项目整理

1.自我介绍

我叫柯贤庭, 来自西安电子科技大学, 本科就读于湖北省宜昌市的三峡大学, 学的是信息与计算科学专业, 本科四年通过自己的刻苦学习, 获得保研资格, 保送到西电计算机学院读研. 本科期间连续四年获取国家励志奖学金,校一等奖学金. 并且在学习之余参加全国大学生数学建模竞赛获取省三等奖, 加入校级学生会(英语学习部),任副部长; 研究生期间, 和志同道合的朋友成立呀噗工作室, 并完成作品yup!(一款社交类的APP), 同时该作品获得2017年互联网+校级二等奖, 华为杯移动终端比赛校级一等奖, 研究生移动终端设计比赛国家二等奖; 负责工作室在2018年的智能pose相机入围微软创新杯半决赛.

2.实习

Java测试实习生:

1).实习简介:2015年7月至9月在上海DerbySoft软件公司实习, 主要从事Java测试工作; 这是一家服务全球各大酒店数据对接以及搜索的公司;

2).主要职责: 单元测试工作

i).接口功能测试:用来保证接口功能的正确性;

ii).局部数据测试:用来保证接口中的数据结构是正确的,比如变量是否有初始值,变量是否溢出等;

iii).边界条件测试:变量有没有赋值, 为null的判断; 如果变量是数值,包括主要边界(最大值,最小值,无穷大)等;溢出边界包括期望异常或拒绝服务;临近边界包括最小值+1,最大值-1; 如果变量是字符串,则考虑引用字符变量的边界,空字符串等; 如果变量是集合则考虑空集合,集合的大小的改变,集合元素的顺序的影响等;

iv).所有独立执行通路测试,保证每一条代码, 每个分支都经过测试, 代码覆盖率保证:语句覆盖(每一条语句都能执行到);判定覆盖(保证每一个分支都执行到);条件覆盖(保证每一个条件覆盖到true和false,即if while中的条件);路径覆盖(保证每一个路径都能覆盖到)

v).测试文档的规范化: 对每一个测试都详细撰写测试用例文档, 对于测试不通过的用例,及时反馈, 与开发人员沟通并商量处理办法.

3)实习收获:通过实习让我了解到了测试的流程, 对Java基础有了更深入的学习,同时懂得了如何和团队合作, 面对问题如何解决.

3.智能POSE相机的服务器端:

1).项目简介: 在不同场景下匹配场景的最佳拍照姿势的APP.

2).主要功能:

i)用户进入应用后选择自拍pose, 在素材库选择pose(按使用热度排序),调整pose图片位置以及大小,开始对比pose自拍;

ii).上传pose到素材库,拍摄者登录后确认上传pose,若为注册账号登录,将跳转到注册登录页面,登录成功后再执行上传pose操作, 系统对上传成功与否进行返回,并对成功上传者给予一定的P币奖励,并跳转至登录页面;

iii).查看pose详情, 点击pose缩略图查看pose详情;

iv).照片分析,场景分析主要是Android客户端实现;

