**西南科技大学**

计算机科学与技术学院

课程设计报告

课程名称：面向对象程序设计方法（Java）

专业班级：卓软1901班

学生姓名：黄婧宇（学生1） 周丽（学生2）

学 号：5120196267（学生1） 5120195366（学生2）

2020 年 12月

1. **设计题目**

基于C/S模式的网络考试系统设计

1. **设计要求**

以Java语言为基础，选择合适的集成开发环境，按照C/S（客户/服务器）模式设计一个适用于局域网内部使用的无纸化在线考试系统，具体需求参见“面向对象程序实践（Java）”课程实验６中的需求“用户需求描述”。

1. **用户需求分析**
   1. 功能点分析
2. 教师端的登录：教师可通过个人账号密码登录考试教师端。
3. 登录验证：对教师及各考生的登录请求进行验证和人物匹配。
4. 教师的题库查询修改操作：教师进行登录验证后可打开，查询，修改考试题目数据库里的数据内容。
5. 教师的考试管理功能：教师通过考试教师端加载读取考生信息并分发试卷和控制整个考试流程。
6. 考试端的登录：各考生通过个人账号和密码进行考试端的登录验证。
7. 重登录功能：个人信息验证失败或考试意外中断的重新登录验证功能。
8. 试卷派发功能：当考生点击“开始考试”按钮，服务端自动发放该场考试的试题卷。
9. 考试实时监测功能：在考生考试的过程中向服务端实时传输考试情况，一面考试意外中断后的考试数据丢失。
10. 倒计时功能：当考试一开始，考试端界面将实时显示考试倒计时。
11. 进度条功能：当考生在等待试卷加载时，进度条显示当前试卷加载进度。
12. 自动交卷功能：当考试截止时间一到，若考生还未交卷，考试端将自动交卷。
13. 分数统计功能：当考生结束该场考试，服务端开始统计考试分数结果并保存在教师端数据库。
    1. 考试网络环境分析

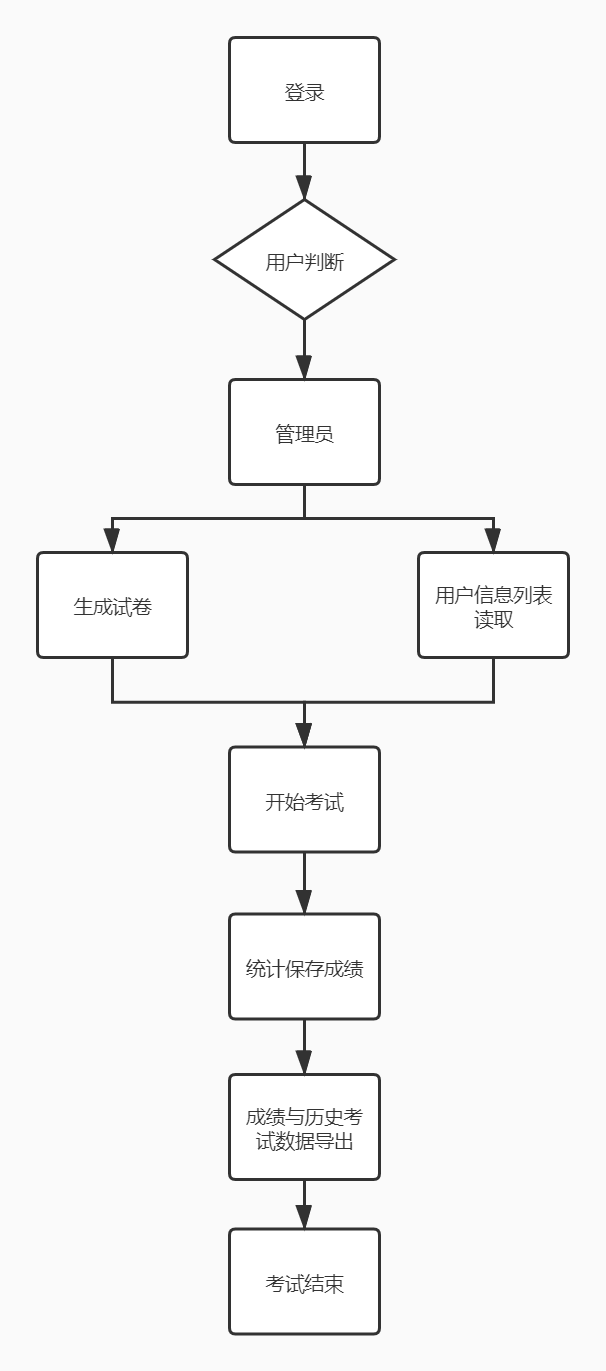
通常的电子化考试采用机房局域网来统一实现管理。

局域网是通过将一组PC连接到指定为服务器上的机器来实现的，是指一组计算机和他的其他设备在物理地址上彼此相隔不远，已允许用户共享比如打印机和内存设备之类的计算机资源的方式互联到一起的系统。

本实验中我们是利用Socket来实现网络进程之间的通信。在本地可以通过PID来唯一标识一个进程，但是在网络中就不可行了。然而TCP/IP协议族已帮我们解决了该问题，网络层“IP地址”可以唯一标识网络中的主机，而传输层的“协议+端口”可以唯一标识主机中的应用程序（进程）。这样利于三组件（IP地址、协议、端口）就可以标识网络的进程了，网络中的进程通信也就可以利用这个标志与其它进程进行交互。

