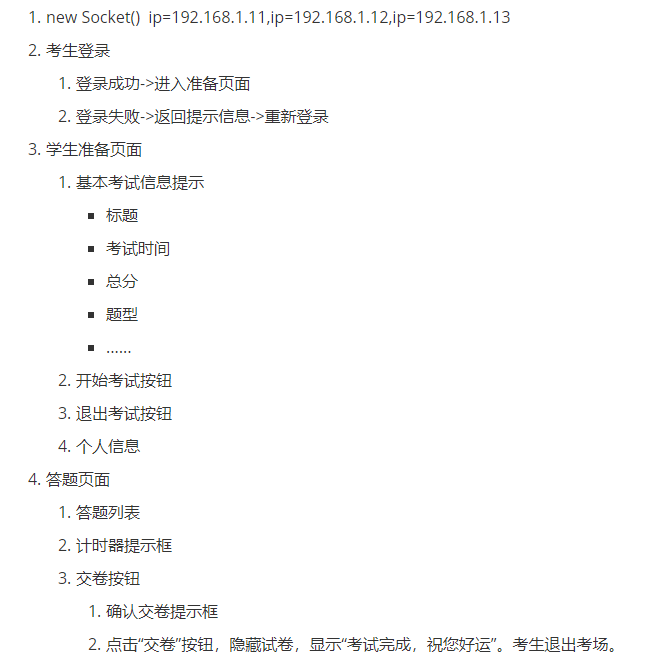
# 考试系统网络程序系统设计

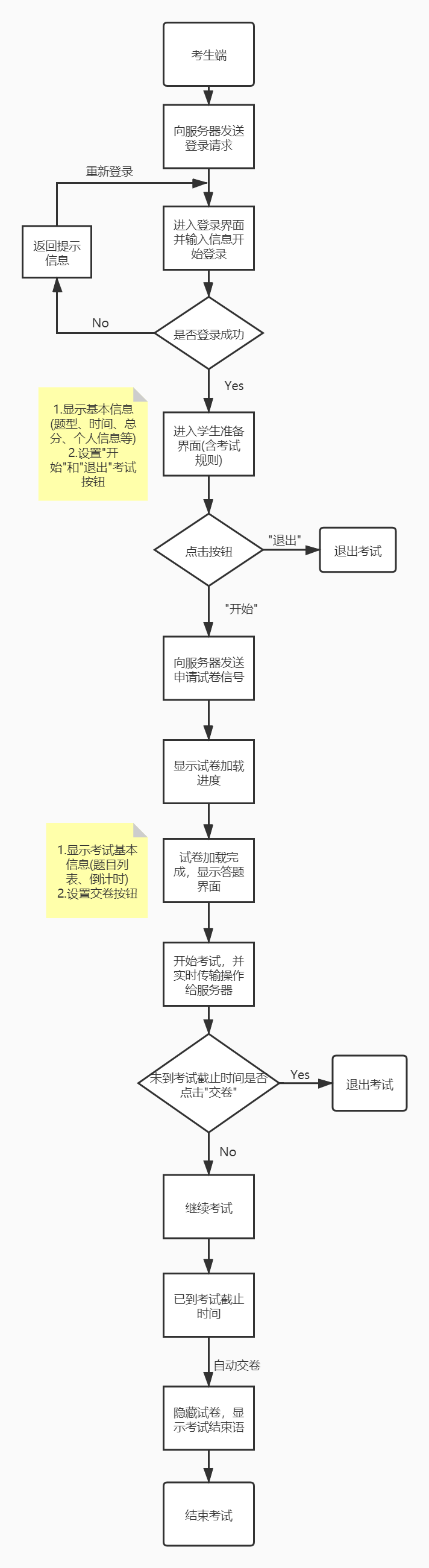
## 客户端

#### 学生端

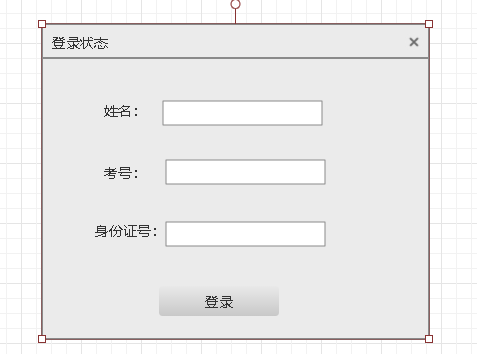
**图一 学生端设计**

****

**图2学生端流程图**



###### 学生登陆



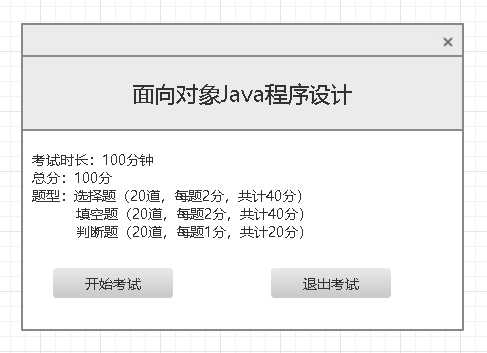
1. 登录成功

进入准备页面

1. 登录失败

返回提示信息+重新登陆

###### 学生准备页面



1.1.2.1基本考试信息提示

- 标题

- 考试时间

- 总分

- 题型

1.1.2.2开始考试按钮

得知考试的基本信息之后，我们这里点击开始考试。

