

一种基于 React Native 框架的换宿系统实现方法

瞿文政,许志明,王嘉茵,倪伟传,万智萍

(中山大学新华学院,广东 东莞 512133)

摘要:打工换宿游是国内外学术界关注的新兴问题。根据国内外打工换宿市场缺少系统化管理的现状,针对用户无法快速、准确获取打工换宿信息的痛点,提出了一种基于 React Native 框架的换宿系统实现方法。系统采用 React Native 高效、快速开发框架,面向 iOS8.0 及以上平台,以 Flexbox 弹性布局与 Koajs 架构下的 MongoDB 数据库结合,使用 ES5 语法,组合 TabBarIOS 导航栏组件等多种第三方组件。同时,在前人的基础上对 Geohash 算法和 Latent Factor 算法进行了改进以提高用户体验。最终,测试表明系统不仅解决了用户的痛点,而且研究成果对基于 iOS 及 React Native 框架的相关研究有很重要的借鉴参考价值,能够帮助换宿游运营者或从业者高效、准确地完成换宿游系统的搭建,同时也可拓展和丰富换宿游系统的研究视角和理论内涵。

关键词:React Native 框架;打工换宿;Geohash 算法;Latent Factor 算法;iOS 系统

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2019)01-0211-05

doi:10.3969/j.issn.1673-629X.2019.01.044

A Working Holiday System Realization Method Based on React Native Framework

QU Wen-zheng, XU Zhi-ming, WANG Jia-yin, NI Wei-chuan, WAN Zhi-ping

(Xinhua College of Sun Yat-Sen University, Dongguan 512133, China)

Abstract: Working holiday is an emerging issue concerned by academic circles at home and abroad. Based on the lack of systematic management in domestic and foreign working holiday market, aiming at the pain point of users' inability to get information quickly and accurately, we propose a method to realize the working holiday system based on React Native framework. The system uses React Native to efficiently and rapidly to develop framework. For iOS8.0 and above platforms, it combines Flexbox flexible layout and the MongoDB database under Koajs architecture. The ES3 syntax is used to combine various third-party components such as TabBarIOS navigation bar components. At the same time, the Geohash algorithm and the Latent Factor algorithm are improved on the basis of the predecessors to improve the user experience. Finally, the test shows that the system not only solves the user's pain points, but also has important reference value for the related research based on the framework of iOS and React Native. It can help working holiday operators or practitioners efficiently and accurately complete the establishment of a working holiday system and at the same time can expand and enrich the research perspective and theoretical connotation of working holiday system.

Key words: React Native framework; working holiday; Geohash algorithm; Latent Factor algorithm; iOS system

0 引言

打工换宿模式在国外深受追捧,但是在国内还鲜为人知。究其原因,一是打工换宿这一符合年轻人个性化旅游发展趋势的风口尚未完全到来;二是国内类似软件平台暂未出现,用户无法快速、准确获取打工换宿信息。换宿平台就是在这样的背景下产生的。

换宿平台是一款服务于国内“年轻”人的打工换

宿交易平台。目的主要是为大学生提供一个集信息发布、过程监督及旅行分享功能的规范化的换宿交易平台。产品定位为国内文艺旅行分享社区,以“打工换宿”为核心主打发现和分享优质旅行及旅行过程中发生的故事并强化用户 UGC 社交属性。除此之外,换宿平台通过高端定制服务实现 B 端和 C 端的交互,打破了传统打工换宿的模式。

收稿日期:2018-02-10

修回日期:2018-06-20

网络出版时间:2018-11-15

基金项目:2017 年度国家级大学生创新创业训练计划基金资助项目(201713902002);2017 年度广东大学生科技创新培育专项资金资助项目(505-pdjh0950);2017 年度校级大学生创新创业训练计划基金资助项目(201713902038)