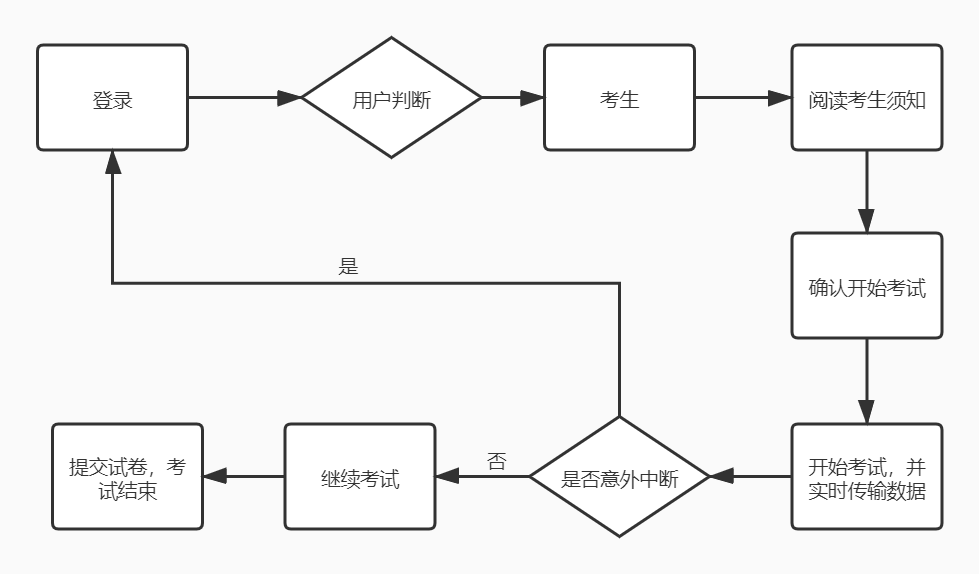
* 1. 服务端的正常操作流程分析

1. 服务端教师登录。
2. 打开链接生成试卷，并设置考试时间。
3. 读取考生列表，并将所有考生初始化为“未登录”。
4. 考试结束，统计并保存考试成绩。
5. 导出成绩及历史做题记录。



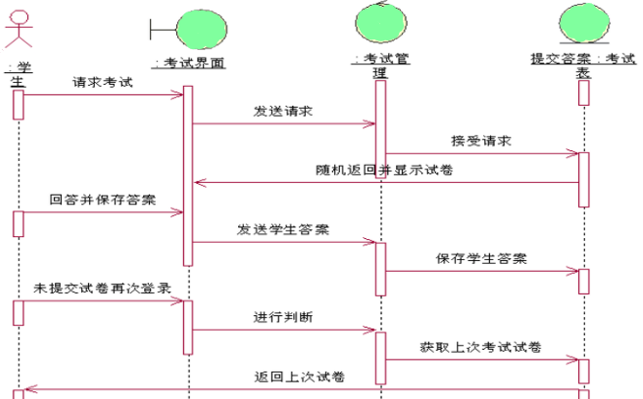
* 1. 客户端的正常操作流程分析

1. 考试端考生登录。(向服务端发送请求)
2. 阅读考试须知
3. 确认进入考试。(向服务端发送请求)
4. 开始考试。
5. 提交试卷。



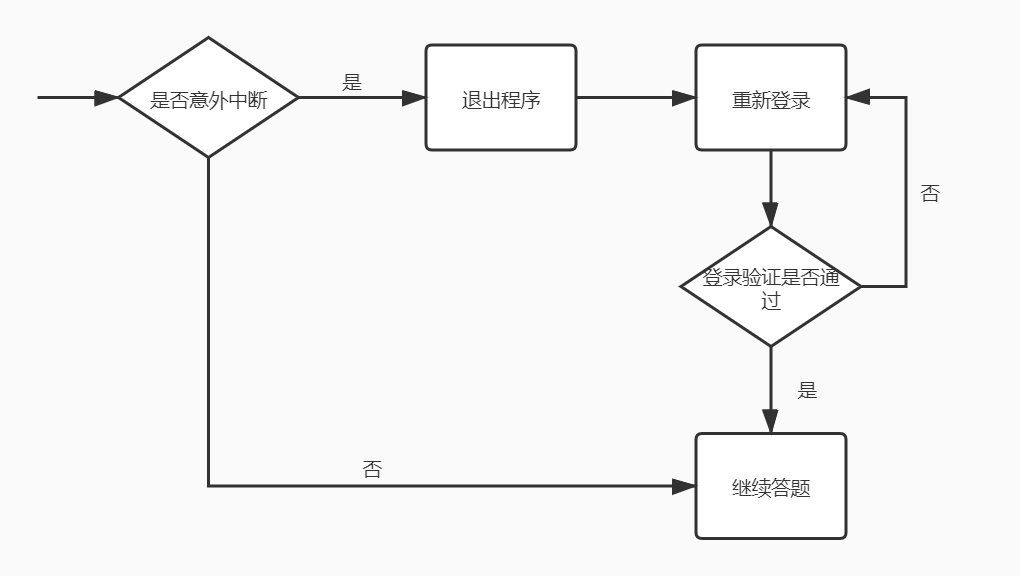
* 1. 服务器端/客户端的各种交互流程分析

1. 当考生向服务端发送考生请求时，服务端发放考试题目数据，并绘制考试界面。
2. 当考生开始答题时，向服务端发送答题数据，服务端数据库收集考试数据。
3. 当考生向服务端发送提交试卷申请，服务端终止考试，并记录考生的考试历史数据。
4. 当考试意外中断考生重新发送登录申请时，服务端根据数据库的历史答题数据重新考试绘制考试界面，考生继续答题。



