一个项目需要存储的信息类型是相同的，每种身份对应的用户信息类型也是相同的，老师和学生这两种不同身份的信息类型部分一致，因此在设计程序时，需要用到类的思想和类的继承结构。

（3）出于安全考虑，和绝大多数系统一样，都需要设置密码登录功能，最好和大多数校内系统一样，先提供一个可直接登录的初始密码，同时可以在进入系统后对密码进行修改。

（4）老师可以访问到所有学生的项目，学生只能访问到自己的项目，这需要在程序中对这两种身份涉及的相关函数进行访问权限上的设计，同时还需要提供增、删、改等功能。。

（5）因为老师角色和学生角色所需要的功能部分一致，界面需求只有略微不同，因此我认为在程序设计中，一些函数和UI界面的复用将大大减少工作量，且便于进行统一修改，提高维护的效率。

（6）因为系统需要UI界面，为了能够设计简洁且美观的界面，我使用了JAVAFX+SceneBuilder进行开发，可以可视化地对各种控件进行布局和调整。

1. 功能模块的设计和划分

总体上来说，根据系统的功能要求，我大概将其分为以下三个子模块：

（1）账号管理子模块

实现使用账号密码登录的系统入口，提供初始密码登录功能和密码修改功能。

（2）个人信息管理子模块

学生信息有：学院、专业、年级、学号、姓名；

老师信息有：学院、工号、姓名。

主要功能为在用户登录后读取并显示个人信息，同时提供个人信息修改功能。

（3）项目信息管理子模块

学生可以查看自己的所有项目，并且可以添加、修改和删除项目信息；

老师可以查看所有学生的项目，并且可以添加、删除学生项目，不过无法进行修改。

同时，两个身份都共享同样的按某类别进行排序的功能。

（此外，老师角色还附加有统计功能。）

1. 开发环境

Operating System：Windows XP

java version : 19.0.2

Java(TM) SE Runtime Environment (build 19.0.2+7-44)

Java HotSpot(TM) 64-Bit Server VM (build 19.0.2+7-44, mixed mode, sharing)

