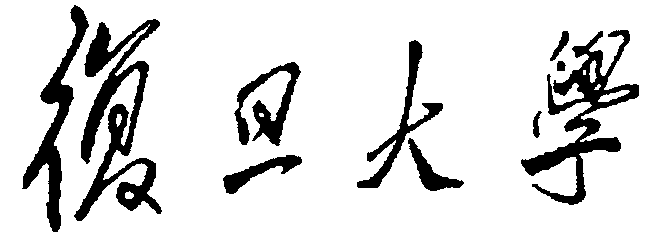
|  |  |
| --- | --- |
|  | 学校代码： 10246 |
|  | 学 号： 14210240100 |



|  |
| --- |
| 硕 士 学 位 论 文 |

**（专业学位）**

|  |
| --- |
| **基于位置的微互助平台设计与实现** |

|  |
| --- |
| **Design** **and Implementation of Micro-Aid Platform Based on Location** |

|  |  |
| --- | --- |
| 院 系： | 计算机科学技术学院 |
| 专业学位类别（领域）： | 计算机技术 |
| 姓 名： | 许家华 |
| 指 导 教 师： | 丁向华 副教授 |
| 完 成 日 期： | 2017年 3月 25日 |

指导小组成员名单

丁向华 副教授

顾 宁 教 授

张 亮 教 授

卢 暾 副教授

# 目录

[目录 I](#_Toc478203237)

[摘要 III](#_Toc478203238)

[Abstract V](#_Toc478203239)

[第一章 绪论 1](#_Toc478203240)

[1.1 研究背景和问题领域 1](#_Toc478203241)

[1.2 国内外研究现状 4](#_Toc478203242)

[1.3本文研究内容 6](#_Toc478203243)

[1.4论文结构 8](#_Toc478203244)

[第二章 相关工作 9](#_Toc478203245)

[2.1 基于上下文信息的推荐系统 9](#_Toc478203246)

[2.2 安全机制 11](#_Toc478203247)

[2.3 激励机制 12](#_Toc478203248)

[2.3.1 显式激励 12](#_Toc478203249)

[2.3.2 隐式激励 14](#_Toc478203250)

[2.4 推荐技术 15](#_Toc478203251)

[2.4.1 基于内容的推荐 15](#_Toc478203252)

[2.4.2 基于协同过滤的推荐 16](#_Toc478203253)

[2.4.3 混合式的推荐 16](#_Toc478203254)

[2.5 小结 17](#_Toc478203255)

[第三章 微互助平台安全机制的设计与实现 19](#_Toc478203256)

[3.1 分类注册功能 20](#_Toc478203257)

[3.1.1 用户模型的设计 20](#_Toc478203258)

[3.1.2 注册的实现 23](#_Toc478203259)

[3.2 群组功能 25](#_Toc478203260)

[3.2.1 任务模型的设计 26](#_Toc478203261)

[3.2.2 群组模型的设计 27](#_Toc478203262)

[3.2.3 群组的实现 28](#_Toc478203263)

[3.3 安全码 31](#_Toc478203264)

[3.3.1 安全码类型的设计 32](#_Toc478203265)

[3.3.2 安全码的实现 33](#_Toc478203266)

[3.4 小结 35](#_Toc478203267)

[第四章 微互助平台激励机制的设计与实现 37](#_Toc478203268)

[4.1 通知功能 37](#_Toc478203269)

[4.1.1 通知功能的设计 37](#_Toc478203270)

[4.1.2 通知功能的实现 39](#_Toc478203271)

[4.2 分享与积分排行 41](#_Toc478203272)

[4.2.1 分享与积分排行的设计 41](#_Toc478203273)

[4.2.2 分享与积分排行的实现 42](#_Toc478203274)

[4.3 小结 44](#_Toc478203275)

[第五章 微互助平台任务推荐策略的制定与实现 45](#_Toc478203276)

[5.1任务推荐中的关键要素的设计 45](#_Toc478203277)

[5.1.1 影响任务接受的因素 45](#_Toc478203278)

[5.1.2 关键要素在微互助平台的获取 48](#_Toc478203279)

[5.2 个性化策略的设计与实现 50](#_Toc478203280)

[5.3 智能推荐策略的设计与实现 52](#_Toc478203281)

[5.4 小结 55](#_Toc478203282)

[第六章 微互助平台系统评估 57](#_Toc478203283)

[6.1用户评价 58](#_Toc478203284)

[6.2 推荐策略效果评估 59](#_Toc478203285)

[6.2.1 推荐效率评估 59](#_Toc478203286)

[6.2.2 推荐准确性评估 60](#_Toc478203287)

[6.3小结 61](#_Toc478203288)

[第七章 总结与展望 63](#_Toc478203289)

[7.1 总结 63](#_Toc478203290)

[7.2 展望 64](#_Toc478203291)

[参考文献 65](#_Toc478203292)

[致谢 69](#_Toc478203293)

# 摘要

随着社会福利制度的不断完善，残疾人的生活质量越来越受到社会各界的关注，志愿活动的形式也越来越丰富。残疾人和志愿者对志愿活动的准确性、及时性和效率等方面也随科技的发展而产生新的要求，这主要是因为在传统的志愿者服务平台存在许多缺陷，主要包括：1）志愿服务结构单一，2）对残疾人日常生活需求考虑不周全，3）服务响应不及时，4）志愿服务不能持久。针对存在这些问题，本文拟通过实现一个基于位置的线上线下结合的微互助系统，以让更多的人参与“随手”公益的方式来解决上述志愿服务存在的问题。该系统通过线上发布微任务，根据位置信息找到就近的相关的志愿者，以实现高效的线下志愿者服务。通过实地调研发现，基于位置的线上线下结合的微互助系统，在实际使用中需要考虑到下述几个方面：1）保证用户的安全性，2）采用相应的激励机制，3）利用位置信息来支持实时任务推荐。

本文针对上述提及的三个方面内容，结合近年来热门的移动端开发平台，包括iOS和Android系统，设计和开发了基于位置的微互助平台。主要的工作如下：

（1）构建了多层次的数据结构模型包括用户、任务、群组等，采用反馈迭代式的设计方法来实现整个平台的开发，通过实地部署服务器、上线到各大应用平台等方式，周期性地调查研究用户的反馈意见来优化平台的界面、拓展功能，保证良好的用户体验。

（2）借鉴现有的安全机制和激励机制，提出了针对线上隐私安全的分类注册和群组模式以及线下执行任务时的安全码功能相结合的方式来对用户的安全进行保障；通过定义相关的通知模型来实现对新任务的推送，采用支持用户分享到第三方平台如微信、微博以及展示用户积分排行榜的方式来鼓励使用者对平台的进一步贡献。

（3）本文通过初步使用，采用访谈研究的方式，得出影响用户接受任务的四大维度。针对这些影响用户的上下文信息，设计获取四大维度的方法，基于用户的历史记录和位置信息，构造用户的推荐模型，采用分值计算法来实现基于位置的任务推荐，最终通过采集真实数据集进行对比分析，验证提出策略的有效性。

**关键字：**位置，安全机制，激励机制，任务推荐策略

**中图分类号：**TP3

# Abstract

With the development of social welfare systems, more attention is paid to the life qualities of the disabled and the forms of volunteer activities have also become increasingly diverse. This creates new requirements for the accuracy and timeliness of volunteer activities, the traditional volunteer service platforms suffer from many defects, including: 1) The homogenous structure of volunteer services. 2) The lack of consideration of the daily requirements for the disabled. 3) The lack of responsiveness for requests from the disabled. 4) Serious brain drain in the voluntary sector. This paper focuses on a location-based micro-volunteering system to resolve these problems. A preliminary study reveals three key aspects to take into account for this type of systems, which include: 1) ensuring safety for all users. 2) Adopting effective incentive mechanisms. 3) Making full use of users location to support real-time task recommendation.

This paper aims to address the above-mentioned issues, with a platform called Micro-Aid based on iOS and Android platforms. The main work and contributions are threefold:

(1) A multilevel data structure model including user, task and model is established. Feedback and iteration design methods are used in the development of Mirco-Aid platform and by means of distributing Mirco-Aid on the Apple store and some Andriod stores. We also interview end users to gain feedback to further improve and optimize the system.

(2) The existing security mechanism and incentive mechanisms are examined. A method, which combines classification registry and group mode online and the security code function offline, is employed to ensure security and safety. Besides, a notification model is realized to push task information to users. Moreover, in the platform, users can find the leader board according to their points and share their tasks to the third-party platform like Wechat or Microblogs to encourage users to contribute more.

(3) An preliminary study is conducted, with which, four dimensions that affect users acceptance of tasks are identified. Accordingly, methods are designed to fetch these dimensions, construct user recommendation model and come up with the task recommendation strategies, mainly based on user history records and location. At the end, the effectiveness of task recommendation strategy is verified by contrastive analysis.

**Keywords:** Location, Security Mechanism, Incentive Mechanism, Task Recommendation Strategy

**Chinese Library Classification:** TP3

# 第一章 绪论

传统的志愿者服务平台和互助平台因为其存在许多不足，使得志愿者的服务没有有效利用，而对残疾人的很多需求不能做到及时响应。一方面存在志愿者结构单一、志愿者人员流失、对残疾人的日常需求考虑不周全等问题；另一方面，对很多可以参与随手公益的日常生活中志愿者没有进行有效利用。为了解决上述问题，我们希望通过一个线上线下结合的基于位置的微互助平台，充分调动日常生活中能随手提供志愿服务的人群的积极性，让更多人成为随手公益的志愿者，为残疾人提供及时有效的志愿者服务。例如，通过考虑志愿者的位置等上下文因素，找到附近有相关经验的合适人员提供志愿服务，以减少耗费在赶路等方面的时间和精力。本文调研了解了现有志愿平台的研究现状，并仔细调查分析志愿者活动的背景以及他们背后的驱动，设计并实现一个基于用户位置的微互助平台来满足用户的要求。

## 1.1 研究背景和问题领域

社会在不断进步，社会福利也在不断提高，残疾人作为特殊的弱势群体深受广大人民群众的关注，社会各界人士也乐于参加各种各样的志愿活动。残疾人是否能够及时得到志愿者的帮助已经成为一个衡量志愿者服务质量的标准。志愿行业也从发达国家渐渐传入了发展中国家，志愿服务作为第三部门的重要组成部分已经成为了下一阶段世界各国主要的发展领域[1]。根据中国残疾人联合会发布的最新调查显示，目前在国内的残疾人人数已经达到了8502万人[2]。但是据不完全统计我国13亿人中的注册志愿者人数仅仅只有5500多万，他们虽然每年提供的志愿服务总时间超过3亿小时[3]，但是平均到每一个残疾人每年却还不到4个小时，这也意味着每一个志愿者的服务对象是多个残疾人。志愿者们通常都是通过各个公益组织举办的活动来帮助残疾人，这就造成了残疾人在公益组织活动的时间内才能得到帮助，但是在日常生活中需要帮助却找不到合适的志愿者。正是由于残疾人和志愿者之间的人数差异，如果只靠这些注册志愿者是完全不能满足残疾人正常的帮助需求。随着无线通信、定位与智能手机等计算机软硬件技术的发展与普及，我国各大政府部门和公益组织也逐渐意识到这个问题。为了高效管理志愿服务，这些组织的管理方式也逐渐由最初线下组织、通过人工来组织管理的方法转变为目前的基于电子信息化的模式，这在一定程度上解决了志愿者与残疾人人数之间的矛盾。但是随着智能手机的兴起，志愿者甚至是有志于志愿事业的普通人越来越希望能够在第一时间接收到周边的志愿信息，以便在减少成本的情况下更好地服务残疾人。在新的要求下，准确性、实时性和效率已经成为了传统的志愿者服务平台重要挑战，其主要根源是由于存在如下的问题：

**1）志愿服务结构单一。**在具有公信力的志愿服务网站如上海志愿者网中，只有注册过的公益组织才能发布志愿信息和志愿者项目。从结构上看，志愿者服务平台的志愿目标和对象十分单一，从层次上仅仅是 “公益组织”与“残疾人组织”这种单一的模式，并没有考虑到并不是每个志愿者或者是有志于公益的志愿人士都是属于公益组织的，这样的设计模式将导致将一些无门无派的志愿者拒之门外，降低潜在志愿者的参与服务的热情。

**2）对残疾人日常生活需求考虑不周全。**传统的志愿者服务平台只能发布和管理一些大型的公益活动，这些活动也仅仅局限于发放衣物、看望孤寡老人等，一方面志愿者大多数情况下并不能利用自己的专业知识和技能来帮助残疾人，另一方面这些活动也没有将残疾人的日常琐事需求考虑进去。这种模式的设计并没有考虑到残疾人的需求，使普通的残疾人在日常生活中不能得到系统所提供的服务。

**3）服务响应不及时。**日常生活中的残疾人需要帮助的内容很多，类型也比较繁杂，残疾人的分布也比较散乱。残疾人与志愿者之间的联系一般是通过公益组织来进行协调的：残疾人首先向公益组织请求帮助，再由该组织来招募志愿者，志愿者经过这些组织的协调来帮助残疾人。一方面，随着残疾人对服务速度和质量要求的提高，这种招募志愿者低效的缺点将会被放大，引起残疾人和志愿者双方的反感；另一方面，通过实际的数据分析，这些日常的琐事其实只需要出很少的力就能得到满足的，而通过上述方式招募到的志愿者由于地理位置的原因可能需要花费很长时间才能到达残疾人周围进行帮助，导致志愿者出现的时候残疾人已经无需他们帮助。传统的志愿者服务平台并不能根据志愿者的位置信息来为志愿者进行任务推荐，使得残疾人的需求不能得到及时响应，志愿者做了许多额外工作，从而导致志愿活动的效率低下。

**4）志愿服务不能持久。**从社会学角度上看，在为社会大众服务的时候，志愿者们也想要得到相应的回报，这种回报既不是那种通过付出劳动后获得的经济利益，也不是在政治上的权力利益，而是存在于服务本身中的激励因素[4]。他们渴望能够从志愿服务中获得精神上的满足和鼓励，而现有的志愿者服务平台往往只是在平台上为每个志愿者添加他们的服务时间，没有加入适当的激励机制，甚至还需要志愿者倒贴交通费用，造成志愿者的付出的额外时间与收获的精神奖励不成正比，无法满足用户的“炫耀”心理并激励他们持续提供志愿帮助，增大了志愿者的流失率，因而不能构建参加志愿服务的长效机制[5]，在当今瞬息万变的大环境中，这种激励措施很难达成高效的志愿服务。

除了传统的志愿者服务平台之外，如今的网上还存在许多互助平台，用户可以自由的请求和提供帮助。但是这种平台的服务大多数为线上服务，其主要活跃群体主要为年轻人，他们分布在世界各地，由于地理位置千差万别，因此平台中绝大多数互助内容线上的“提问-回答”式帮助，而志愿者服务的主要活动形式的线下服务，因此这就成了不可调和的矛盾。

综上所述，导致上述问题的根源是由于科学技术的不断发展，传统志愿者服务平台由于没有充分利用用户的位置信息，进而无法在平台中实现随手公益，调动更广泛的潜在志愿者，使得广大志愿者和残疾人的需求不能得到满足。在实际情况下，人们更加注重与系统实时的交互式体验，系统能够利用位置信息推荐相应的志愿内容，让用户随时随地参与系统的最新活动。因此，针对上述的存在的问题，我们提出了一个基于位置信息的互助平台，它能充分调动日常生活中能随手提供志愿服务的人群的积极性，让更多人成为随手公益的志愿者，为残疾人提供及时有效的志愿者服务。该系统具有如下的特点：