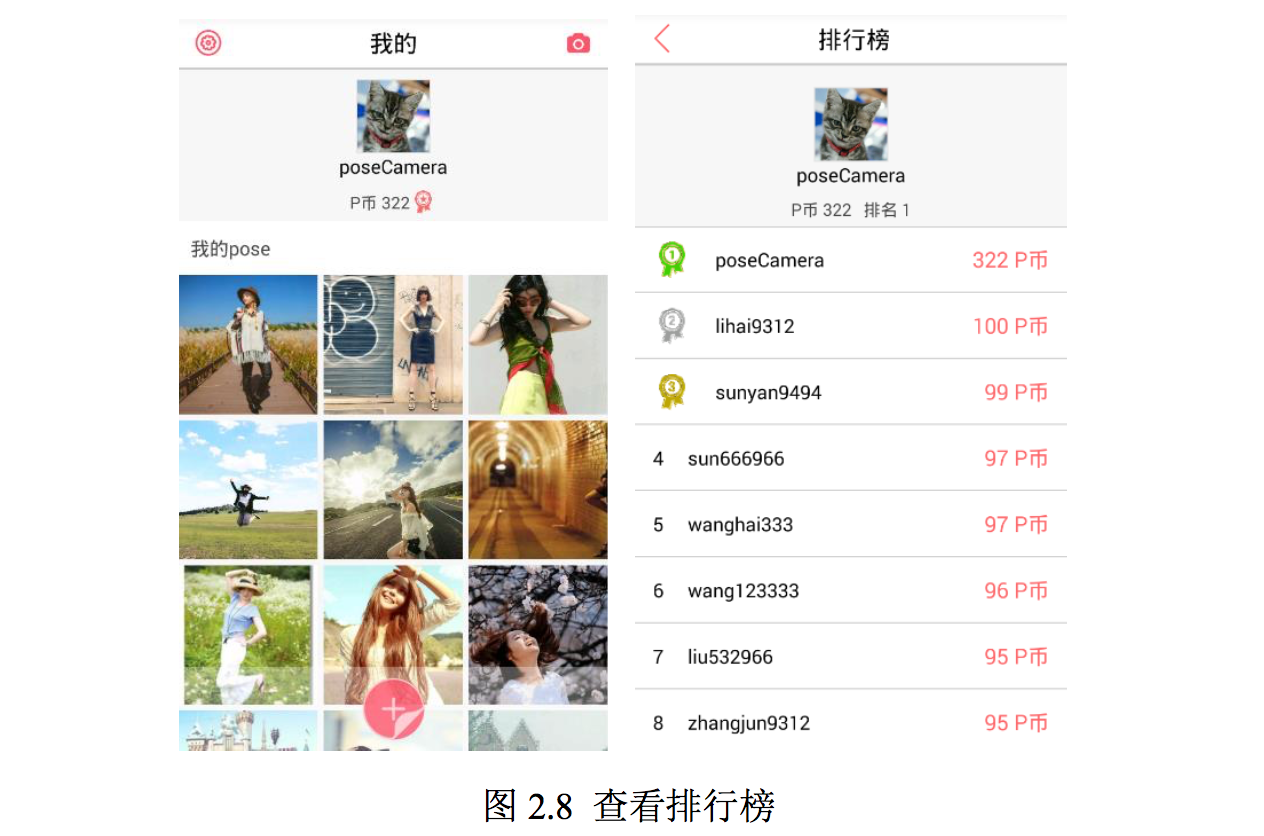
v).用户注册后可以通过pose排行榜查看当前热门pose

**主要功能截图:**











3).主要职责: 服务器端开发,主要采用SSM框架,用spring来管理整个服务器端的结构,SpringMVC用于处理视图层的逻辑, mybatis用于数据库持久层框架.

i).服务器端主要包括用户的管理:用户注册,用户的pose库,用户的P币等级

ii).pose素材的管理,包括素材的热度排行,搜索情况,使用情况等

主要难点: pose素材推荐算法,首先TF-IDF相似度算法进行相似度计算,根据相似度检索数据库照片并按相似度高低排序返回满足条件的图片*ID,* 此过程涉及数据排序，传统的排序算法如冒泡排序、快速排序、堆排序、归并排序等等[5]。它们处理数据量为*n*的平均时间复杂度最好为O(*nlgn*)，当*n*达到大数据规模时，用上述思想无疑会增大时间开销[6]，但传统*Top-K*算法，只需返回指定前k个数据即可，大大缩小排序开销。但若继续增加排序条件：每个元素的特征向量所占权值不同（多权值），且每个元素附带一个额外的信息（热度因子h），设定数据的热度因子会影响到排序的结果，返回的前*k*个值还必须按热度因子排序。那么按照传统做饭，这样做的时间复杂度为O(*nlgn+k2*)。在这种情况下，有必要想一种更加高效可行的办法，只根据需求，输入返回的个数*k*，便能在(*nlogk*)的时间复杂度下找到前*k*个最大（或最小）的数据[7]，即使加了限制条件：返回的前*k*个值还必须按热度因子的大小排序.结构图如下:

****

该项目最终获得2016年第二届研究生移动终端比赛一等奖,华为校园开发者大赛全国亚军.(PS:由于该项目我不是主干开发,为了师兄师姐找工作方便,比赛成员没有写我.)