* 1. 异常情况下处理流程分析

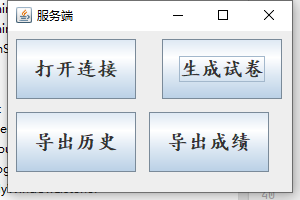
1. 发生意外中断。
2. 退出考试端程序。
3. 重新登录。(向服务端发送请求)
4. 登录成功继续答题。



1. **系统设计**

## 4.1服务器端设计

* + 1. 服务器端主界面设计

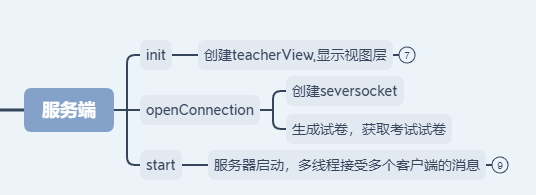




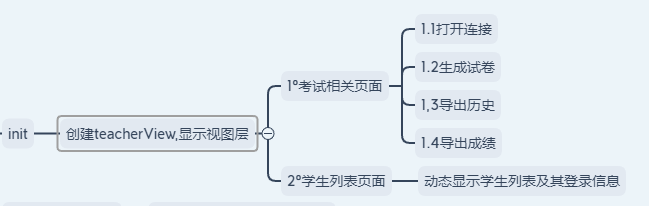
组件使用：JFrame（面板）、JPanel（容器）、JLabel（标签）、JButton（按钮）、TextField（输入框）。

页面设计函数：init——初始化页面

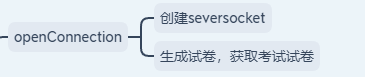
* + 1. 服务器端功能模块设计



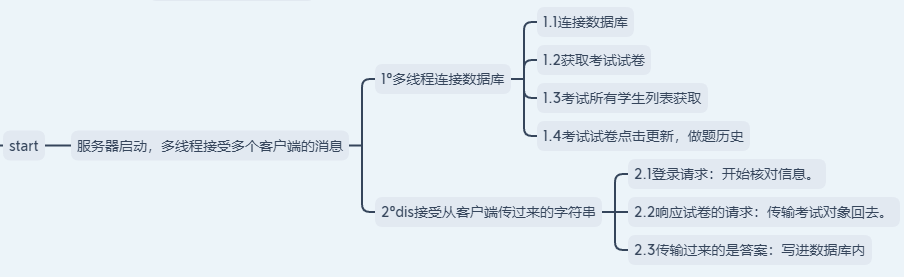
主要功能：①init:创建SeverView Sever的视图层



②openConnection:打开客户端与服务端的连接并生成试卷，在生成试卷时向客户端发送考试时间相关信息，并提供导出功能。导出成绩和历史按钮增加监听事件，1°导出历史：创建文件路径，从服务层得到历史试卷对象，导出其做题记录2°导出成绩：通过刚才获得的历史试卷对象，将考生所选择的选项与正确选项对比算分。