**减少对中间组织者的依赖。**作为志愿者和残疾人之间的协调者，公益组织在实际的志愿活动中存在了不可忽视的作用，它对志愿活动的顺利有序地进行起着重要意义。但是也正是由于它的存在导致了志愿服务结构单一和服务响应不及时等问题，致使满足残疾人需求的小型志愿任务不能得以发布，而且整个志愿活动的招募和完成的过程相对比较漫长。因此，在本文设计的平台中就将减少对公益组织的依赖，并非每一个志愿任务都需要公益组织来进行承接，由此加大了平台的开放性，降低志愿者的门槛，使非注册志愿者也能够参与到志愿活动中，缓解志愿者流失严重的问题，让残疾人周围充满志愿者，从“开源”的角度来解决志愿者人数不足的问题。

**受助者也是施助者。**残疾人在传统的观念中一直处于无助的状态，人们在主观印象中对残疾人的总是处于同情的心态，也在心里对残疾人打上了“需要帮助”的标签。残疾人长期受到社会各界的帮助而无法回馈他人将会产生亏欠心理[6]。在我们的调研过程中发现，他们其实很渴望能够在自己能力范围之内帮助其他人。因此，在系统中我们虽然对注册有分类，但是却并不限制残疾人提供帮助。通过不区分受助者和施助者，让残疾人获得被需要的感觉，使他们明白虽然自身有残疾，但是依然能够为他人提供帮助，依然有自己的擅长之处，这在一定程度上能够帮助残疾人解决心理障碍，获取极大的精神回馈。

**任务智能推荐策略。**传统的志愿活动招募志愿者的情况是等待用户来查看和接受，因此在微互助平台中的智能推荐策略能够帮助转变任务被动等待接受的状态为主动寻求志愿者。这样一方面能够为残疾人找到附近有相关经验的合适人员提供志愿服务，加速服务响应的时间；另一方面利用位置信息能够降低志愿者服务付出的成本，调动他们参与志愿活动的积极性，从“节流”的角度解决志愿者流失严重与日益增长的志愿需求之间的矛盾，做到就近志愿和随手公益。

为了更好地了解用户的想法，我们积极参与上海市的志愿活动，在多次对公益组织负责人以及相关的残疾人、志愿者调查访谈之后，考虑到他们给予的建议，结合系统中“减少对中间组织者的依赖”和“任务智能推荐策略”特点的不足之处，本文总结出了在设计平台过程中应当始终遵循的几个原则：

**1）保证用户的安全性。**残疾人作为社会的弱势群体和目标用户之一，应当对残疾人的信息加以保护，在系统采用必要的安全机制，帮助残疾人更好地从系统中得到服务。此外，公益组织作为一个协调志愿者和残疾人的组织，能够对双方的身份给予相互的保障。正是由于我们降低了对中间组织者的依赖，导致无法利用公益组织来对志愿双方进行安全保证，才需要更加关注用户的安全性，采用系统内部独有的安全机制来保障用户线上的隐私安全以及线下执行任务的人身安全。

**2）采用相应的激励机制[7]。**为了保证系统中能有源源不断的志愿者加入并且持之以恒地做出相应的志愿服务，需要加入相应的激励机制，让整个系统更有生命力和活力，从而能够缓解志愿者不断流失的尴尬局面，挽留已经加入的志愿者，也进一步来充分调动日常生活中的随手提供志愿者服务的积极性，让更多人能成为随手公益的志愿者。

**3）****利用位置信息来支持实时任务推荐。**不管是残疾人还是志愿者，系统的用户在交互的过程中收集必要的位置信息并提供针对性的辅助信息，在必要的时刻能够为任务的匹配提供更加准确的决策。通过这些位置信息和用户的历史数据，能够为志愿者推荐最符合用户需求的志愿活动。一方面，在平台中需要满足残疾人对志愿服务的实时性要求，另一方面，在面对志愿请求时也能够有效利用可以参与随手公益的日常生活中志愿者，结合用户的地理位置信息来帮助他们降低志愿活动付出的额外时间和精力，为残疾人提供快速、高效的服务。

## 1.2 国内外研究现状

技术的发展使得越来越多人开始研究志愿平台，而且研究的深度和广度也在不断加深。为了在设计和实现平台的时候能够对前人的研究有更好的了解、基于前人的研究继续进一步发展，我们调查了目前的志愿者互助平台的发展历程。

志愿互助平台指的是利用现有资源，为其他人提供无偿服务的平台，平台的用户既可以是受助者也可以是施助者。互助平台的概念始于志愿服务，早在上世纪五六十年代，北京市组织的数万名青年志愿者参与志愿垦荒、毛主席提出的“学雷锋”口号，就是当代青年志愿服务的萌芽[8]。随着改革开放的进行，我国也涌现出了真正意义上的志愿者服务。中国最早的志愿者是在由于社区服务的缘故产生的，通过大力推广，在各地民间也逐渐建立了社区志愿者组织。在九十年代初期，随着社区志愿者规模的不断扩展，政府也建立了中央共青团中这个全国性志愿者组织。目前，在国内规模和影响力最大的就是社区志愿者组织以及共青团志愿者组织。相对于国内，国外的志愿服务起源则要早得多。总体上看，现代志愿者服务起源于十九世纪西方国家宗教性的慈善服务，当时欧美等国家为了改善社会秩序，缩小贫富差距，出台了一系列提高社会福利的规章制度，这些规章制度需要大量志愿人员参与到社会服务中，因此政府也号召普通群众参与到志愿服务。随着时代的进步和发展，志愿服务也渐渐形成了一系列的体系制度，有了相应的组织和规范[9]。

随着计算机技术的发展，特别是软硬件技术的革新和移动互联网的出现，传统的志愿服务已经不能满足广大群众和网友的期望。志愿服务的特性也越来越往电子化和信息化的方法发展，不同规模和性质的互助平台也逐渐涌现，呈现出系统化、综合化、网络化的特征。几乎每个发达国家都会建立起各种官方、非官方的互助网站、手机应用来为志愿者以及各个组织提供方便。高效的互助平台如美国的Volunteer Match[10]能提供双向服务，一方面帮助志愿者快捷方便地寻找到就近的志愿信息，另一方面使志愿活动发起人找到符合要求的志愿者。

在国内，如上海志愿者网这个传统的带有官方性质的志愿者服务平台也逐步改革，以微信公共号的形式来实现对现有移动端的对接，使用户能够通过手机端直接查看志愿任务；其他的互助平台如百度知道、微差事、互助吧、益起邦等基于手机移动端的互助平台也已经逐渐流行，它们具有简单易用等特点，用户甚至能够通过不断参与来获取现金、奖品等实物报酬，深受大众的追捧。

尽管在国内外已经出现了相当多的互助平台，但是这些互助平台或仅仅对公益组织开放，如上海志愿者网，仅公益组织才能发放任务；或使用门槛较高，需要注册为志愿者才能接受任务；或仅提供线上服务；或需要给予实际的物质奖励才能找到实际的志愿者。上述的针对志愿服务的互助平台几乎都没有把随手公益作为主要的服务模式，缺乏支持随手公益的有效机制也是目前造成志愿者流失严重的重要原因，无法建立长效的志愿服务机制[11]，而随手公益的关键是考虑志愿者与志愿申请者之间的上下文信息，以做到志愿者服务的最佳匹配，降低成本。本文就针对上下文尤其是用户的地理位置信息做志愿者服务推荐，即充分利用用户的位置信息来为用户推荐活动，以支持随手公益，让志愿者平台更加开放。

## 1.3本文研究内容

针对上述提到的传统志愿者服务平台的不足，本文以志愿者与残疾人的活动为具体的应用场景，在iOS和Android移动端设计并实现了基于位置的微互助平台，该系统具有的功能模块如图1.1所示。



**图1.1 功能模块图**

整个系统分别基于iOS和Android系统，开发设计成当前最热门的手机移动应用。在此基础上利用用户的地理位置信息来搭建微互助平台，构建成三大机制包括安全机制、激励机制以及智能推荐策略。

相比较于传统的公益组织作为中间人来协调的互助平台，在微互助平台中减少了对公益组织的依赖，这也因此出现了安全和激励上的不足和缺陷，因此在系统的实现时首先对现有的平台可能存在的安全问题、激励机制问题进行分析，探究现有的理论方法，对比其他系统中解决类似问题方案，设计出最符合微互助平台的解决方法并加以实现。最重要的一点是在本文中提出一种基于位置的推荐策略，它采用的主要核心思想是利用用户的位置信息，结合用户的偏好以及历史记录来帮助用户自动匹配最符合的志愿任务，一方面解决发布任务长时间无人接受的尴尬，提高志愿任务的效率，加大需求用户对平台的信任；另一方面能够帮助志愿者降低任务的成本，吸引更多的人参与随手公益。

系统采用反馈迭代式开发，通过用户的反馈来对平台的功能、界面进行改进和完善，主要利用目前较为成熟的MVC三层架构原理对系统的前后端交互进行分离的开发方法。在这种开发模式下，服务器端能够根据用户的相关模型和位置信息，借鉴用户的历史数据，从而从安全机制、激励机制以及智能推荐策略这三方面服务于用户。根据本文的研究和相应的实际开发，各大模块逻辑层结构如图1.2所示。



**图1.2 微互助平台框架图**

正如图1.2所示，微互助平台的主要分为四层结构，从上到下分别是界面层、中间的核心逻辑层和模型层以及最底层的数据层。界面层作为与用户交互的展示界面，主要是展示平台的信息，包括任务地点的分布、用户位置的显示、任务信息的展示、群组的设置以及用户偏好的设定等。逻辑层和模型层是整个系统的中间层，也是系统最核心的部分：模型层的功能是构建所有系统中用到的模型，包括用户、通知、任务、群组等，为核心逻辑层作前期的准备建模以及相应的类构建等工作。核心逻辑层主要是由上述提到的三大机制构成：安全机制主要是从线上的分类注册和群组设定到线下的执行任务时候的安全码功能来体现；激励机制主要表现在用户发布任务时对相关用户进行通知推送，对发布和接受任务的用户积分进行增加和减少的管理，相关的任务也能够在第三方平台进行分享；推荐策略首先收集用户的位置信息，结合用户的偏好以及历史记录来分析、整理和推荐最合适的任务给用户。而数据库层则是承担着存放平台中产生数据的作用，包括相关的基本信息表如任务表、用户表以及相关的关系表，为平台提供基础服务。

总体上来看，系统通过上层的界面以及下层的数据，结合中层的模型与核心逻辑，共同构建了系统的三大机制，保障了整个平台的顺畅运转。

## 1.4论文结构

本论文主要分为7个部分，具体的结构如下：

第一章：绪论，介绍了当代志愿服务的研究背景与研究意义，并对志愿互助平台的发展和现状进行分析，同时梳理了整个论文的结构。

第二章：相关工作，首先介绍了基于上下文信息的推荐系统，接着对现有的推荐系统中主要使用的安全机制、激励机制以及推荐技术进行梳理和简要分析。

第三章：微互助平台安全机制的设计与实现，包括设计用户模型、群组模型等基本模块，对系统整体模型框架的进行介绍，研究了安全码类型的选择，最后分别介绍相关注册分类、群组、安全码功能的实现。

第四章：微互助平台激励机制的设计与实现，通过探寻激励机制相关方法，研究在平台中激励机制的设定，实现了对用户的通知以及相关的积分、排行榜功能。

第五章：微互助平台任务推荐策略的制定与实现，通过实验室内部使用，调查影响任务接受的要素，研究平台中要素的获取方式以及相关用户个性化的设计，实现了任务的智能推荐算法。

第六章：微互助平台系统评估，结合用户的访谈信息以及日志数据来探究平台中所实现的安全机制、激励机制和智能推荐算法的性能。

第七章：总结和展望，总结本论文所做的研究，并且对进一步的发展方向进行展望。

# 第二章 相关工作

由于传统的志愿平台中存在的结构单一、志愿者流失严重等问题，因此我们拟研究支持随手公益的微互助平台，以减少对公益组织的依赖性，调动现有和潜在志愿者的积极性，鼓励他们积极参与到系统的志愿活动中。为了更加完善平台的建设和实施，我们多次对公益组织负责人以及相关的残疾人、志愿者进行调查访谈，仔细考虑了他们给予的建议，结合现有的互助平台，发现采用上述的措施后必须解决系统潜在的安全隐患，加强对系统中激励机制的建设，并且需要有相应的任务推荐策略来对提高志愿活动的效率。

利用用户的上下文信息尤其是地理位置信息是支持随手公益的关键，它能够提供实时的任务推荐能够满足残疾人对志愿服务的实时性要求、减少志愿者的额外支出。这方面的内容涉及到基于上下文信息的推荐，因此本章首先介绍了现有的基于上下文信息的推荐系统。接着对推荐系统中存在的安全机制和激励机制的理论与方法以及一些常见的推荐模式进行梳理和分析，以便在设计和实现平台时能够更好利用这些方法。

## 2.1 基于上下文信息的推荐系统

所谓的基于上下文的推荐系统指的是在一定的上下文环境下，能够获取用户所处周围的物理和逻辑信息，包括地理位置、时间、在系统中的位置等，利用这些信息为用户推荐特殊的物品、信息并提供服务的系统。近年来上下文感知计算理论也是计算机领域的研究热点之一，它能够使系统获取周围环境、系统环境等上下文信息，很大程度上解决系统中行为反应不灵敏、不能根据用户的周边信息来进行及时有效的响应等问题。在国内外这个问题已经转变成在特定场景下研究相应的推荐，如在移动社交环境下的推荐[12,13]、旅游推荐[14,15,16]、城市计算推荐[17,18,19]、上下文感知的手机信息检索下的推荐[20,21]等，这些系统存在着一些共同的特性[22]：即都是基于与用户的交互，根据用户的所处的上下文信息，无需用户明显的操作而为用户的行为提供辅助信息，由此来提高系统推荐的质量。此外，基于上下文信息的典型工作过程还包括收集用户的上下文信息，利用用户的历史数据并通过机器学习的方式构建用户模型，针对用户的某些行为提供相关推荐、追踪用户处理推荐的全过程并将这个过程的数据加入历史数据，为进一步推荐做准备。

与传统的面向PC端和web端的推荐系统相比，“上下文感知”是移动端中基于上下文信息的推荐系统中的最主要的一个特性。早在1991年，Mark Weiser就提出了普适计算的概念[23]，在普适计算中，上下文感知计算理论就是作为最核心的领域进行研究。上下文信息的获取有许多种方式，包括通过各种传感器、网络信息、设备状态、用户的设置等[24]。而基于上下文信息系统的历史可以追溯到Roy Want在1992年提出的一个基于红外线技术的系统[25]，它能检测用户所在位置并将电话转拨给离它最近的用户。这也是公认最早的基于上下文信息的系统。之后的数十年中，出现了一系列基于地理位置信息的旅行导航[16,26,27,28,29]，这种导航系统能够根据用户当前的位置信息提供旅行方面的指引建议。而自从上下文信息的概念出现以来，地理位置已经成为了最热门的信息。除了地理位置信息之外，还有存在许多种类的上下文。因此，“上下文”的含义就从此成了众多学者的研究内容之一。“上下文感知”的概念最早是由Schilit和Theimer[30]提出来的，在最初的对上下文信息进行研究的文献中，上下文信息多指地理位置、附近人群的身份、物体（包括时间、方向、日期、用户情绪等）以及这些物体的改变[31,32,33,34]。