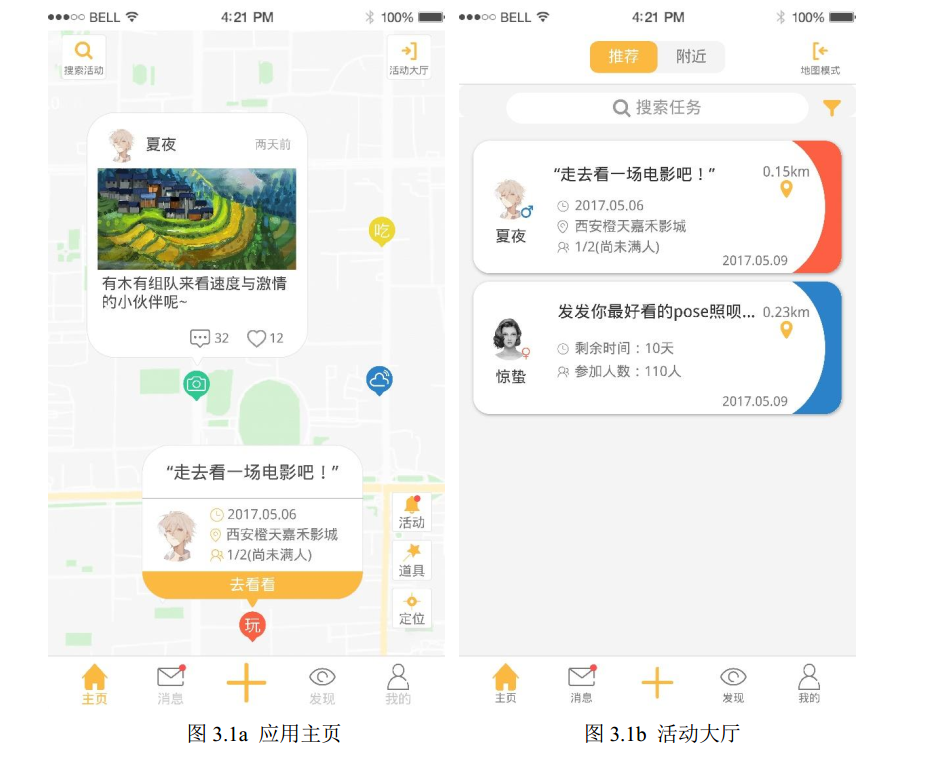
4).项目收获:通过该项目,让我体会到了一款好的产品从产品雏形,中间设计思路的实施,到最终产品成型的过程,如果与团队协作开发,与产品,客户端进行业务以及功能的沟通,收获很大.

4.yup!

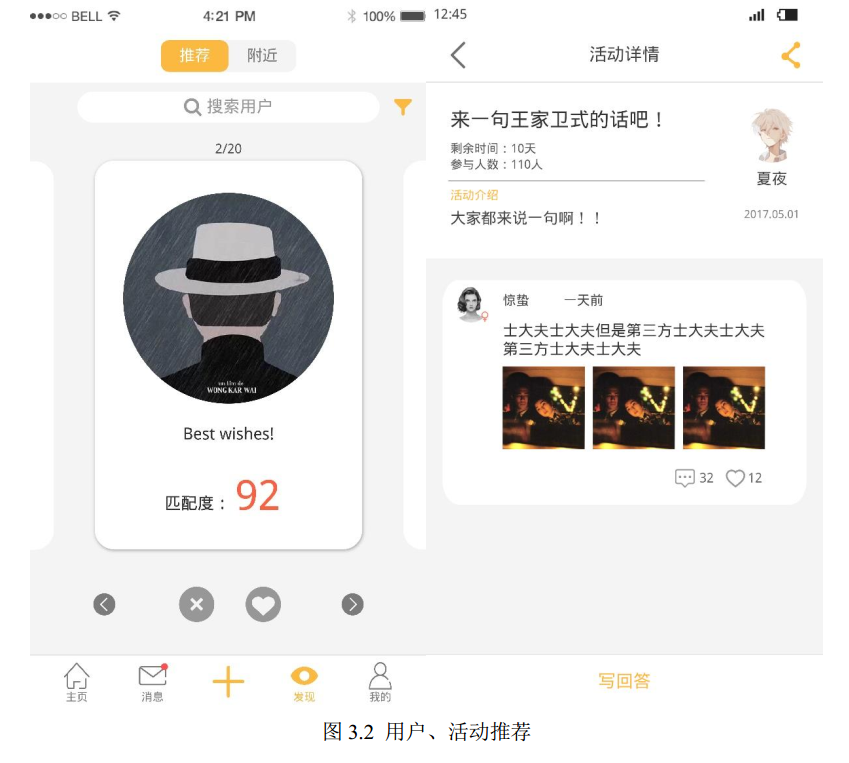
1).项目简介: yup!一款基于地图和只能推荐算的活动社交APP,用户在地图上发布活动,其他人可申请参与,完成活动后,可通过对发起者进行评分,帖标签的方式进行评价,同时yup!御用WHK-TOP算法进行用户以及活动推荐.