点击之后，我们开始响应这个请求，服务器直接将从老师那里得到的试题传给对应的学生，并且在这里是单独开启的线程里进行传送，而我们当前的学生获得服务器的返回结果之后，显示进度条和页面加载。

1.1.2.3退出考试按钮

点击这个按钮之后，直接退出系统，说明考试结束，并把当前学生状态记录

###### 答题页面

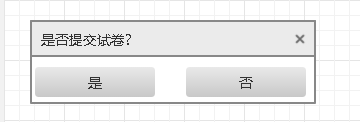


* + - 1. 答题列表

加载出来的试卷，每一次通过点击事件时实时的向服务器发送当前的做题进展，而服务器得到这个结果后显示在老师的页面里面，实时更新

* + - 1. 计时器提示框

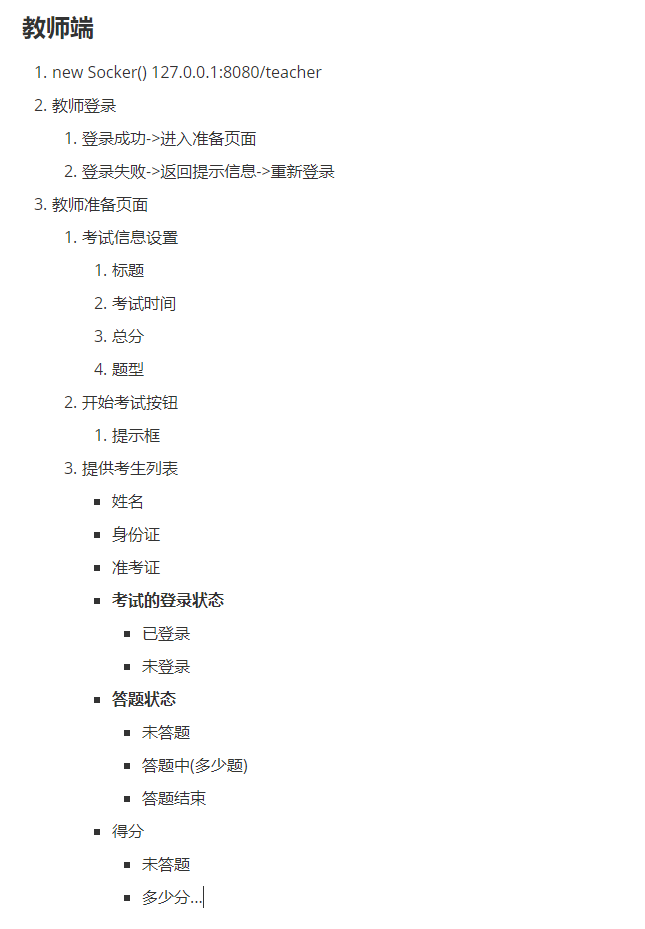
1. 提示框是单独的线程，
2. 能够实现交卷后时间停止
3. 超出规定时间之后自动交卷功能
   * + 1. 交卷按钮