总体来说，Burnett[35]和Gustavsen[36]等人将上述的上下文信息分为两类，即内部信息和外部信息，亦或是Hofer[37]等人所指的物理信息和逻辑信息。外部信息（物理信息）指的是能够通过传感器测量到的信息如位置、亮度、声音分贝、温度、气压、时间等；内部信息（逻辑信息）指的是由用户指定，或者是通过系统与用户的交互捕捉到的信息，如用户的目标、任务、工作流程甚至是用户的情绪状态等。正如上述提到的，大多数基于上下文信息的推荐系统都是利用外部信息来作为主要的内容。Cho等人[xx]采用了用户的位置数据以及两个基于位置的在线社交网络之间的信息来探寻导致人类移动的诱导因素。Quercia和Capra等人[13]通过建立一个“FriendSensing”的框架，利用用户的日志信息以及分析定位信息来为用户推荐个性化的好友。Averjanova等人[14]开发了一个基于地图的手机端推荐系统，能够为用户提供一些个性化的旅游推荐。Ricci和Ngayen等人[15]提出了一个能够在旅行过程中不断进行推荐的系统，该系统通过对话的形式来为用户提供一系列的候选推荐目标供用户选择，并将用户的选择作为进一步的推荐标准。此外，作为上下文信息的重要一部分，也有不少学者对内部信息做出研究，Budzik等人[38]提出的Watson系统和Finkelstein等人[39]提出的IntelliZap系统都是能够根据用户目前已打开的网页、文档等信息来为用户提供服务，提供相关帮助。

基于上下文信息的推荐系统在不断的完善，使用的范围也越来越广泛，上下文信息包括越来越多的内容，在此基础上发展的系统也越来越复杂。但是由于对上下文信息，即使是最常用到的位置信息，在志愿活动中理论框架的研究也并不充足，无法指导开发人员在志愿服务中利用位置信息进行设计与开发，因此限制了位置信息在志愿服务中更深层次的推广，不能开发出令用户满意的推荐系统。而第三部门作为社会重要的一部分，利用志愿者的位置信息来保证志愿服务的高效、有序的运行是十分有必要的。

## 2.2 安全机制

安全是所有用户都十分重视的内容，残疾人本身作为社会的弱势群体，其自身的身份信息、人身安全更应当不容忽视。良好的安全机制能够保证用户对系统的信任，使用户不会因为信息泄露而流失，保证了系统的良好运作。广义上的系统安全包括物理安全、运行安全、信息安全以及安全保密管理四个方面。在推荐系统中主要涉及的是信息安全方面，因此在此针对信息安全方面进行描述。

信息安全的主要目标是保证信息的保密性、真实性和完整性，最常见的保证系统内信息安全的方式包括如下几种：

加密。针对敏感程度较高的内容如用户的密码、职业信息、住址等内容，采用不可逆的加密方法如MD5算法来进行加密计算，即使服务器的数据库受到攻击，黑客得到的内容也是一堆不明所以的乱码，用户的信息安全也得到相应的保护。

权限控制，也称访问控制。权限即用户的权利，权限控制的存在使得用户在登录系统后有些功能可以使用有些功能却无法使用，只有符合相应权限的人才可以使用对应的功能。权限控制一方面可以避免恶意的新用户在系统中胡乱发言等造成系统混乱；另一方面也能够激发用户的进取心，激励用户使用。

显示控制。所谓的显示控制指的是系统将某些内容指定只对某些人开放。与权限控制不同的是，权限控制是用户目前没有该权限，但是经过努力获得权限后就能够接触受限内容；而显示控制指的是所有用户之间没有权限的差别，但是由于自己不属于某个群体而无法看到某些内容。最常见的显示控制为微信的群组功能，用户不属于该群组将无法看到群中的所有消息。

入侵检测技术。入侵检测技术是指采用相应的算法来识别对系统资源的恶意使用行为，包括系统外部的黑客入侵和系统内部用户洪攻击等。该技术能够在系统受到攻击的第一时间进行检测，由此引起系统管理员的注意来加强系统的保护措施，从而保护系统的信息安全。

除了对用户的线上信息进行保护之外，由于本文涉及的系统是一个综合线上、线下的综合性互助平台，因此还需要研究在线下执行志愿活动时所涉及的用户人身安全。在线下执行任务的过程中，最重要的步骤是确认志愿双方的身份，这不仅对残疾人也对志愿者来说都是一个值得关注的内容。最常用的线下确认身份的方法包括：

口令。口令是最简单和方便的互相确认身份的方法。在古代军营中就有着进出对口令的规矩来验证身份。因此，口令双方通过系统生成的不同口令来进行互相问答而无需第三方来进行监管，是一个简单有效的措施。

第三方组织。采用一个志愿双方都信任的第三方组织，通过保证第三方组织的安全性来保证对对方的信任。日常生活中，我们常用到的淘宝网、京东等购物网站就是采用这种方式，网站作为可信任的第三方，发挥着重要的作用。这种方法的安全度相对于口令更高，但是在志愿服务过程中，由于加入了第三方，导致身份验证的过程增长，无法使志愿任务高效地进行。

## 2.3 激励机制

激励机制作为推荐系统中不可忽视的机制之一，发挥着重要的作用。它能通过各种有效的方法来调动用户的积极性和主动性，使其成为系统的活跃用户，帮助实现系统整体的目标，保持系统的活力。Rani和Kumar等人[40]通过长期的研究发现，重复的“行为-奖励”过程会使用户对这个行为养成一种习惯，即使在后续取消了相应的奖励也能使用户保持这种行为。激励按照其特性主要分为显式激励与隐式激励。

### 2.3.1 显式激励

所谓的显式激励是指通过由明显的奖励措施而形成的激励，不单指的是能够看得见的奖励，还包括精神奖励等。最常用的显式激励措施是进行奖励，按照奖励的内容可以分为无形激励和有形激励。奖励的过程通常在用户做出了一个正确的行为或者是达到系统设计者预想的目标时发生。

**无形的激励**即内在激励，又称为精神激励，主要指在精神方面的鼓励和帮助，包括以下几方面的措施：

认可用户的正面行为。一方面对用户的行为进行鼓励，让用户对自身行为有清醒的认识，另一方面为用户营造一种互相信任、互相支持、互相帮助的气氛，提高用户对系统的认同感。

给予用户相应的荣誉。包括相应的积分、称谓、标签等内容，拥有比其他用户出彩的内容。

形成榜样的作用。目标的力量是无穷的，大多数人一旦有了追赶的目标就能够不断自我激励不甘落后，因此，形成正面的榜样并看到自己与其他用户的差距，在这种外界压力之下能够使用户产生前进的动力。

**有形的激励**指的是外在的激励，用户能够通过系统获得一些实在的好处，例如地位、身份以及物质等，主要包括以下措施：

提升用户的权限。加大用户对系统的掌控范围，拥有超越普通用户的权力，使其有别于其他普通的用户，有了这种竞争的良性机制，才能使用户更有动力参与系统。

提高用户的优先级。如果系统中存在类似任务匹配、抢单等机制，可以适当为长期参与的活跃用户提高优先级，提升其匹配任务的速度等。

给予用户物质奖励。物质奖励是最常见和有效的激励措施之一，包括奖金、奖品等内容，使用户获得实质的好处。通过满足用户的物质需求，激发起用户投入系统的动机。

激励机制的形式多种多样，总体来说，它们对系统的两方面的影响，包括助长性和致弱性[41]。

**助长作用**是激励机制的正面影响，它是指在完善的规章制度规范下，用户会持续做出符合系统期望的某些操作。在这样的影响下，系统的用户数量能逐步增加、用户也对系统越发依赖。拥有助长作用的机制是优秀激励机制的必要条件，但是在优秀的机制中不会只存在奖励，也会存在一定的惩罚措施来约束用户的不当行为，达到鼓励良好行为、抑制恶意行为的目的。

**致弱作用**是激励机制的负面影响。尽管在系统中设置激励机制的出发点是希望能够提升用户对系统的依赖，将其转变为深层用户，但是如果系统的机制设置的不够完善，相应的机制不具备普遍性和实施性，系统对用户的过高或过低的要求均会对部分用户的积极性起抑制效果，削弱他们的主动性。因此，必须尽量削减激励机制中存在的致弱作用，建立高效科学可行的激励机制。

因此，在建立相应激励机制时需要满足相应的合理性并注意以下原则：

**目标可达性。**为用户设置目标的时候不能一味地求高求多，而应当做出符合用户能力的设置。

**有形激励与无形激励相结合。**财物上的刺激虽然能够很大程度上给予用户激励，是进行激励的基础，但是毕竟是需要有一定的花费。在一个纯公益的产品中，无形激励作为精神上的鼓励才是根本。因此必须做好二者的有效统一，在此基础上从原有的物质为主的激励逐步过渡到以非物质为主的激励。

**公平原则。**激励的公平性指的是进行激励分配的前提是建立一定的规章制度，满足相应的要求后才能够进行奖励，而且奖励的标准必须在事前就有明确的规范并在用户中广而告之，不因关系的远近亲疏而有所差别，激励影响的效果随着奖励标准的公开性和透明性的增加而不断提升。

**差别原则。**差别原则指的在激励过程中不能实现“平均主义”，而应有所差别，平均分配奖励等于无激励，并不能达到原本的目的。

**奖励的及时性。**及时的奖励有利于将用户的情绪推向最高点，达到趁热打铁的效果，使用户持续投入到系统中，发挥用户的创造性和积极性。如果用户在完成指定活动或行为后马上得到相应的奖励，那么激励的效果最佳，而且激励的效果随着时间的延迟而递减[40]。

### 2.3.2 隐式激励

所谓的隐式激励是相对于显式激励而言的，它通常隐藏在系统的设计中，用户不会轻易察觉到，但是却能够明显感受到系统设计的优势。

Robert E.Kraut等人[7]在讲述关于如何设计一个成功的在线社区的一书中，就提到了一系列的关于激励用户贡献的一些设计措施，主要内容按照分类如表2.1所示。

**表2.1 激励机制设计方案**

|  |  |
| --- | --- |
| 类型 | 设计方案 |
| 选择、分类、突出显示的工具 | 需要用户贡献的列表处于一个社区的显眼位置 |
| 提供易于使用的工具来便于用户查找任务 |
| 问一些用户感兴趣的任务和能完成的任务 |
| 框架设计 | 向特定的用户发任务 |
| 一些简单的任务请求 |
| 强调贡献好处的请求 |
| 造成恐惧心理的请求 |
| 来自地位更高的人的请求 |
| 来自熟悉他们、与他们相似、有吸引力的、高地位的人的请求 |
| 显示其他用户正在做贡献 |
| 提供明确的、具有挑战性的目标 |
| 框架设计 | 带有最后期限的请求 |
| 展示用户贡献的独特性 |
| 显示其他用户做的是补充式贡献 |
| 反馈与奖励 | 有针对用户的表现反馈 |
| 地位、特权和有形的奖励 |
| 互相比较式的用户表现反馈 |
| 根据任务的奖励 |
| 根据任务完成质量的奖励 |
| 设计手段 | 将用户与其他贡献者保持社交联系 |
| 沉浸式经历 |
| 群组结构 | 群组大小控制 |

在其中，提及的一些在界面设计上的内容如在选择、分类等显示上能够突出展示，让用户以熟悉的方式一眼就看到他们感兴趣的任务；还包含了一些显式激励的内容（反馈与奖励）；此外，其中主要提及的是群组结构，文中表示用户如果加入了一个在线社区群组，而且群组人数越少则表示群组内部的人员关系越亲密，会使他们更愿意做出贡献。

## 2.4 推荐技术

所谓的推荐技术，即根据用户已知的操作行为数据，利用设计好的数学算法进行推理比较，得到目标用户想要的信息。推荐系统中最常见的推荐技术主要有基于内容的推荐、协同过滤的推荐以及混合式的推荐技术。

### 2.4.1 基于内容的推荐

大部分人的兴趣爱好一般情况下是不会改变的，而基于内容的推荐技术（Content-based Recommendation）就是根据用户在系统中的历史记录，利用机器学习和内容分析等技术来对用户的特征进行学习分类，分析用户的兴趣点，获取其他用户没浏览过的内容与兴趣点的拟合度，最后依据拟合度的高低推荐相应内容。该方法的理论依据主要来自信息检索和信息过滤，其主要步骤包括：分析目标的历史数据，建立每个用户独特的兴趣模型；分析系统中备选推荐内容，抽取每个内容的独有的特征向量；根据特征向量计算备选内容与兴趣模型的拟合度；根据拟合度选取最大的几项内容推荐给用户[42]。

一般来说，基于内容的推荐技术的主要分析对象是一些比较容易提取的文本内容如新闻、网页、博客等，而提取多媒体数据如图片、视频、音乐等内容具有技术性的困难，关于多媒体数据一般是通过大量用户的标注，系统分析这些标注信息来进行推荐。因此，这项技术一般用于新闻、图书、微博等项目上。此外，基于内容的推荐技术虽然针对用户的兴趣建立特殊模型，并且能够通过不断地增加物品的属性来获得更好的推荐效果，但是从另一方面讲，物品的属性是有限的，很难获得更多数据。另外，基于内容的推荐技术需要分析历史记录来建模，无法为用户发现新的兴趣点，而一旦新用户到来不存在历史记录就难以构建模型进行推荐，无法达到预期的效果。

### 2.4.2 基于协同过滤的推荐

在现实生活中，我们可以发现：与自己爱好类似的人喜欢的内容自己也有很大的可能喜欢。基于协同过滤的推荐（Collaborative Filtering）正是基于这一点发现而产生的。因此，这种推荐方法就是找出和该用户的偏好最相似的其余系统用户，从而根据这个系统用户的偏好内容对目标用户进行推荐，其算法主要包括三个步骤：每个用户对某个内容进行评分，采用相应的算法来表现其对该内容的喜好程度；寻找最近邻居，找到与用户的兴趣最相似的其他用户，此处的兴趣包括喜欢和不喜欢的内容，一般可以采用皮尔森相关系数、余弦相似性或调整余弦相似性三种方法；根据最近邻居的历史列表，找出这个最近邻居评价很高但是该用户还未评价过的项目，向该用户推荐该项目[43]。

与基于内容的推荐相比，这种方法无需考虑备选内容本身的性质和特征向量，而是将用户的评分作为系统推荐的基础，推荐的项目不仅限于文本类，能够推荐难以进行机器自动分析的信息，适合于推荐系统中存在多种类型数据的推荐，可以利用其它现有用户的喜好，规避了基于内容推荐技术中存在的分析不完善和不准确等问题，为用户发现新的兴趣点。而且用户加入系统的时间越长，他们产生的数据越来越丰富，其性能也不断提高。但是由于基于协同过滤的推荐技术的应用主要依靠其它用户的评分等历史数据，因此当系统中的评分数据量不大时，将会发生数据稀疏问题，推荐效果也会因此而下降；另一方面，一旦用户的历史评分增多，系统的复杂度也在增加，计算相似度的复杂性也相应增大，推荐内容所花费的时间也越长，造成效率的降低。

### 2.4.3 混合式的推荐

在实际的推荐过程中通常采用混合式推荐（Hybrid Recommendation）。所谓的混合式推荐是将基于内容的推荐方法和基于协同过滤的推荐方法结合起来的一种推荐方式，包括分别计算将基于内容的推荐和基于协同的推荐后将其结果进行组合；或者是将基于内容的推荐方法中的某些特性集成到基于协同过滤的推荐算法中（反之亦可）。这种推荐技术集中了两种技术的优点，降低了它们之间存在的缺陷。

在混合式推荐的组合方式上，Robin Burke[44]提出了在推荐系统中的七种组合的思路，包括：

1）加权（Weight）：将不同的推荐技术进行权重分配后将之进行整合；

2）变换（Switch）：系统根据自己的需求变换不同的推荐方法；

3）混合（Mix）：将不同的推荐技术得到的内容均展示给用户参考；

4）特征组合（Feature combination）：主要是使用某一种推荐技术，但是结合其他推荐技术的特征；

5）特征扩充（Feature augmentation）：其中某一种推荐技术得到的特征作为另一种推荐技术的输入，由此得出最终的推荐结果；

6）层叠（Cascade）：不同的推荐技术拥有不同的优先级，高优先级的技术在低优先级的算法之上得出进一步更精确的结果；

7）元级别（Meta-level）：将一种推荐技术产生的模型作为其他推荐技术的输入，由此产生出进一步更精确的结果。

上述的这些组合结合了各个推荐方法的优缺点，能够在一定程度上扬长避短，解决推荐系统中的冷启动问题和数据稀疏问题[45]。但是，由于各个推荐算法在实际计算过程中都需要一定的时间进行分析计算，因此，混合式推荐方法对时间的要求就要求很高，这是一种以时间换取推荐精确度的方法。