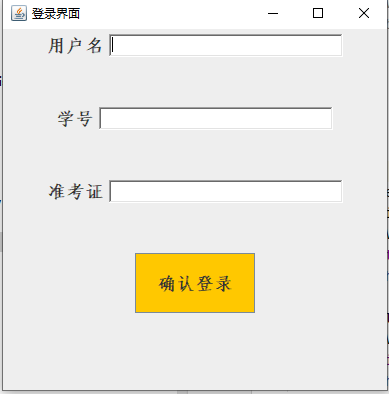


③start:服务器启动，多线程接受多个客户端的多个消息。



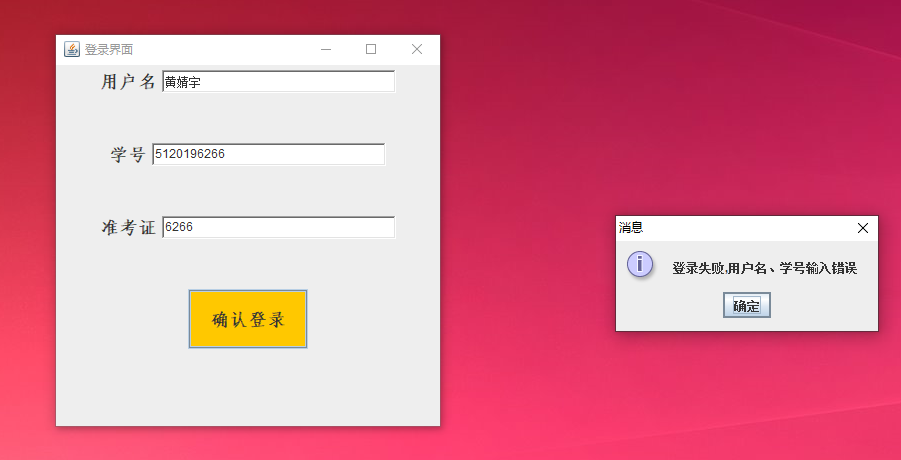
## 4.2客户端设计

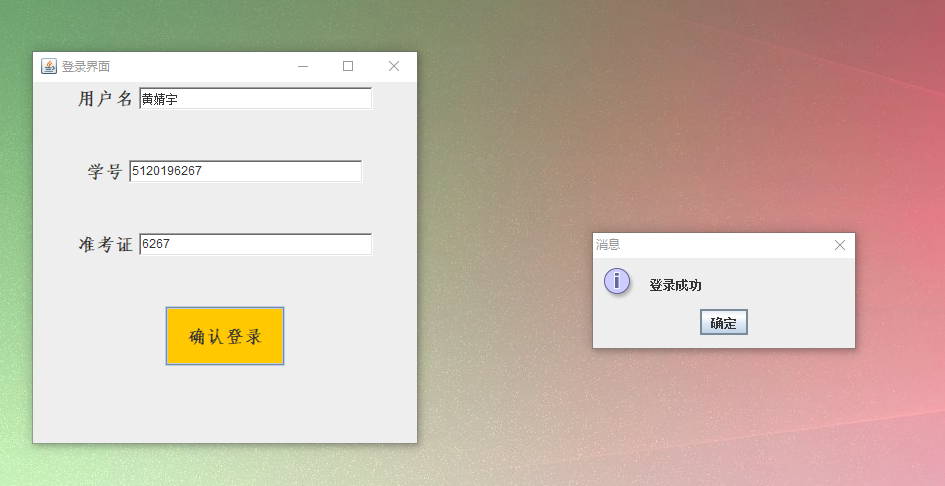
* + 1. 客户端主界面设计
  1. 登录界面



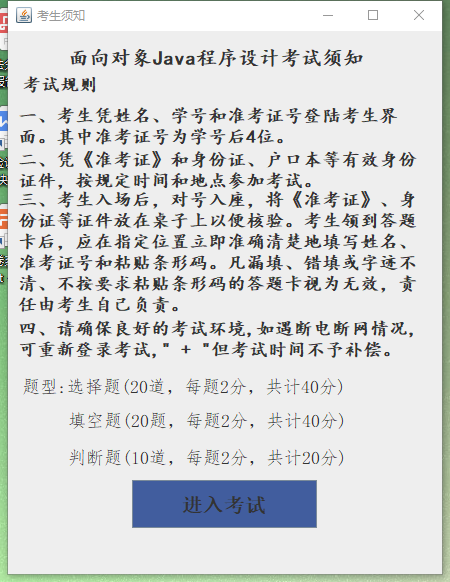
* + 1. 组件使用: JFrame（面板）、JPanel（容器）、JLabel（标签）、JButton（按钮）、TextField（输入框）。
    2. 主要函数：

1. init函数
2. addListener:增加监听事件，点击“退出登录”按钮时，退出
   1. 登录成功与失败界面:





* 1. 考生须知界面
     1. 组件使用: JFrame（面板）、JPanel（容器）、JButton（按钮）。
     2. 主要函数：
  2. init函数
  3. addSubgroup:增加考试须知等各项内容

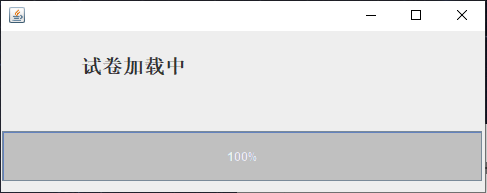


* 1. 进度条界面

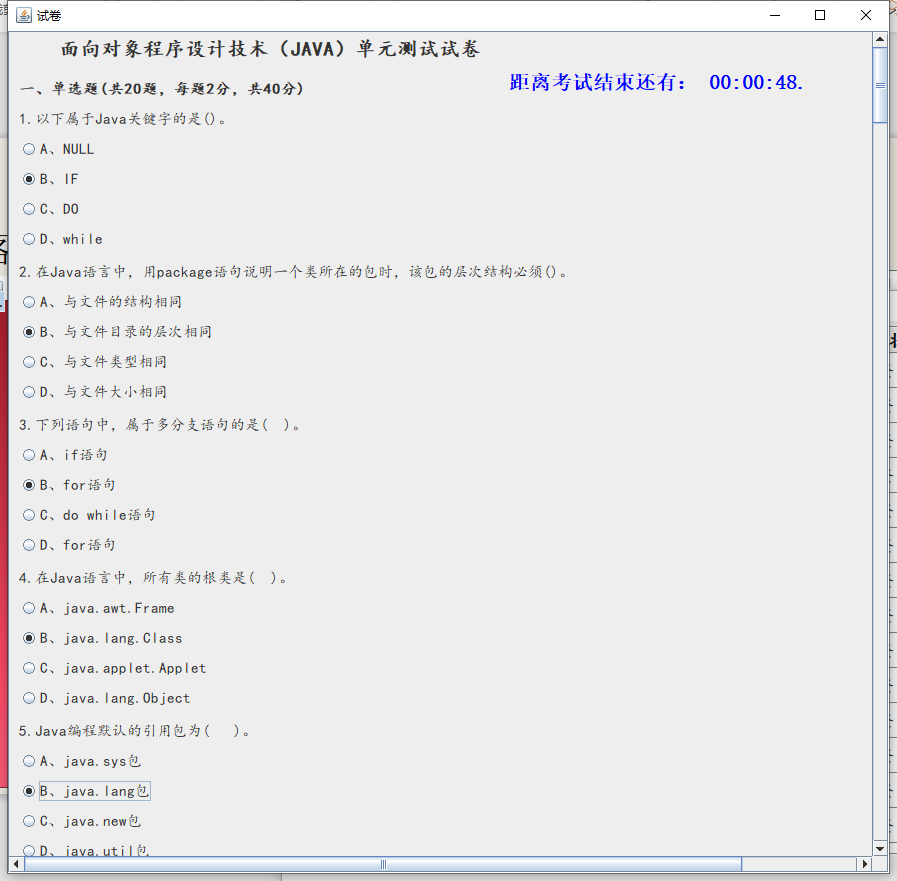
1. 组件设计:JFrame（面板）、JPanel（容器）、JLabel（标题）、JProgressBar（进度条）
2. 主要函数

Init：初始化进度条相关

Run:根据试卷的渲染题目数量等，显示进度条。



* 1. 考试界面
     1. 组件使用: JFrame（面板）、JPanel（容器）、JButton（按钮）、JOptionPane（提示框）、JScrollPane（滚动条）。
     2. 主要函数：
  2. init函数：初始化试卷界面
  3. render函数：渲染试卷界面
  4. countdown函数:倒计时函数