作者简介:瞿文政(1997-),男,研究方向为嵌入式系统开发应用;万智萍,讲师,研究方向为嵌入式系统及物联网。

网络出版地址:http://cnki.net/kcms/detail/61.1450.TP.20181115.1046.014.html

换宿平台技术上采用 React Native 开发框架,以 Flexbox 弹性盒子来进行布局与 Koa 框架下的 MongoDB 数据库结合,使用 ES5 语法,用 React 基础语法来搭建页面,组合多种第三方组件如 TabBarIOS 导航栏组件等实现原生 iOS 应用开发。

文中主要介绍换宿平台的关键技术,包括 React Native 框架、Koa 框架下 Mongo DB 在 iOS 上的实现及 Geohash 算法和 LatentFactor 算法的实际应用,该平台使用户可以快速、准确地获取打工换宿的信息。

1 React Native 框架

React Native 是移动端的跨平台框架,可以满足在 Android 和 iOS 平台下的移动应用的开发工作,提供了基本的跨平台组件^[1]。RN 可以看作是基于 React 之上的一种针对特定平台的技术开发方案,既能进行网页开发也能完成 iOS 和 Android 跨平台 APP 开发。

RN 使用的是 Javascript 开发语言,组件形态的生命周期机制和状态与属性的数据交互机制让应用开发变得更加灵活而又富有章法,不再是过去的 Html/Css/Javascript 杂糅一起,事件、行为、展现,强行分离和抽象的状态^[2]。同时,性能上尽管 RN 使用 Javascript 语言进行 APP 开发,但是编译后却是原生应用,并非 Hybrid 的混合应用,也非 Codava 将网页内容内嵌到 WebView 里面的黏合应用^[3]。最后,使用 RN 使团队之间合作的门槛也得到降低,仅需要 JS 基础即可。

综上所述,React Native 作为 iOS、Android 的开发平台,具有开发效率高及协作门槛低等特点。所以,选择在 RN 框架的基础上进行换宿平台的设计与实现。

在语法技术层面上,RN 的优势也非常明显。首先,

RN 是单纯的 Javascript 组件化,没有掺和不同的语言形态;其次,RN 的技术框架,允许使用者很方便地介入和调用到各个平台下的 API,仅仅借助 JS 就可以实现只有原生语言才可以实现的效果,这也是 RN 框架的核心竞争力;最后,RN 的布局用到的是 Flex 布局,不用去深究 iOS 底层 UIView 的算法及布局,能更快地构建 APP 原型。RN 框架中 JS 调用 OC 模块方法的流程见图 1。

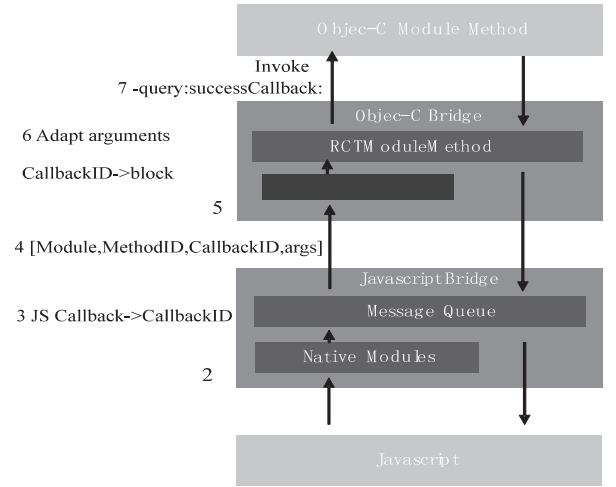


图 1 RN 框架中 JS 调用 OC 模块方法的流程

2 功能需求分析

根据 2 000 余份针对大学生(18 ~ 25 岁)的调查问卷结果显示,大学生打工换宿平台需要满足的 8 个基本功能分别是:查询打工换宿信息/收藏、申请/取消换宿、即时通信、UGC 旅行分享、登录/注册、高端定制服务、实名认证及一键切换房东,功能之间的相互关系框架见图 2。