## 2.5 小结

本章首先介绍基于上下文信息的推荐系统的发展，接着梳理分析了与本文设计实现相关的技术及基本原理，主要分为三部分。第一部分是安全机制的基本理论，主要包含线上的隐私安全和线下的身份识别安全。第二部分是激励机制，包含有显式激励和隐式激励，其中显式激励又分为无形激励和有形激励，分别介绍了这些激烈的基本方法和实现策略。最后一部分是推荐技术，分别介绍了基于内容、基于协同过滤和混合式的推荐技术的应用原理及其优缺点。

# 第三章 微互助平台安全机制的设计与实现

服务结构单一问题是传统的志愿者服务平台中一直存在的严重问题之一，从层次上仅仅是遵循“公益组织”与“残疾人组织”这种单一的模式，并没有考虑到并不是每个志愿者或者是有志于公益的普通人都是属于公益组织的，这样的设计模式打击了一些潜在的志愿者的积极性。究其根源，志愿者服务平台的设计者在进行需求设计的时候就没有将这一点考虑进去，或者是因为担心弱势群体在进行参与志愿活动的时候处于劣势的地位。如果不能通过公益组织这一中间机构来确认志愿者身份的话不能够保证残疾人自身的安全问题，因此才提高了发布和接受志愿任务的门槛，以牺牲用户体验的方式来保护系统用户的安全性，用信任链关系的方式来维持信任关系，即通过信任残疾人组织和公益组织来信任相应的残疾人和志愿者。

因此，要解决传统的志愿者服务平台中服务结构单一问题，就必须减少对公益组织在志愿活动中的依赖，从根本上降低用户发布和接受志愿活动的门槛，提升用户体验，让真正的系统使用者有更多的参与感，而不是完全从“公益组织”这个“中介”组织出发，具体的关系转变如图3.1所示。



**图3.1 关系转变图**

当然，降低了志愿活动发布和接受的门槛就意味着安全性的降低，无法通过信任链的关系来保证对活动双方的信任关系。因此，要实行这一措施的必要前提就是确保和预防参与志愿活动双方的安全，包括线上用户个人信息的隐私安全以及线下人身安全。从微互助平台的角度出发，就是要从最初的注册用户开始，到发布任务的目标群体以及执行任务的人员确认出发来保证用户的安全性，具体如图3.2所示。



**图3.2 微互助平台基础流程**

## 3.1 分类注册功能

### 3.1.1 用户模型的设计

用户是整个微互助平台的基础，作为平台的使用者，设计良好的用户模型有利于平台的顺利开展，提升用户体验等重要功效，包括以下几个原因：

**有助于平台中数据的结构化设计。**微互助平台中最主要的安全机制、激励机制、任务推荐策略等的实现均需要结构化数据的支持，因此，构建基础数据模型的重要性可想而知。只有基础数据的完整性得到保障，才能展现出系统核心逻辑层的功能效果。一旦出现数据缺失就会导致系统的性能降低，同样的道理，数据的冗余也会迫使系统和用户做多余的操作而使效率降低，影响正常的用户体验。因此，完善的用户模型作为基础的结构化数据，可以确定与其他类型数据的关联，构建与其他模型之间的关系。这一点不仅适用于用户模型的建立，对于其他模型如任务模型等内容的建立也是一样的道理。

**体现以用户为中心。**用户模型作为平台基础，以用户为中心建立自然、高效、有序的用户界面有助于用户在后续相关的发布、接受、反馈、推荐任务信息中起到关键作用，因此，探索如何高效建立用户模型具有重大的现实意义。搭建以系统用户为中心的微互助平台，离不开用户模型的建立。只有这样用户才能够自由地与平台中的其他用户进行交互，这体现出了以用户为中心的设计准则，为平台的设计和开发打好坚实的基础。

**展示出完善的交互方式。**只有了解用户的特性和行为，在微互助平台中建立符合用户的特性模型和行为模型，才能加强用户在微互助平台中的交互。因此，在任务模型中也就必须加入以用户为主的模块，构建基于用户认知交互模型之上的志愿任务体系结构。此外，在设计系统的核心逻辑时，所建立的模型与用户的实际行为越相似，那么用户在使用微互助平台的时候所做的额外工作就越少，他们对于平台的认同度就越高，也就越有助于用户高效准确地达成目标，用户体验也就越高。

综上所述，在设计符合微互助平台的安全机制时，第一步就是必须针对用户的特性和行为，分析其行为特点对用户进行建模。这其中首先需要将用户进行分类并对必须保存的用户信息进行归纳，包括他们的任务选择偏好、个人基本信息等内容。

在本文设计和实现的微互助平台中，我们将用户的模型定义为一个五元组，该五元组的表示为：

* ***UM = <UserID, Personality, Role,*** ***Group, Request>***

其中UserID表示的是平台中用户的唯一标识符，在平台后续有关用户数据的生成、汇集、反馈的时候中用到的增加、删除、修改、查找等都需要用这个用户的唯一标识符作为数据库的关联条件。

Personality是一个所有与用户基本信息的集合。其中包含了用户两个方面的特征：一方面是每个平台使用者所具有的客观属性，例如用户名、昵称、密码、年龄、性别、邮箱、身份证号码等一些不可改变或不易改变的内容；另一方面是体现用户主观特征的偏好。所以，我们在平台的设计中就将用户模型中的Personality表示为：

* ***Personality=<UserName, Passwd, Age, Gender, IDNumber, Preference…>***

在表达式中的UserName, Passwd, Age, Gender, IDNumber等内容就是Personality第一方面的客观属性，而Preference则是上文所提及的主观特征。其表示的内容就是用户在选择参与任务的倾向性，为了更好地表达这个概念，我们引入了excel的概念，表示的是搜集到的用户日常生活中的任务种类，并将其描述为：

* ***Preference (UserID) = <Excel(UserID)*︱*Excel(UserID)*** **∈ *ExcelSet>***

Excel(UserID)表示的是属于该UserID用户的擅长内容，每个用户的擅长内容都不能为空，是ExcelSet的一个真子集。这部分的内容需要用户在注册的时候就完成自己的选择。而ExcelSet全集是经过我们长期的观察归纳得到的残疾人用户甚至是普通用户在生活中可能遇到的一些问题类型，包括“拿快递、买东西、查资料、送材料、生活知识、照顾人、其他”共七项内容。

Role代表的是用户的角色，不同的角色拥有的权限不同，在此我们用一个二元组来表示角色：

* ***Role = <RoleType, RoleName>***

RoleType表示的是不同的角色类型，RoleName则是对应具体角色的名称。经过我们的调查分析，在整个志愿者的流程中可能出现了三种角色，包括残疾人、志愿者以及相应的公益组织，在项目中分别用1、2、3来表示其类型。每个不同的Role对应的Personality属性是不同的，例如公益组织还存在联系人、联系人电话等客观属性，而残疾人和志愿者就不存在这些属性。

Group代表的是每个用户所加入的群组，表示的内容包括群组中存在哪些用户、创建者、创建时间等内容，它与用户是多对多的关系，具体内容将在本章下一节的群组功能中介绍。

Request表示的是每个用户所发布和接受的任务，表示的内容包括任务编号、发布内容、接收者、接收时间、通知（Notification）、推荐（Recommend）等客观信息。与群组类似，它和用户模型之间同样是多对多的关系，具体内容也将在下一章节中介绍。

根据上述的描述，微互助平台中具体的用户模型及相关的其他模型的整体框架如图3.3所示。



**图3.3 平台整体模型框架图**

### 3.1.2 注册的实现

我们在上一节详细描述了平台的用户模型及与其相关联的其他交互模型，因此有关用户的数据结构化就将依照上述的表示来实现。在进行代码实现之前，必须对要注册用户进行分类，主要原因在于以下几个方面：

**保证项目的可扩展性。**微互助平台的开发不是一朝一夕就能完善的，而是一个反馈迭代式开发的过程。为了应对未来可能需要进行的修改，设计良好的接口，保证接口的多样性、层次性和逻辑性是十分有必要的，它能够避免将来重新调整程序框架而造成代码被过度工程化的开发。因此，在设计之初就将微互助平台可能存在的角色、权限提前考虑好是由重要意义的，能够在将来的权限分配、界面显示中将不同身份的用户分类展示。

**安全性能上的考虑。**用户的角色不同，相应的权限也不同。用户可以访问而且只能访问自己权限下的资源。虽然目前的系统中并不存在不同的权限问题，但是在此对用户身份进行分类也有助于平台后续扩展工作的进行。此外，对不同用户进行身份上的标识，能够帮助系统针对不同的角色设置不同的安全策略，而且从注册开始就进行分类，填写一些必要的身份信息，在一定程度上有助于保证其他用户在执行线下任务的时候对服务对象和志愿者的身份上的认知，降低双方受到伤害的风险。

综上所述，在用户进行注册时对其身份进行分类并强制要求其填写一些身份相关信息，包括身份证、志愿者证号、残疾证号、手机号码等是十分有意义和必要的。因此，针对不同的注册身份，我们设计的填入信息如表3.1所示。

**表3.1 不同角色用户注册填写信息表**

|  |  |  |  |
| --- | --- | --- | --- |
| 身份 | 志愿者 | 残疾人 | 公益组织 |
| 填写信息 | 手机号（账号）  密码  昵称  身份证号  志愿者证号  帮助内容 | 手机号（账号）  密码  昵称  残疾证号  帮助内容 | 手机号（账号）  密码  组织名称  联系人  联系人电话  联系人身份证号 |

依照上一小节中User模型的表述，在实际的实现中我们将其数据按照如下方式设计：

|  |
| --- |
| class User{  private Integer id;//用户唯一标识符  private String userName;//用户名  private String password;//密码  private String nickName; //昵称（组织名称）  private Integer scores; //积分  private String IDNumber;//身份证号码  private String disabledCertNumber;//残疾人证号  private String volunteerCertNumber;//志愿者证号  private String contacts;//联系人  ……  private String channelID; //手机channelID  private Integer deviceType; //手机类型  private Date lastLoginTime; //上次登录时间  private Integer roleType; //角色类型  } |

上述User类中，身份证号码（IDNumber）和志愿者证号（volunteerCertNumber）用于志愿者填写身份证信息，身份证号码（IDNumber）和联系人（contacts）用于志愿组织的负责人填写身份证信息和联系人信息，残疾人证号（disabledCertNumber）用于残疾人用户填写残疾人证号码。由于在设计的时候不同的角色虽然填写的信息不同，但是均共用同一个User类，因此上述几个特别提到的信息可以为空。

在其中，由于是用户都是采用手机客户端登陆，不同的手机拥有独立的channelID，获取手机的channelID有助于后续的通知功能的开发和实现，而不同手机系统的通知功能的实现方法也不同，因此也需要获取用户的手机类型（deviceType），在平台中deviceType=3代表Android系统，deviceType=4代表iOS。而RoleType则是表示用户的权限身份，在User类中的roleType对应为Role类中的roleID，角色类（Role）的结构为：

|  |
| --- |
| class Role{  private Integer roleID;//角色类型的标识符  private String roleName;//角色名称  } |

注意到在上述的User类中并没有出现我们在上一节中提及的用户擅长内容，这是由于在User类之外，有一个Excel类来管理上文提及的七个主要的任务类型，还存在一个专门的类UserExcelRelation来管理User与Excel类的关系。其中Excel类结构为：

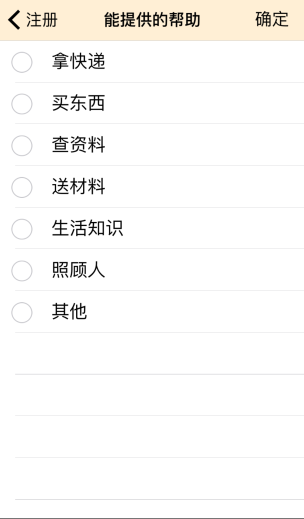
|  |
| --- |
| class Excel{  private Integer excelID;//擅长事物类型ID  private String excelName;//擅长事物名称  } |

UserExcelRelation类的结构为：

|  |
| --- |
| class UserExcelRelation{  private Integer id;//唯一标识符ID  private Integer userID;//用户ID  private Integer excelID;//擅长内容ID  } |

在UserExcelRelation类中，userID为User类中的id，excelID为Excel类中的id，UserExcelRelation类中的每一个数据代表着每个用户与每个擅长内容之间的关联，它们之间的关系是多对多，即允许每个用户拥有多个擅长的任务类型，每个任务类型能够被多个用户所擅长。

因此，经过上述的设计与实现过程后，微互助平台注册模块的界面（包括选择帮助内容界面）如图3.4所示。用户可以自由选择身份进行注册，包括志愿者、残疾人以及公益组织，不同的角色需要填写的内容也各不相同。



**图3.4 微互助平台中的分类注册**

## 3.2 群组功能

群组功能早已经在微信、QQ等即时通讯软件中广泛使用，成为这些软件用户私密聊天的空间。它能够使拥有共同兴趣爱好的用户组成一个小团体，增加用户的活跃性，激励用户对社区的投入[7]。因此，在微信、QQ等即时通讯平台中加入群组是十分有意义的。虽然微互助平台不属于即时通讯软件，但是在其中加入群组功能的主要原因在于以下几方面：

**保证隐私安全。**当智能手机拥有的功能不断增加，传统的设备逐渐被取代，在日常生活中手机的重要性也在日益提升。微互助平台作为一个手机应用，其中涉及到的隐私问题不容忽视。用户可以通过群组发布任务请求，使用户的任务只有群组的成员才能看到，因此在一定程度上保护了用户任务的隐私性，降低用户信息泄露的风险。

**激励用户的使用。**Kraut等人[7]的研究表明，如果一个在线社区中存在群组，而且群组很小的话，用户会更愿意参与社区的贡献。当用户在微互助平台中加入了某个群组中，群组中的成员拥有相同的兴趣爱好，甚至成员的居住范围就在同一个小区，他们就很愿意参与群组中的志愿活动，对活动的激情也会有所提高。

**增加任务的接受率。**研究表明，用户对他们熟悉的请求者发布的任务更愿意接受，他们总是更相信自己熟悉的人。相比于公开任务和来自陌生人的任务，他们更愿意接受来自群组发出的任务，这将大大提高用户发出任务得到接受的概率，极大避免任务无人接受的尴尬情况。

综上所述，在微互助平台中加入群组功能的重要性是显而易见的。因此，在设计群组功能之前，我们首先需要设计任务模型和群组模型，根据模型来对群组功能进行实现和优化。

### 3.2.1 任务模型的设计

任务是整个微互助平台的核心，整个微互助平台就是围绕任务的发布、推荐、接受、完成而进行的，一方面从开发的角度上看，对任务模型进行详细设计有利于系统功能的实现和完善，帮助系统设计人员详细了解任务的进行流程、状态，为后续的开发预留程序接口；另一方面，从用户的角度出发，设计详细的任务模型能够帮助用户搭建高效的任务交互界面，有助于构建基于用户的人机交互平台，加强用户体验。

在上一节中已经从用户的角度简要提到了其和任务的关系，与用户模型类似的是，微互助平台中的任务模型同样也被定义成一个五元组：

* ***Request = < RequestID, Character, Code, Notification, Recommend>***

其中RequestID就是任务的唯一标识符，在平台后续涉及任务数据的生成、状态修改、通知、推荐等过程中用到的增加、删除、修改、查找等都需要用这个任务的唯一标识符作为数据库的关联条件。