Tools: JAVAFX框架 + SceneBuilder可视化布局工具

Javafx version: javafx-sdk-19

1. 程序概要设计
2. 功能模块图如下：

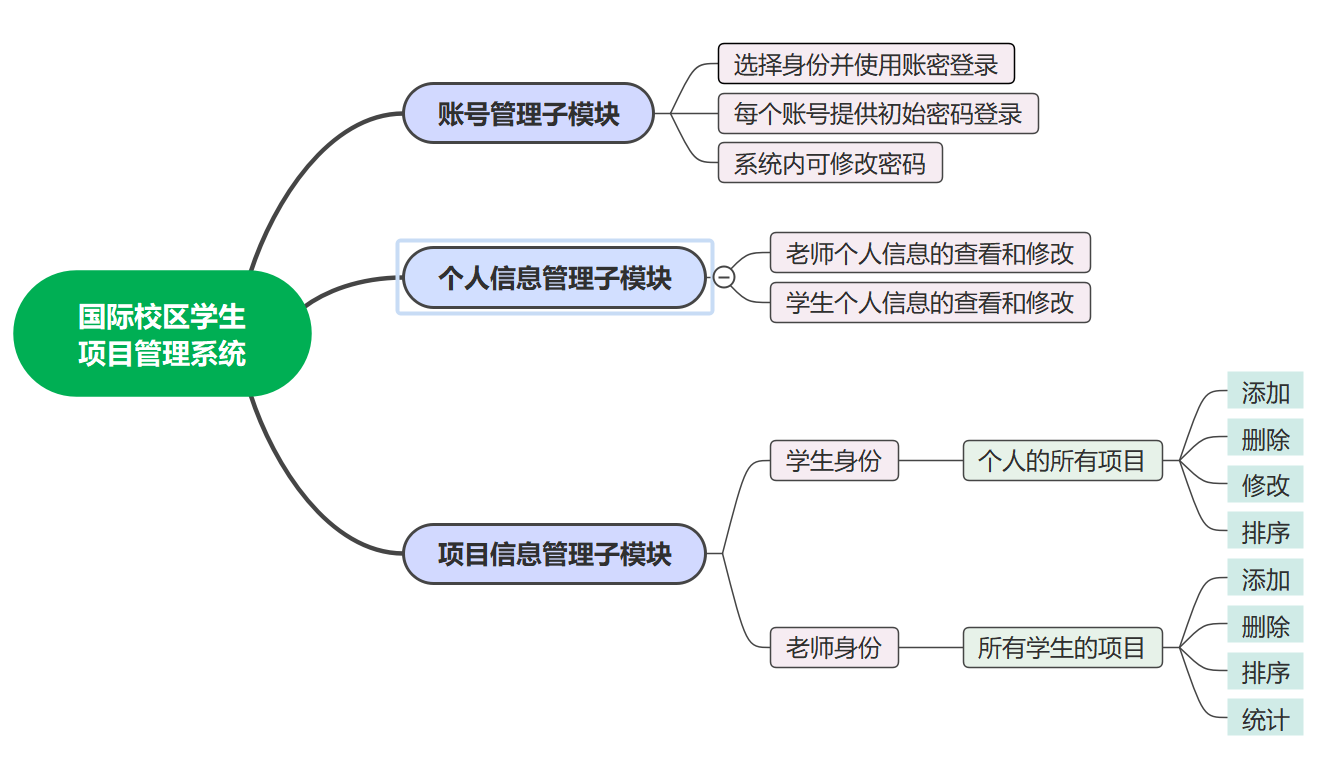


图1 国际校区学生项目管理系统功能模块图

1. 程序系统的总流程图如下：

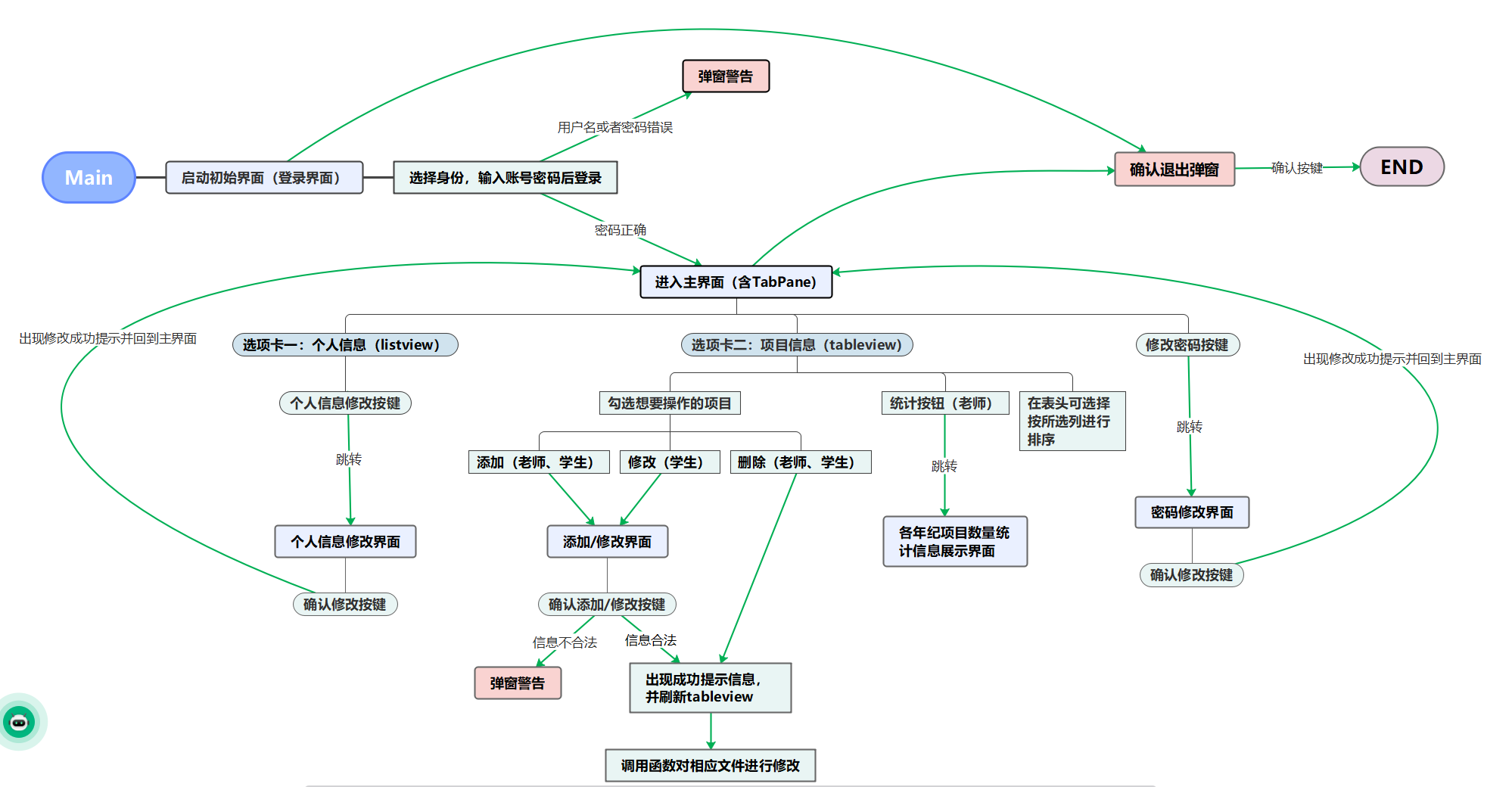


图2 程序（系统）总流程图

1. 部分重要子功能的流程图

（1）登录功能

这里分为两部分来演示，第一个流程图是总体上的登录逻辑，第二个流程图演示的是第一个图中用到的checkexit内部的逻辑。

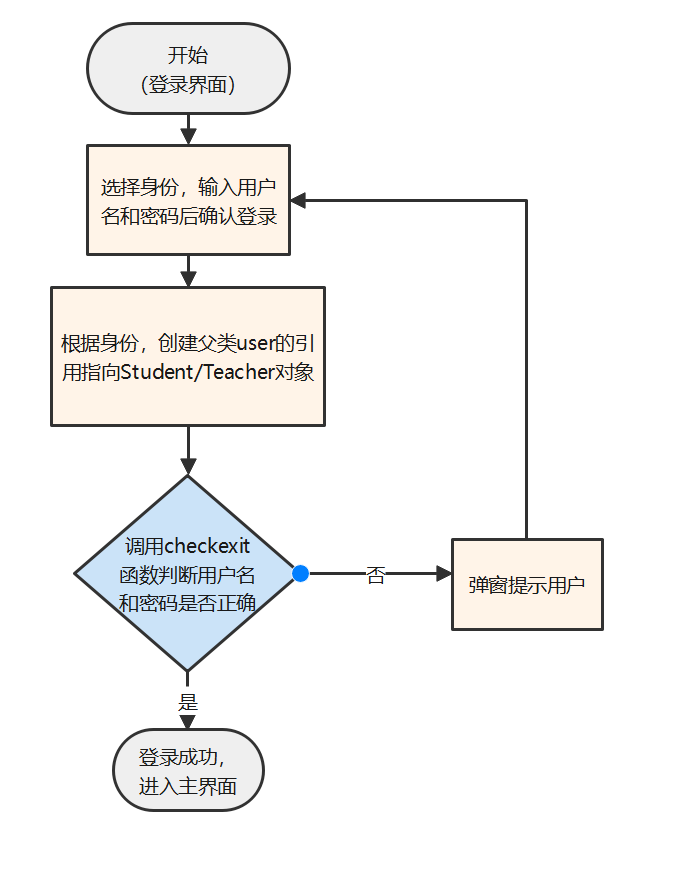


图3 登录总逻辑流程图



图4 关键函数checkexit流程图

（2）个人信息的修改

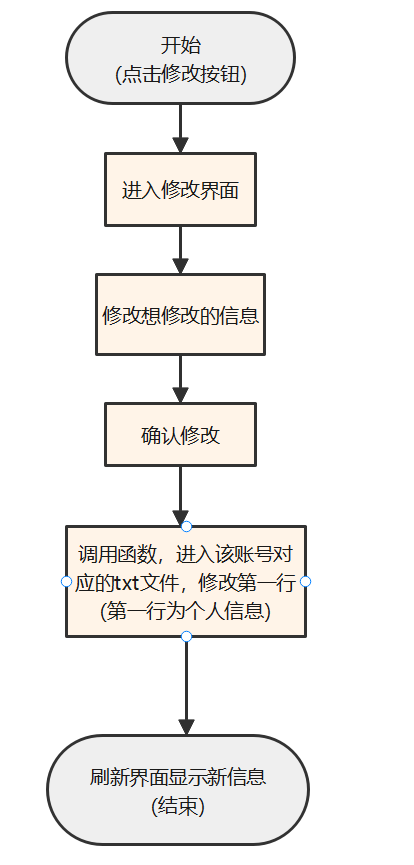


图5 个人信息修改流程图

（3）项目信息的读取

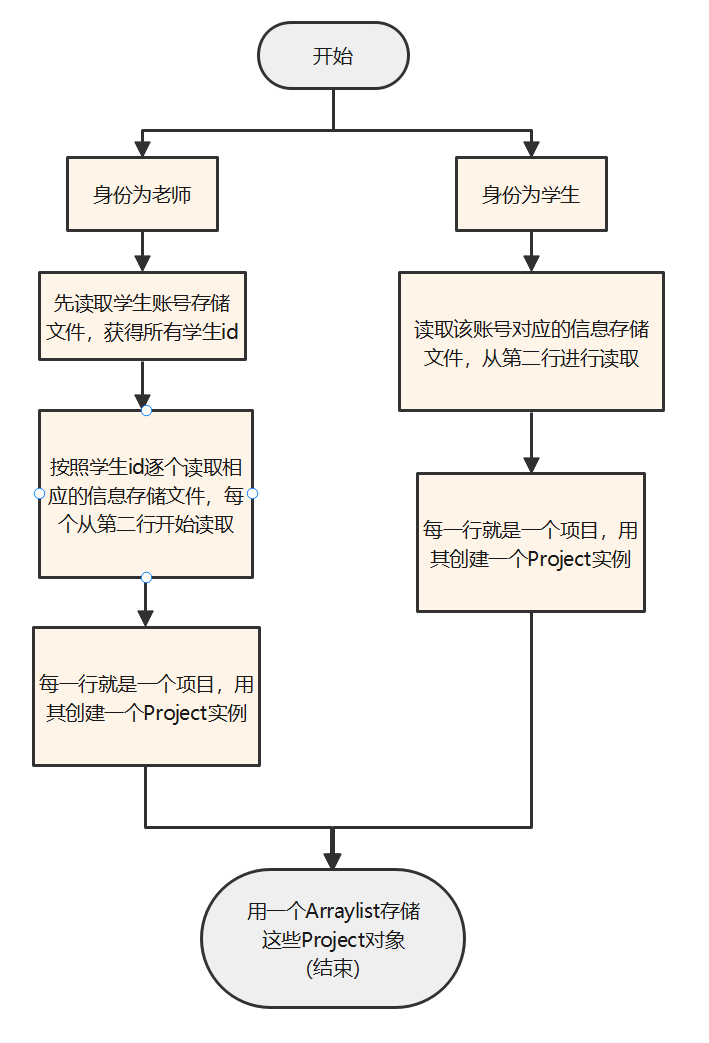


图6 项目信息读取流程图

（4）项目信息的增加/修改



图7 项目信息添加/修改流程图

（5）项目信息的删除

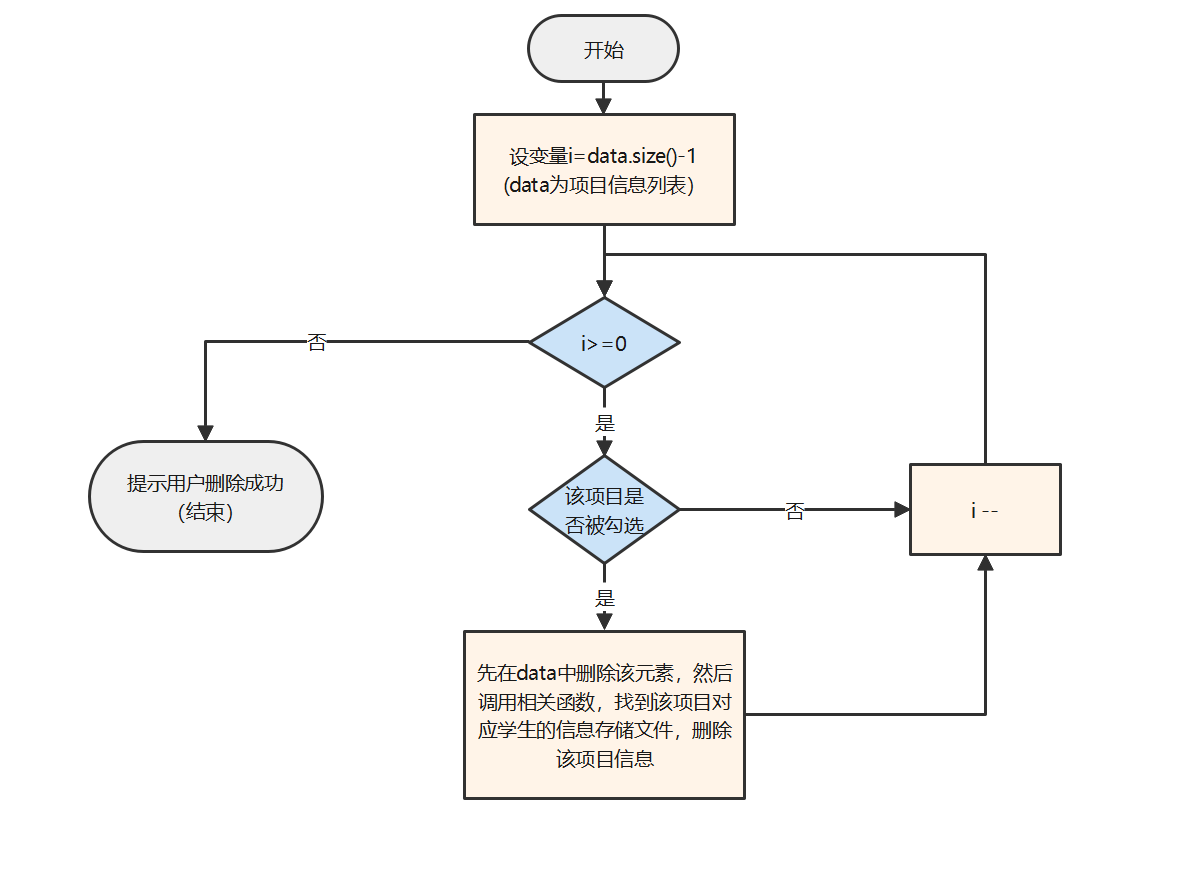


图8 项目信息删除流程图

1. 程序文件结构设计

整个项目采用的是Maven框架结构，下面对重要的目录结构进行说明：

--src

包含了项目所有的源代码和资源文件，以及其他项目相关的文件。

--main

该项目主要的开发内容。

--java/com/example/javafx

存放这个项目的java源代码。

--recources/com/example/javafx

存放这个项目的所有资源文件，包括fxml文件、图片资源和txt文件等。

--target

存放编译后的版本。

因此，整体的文件结构如下：

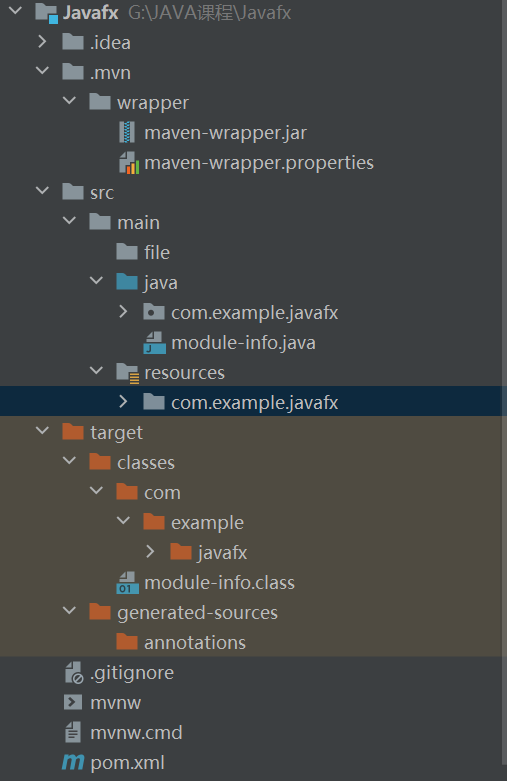


图8 程序文件结构示意图

模块图为：

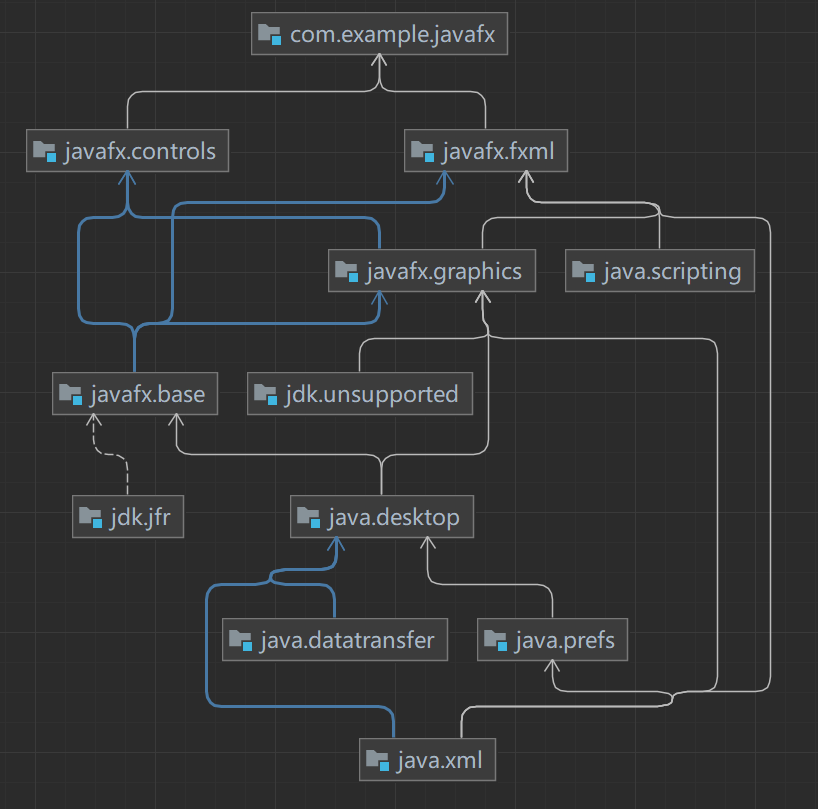


图9 此项目的模块图

1. 类的设计（类图）

（1）用户类的设计

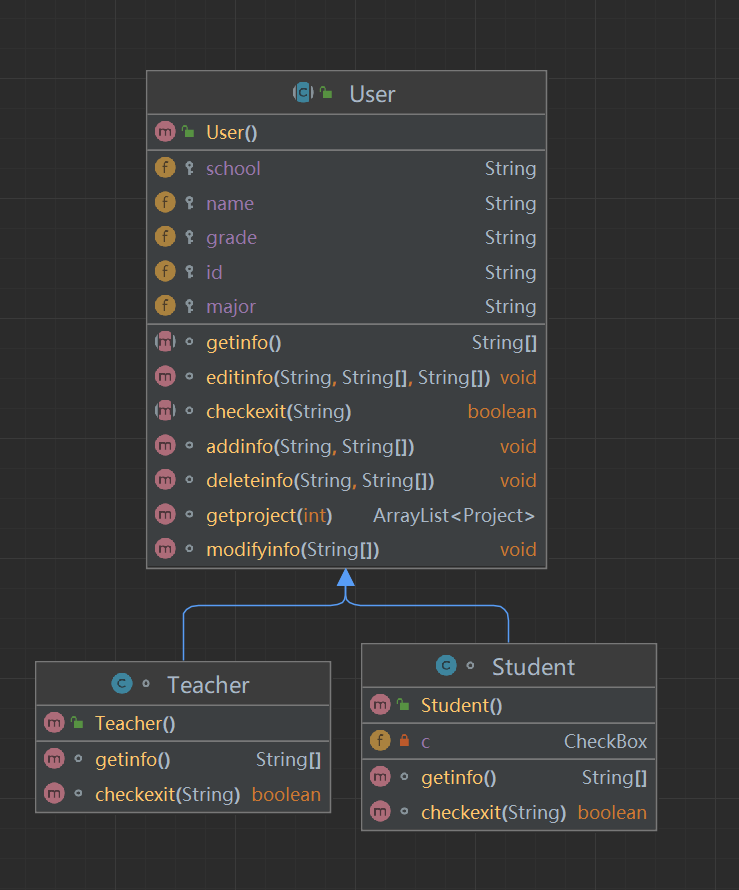


图10 User类及其子类

（2）项目信息类的设计

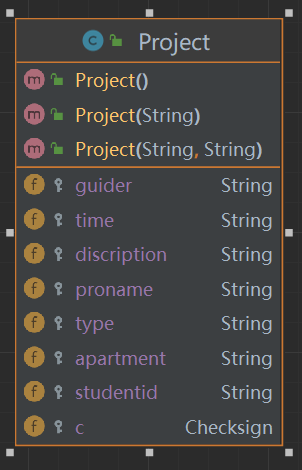


图11 项目信息类

（3）勾选框类的设计

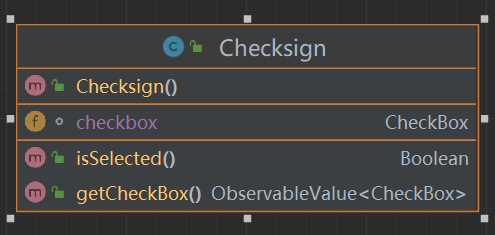


图12 勾选框类

（4）和UI有关的类的设计（Controller）

|  |  |  |
| --- | --- | --- |