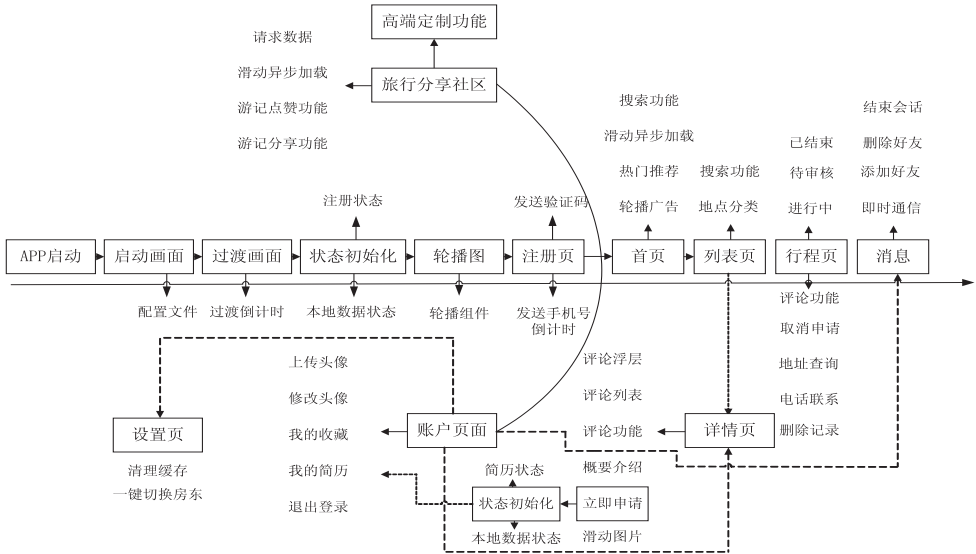


图 2 平台流程结构

每个功能的具体需求如下:
(1) 查询打工换宿信息/收藏:可以按照民宿关键

字、目的地及换宿时间等搜索具体的打工换宿信息,包括换宿概要、费用、评论等,也可以将搜索到的换宿信

息收藏至个人界面。

(2) 申请/取消换宿:提供了一键申请/取消功能,通过预先设置的简历模板对心仪的打工换宿项目进行快速申请。

(3) 即时通信:传统的换宿申请过程中 B 端与 C 端以邮件交流为主,通信效率较低。现在通过线上即时通信来增强信息的交流效率及时性。

(4) UGC 旅行分享:UGC 旅行分享社区为用户提供发布了发布、浏览、点赞及评论原创游记的功能,一方面可以增强用户黏性,另一方面也符合目标人群的行为标准。同时,社区也实现了附近的人、附近的美食及同城攻略等信息的个性化推送功能。

(5) 登录/注册:用户采用手机验证码进行快速登录及注册^[4]。

(6) 高端定制服务:在申请打工换宿的同时,也可以通过付费选择定制服务,以满足用户个性化旅行的需求。

(7) 实名认证:不仅在换宿过程中为年轻人提供安全保障,而且在某种程度上也对 C 端利益保障及行为规范起到一定作用。

(8) 一键切换房东:在设置界面提供 B 端和 C 端用户的一键切换功能,有效解决传统过程中多个 APP 繁琐切换造成用户体验感下降等问题。

3 平台实现

平台是基于 React Native 开发框架在 Macbook OS X 系统下,使用 Sublime、Xcode 模拟器工具、Homebrew 包管理工具结合 Facebook 开源项目 Watchman 和 Flow 及 Nodejs 等进行开发的。后台使用 Mongodb 数据库和 Koa 框架相结合进行上层服务的开发。主要技术关键点是在 RN 框架下调用第三方组件、RN 中获取异步数据请求、RN 中 AsyncStorage 的异步储存、Koa 架构的搭建、Geohash 算法及 LatentFactor 算法的应用等。