Character是一个所有与任务基本信息相关的集合。其中包括了任务的发布者、标题、时间、描述、类型、地址、群组、积分等客观属性，因此，Character的子维度也就定义为：

* ***Character = < Title, UserID, TaskType, Bonus, Group, Status, address…>***

在这些子维度中，特别需要注意的是任务的群组（Group）、悬赏金额（Bonus）以及任务状态（Status），虽然只是任务模型的子维度，但是群组中隐含着系统的安全机制，悬赏隐含着系统的激励机制，都对任务的实现和完成有着重大意义，可以看出任务与群组是多对一的关系，每个任务只能向某一个特定群组发放（“公开”也可以看成是一个群组）。至于任务的状态，一旦创建之后就是未认领状态，如果在任务的截止日期之前有人认领那么就变成已认领，如果认领并过了截止日期则变成已完成，反之如果无人认领任务而到达截止日期则是已过期状态，具体如图3.5所示。



**图3.5 任务状态转移图**

Code表示用户在线下执行任务是用到的安全码，它同样是安全机制的重要组成部分，将在下面章节做详细说明。

Request模型中剩余的两个维度通知（Notification）、推荐（Recommend）表示用户一旦创建新的任务将会有通知给相应的群组，并推荐该任务给相关人员。同样的道理，一旦有用户认领该任务，也同意会有通知出现，具体的内容将在本文的第四章、第五章作详细介绍。

### 3.2.2 群组模型的设计

在针对群组模型的设计过程中，考虑到群组是一个兴趣的共同体，因此在参考了微信群的建立，我们对群组有如下的设计：

**开放式创建。**用户能够随意创建的群组，这样一方面有利于用户对不同群组有着不同的定位，另一方面也能够为用户创建私密的个人空间，保护用户的隐私，增加用户粘性和体验。

**推荐制入群。**用户通过熟人的引荐加入群组而不必通过群主。这样能够形成各自的信任链，即通过信任群里的认识的人来信任群里其他人，加强互相之间的联系，扩大用户在现实中的交际圈，增加任务得到接受的概率。

**自由退出群组。**用户可以自由退出群组，不受到其他用户的约束。正是由于系统采用推荐制加入群组，一旦用户受到推荐就立即加入群组，如果用户受到该群组的骚扰，可以立即自由选择退出该群组。

因此，在微互助平台中设计的群组模型定义为：

* ***Group=<GroupID, GroupName, Creator, CreatTime, GroupUserRelation>***

其中，GroupID表示的是用户创建每个群组后系统自动生成的群组编号，在平台后续涉及群组数据的信息修改、用户关联、信息推送等过程中用到的增加、删除、修改、查找等都需要用这个群组的唯一标识符。

GroupName是群组的名称，群名称并不像GroupID一样作为唯一的标识符，因此允许用户创建一个重复的群名称。

Creator和CreatTime分别代表创建者和创建时间，都是系统在用户创建群组的时候自动与用户关联起来的，一旦用户对个人信息进行修改，系统在群组信息中也会自动进行修改。

GroupUserRelation表示的用户与群组的关系，属于该群组的用户可能有很多，都放在同一个模型中虽然能够使结构变得简单，但是在系统的读写上将会变得很缓慢。因此，系统将用户与群组的关系存放在了一个子模型中，并将GroupUserRelation子维度定义为：

* ***GroupUserRelation = < ID, GroupID, UserID>***

ID是该子维度的标识符，GroupID和UserID分别表示特定的群组和用户，通过定义子模型的方法将User模型和Group模型关联起来。因此，通过GroupUserRelation模型可以看出，Group和User之间是多对多的关系，即每个群组对应着多个用户，每个用户也允许拥有多个群组。

### 3.2.3 群组的实现

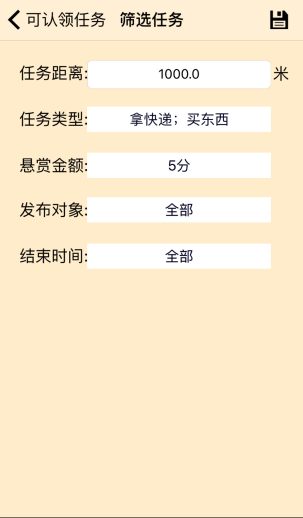
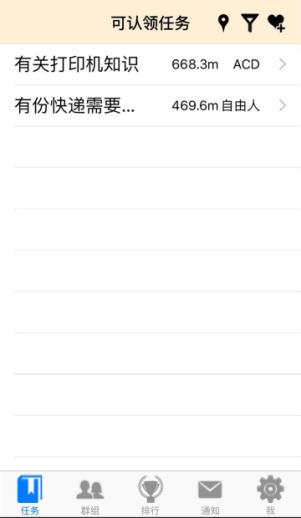
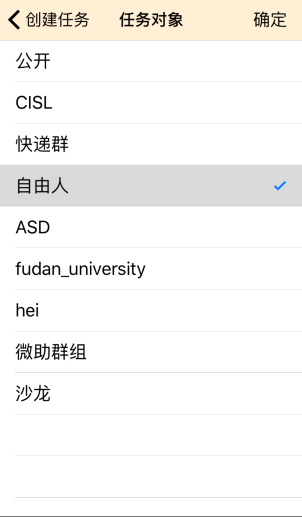
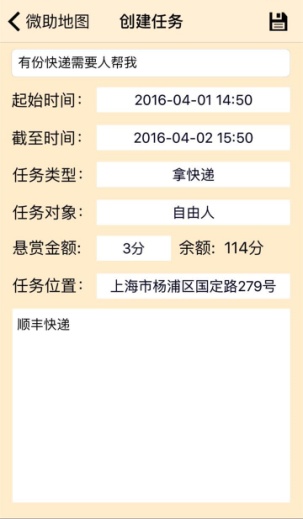
根据上一节的任务模型描述，我们采用下述的方式来设计任务模型：

|  |
| --- |
| class UserExcelRelation{  private Integer id;//任务的标识符  private Integer userID;//发布者标识符  private String title;//任务标题  private String taskType; //任务类型  private String statusInfo; //任务状态  ……  private String groupName;//群组  private Integer recUserID;//任务接受者标识符  private String cipherQuestion;//安全提问  private String cipherAnswer; //安全回答  private Integer isRecommend;//是否推荐过该任务  private Integer isRecommendSuccess; //是否通过推荐接受的任务  } |

上述的内容中，id表示的是模型中的唯一标识符RequestID，cipherQuestion和cipherAnswer是下一节将要详细说明的安全码（Code）， isRecommend和isRecommendSuccess用于任务推荐策略（Recommend），在此只做简单的说明。另一个通知（Notification）在任务模型的数据结构中没有展现出，而是通过在通知模型同利用RequestID来将二者关联起来。剩下的其他项都是属于任务模型中的Character子维度。

在用户创建任务的时候，可以选择发布的群组。默认的情况是“公开”，可以选择用户目前已经加入的任何一个群组作为任务的发布对象，通过这种方式就可以把任务的可见范围缩小为某个特定人群，保证用户的隐私性。

用户可以自由创建任务，需要填写的任务信息包括标题、起始时间、截止时间、任务类型（一共包括上一小结中提到的7大类别）、任务的对象（即上述的发布群组）、悬赏金额（用于奖励接受者，具体将在下一章介绍）以及任务的位置和具体描述。而在创建任务之后则可以在任务列表中清楚看到所有可认领的任务，如果任务比较多的话还能通过筛选功能来实现对任务优先排序，可以按照任务的距离、类型、悬赏金额、发布对象以及结束时间来进行排序，符合要求的任务将靠前展示，方便用户查看和接受特定的任务。具体的微互助平台任务模块的界面如图3.6所示。



**图3.6 微互助平台中的任务模块**

要向某个特定的群组发布任务，首先必须加入一个群组，可以自己创建群组或者受到他人邀请进入该群，群组模型的数据结构设计为如下：

|  |
| --- |
| class Group{  private Integer id;//群组唯一标识符  private String groupName;//群名  private String creator;//创建者  private Date creatTime;//创建时间  private String creatorNickName;//创建者昵称  private Integer size;//大小  } |

注意到上述内容中出现了在群组模型中没有提及的creatorNickName和size这两个变量，这是由于在平台中显示用户模型的时候，系统将自动获取该群创建者的昵称以及加入的总人数。

此外，UserGroupRelation的结构为：

|  |
| --- |
| class UserGroupRelation{  private Integer id;//标识符  private Integer groupID;//群组ID  private Integer userID;//用户ID  } |

通过上述的UserGroupRelation类将群组和用户相关联起来。而自由创建群组、邀请用户加入群组以及退出群的功能则是通过对Group以及UserGroupRelation对应的数据库表进行相应的增加、删除以及修改操作来进行。

新建群组的过程比较简单，只要输入群名称即可。点击某个群组可以查看已经加入该群组的用户，导航栏显示的是群名称，括号内的数字表示群中的人数。已加入群组中的人都可以任意邀请其他用户加入该群，或者自由选择离开该群。具体的群组相关界面如图3.7所示。



**图3.7微互助平台中的群组模块**

## 3.3 安全码

前文提及的分类注册以及群组功能分别在注册和使用功能的时候对用户的信息给予保护，一定程度上保证了用户的隐私不受侵犯。这些功能都是为用户的线上安全提供保障，除此之外还需要对用户线下的安全加以考虑。

一方面，微互助平台是一个开放式的平台，其中涉及的任务包括线上的帮助以及线下的请求。线下的帮助常常涉及任务双方的人身财产安全，虽然用户的注册需要进行身份验证，但是难免出现不法分子钻系统的漏洞，危害系统的正常运转。

另一方面是让线下任务的双方在进行志愿活动前进行身份确认，为任务发生过程中可能出现的状况作担保。微互助平台主要的目标用户之一是残疾人，残疾人作为社会的弱势群体没有很强的自我保护能力，对那些想帮助他们的人戒心将会降低。因此必须要在执行志愿活动的过程中确认到为其服务的志愿者就是接受了该任务的用户。为了防止出现有不法分子伪装成任务承接人的情况出现，需要对志愿者的身份进行甄别。同样的道理，也必须对志愿者的安全负责，确认服务对象的身份，以免出现冒名顶替的现象。

因此，在进行线下任务时，不管是任务的发起人还是接受者，首先要做的就是对对方的身份进行验证。这就要求我们的系统存在一个简单易行的身份验证功能，在微互助系统中就是通过安全码功能来实现。

所谓的安全码，就是用户在进行任务前生成的一对特殊的安全代码，代码的类型包括数字、英文甚至是汉字等。它用于证实任务双方的身份，从而防止用户欺诈等行为。

### 3.3.1 安全码类型的设计

在对安全码类型进行设计的时候，考虑到残疾人作为主要用户，他们的平均文化水平不高，因此排除掉英文字母的选项。而微互助平台的主要用户的中国人，因此可以考虑采用汉字。接着我们分别实现了随机汉字的算法和数字的算法，并对其进行对比试验。

算法3.1详细描述了随机生成简体中文的方法。

|  |
| --- |
| 算法3.1： 随机生成汉字 |
| 输入：*len* ：汉字长度 *hightPos:* 汉字高位 *lowPos:* 汉字低位  输出：*result*：生成的汉字  算法过程：   1. for each i < len do 2. hightPos = (176 + Math.abs(random.nextInt(39))); //获取高位值 3. lowPos = (161 + Math.abs(random.nextInt(93))); //获取低位值 4. b[0] = (new Integer(hightPos).byteValue()); 5. b[1] = (new Integer(lowPos).byteValue()); 6. str = new String(b, "GBk"); 7. result = result + str 8. end for 9. return result |

汉字的编码在计算机中区位码是由四位数字组成的，区位输入法可以输入GB2312-80的全部汉字，因此只需要随机产生一个四位的数字，并把这个数字换成区位码就成输出一个汉字。汉字区位码可以分为高位与底位，每个不同的区位码代表一个汉字，汉字包含简体和繁体，区位码的位数越靠前生成的汉字繁体的机率越大。因此，我们在实现随机生成汉字的过程中高位从十进制的176开始取，低位从十进制的161开始，这样就能避免大部分的繁体字以及生僻字。

接着，我们分别随机生成了10条长度为6的汉字以及长度为6的数字，并邀请了10为用户对来读取这些内容，统计后的用时如图3.8所示。

**图3.8 用户读取安全码用时对比曲线图**

正如图3.8所示，我们对上述的图表进行如下分析：

从整体来看，对数字的读取时间远小于读取汉字的时间，尽管用时的减少幅度不尽相同，这可能是由于如下原因造成：

一方面，尽管在算法中已经尽量规避了繁体字，但是所组成的汉字还是存在某些字比较生僻，例如“瓶硒予埂傲胞”，使用户难以辨识；而数字则是只有0到9共十个，组成的内容并不算多。

另一方面，虽然每个汉语中字都不难辨别，但是组合起来词不达意，是一堆毫无意义的汉字拼接，无法组成通顺的一句话，给用户的识别带来一定程度上的困扰，造成用时过长。

综上所述，考虑到用户在实际应用中的易读性和可用性，我们决定在系统中采用数字作为安全码来验证用户的身份。

### 3.3.2 安全码的实现

在上一节的任务模型中已经提及了cipherQuestion以及cipherAnswer两个变量，分别代表的是安全码的提问与回答，任何一方提出安全提问，只有回答正确才代表身份验证通过。

安全码选择6位数字进行验证，之所以选择6位是因为一个6位的安全码被破解的概率是百万分之一，更短的话破解概率将10倍地提升，更长则会增加用户读数的时间。

与生成汉字的算法类似，算法3.2随机生成数字的方法。

|  |
| --- |
| 算法3.2： 随机生成数字 |
| 输入：*len* ：数字长度  输出：*result*：生成的数字  算法过程：   1. for each i < len do 2. int = random.nextInt(10)//随机生成数字 3. result = result + int 4. end for 5. for each i < len do 6. pos = pos \*10//计算位数 7. end for 8. result = (Math.random()\*9+1)\*pos//得到随机数 9. return result |

这就是安全码的实现，它与任务模型是一一对应的关系，二者是互相绑定的。在安全码的实现上，首先就是需要保证安全码的可见范围。一旦任务被接受之后系统就将自动生成对应的安全码，而且只有任务的发起人和接受者才能在任务详情中查看，用户在执行线下任务的时候可以通过安全码功能来进行提问和回答，以此来确认双方的身份。此外，考虑到安全码的时效性，一旦任务的一方觉得安全码不在安全的时候，都可以选择随时更换。具体的安全码相关界面如图3.9所示。



**图3.9 微互助平台中的任务安全码**

## 3.4 小结

本章介绍了微互助平台中的安全机制设计以及实现。第一部分是分类注册，包括基础用户模型的介绍以及整个系统中的模型框架，接着介绍了其设计的原因以及相应的实现。第二部分是群组功能，介绍了任务模型的构建以及群组模块的相关实现。第三部分是安全码的形成，包括对安全码的类型选择进行相关的对比实验，找寻最优的安全码类型。三个部分相辅相成，共同为平台的安全机制提供基础保障。

# 第四章 微互助平台激励机制的设计与实现

所谓的激励机制，就是在系统中运用特定的方法和管理体系，使用户对系统的参与度最大化的过程。传统的志愿平台中因为其先天的设计不足，不能够使志愿服务的双方得到足够高的激励，因此一方面导致残疾人用户的服务请求不能得到快速相应；另一方面参与活动的志愿者满足不了相应的自身的精神、物质需求而导致其服务不能持久。究其根源，就是传统的志愿平台在对志愿活动双方用户的激励机制考虑不够周到，不能够完全明白他们参与志愿活动的驱动，因此就必须采用特定的激励机制来鼓励志愿者和更多的普通人参与志愿活动并持之以恒地发展下去。