| 图13 确认关闭弹窗 | 图14 个人信息修改界面 | 图15 登录界面 |
| 图15 主界面 | 图16 密码修改界面    图17 统计界面 | 图18 项目添加/修改界面 |

1. 程序详细设计

注：①在对文件进行操作时，我是先调用System.getProperty("user.dir")函数获取了项目所在路径，然后得到main/resource所在路径，从而得到这些txt文件的路径进行访问。

②在文件存储中，我将每行作为一条信息，每个信息中不同种类的信息用‘，’分隔开。

1. 账号管理子模块中一些关键功能的实现

（1）登录

我的设计思路是每一个用户在第一次登录时初始密码统一预设为“123456”，登录后就会自动创建账号，同时可以在系统内修改密码，具体实现如下：

在Main中，我定义了两个成员变量：static修饰的的User类型变量user和static修饰的int类型变量identity，前者会在用户登录时根据用户身份指向Student类型对象或Teacher类型对象，后者则直接代表身份（1为学生，0为老师）。

在登录时，当检测到登录按键按下时，绑定的事件函数会被调用。首先，根据用户输入的类型对user和identity进行赋值，然后调用checkexit函数进行检测，checkexit函数返回值是boolean类型的变量，这里用if else语句进行判断，如果返回true则表示登陆成功，进入主界面，如果返回false就会有弹窗提示，告诉用户用户名或者密码不正确。

在checkexit函数中，首先使用正则表达式对用户输入的用户名进行判断，因为这个系统以学号/工号为用户名，我假设固定格式为12位，如果格式不对则直接返回false，就会有弹窗提示。

接着，我会根据身份打开对应的账号存储文件：user\_student.txt或者user\_teacher.txt，这两个文件分别存储了学生的账号及密码和老师的账号及密码，每一行对应一个账号。我遍历读取文件的每一行，存入ArrayList中，然后遍历它判断账号是否存在，存在则核对密码，密码正确返回true，密码错误返回false。如果账号不存在，证明是第一次登录的账号，若密码为“123456”，则创建该账号对应的信息存储txt文件，也就是以学号/工号命名的txt文件。这个文件第一行是个人信息，从第二行开始是项目信息，同理密码正确返回true，密码错误返回false。

弹窗提示如下：

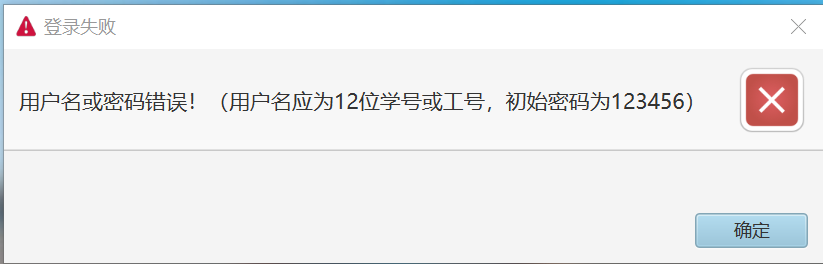


图19 登录失败的弹窗提示

（2）密码可见模式可调

因为出于安全考虑，密码输入框的密码在输入时是不可见的，javafx中也有这样一个控件叫做PasswordField，但是用户可能因为无法看见密码具体内容而无法判断自己有没有出现输入错误，因此密码可见模式可调功能就派上用场了。和目前主流的登录界面一样，我也在密码框旁边设置了一个小眼睛，用户可以点击这个小眼睛，设置它的开闭，也就是密码的可见模式。

具体实现方法为：