* + 1. 客户端功能模块设计

## 4.3服务器/客户端交互功能设计

* + 1. 交互方式总体设计

通过两端的Socket连接，发送不同编码的消息，实现两端的交互。

消息具体格式设计如下：

消息编码：

例1："当前请求的用户准考证为 " + stuAdm + " 学号为 " + stuNum + "选择的选项id为 " + stuSelceted 🡪 客户端向服务端发送登录请求。

例2："考试开始已响应" 🡪 客户端通知服务端开始考试。

例3："试卷传输完毕" 🡪 客户端通知服务端试卷加载完毕。

* + 1. 服务器端交互消息详细设计

|  |  |  |  |
| --- | --- | --- | --- |
| 消息编码 | 参数表 | 消息含义 | 其它说明 |
| 001 | 登录 | 通知客户端登录成功 |  |
| 002 | 开始 | 通知客户端考生开始 |  |
| 003 | 试卷 | 向客户端发送试卷信息 |  |
| 004 | 结束 | 通知客户端考生已结束，并发送祝语 |  |

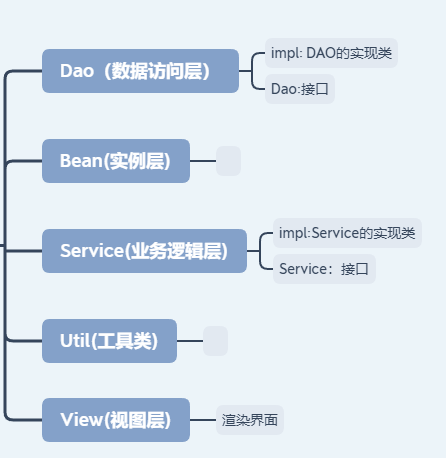
* + 1. 客户端交互消息详细设计

|  |  |  |  |
| --- | --- | --- | --- |
| 消息编码 | 参数表 | 消息含义 | 其它说明 |
| 001 | 姓名，准考证号，身份证号 | 请求服务器验证考生信息 |  |
| 002 | 考试请求 | 请求服务端发放试卷 | 表示客户端已准备好开始考试 |
| 003 | 试卷请求完毕 | 通知服务端试卷已加载完毕 | 表示服务端可开始向数据库传输实时答题数据 |
| 004 | 判题请求 | 向服务器传输答题数据 |  |

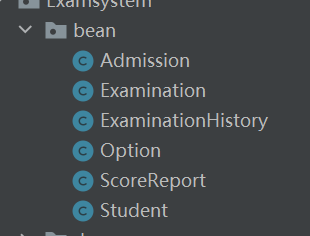
## 4.4异常情况下的处理功能设计

当登录，准备页面，答题页面三者任意一个窗口关闭，都可以模拟断网的情况，由于readUFT的关系，此时客户端与服务端自己的联系已经断开。因此当窗口关闭时，客户端与服务端直接的连接断开时，向客户端弹出提示框，为用户提供对应的提示信息。

1. **详细设计**



## 1. Bean层：实例层，在该层中实现将相关联的数据组装在一起。



（1）.Admission类：

属性：考号：admNum、学号：stuNum

（2）.Examination类:

属性：试卷名称：examName、试卷序号：examQuestionNum、问题名字：examQuestionName、问题描述examQuestionDescribe 、题号questionNum、问题描述 questionDescribe、正确答案questionRightAnswer 、当前问题分数 questionScore、选项序号 optionNum、选项描述 optionContent

（3）. ExaminationHistory类：

属性:历史id hisId、学号hisStuNum、准考证号hisAdmNum、历史试卷id hisExaId、历史题目id hisQtId、历史选项id：hisOId、历史学生选项 hisStuSelected、创建时间 createTime、最后更新时间lastModifyTime

（4）.Option类：

属性：选项id optionId、选项内容optionContent、选项正确性correct

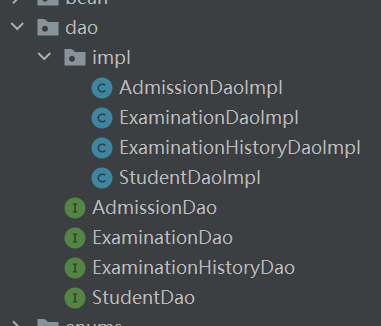
（5）.ScoreReport类：

属性：学生姓名：stu\_name、准考证号：adm\_num、学号：stu\_num、学生性别:stu\_gender、多选题分数：sum\_multiple\_score、单选+判断分数：sum\_single\_score、总分：sum

（6）.Student类：

属性：学生姓名：stuName、学生学号：stuNum、学生性别：stuGender

## 2. Dao层：从数据库取数据。



(1).impl：Dao的实现类

<1>.AdmissionDaoImpl（身份证数据持久层实现类）：

findAll通过身份证号查询数据库，并为admission对象设置对应的属性。

<2>.ExaminationDaoImpl（考试数据持久层实现类）：

从数据库中找到所有问题的相关对象及属性，并设置给examination对象。

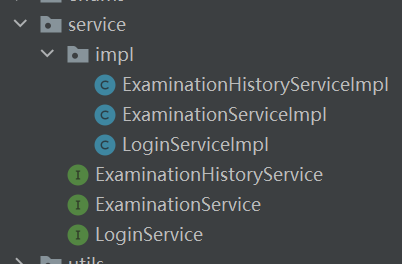
<3>.ExaminationHistoryDao（Impl）（考试历史数据持久层实现类）：