## 4.1 通知功能

### 4.1.1 通知功能的设计

作为移动端应用运营最重要的营销手段，对应用消息的通知推送受到越来越多移动应用开发商的重视。在现如今信息爆炸的年代，人们手机中的应用种类繁多，每部手机收到的通知推送也越来越多，之所以移动应用广泛使用通知推送是因为应用的通知推送具有如下的特点：

**数量大。**每个应用的活跃用户数量不尽相同，少则上百，多则上亿。如果这些应用的活跃用户都授权应用进行通知推送，那么每次推送的数量就能够涉及上千万，这相对于传统的媒介速度更快、范围更大，造成的影响也更广。

**通知精准。**只有安装了手机应用的用户才能接收到该应用的通知信息，这些用户都是系统的活跃用户或者是有志于参与这类应用中内容的用户，也是应用商最想影响人，是潜在或正在进行的消费者，这相比于传统的媒介“广撒网多捞鱼”的策略不同，是不用经过重重过滤就能得到的目标用户。

**免费使用。**应用的通知与否掌握在应用开发商手中，只要开发商的代码符合android或者iOS的协议是无需任何花费的，是一个免费的广告。相比于传统媒介动辄上万的广告费用不可同日而语。

正是由于通知推送的上述特点，手机应用受到应用开发商的热捧。而通知推送对移动应用来说也是一把双刃剑[46]。从正面意义上出发，在微互助平台中使用通知机制具有如下优点：

**能够提高产品的活跃度。**用户的活跃度一定程度上决定了产品是否取得成功，用户数量是产品的根本价值体现。在纷繁复杂的手机应用中，利用通知推送能够吸引用户眼球，激发用户对微互助平台的黏性。

**带动模块的使用率。**根据二八法则，80%的用户只会用到产品中20%的功能，因此运营人员有必要通过通知的手段来引导用户对剩下模块的使用。酒香也怕巷子深，通知功能的正是一个引导用户对相应模块关注的绝佳途径，例如在微互助平台也能够通过推送来带动用户使用群组功能。

**增加用户粘性。**用户对手机应用的使用率即用户粘性，是评估用户对产品信任度和忠诚度的重要标准，消息推送在作为手机应用服务的一个主要组成部分，能够帮助用户对应用内的最新进展进行通知，促进用户关注系统的动态，提高用户的使用率和忠诚度。

**激发潜水用户。**应用的用户绝大多数不是活跃用户，因此利用通知机制来唤醒那些长期不在线的潜水用户，挽留流失用户，有助于解决微互助平台中志愿者流失率大的问题，因此合适的通知能够有效唤醒这部分潜水用户。在微互助平台中对群组中的任务进行广播通知，将会大大提高任务得到接受的速度，不仅能够增加任务发布者的信心，也能够吸引群组中的潜水用户。

虽然通知存在较多优点，但是一旦设计不完善就会出现相应的不利影响：

**对用户造成打扰。**用户的忍耐程度是有限的，一旦通知的信息不是用户关注的内容而且频繁打扰到用户对手机的使用，那么必定会激怒用户并招致用户对应用失去信心。

**用户对通知信息无所谓。**羊来了的故事告诉我们，盲目地向用户推送不感兴趣的内容用户将使用户对系统的推送信息失去信心，从而对应用产生心理抗拒。久而久之即使是真正有价值的通知信息也得不到用户的关注。

**产品丧失用户信任。**用户的下载使用代表着对商品的信任与好奇，透支这份信任必然会丧失用户对产品的好感，造成口碑不断下降以及难以挽回的损失。

综合上述的用户心理，他们对通知的心理反应如图4.1所示。



**图4.1 用户对手机通知的心理反应**

正是由于通知推送的上述特点，在手机推送已经逐渐泛滥的当今社会，如何才能做到最优的推送策略，不让用户反感而是欣然接受已经成为了众多开发者的难题。因此，在设计微互助平台的通知功能中必须遵循如下原则：

**细分通知的对象，不随意推送全部用户。**首先在用户创建任务的时候就引导用户选择任务的目标群组，在对任务内容进行推送的时候就可以做到有的放矢，不必要全部任务都通知给全部用户。

**推荐用户感兴趣的内容。**用户永远只关心自己感兴趣的内容，只有用户加入的群组才表示用户与该群组的其他用户志同道合，是用户的兴趣所在，因此只向用户推荐其所属的群组。

**方便用户使用。**如果用户通过某一条推送信息打开应用，那么一定是因为对这个任务感兴趣。因此，为了提升用户体验，就必须使用户通过该任务的推荐打开的应用界面一定是该任务。

综上所述，对微互助平台中的通知模型进行了如下的定义：

* ***Notification = < NotificationID, RequestID, Content>***

其中NotificationID就是通知的唯一标识符，在平台后续涉及通知数据的生成、推荐等过程中用到的增加、删除、修改、查找等都需要用这个通知的唯一标识符作为数据库的关联条件。

RequestID表示的是任务的唯一标识符，代表该通知是为这个任务发布的，通过这个编号用户能够获取任务的标题，并能够在点击该通知时直接链接到相关的任务详细信息页中。

Content是一个通知基本信息的集合。其中包括了通知的标题、时间、描述、任务名称、群组名称等客观属性，因此，Content的子维度也就定义为：

* ***Content = < Title, Group, time, userID, userName>***

通过userID这个表示用户的唯一标识符能够获取用户的个人相关信息，包括昵称等内容。

### 4.1.2 通知功能的实现

根据上述的通知模型，我们将通知模型的数据结构设计为如下：

|  |
| --- |
| class Notification{  private Integer id; //通知标识符  private Integer userID;//任务发起人的标识符  private Integer RequestID;//任务标识符  private String title;//通知的标题  private String requestName;//任务的标题  private String group;//任务的群组  private Date time;//通知发起时间  } |

上述内容中的id即为通知模型的唯一标识符，而userID、title、requestName、group、time等信息是上述通知模型中的content内容，上述的内容共同组成了通知的数据结构。

要实现通知功能首先必须获取每个手机对应的channelID。不同的手机系统对channelID的获取方法不同，微互助平台中的通知功能是借助百度云推送平台来实现的。首先必须在云推送平台中注册相应的申请对应手机系统的api\_key，获取通过该平台来进行推送的权限。其次，通过在andoird和 iOS的开发代码中加入相应的开发包来将手机与云推送绑定起来并获取每个手机的channelID，以Android平台为例，具体绑定过程如下：

首先必须在系统登陆界面的onCreate()函数中加入了PushManager.startWork函数来确保每次启动程序就启动推送功能，并通过第二段的onBind()函数来进行绑定，获取channelID，在用户登陆过程中将channelID的值传输给服务器，服务器利用用户的channelID进行推送。

绑定之后就可以将信息推送给用户。以推送给某个群组为例，在通知推送之前，获取要推送的用户群组名称并存入变量tagName中，接着构建通知信息的请求并发送该请求：

|  |
| --- |
| PushMsgToTagRequest request = new PushMsgToTagRequest().  addTagName(tagName).//群组名称  addMsgExpires(new Integer(3600)). //过期时间  addMessageType(1).//消息类型：通知  addMessage(notification.toString()).//通知内容  addDeviceType(3);//手机系统型号  PushMsgToTagResponse response = pushClient.pushMsgToTag(request); |

通知信息的具体内容为“群组名称+任务标题名”，这样做是为了便于用户辨别接收到的通知属于哪个群组以及是否愿意查看，至于用户接不接受通知则是在Android系统和iOS系统的设置中自行开启和关闭的。除了接收到的通知，用户还可以通过通知列表查看到接收到的所有通知，其中在每一列都显示除了任务的名称、群组、发起人以及通知的时间，在列前用不同的颜色表示任务状态，红色表示未接受的任务，黄色表示正在进行中，灰色表示已过期，绿色表示已完成。手机接收到通知的界面以及微互助平台的通知列表如图4.2所示。



**图4.2 微互助平台中的通知模块**

## 4.2 分享与积分排行

### 4.2.1 分享与积分排行的设计

上述的通知功能除了在一定程度上增加用户对微互助平台的贡献外，更多的作用还是在于帮助有需求的用户快速完成任务，因此，需要一些其他的功能来刺激用户的使用。

将应用的相关信息分享到微信、微博等第三方的社交平台是各大手机应用商常用的手段，它具有如下的特点：

**提高手机应用的知名度。**第三方社交平台如微信等拥有上亿的用户，通过向微信朋友圈、微博粉丝来展示手机应用等方式能够大大提高应用的知名度，而且第三方社交平台一般都是免费开放的，是一个无需任何花费就能够向广大网友做广告的方式，这对于没有充足资金的应用开发者来说无异于福音。

**增加应用的用户。**相比较于通过地推等方式来进行用户推广，微信朋友圈拥有与使用者相同的爱好，通过用户的分享来帮助应用推广可以在一定程度上吸引其他潜在用户，解决用户渠道缺失问题，使应用的开发团队可以更专心做自己的产品而不必费太多心思在推广上。以微互助平台的用户为例，如果平台用户发布了一个任务并将其分享到微信朋友圈中，如果能够吸引该用户的朋友来帮其解决困难，这就意味着系统增加了一个用户。

**吸引用户持续贡献。**通过微信朋友圈和微博的点赞、评论功能，让用户的其他好友、粉丝来对自己分享的内容进行点赞和评论，能够在一定程度上满足用户的虚荣心。特别是对于微互助平台这种致力于志愿活动的应用来说，分享的内容大多数是帮助了其他人的任务，因此通过这些功能更能够让朋友圈中的其他用户对其行为进行表扬，这种精神上的激励能使用户持续活跃于微互助平台上。

利用第三方平台对微互助平台中的项目进行分享属于在微互助系统外部的激励机制，除此之外，还需在微互助平台的内部构建一个激励机制的相关规则来吸引用户持续为系统做贡献，而传统的积分排行等方式就是十分有效的手段。在微互助平台中采用积分排行的主要原因如下：

**刺激用户保持贡献度。**除了对用户每次发布任务和接受任务给予一定量的积分外，任务的发布者可以添加一定的积分进行任务悬赏，以此鼓励用户接受任务。此外，在微互助平台中存在的排行榜能够根据用户的积分进行排行。如果用户的名次靠前，那么用户在心理上就会有一种满足感。这种精神上的满足能够刺激用户保持接受和发布任务，实现用户在自我价值的实现和虚荣方面的心理需求。此外，一方面对于积分落后的用户可以看到与其他用户的差距从而产生追击和赶超的心态，另一方面对于积分领先的用户也会有一种被追击的紧迫感。对大多数用户来说，积分功能的存在使平台变得游戏化，增加了平台的趣味性和可玩性，从而为了保持甚至提高排名来持续完成平台发布的任务，刺激用户的使用。

**保证项目的可扩展性。**虽然目前积分在激励机制上的作用只是为了展示用户排行榜，但是这绝不可能是微互助平台中将来的唯一激励机制。它可以利用用户对积分的渴求来构建积分商城等方式来使积分物质化，增加用户的黏性、吸引更多用户，甚至积分来可以作为将来用户权限扩展的参考条件。因此现在有必要率先实行积分功能来为将来的服务接口做先期的准备工作。

综上所述，分享和积分排行的功能一内一外分别在一定程度上刺激了用户对微互助平台的使用，二者相辅相成，缺一不可。

在微互助系统中，积分机制的设计主要体现在用户模型（User）以及任务模型（Request）的属性中：

在前文提及的UserModel模型中，存在一个子维度Personality，其中除了用户名、密码、昵称等信息外，还有用户的积分Score，用来维持用户积分的增加和删除，设计时我们将其初始化为100分，一旦用户发布或接受任务就增加2分，此外还有一定的悬赏积分。该悬赏积分就是在Request模型中定义的Character来表示。用于表示某个具体任务的悬赏积分。通过上述贯穿整个用户注册、发布、接受任务等流程来实现我们的积分机制。

### 4.2.2 分享与积分排行的实现

在微互助平台中实现分享功能是利用了友盟+开放平台中的社会化分享模块来进行实现的。考虑到目前各大社交平台种类繁多，我们仅对用户使用率以及占有率进行考虑，因此选择微信、微信朋友圈以及微博作为分享目标。

要在这些社交平台上分享应用的信息，应该首先分别在微信开放平台和微博开放平台上注册微互助这款应用，通过审核后才能拥有分享的权限。

其次，在Android和iOS的代码中加入友盟+SDK的相应开发包并配置相关选项以及URL。以iOS程序代码中配置微信为例，首先必须在程序APPdelegate的入口方法didFinishLaunchingWithOptions中设置WXAppId、appSecret和url。其中WXAppId表示在微信开放平台中申请到的应用ID，appSecret表示该应用的安全密钥，url则是其他微信用户点击该分享后打开的链接，此处填写的是微互助平台在Apple Store上的下载地址。

接着必须在APPdelegate中实现系统回调方法，如果其他用户看到该分享并点击，将自动打开应用或者转向下载的地址。

最后才是在平台中点击分享按钮后使用的方法，首先对任务的接受用户进行判断，如果是分享者接受的任务，则任务的标题名是“我在微助平台又完成了一个任务，快来加入我吧～”，否则是“我在微助平台发布了一个任务，快来帮我完成吧～”，并将其设置为[UMSocialData defaultData].extConfig.wechatSessionData. title，最后才将该定义该方法的具体内容如位置、图片等信息。此处的图片我们设置为用户的头像，具体的代码如下所示。

|  |
| --- |
| [UMSocialData defaultData].extConfig.wxMessageType = UMSocialWXMessageTypeApp;  [[UMSocialDataService defaultDataService]  postSNSWithTypes:@[UMShareToWechatSession]//微信好友  content:titleLabel.text//文本  image:fromView.currentBackgroundImage//头像  location:nil//地址  urlResource:nil  presentedController:self  completion:^(UMSocialResponseEntity \*response){  if (response.responseCode == UMSResponseCodeSuccess) {  NSLog(@"分享成功！");  }  }]; |

而积分排行功能的实现则正如上一小节所说，是通过在注册用户时默认用户积分进行选择，积分的增加和减少则是通过创建任务时让用户进行选择以及后台的服务器自动增加。在微互助平台中显示的排行信息则是有排行、昵称以及积分组成。具体的应用程序截图如图4.3所示。



**图4.3 微互助平台中的分享及积分排行**

## 4.3 小结

本章主要介绍了微互助平台中的激励机制的设计以及实现。第一部分是通知功能，通过分析用户接受到通知的心理活动来设计用户接受到的通知信息，并利用百度云推送API来实现通知的推送功能；第二部分是分享及积分排行功能，通过构建平台内部的激励机制来吸引用户持续做贡献，利用“友盟+”的SDK来实现对微信和微博这两个热门的社交平台上进行分享并在系统内容构建积分系统，完成对用户积分的排行。两个激励机制内外兼容、相辅相成，为微互助平台的激励机制提供强有力的支持。

# 第五章 微互助平台任务推荐策略的制定与实现

前两个章节已经详细描述了微互助平台中的基础，即安全机制和激励机制。本章主要讲述如何利用平台中用户的数据来为用户进行任务推荐，包括相关推荐策略的制定以及实现。

任务推荐策略作为微互助平台的核心内容，其主要功能是根据用户的位置信息，结合偏好以及历史记录，包括注册时填写的信息以及加入的群组、接受过的任务等信息，为用户筛选并推荐最适合的任务。其存在有着以下优点：

首先任务推荐策略能够增加任务的实时性。在微互助平台中加入了任务推荐策略，能够为用户推荐最合适的任务，能够减少任务发布者的等待时间，将任务从被动等待接受改变为主动请求接受，大大提升用户接受任务的概率，增加任务发布者对微互助平台的信心，而不是在发布者久等无响应之后逐渐放弃使用微互助平台，提高了用户对微互助平台的黏性，避免残疾人用户的流失。