①在密码输入框处放置两个输入框控件，一个为输入不可见的PasswordField，一个为输入可见的TextField，两个控件的位置重合在一起。

②在输入框的右侧放一个小按钮，按钮的图片样式初始时设为闭着的小眼睛，对应着，PasswordField的visible属性设为true，TextField的visible属性设为false，也就是说默认情况下是密码不可见，同时我还在这个类中设置了一个boolean类型的成员变量check，初始值为true，用于辅助判断。

③当用户点击小眼睛时，按钮被触发会调用相应事件函数，在函数中先检测check的值，如果为true则让TextField可见而PasswordField不可见，同时获取PassswordField中用户输入的文本，传入TextField中，再把check置为false，同时把按钮上的图片换为睁开的小眼睛;如果check为false，则反过来进行操作。

效果如下：



图20 密码不可见



图21 密码可见

（3）修改密码

我在主界面右上角设有修改密码按键，便于用户对自己的密码进行修改。同样，为了避免用户一次输入有误设错密码，也提供了第二次输入确认密码的输入框。

代码中的逻辑为：首先对两个密码输入框中的密码进行比对，如果不一致就出现弹出弹窗警告提醒用户核对密码，如果一致就进行新密码的存储。同样是根据身份打开对应的账号信息存储文件，逐行读取，存入ArrayList中后遍历判断，找到对应的学号/工号（即用户名）就将原密码修改为新密码，这里用到的是简单的字符串拼接并赋值。

1. 个人信息管理子模块中一些关键功能的实现

（1）个人信息的读取

关于个人信息的显示，我使用的是listview，首先设了一个String类型的ObservableList成员变量dataList,与listview相关联。然后调用user的getinfo()函数来获得个人信息。

getinfo():User类中的抽象函数，在Student和Teacher中分别得到实现。除了信息类别不同，这两个子类中此函数的代码逻辑是一致的。首先，结合路径与文件名（学号/工号.txt）得到文件路径，因为我用第一行存储个人信息，所以这里可以直接使用RandomAccessFile来读取文件，首先调用seek函数将光标移到开头，然后读取一行，这一行就是个人信息，因为存储时采用‘，’作为分隔符分开几种信息，所以这里直接使用split函数取出信息，赋值给user的各个变量，最后配上小标题后按顺序存入一个字符串数组中，传回主函数。

（2）个人信息的修改

对于两个身份，我使用的个人信息修改界面是相同的，不过在界面初始化时会根据身份将对应的控件设为visible，不需要的控件设为invisible。

①在点击修改按键后，个人信息修改界面就会打开，因为之前初始化时已经把个人信息存进user的变量中了，所以这里不需要再读取文件就能直接获取。首先将各输入框中的文字设置为旧的个人信息，用户修改后点击确认按键，就会进入绑定的事件函数。

