

##小组成员

> 031502107 陈家权 031502147 庄加鑫

##问题描述

> 各个部门在开学初占据学校青春广场有利位置,通过张贴海报、发传单等形式向学生宣传;对某个部门感兴趣的同学,填写加入部门申请表交给各部门负责人。各部门负责人通过一种说不清道不明的算法对申请的学生进行人工筛选,人工筛选留下的学生也面临被淘汰问题。筛选和淘汰的规则如下:

现状困扰的是:流程繁琐复杂,各个部门手工发放申请表,手工收集汇总,各个部门之间信息沟通不畅,导致不少学生加入几个部门后,由于活动时间冲突而被淘汰,浪费时间和精力。学生在加入部门前对部门的情况了解有限;部门在学生申请之前对学生也不了解,稀里糊涂,不可言说,就接收了,导致后续配合存在隐患和困扰。

现在,现在,现在,我们很想做这样一个系统,请你和你的“对友”设计一个原型系统,让部门选择的过程能够信息化起来,让学生和部门之间可以双向选择。

##任务要点

>部门纳新人数和面试时间必须事先申报确定;

部门活动时间包括常规活动时间(如每周三 19 点-20 点)和临时活动时间,常规活动时间在纳新时候就要公布;

如果一个学生常规部门活动时间请假超过 6 次,将面临被淘汰;

学生最多加入 5 个部门,但是要考虑部门活动时间冲突次数;

未参加部门面试的学生不能纳入部门。

##探索思路

>

构建之法这本书第 8 章中提到: NABCD 模型是进行竞争性需求分析的有效方法。现在来说这次现实的需求分析:

1.N(Need, 需求): 设计一个系统,让部门选择的过程能够信息化起来,让学生和部门之间可以双向选择

2.A (Approach, 做法): 争取又快又好的实现系统,给需求者一个满意的答复

3.B (Benefit, 好处):

学生使用我们的系统,。各个部门的活动时间和纳新人数在宿舍便可了如指掌。不必去青春广场填写部门申请表了,在宿舍动动手指就 ok 了。

部门的管理员使用我们的系统,可以在线发布部门纳新的相关信息,省去了张贴海报、发传单所需的人力物力,也可在线发布审核申请信息。

4.C(Competitor, 竞争) 看清我方优势和劣势,做出让需求者满意的系统

5.D (Delivery, 推广) 待续...

明白了用户的需求,满怀一腔热血的想要干一番事业。然而,迎头就是一盆冷水,原型设计是什么?又如何去实现原型模型?问题接踵而至。唯一的解决办法就是去学习呗。

和队友讨论了一下该用什么设计工具,一番商讨之后决定采用 Axure Rp 。接着就是下载安装 Axure Rp ,安装好之后,在网上搜索了关于 Axure Rp 的基础教程,找到了[Axure Rp 核心训练](<http://www.webppd.com/thread-9347-1-1.html>),文件共享之后,按着文档里所说的操