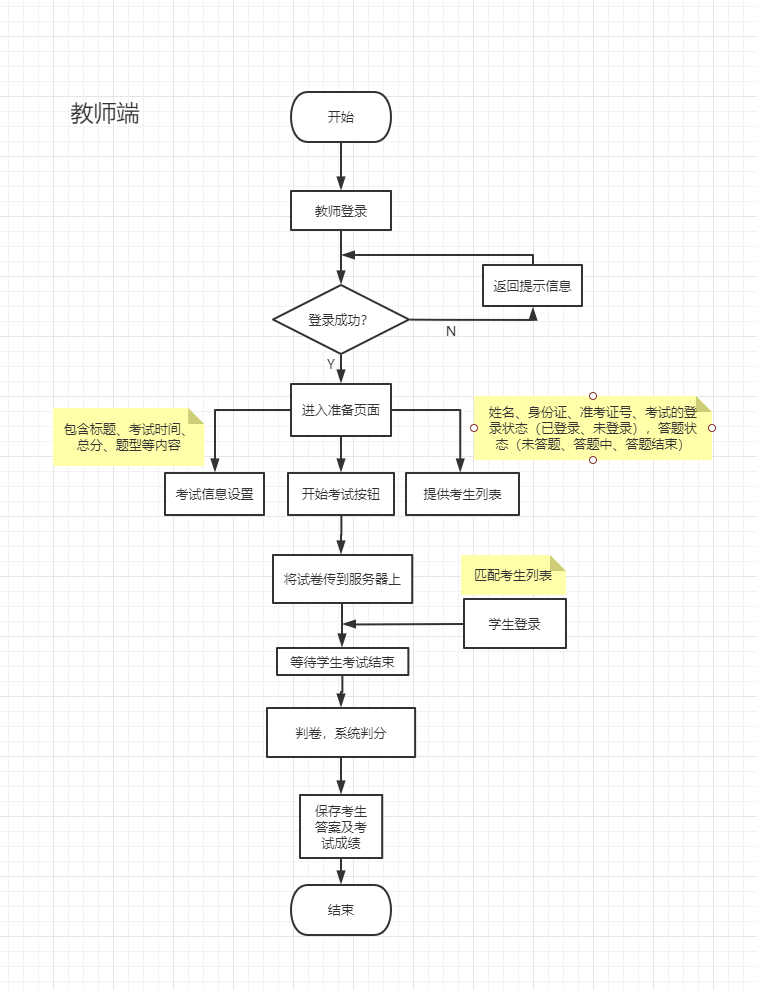
1. 选择判断，是不是确认要交卷，
2. 点击“确认”之后，传输当前的答题结果给服务器，服务器将这个结果存储在学生列表；的考试成绩属性里面，进行存储；隐藏试卷，显示“考试完成，祝您好运”。考生退出考场。
3. 点击“继续答题”，回到答题页面；
4. 点击“退出考试”，存储当前分数，传送给服务器，服务器接收后传给用Hash表存储的学生列表的对应的那个学生的最终成绩属性；