②在这个函数中，首先对身份进行判断，然后把修改后的个人信息存入一个字符串数组中，然后将这个数组传入modifyinfo()函数。

③和getinfo()函数一样，modifyinfo()函数同样也是User类中的抽象函数，两个子类中分别实现。在这个函数中，先调用了Files.readAllLines函数，读取txt文件中的所有行，存到ArrayList中，然后对第一个元素进行修改，最后再重新写入txt文件，实现对第一行个人信息的修改。

1. 项目信息管理子模块中一些关键功能的实现

（1）项目信息的读取

为了更方便地对项目信息进行操作，我定义了Project类，定义了两个构造函数。

①Project(String content) ②Project(String id,String content)

这两个构造函数分别对应了不同的应用场景，对于学生身份，项目信息中不需要含有学号信息，对于老师身份，每条项目信息都需要还有学生学号的信息。

对于项目信息的展示，我使用的是tableview，也定义了一个Project类型的ObservableList成员变量data与其相关联。tableview在主界面初始化时得到初始化，第一步是判断身份，然后采取不同的逻辑：

①学生：根据学号读取对应的信息存储txt文件，调用Files.readAllLines函数将每行信息存入String类型的ArrayList中，然后从第二个item开始，调用第一个构造函数，利用每个item的信息创建Project实例，最后返回一个存储了此学生所有项目的ArrayList。

②老师：首先，需要读取user\_student.txt文件，获取所有学生的账号，然后逐个打开对应的信息存储文件进行读取，大体步骤同上。有一个区别就是在构建Project实例时，因为需要加入学生学号信息，所以会调用第二个构造函数，以便传入ID。

（2）项目信息的增加/修改

因为项目信息的增加和修改界面的构成是相同的，所以我也是复用了同一个UI界面，不过在此界面对应的Controller中，设置了一个String类型的成员变量mode，用来辅助判断。具体步骤如下：

①在用户点击增加/修改按键后，就会进入增加/修改界面，这里在初始化该界面时会对mode进行判断，从而调整确认增加/确认修改的按钮的可视属性，同时修改界面还需要根据勾选情况找到要修改的那一栏，在输入框中显示旧的项目信息。因为修改功能只能针对某一栏信息，所以不能在同时勾选多个项目时进行修改，否则会出现如下提示：

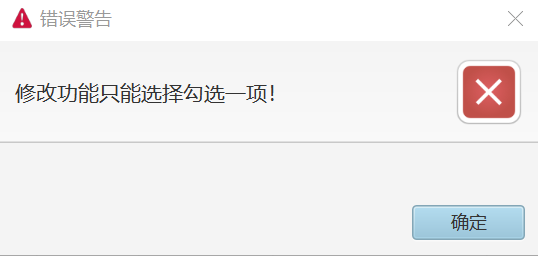


图22 多项修改的弹窗警告

②在确认按键点击后，会进入绑定的事件函数，前半部分增加和修改的操作是一样的，都是对信息合法性的判断，我设置的原则为项目名称不能为空和项目时间必须有效。

③第一个判断的是项目时间的有效性，这里我使用了正则表达式，确保输入的日期格式是（yyyy-MM-dd），这一格式我已经放在了日期输入框的提示词中。如果输入的日期格式不对，会出现弹窗警告，对于日期格式的弹窗警告如下：



图23 日期格式错误的弹窗警告

检查完日期格式之后就是检查日期是否合法，这里使用了SimpleDateFormat的构造函数帮助判断，这里使用了try…catch语句块，如果成功构造证明日期是有效的，如果捕捉到了错误进入catch语句，证明日期无效。具体函数如下：



图24 检查日期合法性的函数

日期不合法（如月份大于12、日期超过该月理论的最大天数等）的弹窗警告如下：

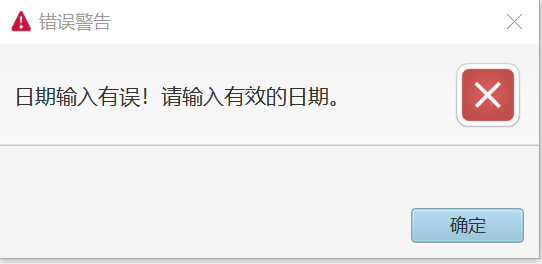


图25 日期不合法的弹窗警告

④若项目时间是有效的，则继续判断项目名称是否为空，这里检测的是：namefield.getText().trim().equals("")

如果为false就证明项目名称不为空，可以进行进一步操作，如果为true则弹窗警告用户。

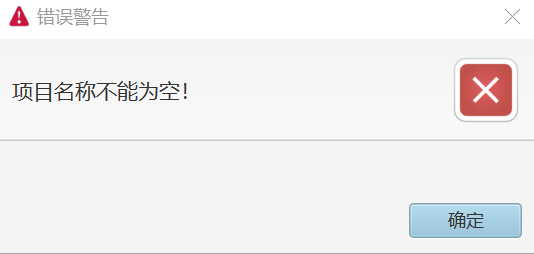


图26 项目名称为空的弹窗警告

此外，为了程序正常运行不报错，这里继续对其他输入框的输入内容进行判断，如果为空则设为空格再进行存储，否则会遇到null的情况出现报错。

如果用户输入的信息合理性通过了检验，程序中就会对和tableview关联的ObservableList进行对应的操作，让显示的项目信息得到更新。

⑤接下来就是文件层面的操作，这里添加和修改两种操作有略微的区别。

* 添加

这里调用了User类的addinfo函数，这个函数主要有两个参数，第一个是此项目对应的学生学号，第二个参数是一个含有项目信息的字符串数组。然后和之前的文件操作相同，逐行读取对应的信息存储文件，存入ArrayList中，再将项目信息按格式拼接为字符串，调用add函数加入到ArrayList中，最后再重新写入txt文件。

* 修改

这里调用的是User类的editinfo函数，有三个参数需要传入：第一个为学生学号，第二个为修改后的项目信息（字符串数组），第三个为旧的项目信息（字符串数组），接下来的操作与添加类似，不过最后一步是遍历ArrayList的每行，找到与旧的项目信息匹配的item，将其替换为修改后的项目信息，然后再重新写入txt文件。

（3）项目信息的删除

因为我在tableview中还添加了勾选框这一列，因此勾选项目会更加方便。修改功能只能选择一个，但是删除功能不受这个限制，可以多选进行批量删除。

因为我是遍历data进行判断，进行删除时会缩短data长度，会发生索引溢出的报错情况，所以为了解决这一问题，我采用的是倒序遍历：

①从后往前进行判断，调用当前item的checkbox的isSelected()函数对是否被选中进行判断，若是被选中，先调用user的deleteinfo函数对文件进行操作，然后在data中删除该项，实现。

②User类的deleteinfo函数一共有两个参数，第一个是学生学号，第二个是由项目信息构成的字符串数组。进入函数后，同样是逐行读取对应的信息存储文件存入ArrayList中，然后遍历寻找和传入的项目信息一致的item，找到之后记录该item对应的索引并退出循环，出了循环后调用remove函数删除该item，最后把整个ArrayList重新写入txt文件。

（4）项目信息的排序功能

因为tableview自带有升降序排序功能，但是排序逻辑是系统自带的，这里需要对tabelview的排序逻辑进行重写，在setSortPolicy方法中进行修改。

①对于获得时间、项目名称和指导老师，我直接调用的是String类中重写的compare函数，默认是按照ASCII码值进行排序，所以无论是日期还是时间，因为格式已经固定好，且都满足从左往右进行判断，所以可以直接使用String的排序规则。

②对于项目类型，我重新写了升降序逻辑，设院级项目为最小项，国际级为最大项。具体逻辑为定义一个字典，key为项目类型，value为0到5的数字，分别对应各级类型，然后比较这个数字即可。