其次，能够帮助用户减少点击次数。在微互助平台中虽然存在任务列表、任务筛选等帮助用户查看任务和过滤无用的任务信息等功能，但是用户总是希望自己能花最少的努力就产生最好的结果。因此，通过任务推荐策略，根据用户的历史记录来推荐最符合的任务能够帮助他们略过任务筛选的过程而直接接受任务，为用户提供最贴心的服务，对增加用户黏性起至关重要的作用。

最后，能够降低用户的参与志愿活动的成本。为用户推荐特定的任务最主要的考虑因素是用户与任务之间的距离。系统通过线上发布微任务，根据位置找到就近的相关的志愿者，以实现高效的线下志愿者服务，使用户能够因为顺手而开始参与志愿服务，降低了用户的成本，也在一定程度上加强了用户参与志愿服务的信心，从而吸引更多人参与到系统的志愿活动中，做到全民志愿。

因此，制定相关的任务推荐策略对平台不断发展新用户具有重大意义。在本章将具体讲述关于任务推荐策略中的影响因素、个性化策略以及相关的实现。

## 5.1任务推荐中的关键要素的设计

### 5.1.1 影响任务接受的因素

制定任务推荐策略时，首先需要了解什么是影响用户接受任务的因素，这将有助于系统的设计者充分了解用户对志愿任务的预期以及这些因素对接受任务的影响程度。

为了充分了解用户的心理，我们进行了初步部署，并邀请了积极参与系统的发布任务和接受任务的12名用户进行了一次访问调查。具体的过程和访谈情况将在下一章系统评估中进行详细说明。

经过对12位在部署期间比较活跃用户的采访，我们得到了影响用户接受任务的几大因素，包括如下方面：

**是否有能力完成。**这对于用户来说几乎可以算是决定性的因素，如果用户无法完成该任务，即使接受了也没有任何意义。只有发现自己有能力完成后才会进一步考虑是否接受任务，包括距离等因素，关于这点也要求任务的发布者在填写任务信息的时候将任务描述清楚。此外，如果用户处于忙碌的状态，即使用户与任务地点的距离很近、与发布者的关系再好、任务的类型十分喜欢，但是也几乎不会接受该任务。

**任务的距离。**任务的距离是影响用户是否接受任务的关键点，在使用的过程中，几乎所有用户都表明如果任务的目的地不是很近或者跟自己顺路的话是不会接受该任务的，例如有人表示“*关于距离，（对我来说）一般自行车就是上限了，不喜欢坐公共交通。*”。只有任务的位置与用户的位置超过一个最高接受的范围时才使用户不会对任务产生抗拒心理，能够帮助用户计算提供帮助所需花费的时间、精力等，有助于用户降低自己提供帮助的成本。在经过对参与用户的反复询问，一公里是比较理想的任务距离。

**任务发布者与用户的关系。**任务发布者与接受者的关系对是否接受任务有着相当大的影响。人们往往倾向于帮助自己认识或者熟悉的人，甚至愿意为认识的人做更多的付出来提供帮助。对于自己的老师、上级有些人甚至表示出更愿意接受他们的任务，例如受访者中有人表示“*老师发的任务影响力会更大一点，更倾向于接受。而陌生人发的任务基本不会去领取*”。与之相反的是，在使用中表现出的现象是来自陌生人的任务经常无人接受。因此在实现任务推荐策略的时候有必要将用户与发布者的关系考虑进去。

**用户的偏好。**用户在发起志愿任务的时候都必须填写一个属性——任务类型。而用户在注册的时候也要求填写自己擅长的任务类型。因此，系统中可以针对用户，通过匹配的方式找到该用户是否擅长这个任务，通过查看任务类型来帮助用户了解该任务的成本、时间等内容，一方面能够帮助用户决定是否接受推荐的任务，另一方面由于推荐的任务是自己擅长的内容，能够提高推荐任务的成功率，避免做无用功。

**用户加入的群组。**处于相同群组中的用户拥有近似的任务喜好和朋友圈等。如果用户在同一个群组中，就可以认为他们相识或者有共同的朋友。相比于公开任务和来自陌生人的任务，用户也会更愿意接受来自群组内部发出的任务。这是因为除了兴趣类似之外，还存在群组的用户都是经过其他用户推荐才进入群组，他们之间存在一定的信任关系，因此通过群组而发布的任务也满足了平台的安全机制，也必须将任务的群组作为推荐策略的重要考虑因素之一。

**任务的悬赏分数。**积分排行是微互助平台激励机制的重要表现形式之一，在相同的条件下，用户必然是希望获得的积分越高越好。不过目前微互助平台的积分仅用于排行等功能，仅作为精神上的激励，正如受访者表示“*做任务的时候不看积分，因为目前积分除了排行没有其他用处，我不看重排行*”。因此任务的悬赏只能算是最后考虑的因素，在项目后续的改进中可以增加积分的用处，例如引入积分兑换商城等来加大积分的权重。

上述的几个因素基本上决定了用户是否接受任务。但是有些内容是无法进行量化表示的，比如与用户的关系、是否有能力完成等，这些属于用户的能力与主观问题，因此在进行推荐策略的设计时仅考虑距离、偏好、群组以及分数，这些信息作为用户的上下文信息对任务的推荐策略起着关键的影响。除此之外，无论用户是否接受推荐任务，都会将结果记录下来，用于下一次推荐时进行参考，具体的关系如图5.1所示。



**图5.1 影响推荐的关键因素**

### 5.1.2 关键要素在微互助平台的获取

上述影响用户是否接受任务的关键要素包括距离、用户偏好、群组以及任务分数四项内容。在本小节中将具体描述如何获取这几项内容：

用户与任务之间的距离的获取首先依赖于百度地图API，百度地图API是由百度公司领导开发和设计，使用者能够利用该API接口来调用百度地图的种种工具，包括JavaScript接口、网络服务接口、Android接口、iOS接口、基于位置的云服务等，能够展现基础的地图、用户定位、目标位置搜索、地理位置编码、路线规划、热力图展示等，其目标使用设备包括各大系统的PC端、触控移动端、后台服务器等[47] 。因此，我们利用百度地图API来获取用户实时的位置。

使用百度地图定位的相关SDK时首先需要在在百度地图API的网站上分别申请iOS端和Android端相关应用的密钥，接着在各自的工程文件中配置相应的环境并使用。

以iOS平台中的百度地图开发为例，第一步是配置开发环境，包括在Info.plist 文件中设置NSAppTransportSecurity、导入.framework开发包以及在项目中引入所需的系统库、添加“-ObjC”的Flags设置、加入mapapi.bundle文件、在需要展示百度地图的地方引入相关的头文件等。头文件的内容分为基础定位、云检索、雷达、热力图等，在获取用户位置时我们只需做与定位相关的操作，因此只需添加基础定位头文件BaiduMapAPI/BMapKit.h。

接着，由于需要在首页中展示地图。展示地图需要初始化BMKMapManager，包括以下步骤：

1）在AppDelegate.h文件中添加对BMKMapManager的引用定义；

2）在AppDelegate.m文件的didFinishLaunchingWithOptions函数中添加对BMKMapManager的初始化，并填入申请到的授权Key；

3）在要展示地图的地方创建地图对象BMKMapView；

4）在函数中设置代理。

在展示地图后就是对用户的位置进行定位，同样需要在重写viewDidLoad函数来初始化定位服务，并利用didUpdateBMKUserLocation方法实时更新用户坐标。

经过上述的设置后，用户就能够看到自己在地图上的位置了，同样的道理也能够标识出任务的位置，具体如图5.2所示，图中绿色代表用户的所在的位置，其上的气泡就是系统自动定位的结果，而红色代表附近的任务，同样也可以通过点击来进一步了解任务的详细内容。



**图5.2 微互助平台中的用户位置展示**

在获取用户的位置后，需要得到计算附近一定范围内的任务，因此，我们根据需要得到的范围信息，例如一公里，来获取最大最小的经纬度值。由于地球的形状近似一个球体，关于球面上两点之间的计算公式可以使用Great-circle distance算法[48]。在具体实现时我们将其封装在boundingCoordinates方法中，由此来获取以用户为中心，一公里范围内的最大和最小经纬度。接着，利用这两对最大和最小的经纬度值来筛选数据库中的任务信息，以此为备选任务作为进一步的推荐。

|  |
| --- |
| 算法5.1： 寻找一定距离内的所有任务 |
| 输入：*distance* ：距离  输出：*resultList*：查询到的任务列表  算法过程：   1. GeoLocation coordinates[] = local.boundingCoordinates(distance, GeoLocation.EARTH\_RADIUS\_IN\_METRE);//计算地图上距离一个经纬度点一定距离以内的最大最小经纬度 2. taskFilter.setMaxAndMinLatAndLon;//设置最大最小的经纬度 3. StringBuffer sb = new StringBuffer("finding instance with filter :"); 4. sb.append(" latitude between [" + taskFilter.getMinLat() + "] to [" + taskFilter.getMaxLat() + "] longtitude between [" + taskFilter.getMinLon()+ "] to [" + taskFilter.getMaxLon() + "]");//创建SQL语句 5. criteria.add(Restrictions.between(LATITUDE, taskFilter.getMinLat(), taskFilter.getMaxLat()));//添加参数 6. criteria.add(Restrictions.between(LONGITUDE, taskFilter.getMinLon(), taskFilter.getMaxLon()));//添加参数 7. resultList = findByCriteria(criteria); //查询 8. return resultList |

而其他几个任务相关的推荐维度群组、用户偏好以及任务悬赏金额等内容的获取，则是直接通过数据库的查询函数来分别获取，其中用户的群组信息通过UserGroupRelation类的关系来获取，任务的群组、悬赏以及类型都是属于Request模型中的Character子维度，而用户的偏好则是同样是通过UserExcelRelation类来获取，三者都是通过数据库的查询函数实现的，以查询用户群组为例，具体如下算法5.2所示，其他的查询都是采用类似的方法。

|  |
| --- |
| 算法5.2： 查询用户所属群组 |
| 输入：*userID*：用户的ID  输出：*resultList*：用户加入的群组列表  算法过程：   1. relationList = relationDao.findByUserID(userID)//从数据库中查找所有与该用户相关的群关系列表 2. for relation in relationList do//整理list，返回一个群组的列表 3. groupID = relation.getGroupID() 4. group = groupDao.findByID(groupID) 5. add group to resultList 6. return resultList |

## 5.2 个性化策略的设计与实现

所谓的个性化策略，是针对全局策略相对而言的。全局策略指的是在上文中提及的通知功能，所有的加入特定群组的用户都能收到该群组中其他用户发起任务的通知。但是在实际的应用中，不同的用户对微互助平台的要求不尽相同，他们对接收到的通知内容并不是全盘接受的，他们拥有不同的需求。面对不同用户的个性化需要，在微互助平台中应当通过支持用户的自由设置，做到每个用户都能拥有自己一套的私人订制，保证用户除了全局的通知之外还能拥有一套符合用户偏好的推荐策略。这种做法要求在平台的需求分析阶段就能够将全部内容考虑进去。因此，本节的主要内容就是设计基于用户自主设置的方式来实现支持个性化的策略。

推荐策略的关键要素包括距离、群组、悬赏积分、用户偏好类型等几大维度，当然这些信息有些内容例如任务距离等，是用户在日常生活中不会专门因为任务的原因而专门改变自己目的地的，这也不是微互助平台中任务推荐策略的初衷；或者是如任务的积分悬赏，用户自然会认为越多越好。因此这些因素是由系统来为用户选择的。而用户的喜好和群组则是可以通过由用户自由选择来实现的，这些内容就是用户个性化的主要内容。

除此之外，微互助平台的设计是以用户的需求为中心，因此，用户是否接受推荐、接受推荐的时间间隔是多少也应该交由用户自己来决定。这种方式使用户有取消受推荐的权利，否则一旦用户遇到反感的推荐内容想要加大推荐间隔甚至是关闭推荐时，如果用户找不到关闭的入口，将会义无反顾地卸载应用，导致用户流失。因此就十分有必要在微互助平台中实现这种功能，平台是通过将上述的内容保存在本地的方法来实现的。

将数据存储在本地一方面能够节省用户的流量，只传输一些有用的信息，降低微互助平台对用户流量的损耗，做到随意存取；其次，能够方便快捷地存储和提取信息，众所周知，每个服务器端不可能无缘无故就发送推送信息，而是通过前端的请求来实现的，一旦移动端的轮询速度加快势必由于服务器端的延迟而降低用户体验，因此将存取的开关入口设置在移动端能够加快读取数据的速度，做到真正的随用随取。而这种做法的缺点就是，一旦用户删除应用或是换一个移动端，那系统保存下来的数据将无法迁移到新安装的应用中。在平台采用这种方式的原因是一方面使用者不会很经常换手机或删除应用，另一方面还是利大于弊，用户体验更加重要。

这种方法具体的操作步骤是，在每次用户设置存储之后，利用Xcode自带的本地数据库NSUserDefaults类来将数据保存在本地，保存数据的步骤具体如下：

1）定义一个用户偏好对象；

2）分别设置分数、类型、是否接受推荐以及推荐时间间隔等内容的key-value值；

3）使用同步保存的方法（synchronize）来保证key-value值的成功存储。

在使用存储的时候，同样，也是先定义一个用户的偏好对象userDefaults，再使用objectForKey方法将存储的默认值从本地数据库中取出来放到事先定义好的变量中，并利用预先定义好的Localized()方法来保证在本地保存的数值以正确的语言显示。

因此，微互助平台中默认值设置界面如图5.3所示。



**图5.3 微互助平台中的用户默认值设置**

在上述的默认值设置中，任务类型、悬赏金额、任务对象均代表用户在创建新任务的时候预设的默认值，系统语言则是代表选择系统的语言，目前支持中英两种语言，与上述的Localized()方法相对应，这些都与本章内容推荐机制无关，在此不再赘述。而推荐间隔、是否接受推荐则是取决于用户的选择，默认设计为接受推荐并且推荐时间间隔为60秒。

## 5.3 智能推荐策略的设计与实现

在获取了推荐策略的四大维度后，最主要的就是要实现智能推荐策略。智能推荐策略的实现包括下述几大步骤：

1）前端轮询。上文曾提到，服务器端的设计原则是不可以随意往前端发送信息信。因此，通过移动端每隔一定时间不断对用户的位置信息进行轮询，执行查询用户位置的功能，具体如算法5.3所示。

|  |
| --- |
| 算法5.3： 轮询用户位置 |
| 输入：*无*  输出：*无*  算法过程：   1. if isAcceptRec is true 2. recInterval = get userInterval from userDefault//得到推荐时间 3. if recInterval<=0 4. recInterval = 60//默认为60s 5. end if 6. timer = startTime(recInterval)//开始轮询 7. end if |

上述代码中的isAcceptRec表示用户是否接受推荐，如果不接受推荐将不会对用户位置进行轮询，recInterval表示用户设置的默认推荐时间，如果用户没有设置则默认为60秒。每隔60秒将对用户进行定位。