3.1 RN 中调用第三方组件

RN 框架常借助第三方组件实现软件的各种功能,从而达到快速、高效的开发。换宿平台的登录及注册功能是通过借助第三方平台-螺丝帽发送一个 4 位手机验证码的方式实现的。螺丝帽官方是使用 Https 传输协议,通过 Https Request 发起一个 Options 请求,是一个回调方式。但是,对于 Koa 数据库框架,需要通过封装把它转化成一个 Promise,这样可以更方便地通过 Yield 调用这个异步过程。Speakeasy 方法是在生成验证码的过程中采用加密功能,避免被模拟生成验证码的这套算法。短信 API 接口关键代码为:

```
var https=require('https')//
```

```
//通过 Https 的 Require 发起一个 Post 请求
var querystring=require('querystring')//
var Promise=require('bluebird')
//引入 Promise 对代码进行封装
var speakeasy=require('speakeasy')
//生成验证码
exports.getCode=function() {
//对外暴露生成验证码的接口
var code=speakeasy.totp({
//对 totp 方法传入一个对象
secret:'Yisuisnice',
//对生成验证码的过程进行加密
digits:4})
//短信字符数为 4 位
return code
//返回
}
```

3.2 RN 中获取异步数据请求

iOS 应用中数据请求总共有四种,分别是同步请求、异步请求、Get 请求、Post 请求。异步请求解决了同步请求中线程阻塞的问题,当发生异步请求之后,程序会建立新的线程进行操作,用户交互丝毫不会受到影响^[5]。

RN 里常用的获取异步数据的网络请求有 Fetch、WebSocket 和 XMLHttpRequest。该换宿平台中异步数据请求是通过在 componentDidMount 中调用 fetchData 的方式进行异步数据的获取。

3.3 RN 中 AsyncStorage 的异步储存

在 Android 的开发中,大多数开发者会使用 SharedPreferences 以键值对的形式对用户的账号密码进行存储。而在本次 RN 开发中,选择了同样采用键值对存储方式的 AsyncStorage(异步持久化的方案)轻量存储系统来进行数据及用户操作路径的存储。AsyncStorage 跟浏览器的 LocalStorage 本地存储有相似之处。因为 AsyncStorage 是全局操作的,所以文中对 AsyncStorage 抽象封装后才进行使用。

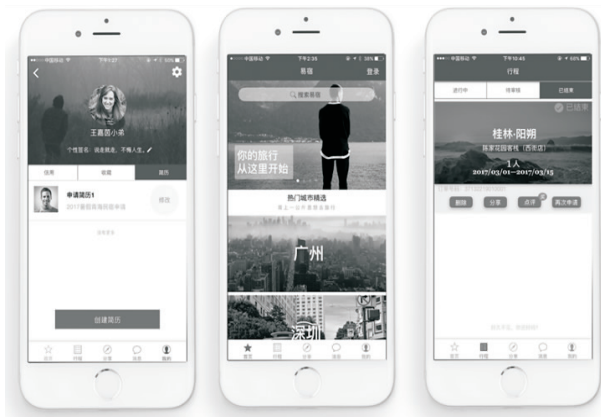


图3 平台展示图

3.4 Koa(Koajs) 架构的搭建

Koa 是使用 ES5 规范的 Generator 和异步编程的轻量级 Web 开发框架^[6]。Koa 不在内核方法中绑定任何中间件, 仅仅提供了一个轻量优雅的函数库, 使得编写 Web 应用变得得心应手, 大大提高了软件的开发效率。

在 Koa 架构的搭建过程中, 首先安装 Koa 模块, 然后在应用的最开始部分放入 Koa-logger 开发环境下的日志中间件, 统计整个请求所花费的时间及记录状态等。同时, 为了保持用户会话状态, 安装了基于 cookie 的会话中间件 Koa-session; 表单数据的解析上是使用 Koa-bodyparser 来生成对象结构数据; 路由中间件使用 Koa-router 为不同的 Url 地址分配不同的规则, 从而调用不同的函数; 使用针对 Mongodb 数据库的对象建模工具 Mongoose 实现与数据库 Mongodb 进行通信; 数据加密上采用的是哈希算法库 Sha1 及基于安全算法生成随机数字的工具库的 Speakeasy 生成短信验证码^[7]; 使用 Uuid 生成不会重复的 ID; 同时, 为避免 XSS 恶意攻击选择使用 XSS 模块; 最后, 使用 Promise 规范(标准) 实现对封装后的 Promise 库 Bluebird 提供优雅的 API 设计和更好的性能。