* + - 1. findAll：查询数据库，并去掉其中有“&&”特殊符号的题目数据(对应多选题取消选择的选项)。
      2. update：通过传入的optionId,查询选项，得到其questionId和content，并更新一整条数据，共两个字段
      3. updateWithStr：通过传入的optionId,检索到它的questionId,content,更新一整条数据，只需要更新所选的选项的那部分，查询选项，得到题目ID，重新更新这一条数据，换成&&
      4. insert：通过传入的optionId,检索到它的questionId,通过questionId取出正确答案和题目分值，插入一整条数据，共5个字段。
      5. isQuestionExistByHisOId：通过optionId得到questionId 判断这道题，在不在历史记录中。(当前这个人是不是在这里)， 如果在历史记录中，我们就更新这道题的选项，如果不在，我们就新增
      6. calculateScore：查询选项得到题目id，并计算分数。
      7. historyMapping：整合历史考试数据，并返回一个数据对象。

<4>.StudentDaoImpl（学生数据持久层实现类）：

findAll:从数据库中找到所有学生，并且返回学生列表。

## 3. Service（业务逻辑层）



<1>.ExaminationHistoryServiceImpl(历史试卷对象层)：

* 1. .getAllHistory：返回历史对象的list
  2. .onQuestionOptionChangedDan：点击事件,更新历史试卷
  3. .onQuestionOptionChangedDuoChecked: 点击事件,更新历史试卷,不能自动更新的，只需要新增
  4. .onQuestionOptionChangedDuoNoChceked: 点击事件,更新历史试卷,并更新当前选项选中的答案为null
  5. .adjust（判分）:导出历史：创建文件路径，从服务层得到历史试卷对象(含姓名、准考证号、性别、总分等属性)，导出其做题记录

<2>. ExaminationServiceImpl(试卷对象服务层)：

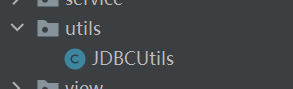
getExamination:得到数据库全部考试试卷对象

<3>.LoginServiceImpl(登录服务层)：

Login：①. admissio为空：返回“登录错误”。

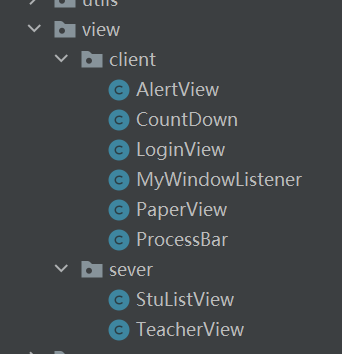
②.admission不为空:当admission的学号等于所查询的学号时，返回当前的状态为SUCCESS

## 4. Utils（工具层）



1. beginConnect：通过网址和账户密码与远程数据库建立连接。
2. killConnect：关闭服务端与数据库的连接。

## 5. View（视图层）



<1>.AlertView（考试须知页面）:

* 1. Init：初始化页面相关容器。
  2. addSubgroup:绘制考试须知相关的规则。

<2>.CountDown（倒计时页面）:

根据考试时长推算出结束时间，再根据实时时间反推剩余考试时间。

1. Execute：推算获取相应的结束时间、当前时间、剩余时间，并返回剩余时间的一个字符串。
2. changeFormat：将剩余时间的字符串转换成需正确显示的格式。

<3>.LoginView（登录界面）：

* 1. Init:初始化页面相关容器。
  2. addSubgroup:增加登录相关属性

<4>.MyWindowListener(断网检测)