2).主要功能介绍:

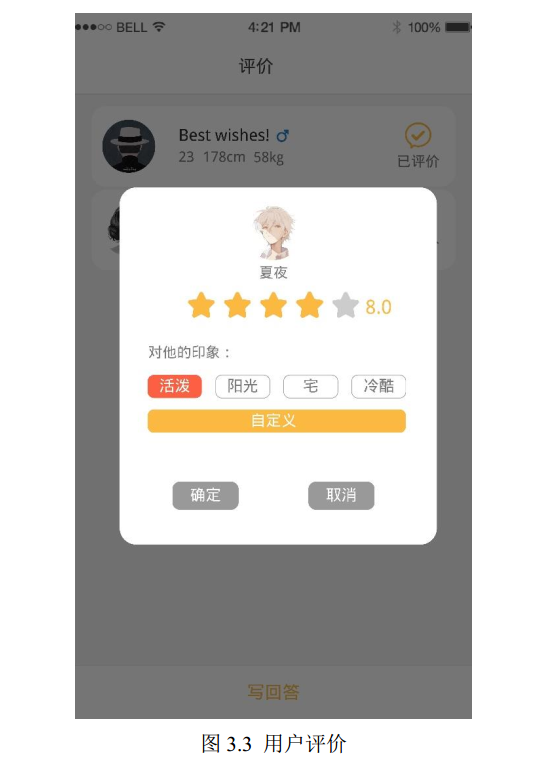
i)基于地图的活动交友:为用户提供腹肌丰富的活动信息,用户可基于活动进行交友, 用户可以通过地图发布活动，发布活动后，周围的用户可以选择申请加入该  
活动。活动的内容包括线上活动和线下活动, 线下活动在法律规定范围内包括但不限于美食、电影、玩乐等.



ii). 智能化推荐匹配的用户和活动，促进深度交友, 系统推荐算法会基于用户的基本信息， 个性标签，评分等来推荐最为匹配的人。标签和评分通过每次发布任务或参加任务后参与者或发起者对其的评价产生。  
同时用户 A 在完成某个任务后，会对其中和他一起参加活动的人进行打分。系统也会对这次打分进行记录.



iii). 评价系统，活动后双方互相评价，让用户更加个性化, 在完成线上/线下活动后，发起者和参与者可以进行互评，互评内容包括发起者发起活动的有趣程度，以及根据活动进行中的互动，对他贴个性的标签.



3).主要职责: 负责服务器端开发, 接口文档的设计, 后台管理系统的设计以及实现

i).接口文档设计: 关于开发的全局配置信息, 返回状态信息,错误信息, 数据库表的设计, 对每个接口的详细设计, 设计到哪张表, 传入参数是否必须,类型,返回结构格式, 返回正确信息状态码,错误信息状态码.

ii).采用MVC设计模式, SSM框架, 控制层使用SpringMVC, 对客户端的请求通过RequestMapping映射到控制层的处理方法, 然后调用service层的业务处理方法, 在service层调用dao层的mapper方法, 进行数据库持久化操作. 最终返回结果给客户端, 数据格式采用json传输;

**iii).用户推荐算法**, 通智能pose相机的方式; 活动推荐算法: 首先从用户活动矩阵汇总找出与目标用户相近的k个用户,然后提取出该用户喜欢的活动,并通过公式计算用户对每个活动的相似程度,同时带入用户评分准则. 改进的用户协同过滤算法:

借鉴了用户协同过滤算法的思想，并加以改进。传统的基于用户的协同过滤算法（User Collaboration Filter，UserCF）核心思想为：在一个在线推荐系统中，当用户u需要个性化推荐时，可以先找到和他有相似兴趣的其它用户，然后把那些用户喜欢的、而用户v没有听说过的物品推荐给v，这种方法称为基于用户的协同过滤算法。但其未考虑到项目热门程度不同，在计算用户相似度时，赋予所有项目相同的权值。常见计算用户兴趣相似度有求Jaccard系数和余弦相似度两种计算方法，就后者举例：N(u)表示用户u曾经评过分的项目集合，N(v)表示用户v曾经评过分的项目集合，计算公式如下所示：