3.5 Geohash 算法实现“附近”功能

Geohash 是一种地理编码格式, 它沿着经度和纬度的方向交替二分地球表面, 并使用一个二进制数(Geohash 编码) 表示所形成的互不重叠网格。

Geohash 编码的优点在于既能表明用户的大概位置, 又不至于暴露自己的精确坐标, 有助于隐私保护; 而且只需使用一维索引即可表示一个矩形区域; 另外, 编码的前缀可以表示更大的区域, 更方便查询附近的区域。

文中采用 Geohash 算法对用户的位置坐标进行多次编码, 每次形成一个由 n 个编码组成的长度为 n 的 Geohash 编码。达到位置从粗划分到细划分的变化, 从而进行精确计算, 获取附近(距离小于 10 km) 的物理信息, 并进行地理位置移动距离计算, 实现 UGC 社区中的“附近”功能。

Geohash 算法实际就是将一个经纬度信息, 转换成一个可以排序, 可以比较的字符串编码。具体实现方法是先根据用户当前坐标计算 Geohash, 然后取其前缀进行查询, 即可查询附近的所有地点^[8-9]。

3.6 Latent Factor 算法的个性推荐

Latent Factor Models 是基于潜在因子模型将用户和资源的特征同时映射到相同的潜在因子空间

(Latent Factor Space), 以使得它们可以直接比较的一种算法^[10]。

在换宿平台 UGC 社区中通过使用 Latent Factor(潜在因子) 算法, 对用户进行精准的地点及游记等个性化推荐。原理就是通过对每个用户的偏好进行分类, 比如 A 喜欢清静的、理想、民族风等元素, 如果有一个地点带有这些元素, 那么就将这个地点或游记推荐给该用户, 也就是用元素将用户和地点或游记连接起来。但是, 每个人对不同的元素偏好不同, 同时每个地点或每篇游记包含的元素也不一样, 所以采用以下的矩阵进行简单分析。

3.6.1 用户-潜在因子矩阵 Q

用户-潜在因子矩阵 Q 表示不同的用户对于不用元素的偏好程度, 从 0 到 1 依次表示偏好的强弱, 1 为非常偏好, 0 为无偏好, 如表 1 所示。

表 1 潜在因子矩阵 Q

用户	民族风	理想	孤独	红色	山水	都市
张三	0.6	0.8	0.8	0.1	0.1	0.7
李四	0.1	0	0.9	0.1	0.2	0.2

3.6.2 潜在因子-地点矩阵 P

潜在因子-地点矩阵 P 表示每个地点或分享所含有的各种元素。比如表 2, 丽江是一个偏民族风和理想孤独的地方, 含有民族风这个元素的成分是 0.9, 理想的成分是 0.8, 孤独的成分是 0.8……如表 2 所示。

表 2 潜在因子矩阵 P

地点	民族风	理想	孤独	红色	山水	都市
丽江	0.6	0.8	0.8	0.1	0.1	0.7
阳朔	0.1	0	0.9	0.1	0.2	0.2
遵义	0.3	0.6	0.3	1	0.4	0.3

利用这两个矩阵, 可以计算出张三对丽江的喜欢程度是: 张三对民族风的偏好 * 丽江含有民族风的成分 + 对理想的偏好 * 丽江含有理想的成分 + 对孤独的偏好 * 丽江含有孤独的成分 + ……。

最后, 将每个用户和每个地点按照该方法进行计算, 就可以得出不同用户对不同地点的可推荐值矩阵 \tilde{R} 。(注: 这里的破浪线表示估计的评分, 下文不带破浪线的 R 表示实际的评分)^[11]