①通过readUTF读取，当两者之间联系断掉的时候，readUTF无法读取到内容，以此来模拟断网，向客服端发出提示。

<5>.PaperView（试卷界面）

1. init：初始化页面相关所需的容器。

1.1initJFrame:初始化JFrame

1.2initContPane:初始化容器

1.3initScroll:初始化滚动条

1. Render：渲染试卷

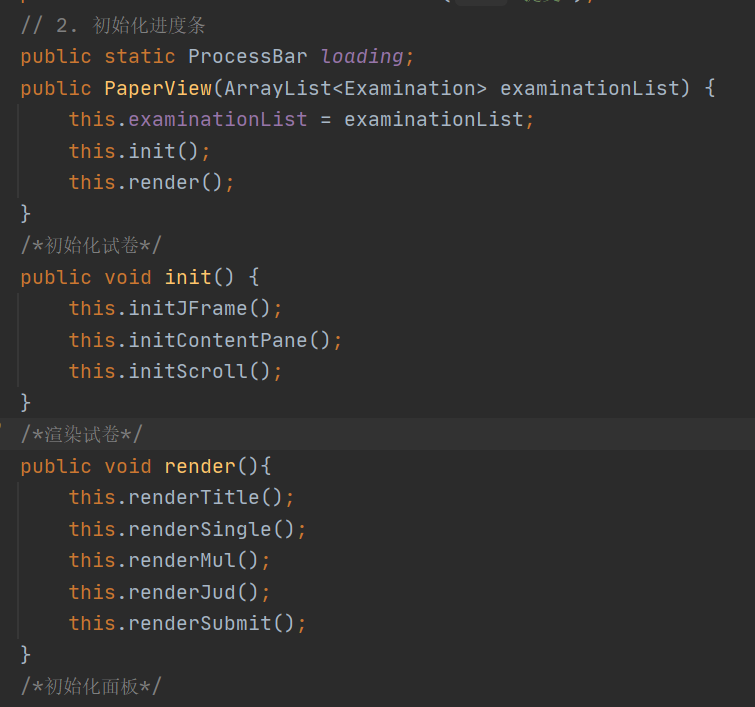
2.1 renderTitle:渲染试卷标题

2.2 renderSingle:渲染单选题

2.3 renderMul:渲染多选题

2.4 renderJud:渲染判断题

2.5 renderSubmit:渲染提交按钮

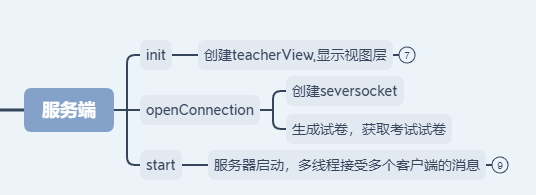


<6>.ProcessBar:进度条界面

①．Init：初始化进度条相关

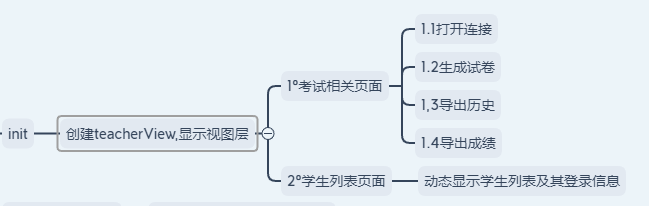
②．Run:根据试卷的渲染题目数量等，显示进度条。每当题目渲染了一个之后，则渲染的试题个数增加，进度条的进度更新。

## 6. Teacher

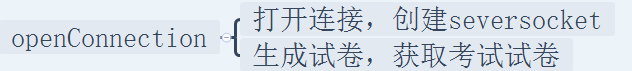


服务端有以下三个功能：

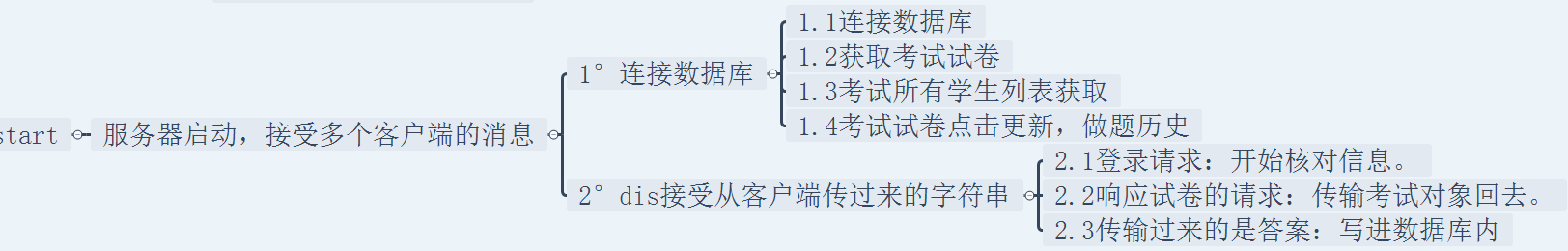
* 1. Init（创建teacherView视图层）：当前部分有两个页面，分别是考试相关页面和学生列表页面。考试相关页面与考试前后的相关准备有关，学生列表页面可以根据学生的登录情况实时更新学生的登录状态。



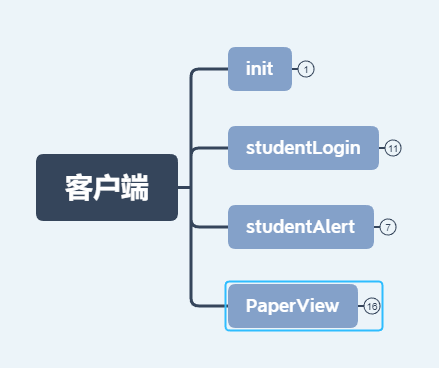
* 1. openConnection(打开链接):这个部分负责创建severSocket，并且在点击生成试卷按钮后从数据库获得考试试卷对象。



* 1. start（启动服务器，接受多个客户端的信息）：首先与数据库建立连接，再通过DataInputStream接收从客户端传过来的字符串，字符串传输3种请求。1.当传送过来的字符串中只有一个‘&’时，为登录请求，开始核对数据库中的信息，更新学生列表的信息。2.当传送过的是“考试开始已响应”，则传送试卷对象回去。3.传送的是答案过来，多个“&”符号，若‘&’结尾，即表示已经选择过，标记。3.1单选和判断题：直接传题号去新增或更新3.2多选题:是否标记，未标记直接传id和对应的选项，否则传id，将已经选择的改为‘&&’，更新数据库。