用户项目评分表

|  |  |  |  |
| --- | --- | --- | --- |
| 用户 | 评过分的项目 | | |
| A | a | b | d |
| B | a | c |  |
| C | b | e |  |
| D | c | d | e |

因此，若要计算用户A和B的兴趣相似度，公式应该为：



热门项目将对推荐结果造成负面影响。例如两个用户都曾买过《新华字典》，但因为《新华字典》作为基本工具书，绝大对数人都曾购买过，所以不能很大程度上说明他们兴趣相似，但如果他们俩都购买过同一本冷门书籍，则可认为他们兴趣相似。用N(i)表示所有评分项目中包含项目i的集合，给出改进后的相似度计算公式：



对项目元素个数|N(i)|求对数后再区倒数，降低了热门项目对用户相似度计算的干扰，有助于提高推荐结果的精度和个性化。

则用户u对项目m的感兴趣程度计算公式为：



其中，S(u,K)包含和用户u兴趣最接近的K个用户，N(m)是对项目m评过分的用户集合，wuv是用户u和用户v的兴趣相似度，rvm代表用户v对项目m的兴趣（因为使用单一行为的隐反馈数据，所以rvm=1）

**iv).活动推荐:**

举个例子，假设我们要给 A 推荐物品，选取 K = 3 个相似用户，相似用户则是：B、C、D，那么他们喜欢过并且 A 没有喜欢过的物品有：c、e，那么分别计算 p(A, c) 和 p(A, e)：

http://images.cnitblog.com/blog2015/70278/201504/292347569434600.png

http://images.cnitblog.com/blog2015/70278/201504/292349095052333.png

看样子用户 A 对 c 和 e 的喜欢程度可能是一样的，在真实的推荐系统中，只要按得分排序，取前几个物品就可以了。

对于项目中的推荐算法:

1).每个用户拥有标签(在注册的时候用户自己选系统提出的标签), 人为规定每个标签应该占的权重, 用户热度采用每个用户的粉丝数代替, 然后根据用户标签进行用户推荐,按相似度排序进行推荐.

v).活动评分:引入IMDB评分准则, IMDB的排名评分标准应该是相当完善的, 根据贝叶斯统计公式算出来加权分，对加权分有影响的因子有普通方法计算出来的平均分，投票人数和目前所有电影的平均分;这样的计算结果显示的是电影之间的相对关系， 可以说跟豆瓣、时光这种靠平均统计打分，更能体现出电影的实际质量。具体的计算公式如下：weighted rank (WR) = (v ÷ (v+m)) × R + (m ÷ (v+m)) × C

其中：R = average for the movie (mean) = (Rating) （是用普通的方法计算出的平均分） v = number of votes for the movie = (votes) （投票人数，需要注意的是，只有经常投票者才会被计算在内， 这个下面详细解释） m =minimum votes required to be listed in the top 250 (currently 1250)（进入imdb top 250需要的最小票数，只有三两个人投票的电影就算得满分也没用的） C = the mean vote across the whole report (currently 6.9) （目前所有电影的平均得分） 另外重点来了，根据这个注释： note: for this top 250， only votes from regular voters are considered. 只有‘regular voters‘的投票才会被计算在IMDB top 250之内，这就是IMDB防御因为某 种电影的fans拉票而影响 top 250结果，把top 250尽量限制在资深影迷投票范围内的主要方法。regular voter的标准不详， 估计至少是“投票电影超过xxx部以上”这样的水平，搞不好还会加上投票的时间分布，为支持自己的心爱电影一天内 给N百部电影投票估计也不行。因此，细心的人可以注意到，列入IMDB top 250的电影，其主页面上的分数与250列表中的分数是不同的。

vi).后台管理系统的实现,主要实现了用户活跃数,用户登录此时的统计; 用户活动的查看,用户的管理,活动的审查; 系统消息的推送(用到了极光推送,IM即时通讯), 用户活动的管理等.前端页面主要用到js,html,jQuery,ajax等技术. 数据库采用mysql数据库.

5.科研项目:

1).本科期间的论文: DSLC-FOA: Improved fruit fly optimization algorithm for application to structural engineering design optimization problems; 主要提出改进FOA算法求解结构工程优化问题.

2).研究生期间: 虹膜图像分类, 首先提取图像的LGBP,GLCM特征,然后用到LDA,KNN,SVM, CNN进行分类,论文已投稿.