1. 程序测试
2. 登录界面演示：

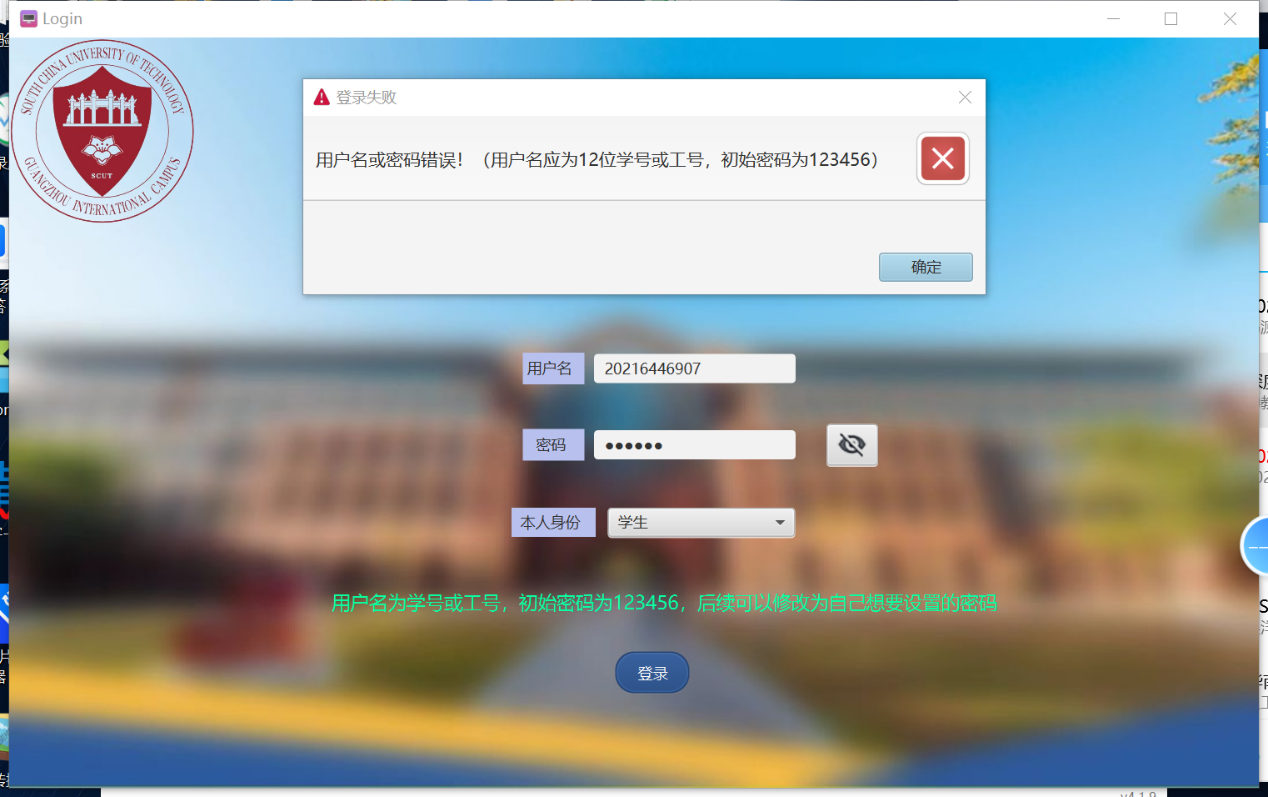


图27 登录失败

1. 个人信息修改

（1）学生

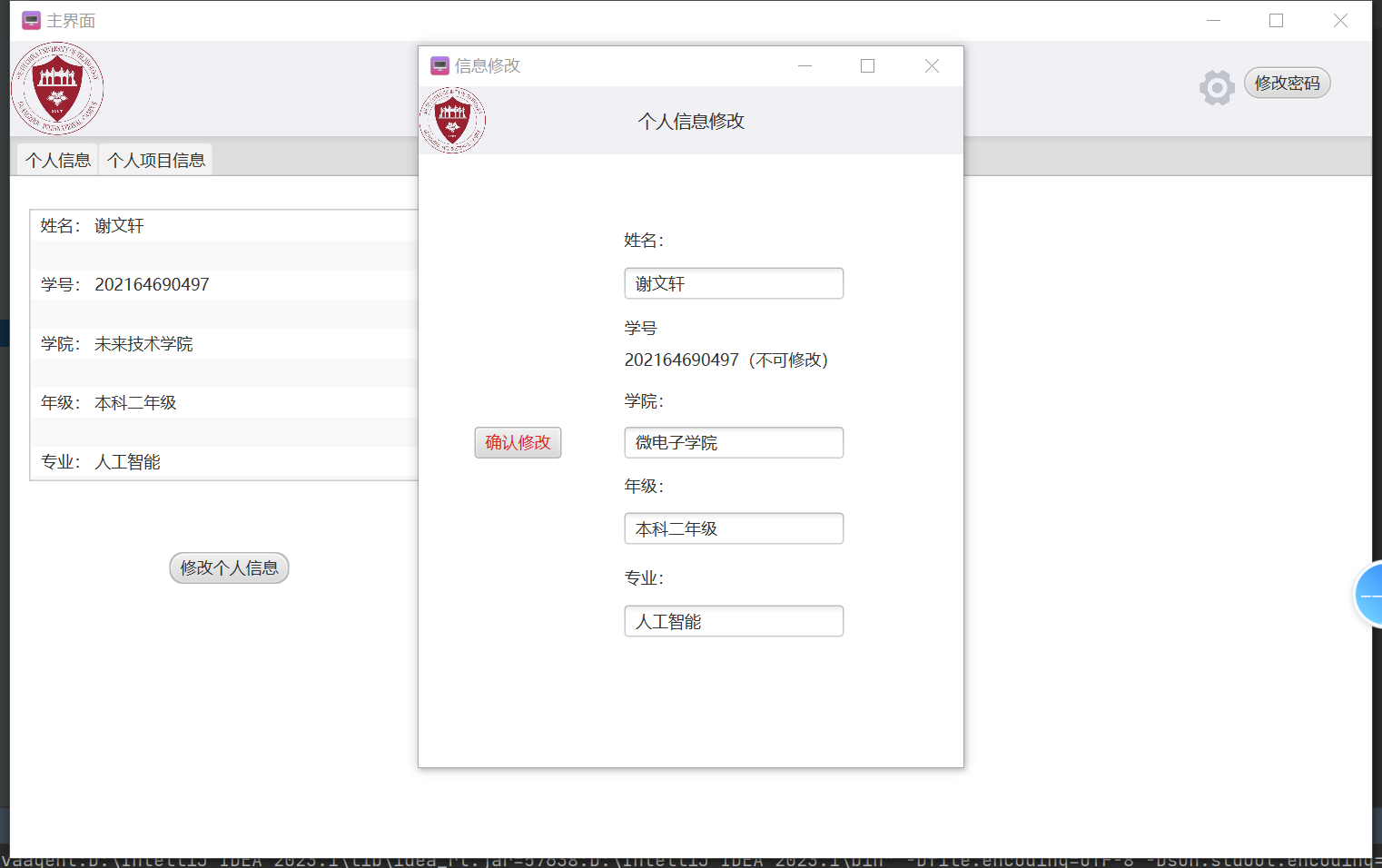


图28 个人信息修改界面

这里把未来技术学院改为微电子学院，修改后的界面如下：



图29 个人信息修改成功

（2）老师

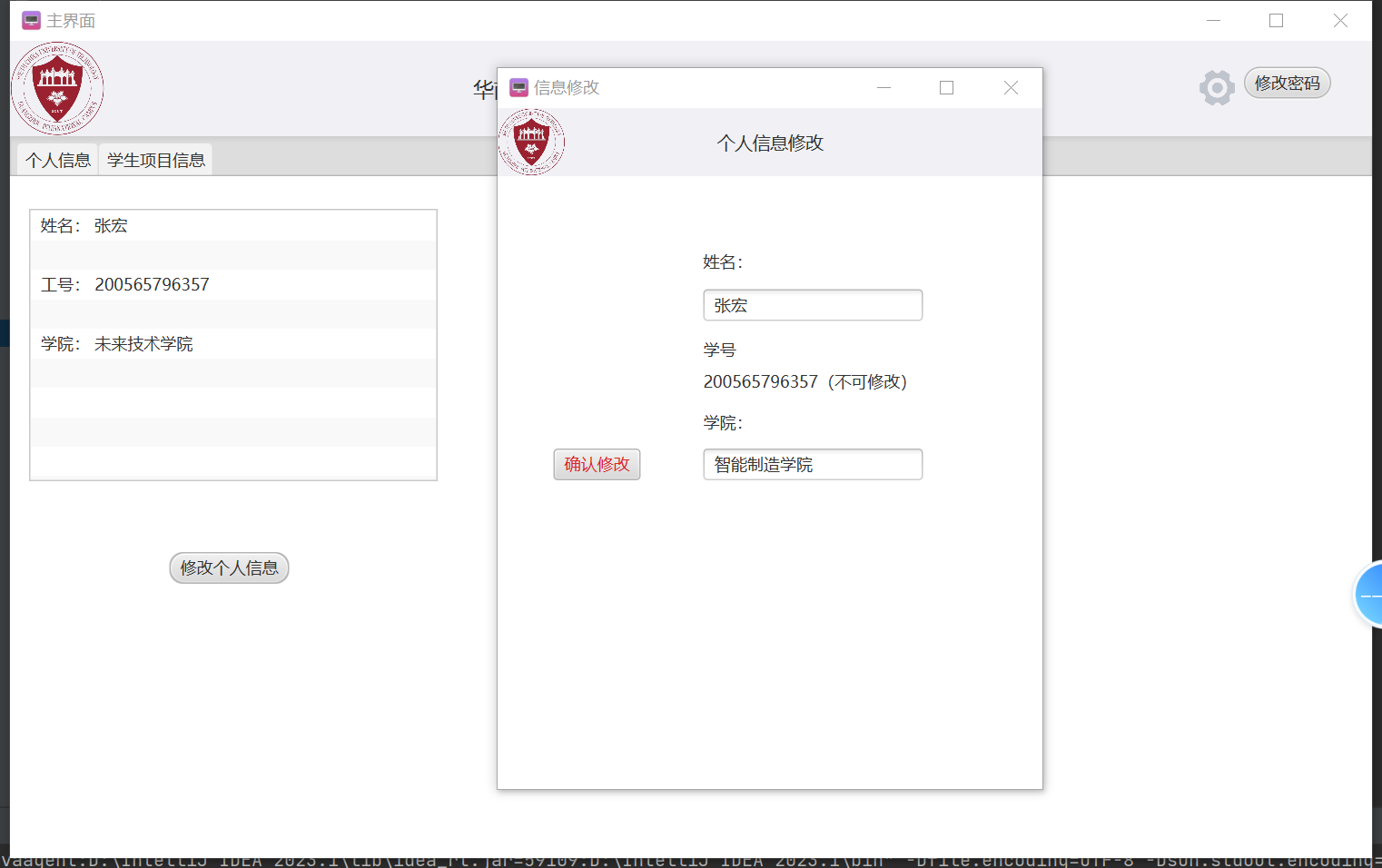


图30 个人信息修改界面



图31 个人信息修改成功

1. 项目信息修改：

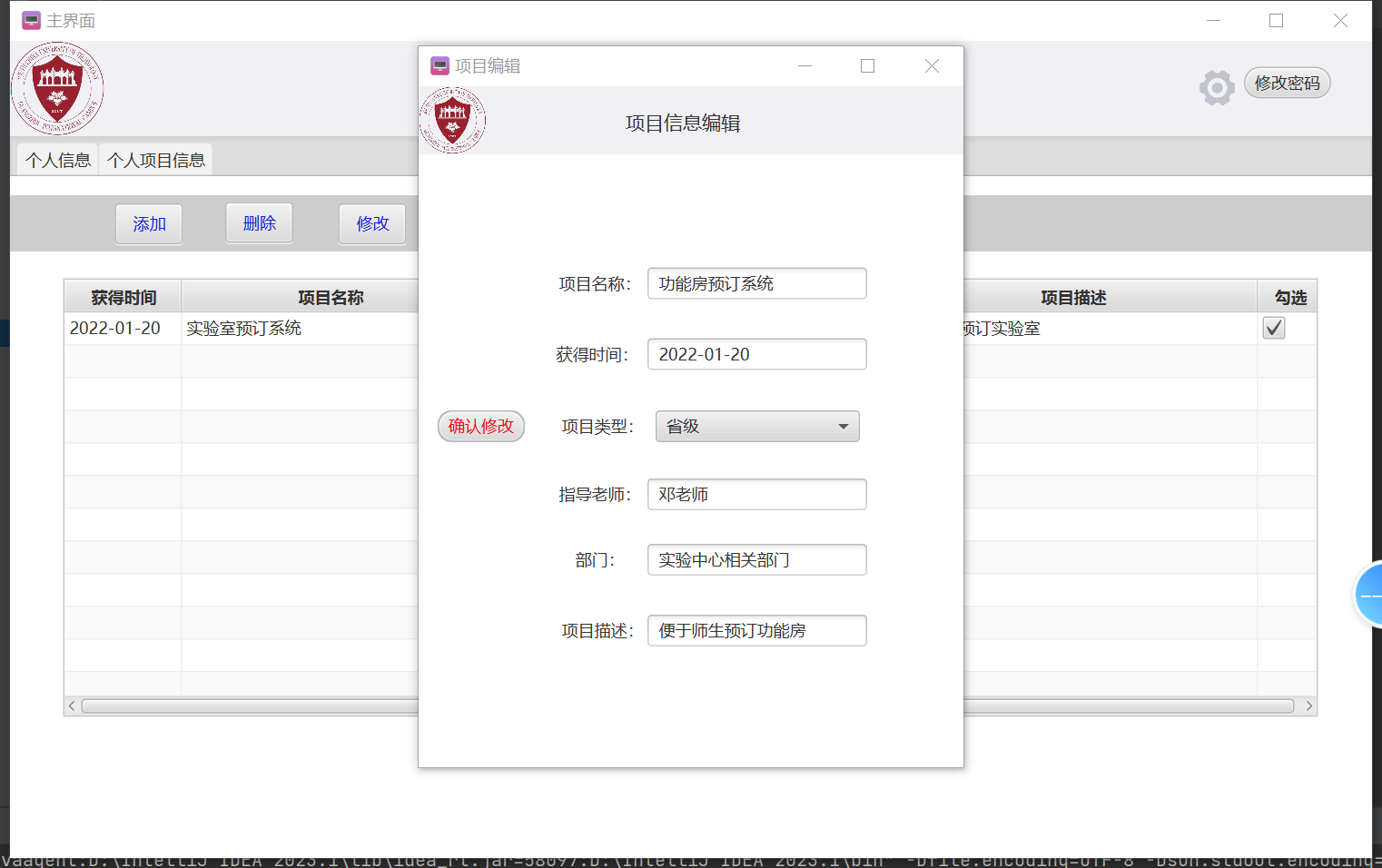


图32 项目信息修改

这里把实验室预订系统改为功能房预订系统，修改后：



图33 修改后的项目信息

1. 项目信息删除：

（1）学生

将上面的功能房预订系统删除。



图34 删除后

（2）老师

删除所选项，删除前：



图35 删除前



图36 删除后

1. 项目信息添加：

（1）学生

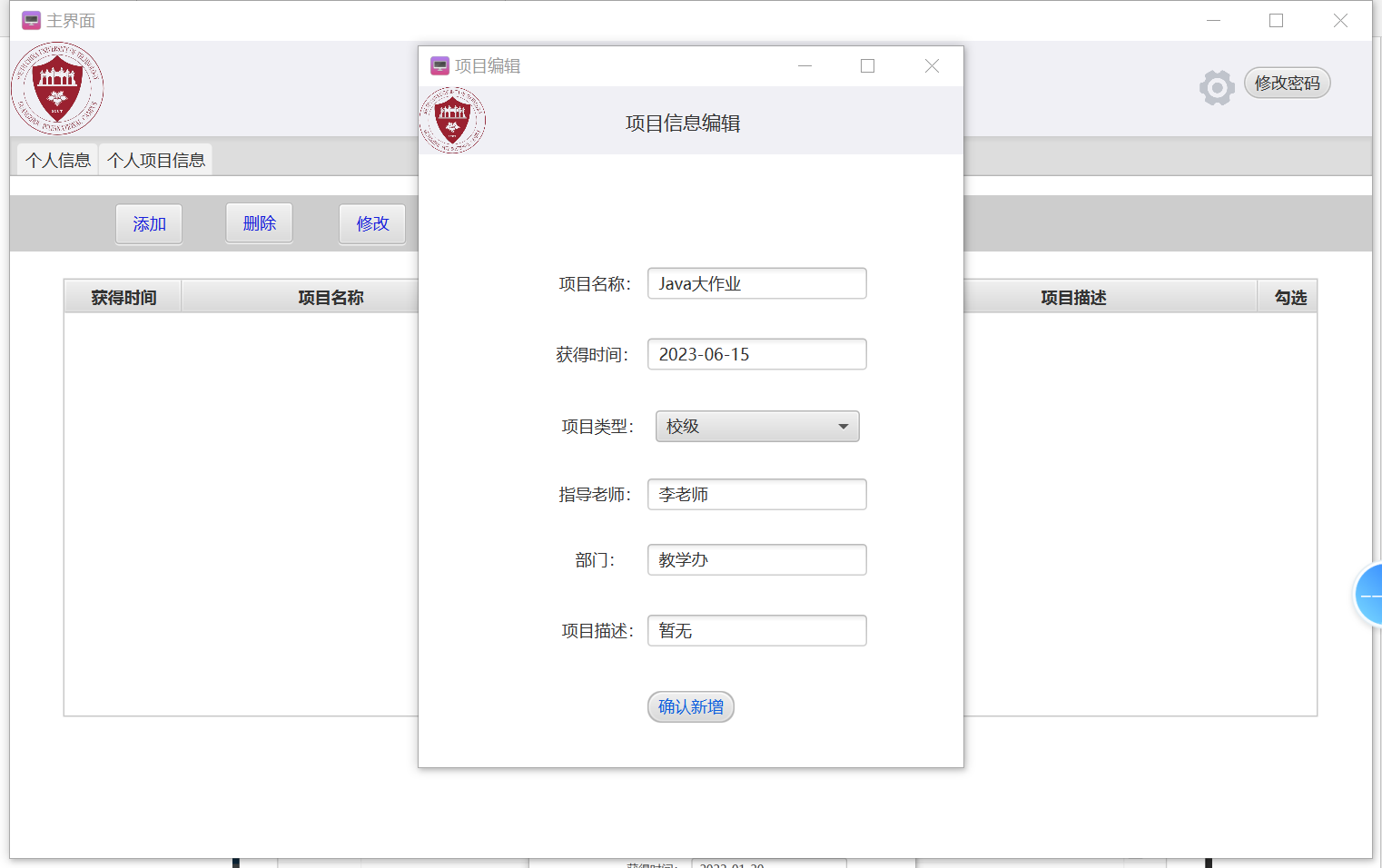