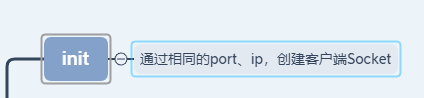


## 7. Student

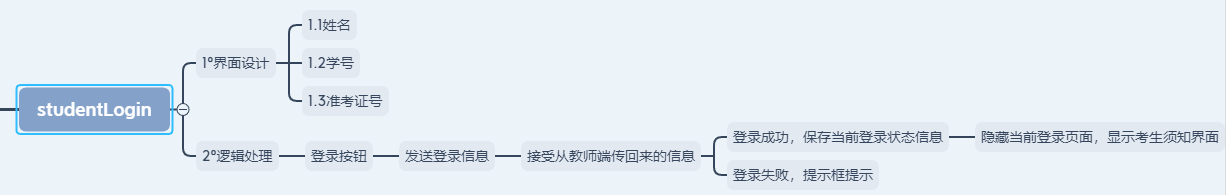


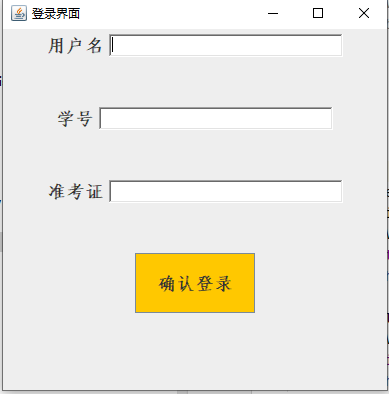
客户端分为以下4个部分：

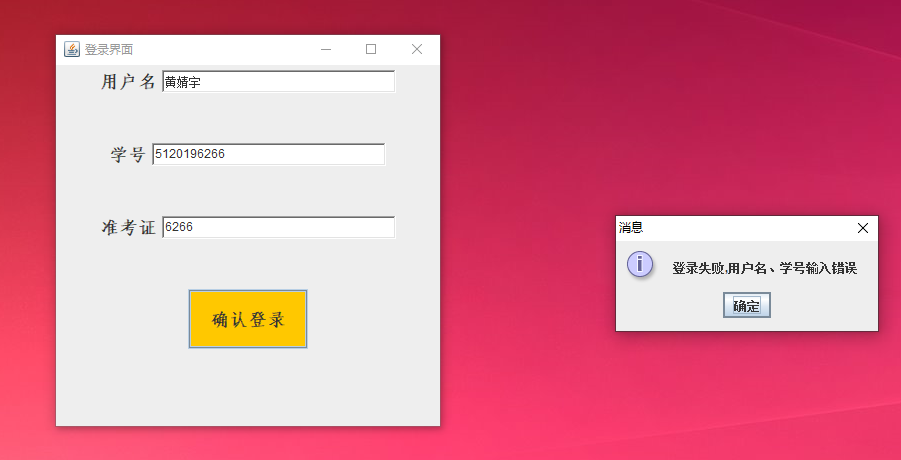
1. 初始化线程：通过相同的端口和ip，创建客户端的socket

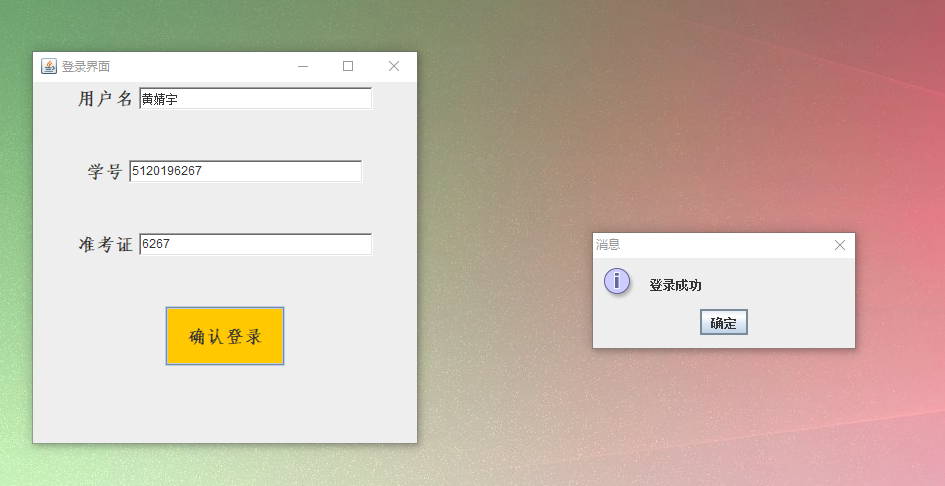


1. StudentLogin（学生登录功能）：关于学生登录功能，它的界面中需要输入姓名、学号和准考证号，点击登录按钮之后，将其拼接向服务端发送登录请求，接受从教师端传回来的信息，若登录成功，则保存当前登录状态信息并进入考试须知页面，若登录失败，则弹出提示框提示登录失败。

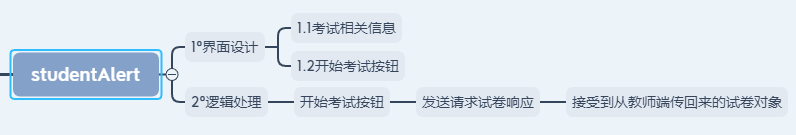


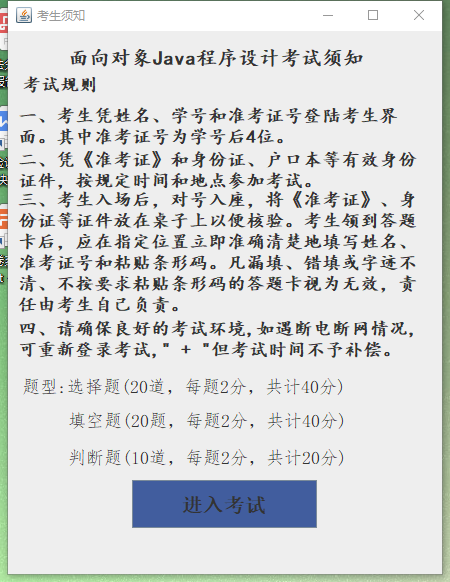




****

1. StudentAlert（考试须知）：关于考试须知功能，其界面上显示了与考试相关的规则及时间等，以及考试登录按钮。在点击了考试登录按钮之后，向服务端发送请求试卷响应，然后接收到从服务端传回了的试卷对象。





1. PaperView（试卷）：关于考试试卷功能，首先进入考试界面，显示进度条、试卷等内容，并打开断电检测，为每个页面增加监听事件，监听客户端与服务端之间的联系是否端口。之后为试卷上的每一个选项按钮增加监听事件，实时更新做题记录。交卷事件：有以下两种情况会交卷：1.点击提交按钮，弹出确认框，确认后提交。2.倒计时事件：在自己到了之后，自动执行submit.doClick函数交卷。