#### 教师端

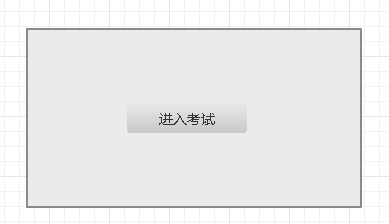
**图三教师端设计**



**图4 教师端流程图**

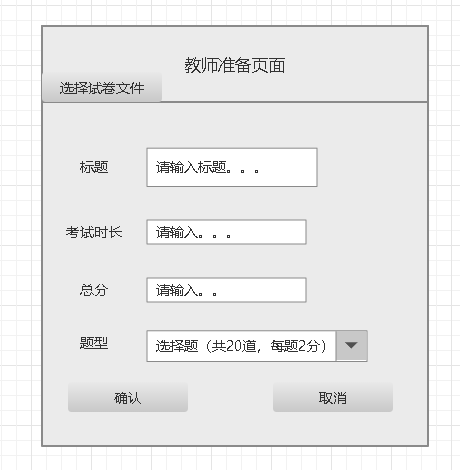


###### 教师登陆



点击进入考场按钮，进入准备界面

###### 教师准备页面



* + - 1. 试卷信息设置

1. 标题

2. 考试时间

3. 总分

4. 题型

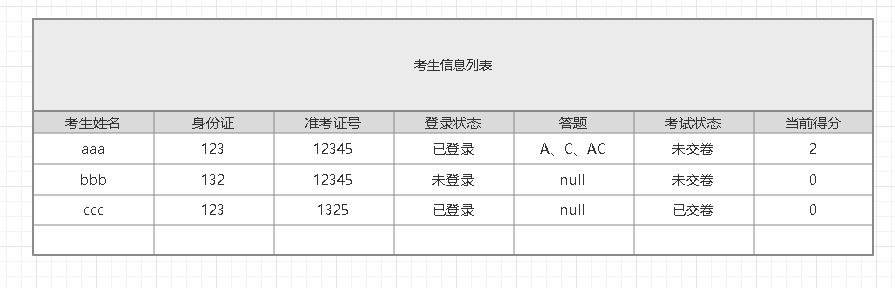
初始化学生准备界面的样式

* + - 1. 点击生成试卷

点击“确认”， 开始生成的试卷out线程出去，传给服务器，不显示出来；

点击“取消”， 退出当前试卷生成窗口

* + - 1. 提供考生列表



- 姓名

- 身份证

- 准考证

- \*\*考试的登录状态\*\*

- 已登录

- 未登录

- \*\*答题状态\*\*

- 答题答案

- \*\*考试状态\*\*

- 已交卷

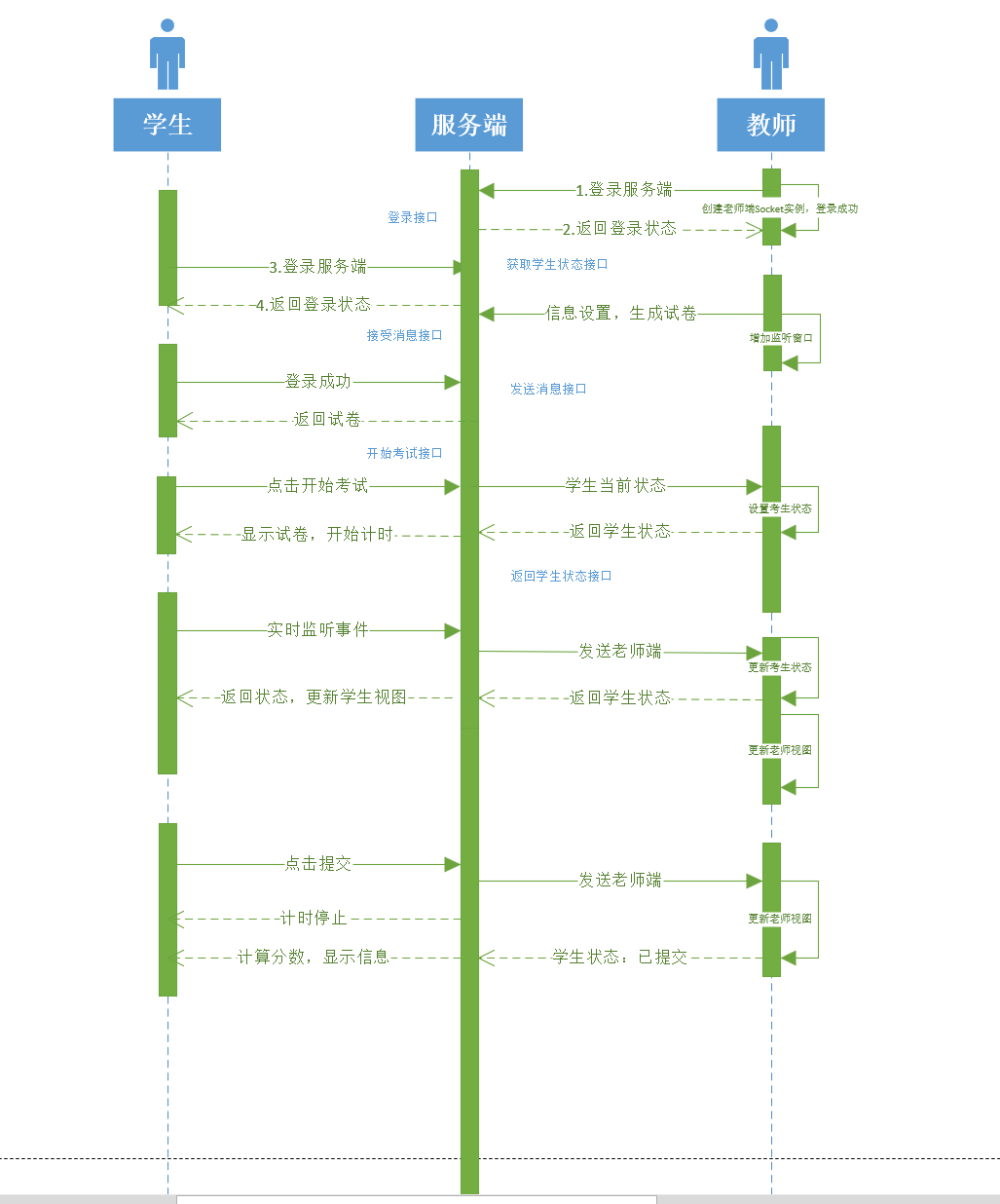
-未交卷

- 得分

- 分值

提供的考生列表，用来记录我们考生的全部信息（Hash表），便于后期查找，用来判断考生登陆是否正确

## 服务端



###### ServerSocket

###### 获取一个学生状态的接口（列表）

存储当前连接对象的socket信息 <List>，用来后面遍历实现SocketTeacher和SocketStu之间的通信

###### 发送信息接口

将服务端的学生状态，发送到客户端，再传递给老师端。out线程

###### 接收信息接口

获取客户端的信息。In 线程

###### 登陆接口

点击考试开始，事件监听，

###### 开始考试接口