练一遍，慢慢摸索熟悉 Axure Rp 的用法。

>

解决方案

>

我们设计了一个福州大学学生会部门系统的原型。这个原型有三个界面，分别是登录界面，部门选择界面，学生选择界面。

1、无论是学生还是部门管理人员，都可以从登录界面登录，进入到相应的管理界面进行操作。



2、部门管理人员登录进去可以看到申请加入本部门的学生信息，可以根据该学生参加面试的情况和已加入的部门数来确定是否让该学生加入。



3、学生登录进去可以看到全部的部门信息和自己已成功加入的部门信息，学生可以填写信息后来申请加入部门。

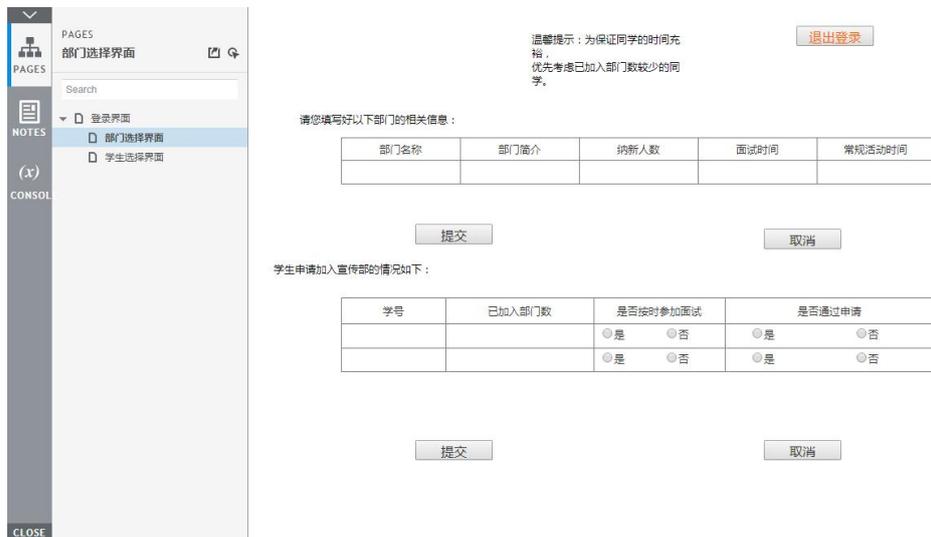


##改进, 完善

>之后在复看原型模型过程中, 发现了一处逻辑 BUG, 部门的相关信息应该由部门的管理人员在部门选择系统发布, 而在初代原型模型并未体现出来, 因此需要改进。

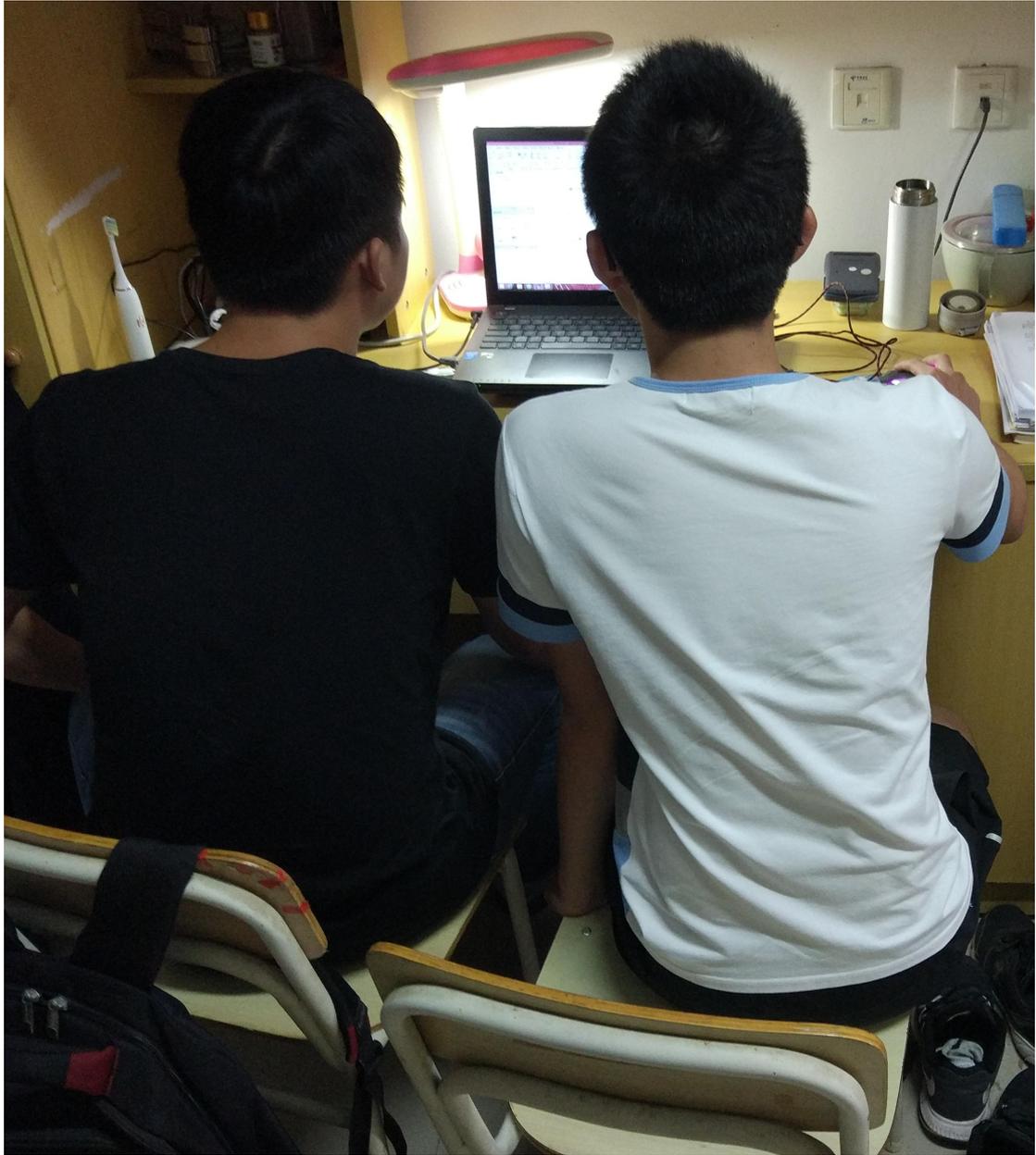
部门管理人员登录进去后可以发布本部门的相关信息, 其中包含部门名称, 简介, 纳新人数, 常规活动时间和面试时间。

改进后的界面:



##结对照片

同班同学嘛, 说组队就组队 (-> _ ->)



PSP 表格

PSP2.1		Personal Software Process Stages	
预估耗时 (分钟)	实际耗时 (分钟)		
-----	-----	-----	-----
Planning			计划
30	20		
• Estimate			• 估计这个任务需要多少时间
30	20		
Development			开发
360	420		

• Analysis		• 需求分析 (包括学习新技术)
60	70	
• Design Spec		• 生成设计文档
240	270	
• Design Review		• 设计复审 (和同事审核设计文档)
60	80	
• Coding Standard		• 代码规范 (为目前的开发制定合适的规范)
范)		
• Design		• 具体设计
• Coding		• 具体编码
• Code Review		• 代码复审
• Test		• 测试 (自我测试, 修改代码, 提交修改)
改)		
Reporting		报告
60	60	
• Test Report		• 测试报告
• Size Measurement		• 计算工作量
• Postmortem & Process Improvement Plan		• 事后总结, 并提出过程改进计划
合计		
450	500	