图37 项目添加前



图38 项目添加后

（2）老师

这里演示一下日期填错的警告：

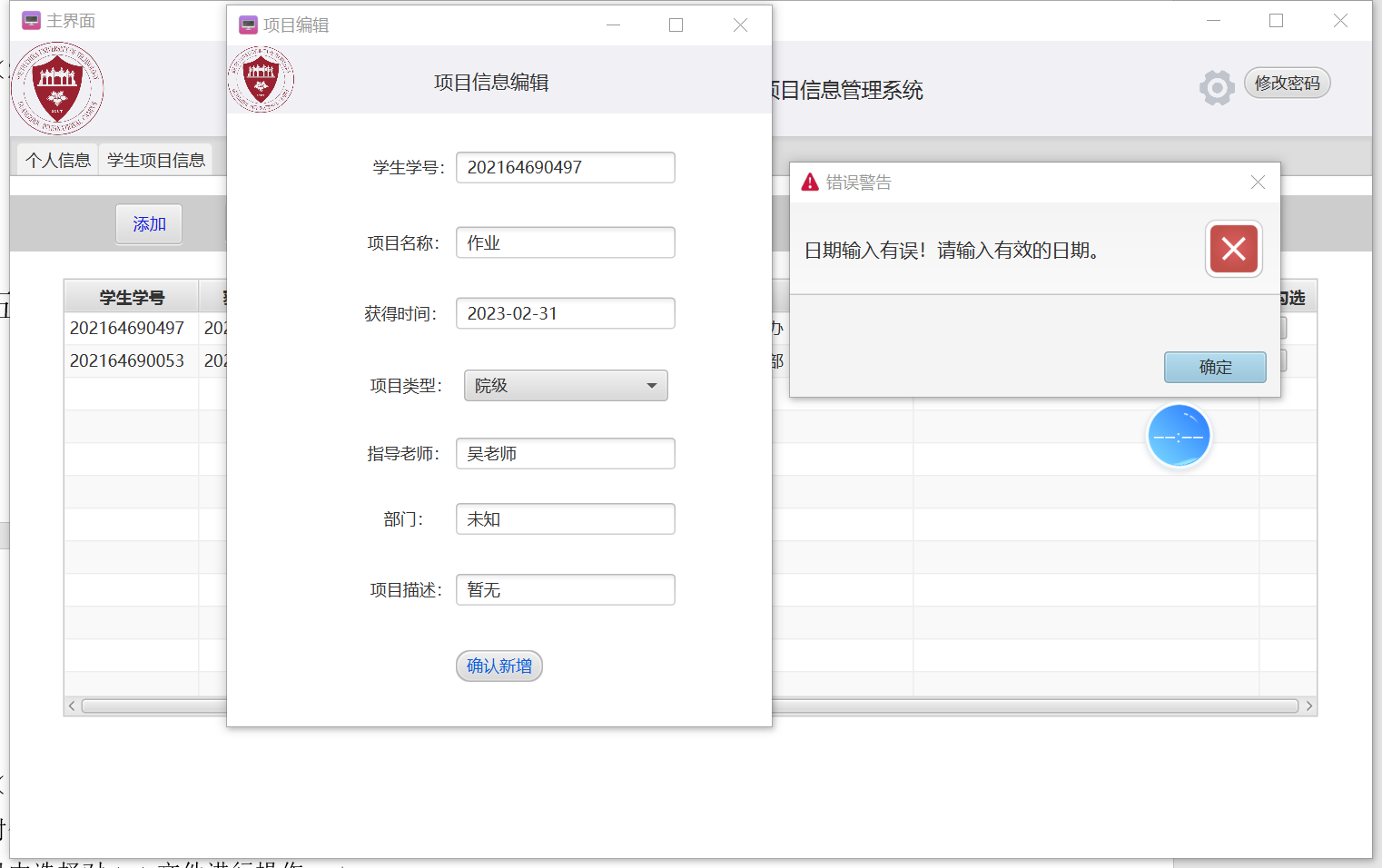


图39 项目添加前（日期有误）

将日期改为20号，添加后：



图40 项目添加后

1. 按照年纪统计项目信息：



图41 统计界面

1. 排序：



图42 按学号排序



图43 按时间排序



图44 按指导老师排序



图45 按项目类型排序

1. 小结
2. 不足之处：

（1）因为这个系统在实际使用时需要存储大量数据并需要实时检索，所以使用数据库文件对信息进行存储会更加规范且方便，不过由于我电脑之前没有安装mysql软件，因此我在项目中选择对txt文件进行操作。

（2）代码中有些算法比较冗杂，可以优化变得更加简洁；有小部分重复性的代码没有进一步封装成函数。

1. 优点：

（1）我在程序中灵活运用了类的继承结构和类的组合，还使用了抽象类，实现了多态，这让我的程序思路清晰、结构严谨。

（2）因为老师和学生两种身份所需要的大部分功能是一致的，因此我并没有针对某种身份单独设计UI界面，而是复用同一个界面，在实际运行时检测用户身份来调整控件的可见性。这样做避免了一些重复劳动，减少了不必要的工作量，同时使维护成本降低。

（3）我考虑到了许多和用户交互友好的小设计，比如登录时的小眼睛（密码可视）、每次操作成功或者失败都有对应的成功提示和弹窗警告、退出系统时需要确认（防止用户误触）等等。

（4）封装性良好，我程序中的绝大多数变量都是protect权限和默认权限的，因此别的包无法访问我内部的各种成员变量，更加安全。

1. 收获：

（1）我熟悉了java项目的开发流程，独立完成了根据要求设计系统的任务，从功能模块的设计和划分，再到根据流程逐个完善子模块，我的实战经验大大提升。

（2）在这次项目中，我初步掌握了javafx的简单开发，掌握了Scene Builder工具的使用，能够设计出自己想要的功能和界面。

（3）我熟悉了Maven框架的结构和相关知识，这在以后的项目开发中会给我带来便利。

1. 参考目录

[https://fxdocs.github.io/docs/html5/](#_top) （JavaFX Documentation Project）