表 3 可推荐数值 \tilde{R}

用户	丽江	阳朔	遵义
张三	2.02	1.47	1.25
李四	0.99	0.7	0.54

因此对张三推荐四个地点中得分最高的丽江, 对李四推荐得分最高的丽江。矩阵可表示为:

$$\tilde{R} = QP^T \tag{1}$$

由于实际应用中所获得的数据是异常巨大的,而且只能通过用户行为数据来进行分析,无法使用户自己进行分类并返回自己的偏好系数。所以,在实际分析时获得的是实际评分矩阵 R ,这是一个非常稀疏的矩阵。所以需要对该矩阵的 UV 进行分解。也就是将实际的评分矩阵分解为两个低维度的矩阵,再利用 Q 和 P 两个矩阵的乘积对实际矩阵中的未知评分进行预测,实际上就是求解下面的目标函数的过程^[12]。

$$\min_{P,Q} \sum (r_{ui} - q_i P_u^T)^2 \tag{2}$$

4 应用测试

从全面的角度出发,平台应用测试阶段进行了功能测试、兼容性测试、稳定性测试、安全性测试、分发测试及 BUG 遍历测试^[3]等。目前平台已经完成了在百度 MTC 测试上通过获取 UDID 进行的 100 份的分发内测及 BUG 遍历测试。测试结果证明,换宿平台已完成基本功能的开发和运行。主要功能测试结果如表 4 所示。

表 4 平台主要功能 BUG 遍历测试结果

功能模块	必现 BUG	偶发 BUG	总 BUG 数
登录/注册	0	1	1
首页	0	0	0
行程	0	4	4
我的	2	0	2
消息	0	1	1
社区	4	3	7

5 结束语

文中提出了一种基于 React Native 框架集合了 Geohash 算法和 Latent Factor 算法的 iOS 换宿平台开发方法,该平台不仅解决了用户无法快速、准确获取打工换宿信息的痛点,还通过高端定制服务实现了 B 端和 C 端的交互,同时一定程度上强化了用户 UGC 原创属性。在后期的研究中考考虑继续加入 AI 客服系统及 VR 虚拟场景预览等功能。最后,文中阐述的开发

思路与关键技术,打破了传统 APP 开发的模式,以更加快速、高效、低门槛的方式进行 APP 原生开发,对基于 iOS 及 Android 或 React Native 框架的相关研究有重大的借鉴意义。

参考文献:

[1] 吴幅幅. 移动应用跨平台开发框架研究及在社保权益中的应用[D]. 济南:山东大学,2017.

[2] TILKOV S,VINOSKI S. Node.js:using JavaScript to build high-performance network programs[J]. IEEE Internet Computing,2010,14(6):80-83.

[3] 梁思率,杨树国,王臻. 基于 Android 和 ios 平台的高校仪器预约 APP 设计与实现[J]. 实验技术与管理,2016,33(5):248-251.

[4] 任克强,李晓亮,谢斌. 基于 Android 的手机导航系统设计与实验[J]. 实验技术与管理,2014,31(5):131-135.

[5] 王艳君. 基于 iOS 平台的漫画阅读手机客户端的设计与实现[J]. 现代电子技术,2014,37(19):121-124.

[6] 向隆刚,王德浩,龚健雅. 大规模轨迹数据的 Geohash 编码组织及高效范围查询[J]. 武汉大学学报:信息科学版,2017,42(1):21-27.

[7] 郑静静,叶焱,刘太君,等. 基于 Flex、Red5 和 MongoDB 的视频直播、录制及存储系统设计[J]. 计算机应用,2014,34(2):589-592.

[8] NI Jinfeng,RAVISHANKAR C V. Indexing spatio-temporal trajectories with efficient polynomial approximations[J]. IEEE Transactions on Knowledge and Data Engineering,2007,19(5):663-678.

[9] CUDRE-MAUROUX P,WU E,MADDEN S. TrajStore:an adaptive storage system for very large trajectory data sets[C]//IEEE 26th international conference on data engineering. Long Beach,CA,USA:IEEE,2010:109-120.

[10] 王建芳,张朋飞,谷振鹏,等. 一种优化的带偏置概率矩阵分解算法[J]. 小型微型计算机系统,2017,38(5):1081-1085.

[11] ZHANG Zike,LIU Chuang,ZHANG Yicheng,et al. Solving the cold-start problem in recommender systems with social tags[J]. Europhysics Letters,2010,92(2):1-6.

[12] 杨光. 手机 App 安全性测试初探[J]. 计算机与网络,2014,40(11):46.