2）发送定位信息给服务器。在获取用户的经纬度信息后，将代表用户唯一标识符的ID和相关的位置信息发送给服务器。

3）服务器综合推荐任务。服务器端在接收到移动端发送的用户位置信息后，对用户的历史记录进行获取和分析，结合用户的偏好信息和任务的位置信息，匹配与用户最符合的任务，具体的算法5.4所示。

|  |
| --- |
| 算法5.4： 任务推荐算法 |
| 输入：*latitude*:纬度 *longitude*:经度 *userID:*用户ID  输出：*task*:推荐的任务  算法过程：   1. set statusList as 0 which means task not be accepted//设置筛选任务状态 2. taskFilter.setStatusList(statusList) 3. groupList = groupService.getJoinedGroup(userID)//得到用户加入的群组 4. for group in groupList do 5. *publicityList*.add(group.getGroupName()) 6. end for 7. allTaskList = taskService.findByTaskFilter(taskFilter) //未被接收的任务 8. for tempTask in allTaskList//筛选出不是自己发布的任务 9. if(!temp.getUserID().equals(userID) && recommendService.findBy TaskIDAndUserID(temp.getId(), userID)==null) do 10. filteredTaskList.add(temp) 11. end if 12. end for 13. if filteredTaskList.isEmpty() do//如果附近没有任务 14. return null 15. end if 16. else do //如果附近有任务 17. excelList = excelService.findExcelByUserID(userID) //用户擅长类型 18. for excel in excelList do 19. *excelType*.add(excel.getTaskType()) 20. end for 21. recTaskList = taskService.findByRecUserID(userID) //接受过的任务 22. for taskInfo in recTaskList do//用户接受过的任务类型 23. *taskType*.add(taskInfo.getTaskType()) 24. end for 25. for i < filteredTaskList.size() do//计算符合要求任务的分值 26. score = calculateScore(filteredTaskList.get(i), *publicityList*, *excelType*, *taskType*) 27. scoreMap.put(I, score) 28. end for 29. biggestValueInteger = biggestValue(scoreMap)//得到分值最大的任务 30. taskID = filteredTaskList.get(biggestValueInteger).getId() 31. task = taskService.findByTaskID(taskID) 32. taskService.updateTask(task); //更新任务信息 33. return task 34. end else |

在算法5.4中，首先设置taskFilter筛选条件，对要从数据库中获取的任务进行初步筛选，包括任务的类型为“未接受”、查找1000m内的任务、不是该用户发起的任务等。如果存在任务，将用户曾经接受过的任务放到taskType列表中、将用户加入的群组放入publicityList中、擅长的内容放在excelType中。此后，遍历每个符合最初筛选的任务，通过calculateScore方法来计算每个任务的分数，最后服务器返回分数最高的任务作为匹配的结果，并更新该任务的相关字段，将推荐的相关内容在相应的表中保存。

在上述的算法中，不得不提一下calculateScore这个计算任务分数的方法，他的具体内容是，按照四大维度的权重从大到小为分距离、群组、擅长内容以及任务悬赏。只要在一公里内的任务，距离越近分数越高，最高10分，最低0分；一旦该任务属于用户的群组（不是“公开”）则加6分，属于用户擅长内容增加4分，属于用户帮助过的内容增加2分，任务分数越高增加的分数也越多，最高2分，最少0.4分。具体如算法5.5所示。

|  |
| --- |
| 算法5.5： 计算任务分值 |
| 输入：*task*：任务 *groupList*：用户加入的群组 *excelList*：用户擅长类型 *typeList*：用户已接受任务类型  输出：*result* ：任务分值  算法过程：   1. result = (1000.0-task.getDistance())/10.0; //计算距离得分 2. if groupList.contains(task.getPublicity()) do//计算加入群组得分 3. result = result + 6; 4. end if 5. if excelList.contains(task.getType()) do //计算擅长任务类型得分 6. result = result + 4; 7. end if 8. if typeList.contains(task.getType()) do //计算已接受任务类型得分 9. result = result + 2; 10. end if 11. result = result + task.getScore()/2.5//计算任务悬赏得分 12. return result |

4）移动端收到服务器端返回的任务信息后，将给用户弹出一个提示框，用户可以选择是否接受，具体如图5.4所示。



**图5.4 微互助平台中的任务推荐策略**

## 5.4 小结

本章主要介绍了微互助平台中的推荐策略的制定与实现。第一部分通过初步部署，探究了推荐策略的四大维度距离、群组、用户偏好以及任务悬赏积分，分别介绍了它们的作用、与推荐策略的关联和在微互助平台中的获取方式；第二部分通过介绍微互助平台中的个性化策略，来实现用户自由对推荐策略的控制；第三部分则是推荐策略具体实现的整体过程。系统通过整合多维度的影响因素来保证推荐策略的高效、有序进行。

# 第六章 微互助平台系统评估

我们在进行系统的设计前多次与公益组织负责人以及相关的残疾人、志愿者交流，发现在系统设计之中需要注意潜在的安全隐患，加强对系统中激励机制的建设，采用相应的任务推荐策略来对提高志愿活动的效率。前面几个章节分别从原理阐述、模型建立、系统构建等方面详细描述了微互助平台的安全机制、激励机制以及智能推荐策略的设计与实现。但是在实际的使用中是否能够达到预期的目标、解决上述所提及的问题，这需要我们进一步的验证和评估。

首先我们将微互助平台发布在Apple Store（iOS版）以及360应用商店、应用宝和豌豆荚（Android版）中，并且邀请实验室的同学、相应的志愿者、残疾人、公益组织做初步的使用。整个过程中，我们邀请比较活跃的用户进行相应的访谈，结合他们的用户日志信息，来帮助我们发现一些bug，探究激励机制和安全机制在实际应用过程中的使用效果以及用户对推荐策略的反应，验证推荐策略的效率。

整个评估研究过程持续时间超过了半年，在此期间我们根据用户的反馈不断改善界面、修复bug，并记录下用户在此期间的使用日志。整个过程中注册的总用户数超过了400人，其中包括了32位残疾人、12位公益组织的负责人。总共发布了超过200个志愿任务。针对这些用户，我们选取了其中12位比较活跃的用户进行了进一步的采访调查，其中包括两位残疾人和一位公益组织负责人。主要针对的问题分为两类，包括从发布者和接受者两方，具体如下：

1、如果发起过任务请求：

1）是什么样子的任务，当时是什么情况？

2）有没有获得即时的响应？

3）获得响应后，当时是什么感受？没获得，那你怎么办的？

4）回忆一下上次需要帮助的时候，是通过什么样的方式获得的帮助？

5）通过这种方式获得的帮助与微互助有什么不同？

2、如果接受过任务请求：

1）看到微互助平台上的请求有主动提供过帮助吗？

2）是什么情况，为什么？

3）收到过微互助平台帮助的信息吗？收到后是什么反应？

4）是谁发的信息对你有影响吗？

5）位置信息对你有影响吗？

6）除了发布者以及位置信息之外，什么是判断你接不接受请求的因素？

在本章中，我们将根据上述访谈结果，结合全部的用户日志来分析微互助平台中安全机制、激励机制以及智能推荐策略的效果。

## 6.1用户评价

为了了解微互助平台在现有的志愿平台中的水准，我们邀请了20位使用过上海志愿者网、益起邦和微互助平台的用户，针对各大平台的安全性、激励性、任务时效性以及总体体验四大方面进行评分。打分的标准为0-5分，其中0分为最差、5分最好，而且允许小数点后一位，具体的结果如图6.1所示。

**图6.1 各大平台中的模块对比**

从图6.1可以看出，用户对于微互助的总体评价在现有的志愿平台中属于中间的水准，用户在一定程度上可以接受现有的平台，但是仍然存在一定的缺陷，具体分析如下：

首先，关于平台的安全性上，微互助平台明显处于劣势地位，因为志愿者网是一个官方的平台，在发布任务和接受任务上也比较严格，因此在安全性方面存在天然的优势；而益起邦和微互助平台则是由于仅仅采用了手机号码注册来进行验证，而且面向所有人群开放，无法能够做到高级别的安全性保证。

在平台的激励性上，益起邦处于明显的优势，这是由于在益起邦中对用户执行任务采用了物质奖励，因此吸引了更多用户；而微互助和上海志愿者网则是只有无形的奖励，无法对有物质需要的用户产生吸引力。

然而从平台中任务的时效性上看，微互助平台处于一个领先的地位。残疾人用户也表示“*以往我需要帮助需要很长的时间，甚至没人搭理我。但是在这个平台（微互助）中我能够很快就得到了响应，甚至有时候帮助我的人竟然就在小区里面，我感到很高兴，大家都很关心我们*”。因此，微互助平台确实能够帮助残疾人快速解决日常生活中遇到的琐碎事务，做到任务的及时响应和全民志愿。

综上所述，对于平台中的激励机制和安全机制，用户并没有觉得有什么突出之处，甚至还不如现有的其他平台。但是从任务的实效性上看，微互助平台在现有的志愿平台中属于领先的地位，能够为残疾人减少任务发布的时间。因此，对于平台中现有的激励机制和安全机制应当进一步加强，让用户对平台的安全性和激励性更有信心。

## 6.2 推荐策略效果评估

上一节中讲述了用户对系统的安全机制和激励机制相比较于现有的志愿平台并不突出，但是对任务推荐策略比较满意。因此，本节将利用用户的日志信息来具体分析系统中推荐策略的效果。

### 6.2.1 推荐效率评估

在衡量推荐效率上，最简单的方法是查看一个任务从发起到接受所花费的时间，时间越短则代表效率越高。因此，在此可以比较实施任务推荐策略的前后所有被接受的任务从发起到接受花费的时间，根据对比的结果来评估本文所设计和实现的智能推荐策略能否可以在推荐准确性和接受速度上达到期望的结果。上述的时间能够体现出两点内容：一方面代表我们的推荐任务准确性比较高，是符合用户需求和喜好的任务；另一方面减少了用户对平台中任务的查询等操作。

在本文所采取的具体评估方法是针对所有的任务，根据其记录分为系统推荐过和系统没有推荐过的任务，对比各个任务类型平均接受时间耗时，最终的统计结果如图6.2所示。

**图6.2 平均任务接收时间对比曲线图**

正如图6.2所示，我们对上述的图表进行如下分析：

从整体来看，经过智能推荐的任务平均接受时间远少于没有被推荐的任务花费的时间，尽管其用时的减少幅度不尽相同，但都有明显的降低。这可能是由于如下的原因造成的：

对于“拿快递”和“送材料”这一类型的任务，减少时间较多主要是由于在原来没有智能推荐的情况下，大多数用户查看任务的时候发现并没有在自己的周围，而在周围的人可能没有查看到该任务，因此接受的时间较长。而采用智能推荐之后，有目的性的针对任务地点附近的用户进行推荐，因此任务被接受的时间大大减少。

“查资料”减少的幅度最大，其主要原因是由于在实际情况下，用户在发送该类型任务的时候主要的任务地点就是用户的位置，大多是住宅区附近，因此在周围的用户也较多，经过任务推荐策略会进行除了通知之外的推荐，而且这种类型的任务大多比较简单，因此接受的时间也大大减少。

总体上，由上图的曲线以及相应的分析中可以得到在本文提出的智能推荐策略相较于原本的微互助平台是行之有效的方法，大大降低了用户等待任务被接受的时间。

### 6.2.2 推荐准确性评估

上一小结中的推荐效率评估从志愿任务的发起人角度出发，证明了通过微互助平台的智能推荐策略能够降低用户的等待时长。但是从志愿者的角度出发，志愿者不希望收到过多不想接受的任务推荐，受到“推荐轰炸”而对平台失去信任，关于这点是无法通过上述的推荐效率来体现。智能推荐策略最主要的目的是为用户推荐符合用户需求的志愿任务。关于策略是否能够达到预期目标，判断的标准在于志愿者是否有直接接受平台所提供的推荐任务。因此，在此我们利用推荐模型中设计好的is\_accept字段来判断用户是否接受推荐。在此统计了发布智能推荐策略后，根据数据库信息得到的10个用户接收到的推荐和接受推荐任务的次数对比图，具体如图6.3所示。

**图6.3 用户任务推荐接受情况对比曲线图**

根据图6.3的描述可以很清楚的发现智能推荐策略所推荐的任务被接受的概率总体维持在60%左右。根据对相关用户的采访调查，造成上述原因的结果包括下述几点：

首先，本文所设计的推荐算法还比较简单，无法获知用户目前所处的状态。例如用户可能目前比较忙，但是系统向其推荐信息，这样的情况下用户基本是不会接受人物推荐的。

其次，虽然我们将任务的距离作为主要考虑的因素，但是可能用户收到推荐的时候正在走路，而任务的目的地与用户的方向背道而驰，因而可能产生不接受任务的状况。

最后，关于剩余被接受的60%推荐任务，经过访问，在这种情况下的用户或者是处于与任务目的顺路的方向，或者任务的距离与用户十分接近。

综上所述，目前的任务推荐策略达到了60%的准确率，在一定前提下能够为用户提供正确的任务推荐，但是在一些特殊情况下的推荐准确率并不高。因此，在今后的工作中需要全面研究用户的行为状态，充分利用用户的其他上下文信息，例如目的地等信息，来提升推荐的准确率。

## 6.3小结

本章针对所设计和实现的微互助平台是否能够解决传统互助平台中的安全、激励和推荐等问题，通过对平台的部署，对用户进行调查访谈，邀请用户做对比现有的上海志愿者网和益起邦平台来进行针对性的评价，并结合用户的日志信息来分析微互助平台中的安全机制、激励机制及任务推荐策略，通过多方面的对比，探讨了出现各种情况的原因，验证了系统所提出机制和策略的有效性，也探查了相应的缺陷，为后续进一步的提高系统的性能提出了新的改进建议。

# 第七章 总结与展望

## 7.1 总结

本文针对目前传统的志愿者服务平台和互助平台在设计上存在志愿服务结构单一、对残疾人日常生活需求考虑不周全、服务响应不及时和服务不能持久等问题，以志愿者与残疾人的志愿活动为具体的应用场景，在iOS和Android移动端设计并实现了基于位置的微互助平台。论文主要依照发现相关问题、寻求问题的解决思路、确定解决方案、实现设计的步骤，做了以下工作：

（1）阐述了目前在志愿者和残疾人互助上的服务响应时间、考虑范围、服务持久性等内容需要进一步的提升的现状，分析了传统的志愿者服务平台和互助平台的不足的主要原因是没有利用好用户的位置信息，提出实现一个基于位置的线上线下结合的微互助系统，该系统通过线上发布微任务，根据位置找到就近的相关的志愿者，以实现高效的线下志愿者服务。此外，还调查了互助平台的发展。

（2）综合调研了和基于上下文信息的推荐系统发展，接着对推荐系统中存在的安全机制、激励机制以及推荐技术进行梳理和简要分析，进一步说明了搭建基于位置信息的互助系统来解决目前发现的问题能够行之有效。

（3）探究了平台中安全机制的不足，对志愿过程中的残疾人、志愿者、公益组织的角色以及任务、群组、通知、推荐等模型进行优化重构，提出了群组的想法来保证系统用户在线上的隐私安全；提出安全码的功能，通过对比试验研究安全码合适的类型并将其在系统中实现，进一步保证了用户在执行线下志愿活动的安全性。

（4）通过研究和定义通知模型来实现平台中对新消息的推送功能，采用支持分享和积分排行榜的方式来鼓励用户对平台的不断贡献。

（5）总结出影响用户接受任务的四大维度：距离、用户偏好、群组以及任务悬赏分数。针对这些影响用户的上下文信息，设计获取这四大维度的方法，并采用分值计算的方法来实现基于位置任务推荐策略。此外还针对用户的实际需要设计了个性化方案。

（6）通过调查访谈、邀请用户进行评分、分析用户日志信息等方法来对比现有的上海志愿者网、益起邦平台来探究微互助平台的优势和劣势，证明了平台在现有市场上能够具有一席之地，也为后续进一步的提高系统的性能提出了改进建议。

## 7.2 展望

用户体验是所有软件都注重的因素，而本文利用用户的位置信息，通过构建具有安全机制、激励机制和推荐策略的微互助平台，在一定程度上满足了用户对软件的需求和体验。由于受到时间和精力的限制，微互助平台虽然已经在Apple Store以及各大android应用商店上线，但是对于这三个方面上只是做了一个初步的设计和开发，在具体的时候中可能还需要进一步的改进，具体包括：

（1）完善用户界面的整洁性和美观性，可以考虑在项目中的微助地图上开发基于残疾人出行的无障碍地图，发动全社会人群来帮助残疾人来标记附近的无障碍设施，可以进一步增强残疾人用户对微互助平台的依赖。

（2）在安全机制上，可以采用用户认证的功能来进一步加强用户在执行志愿任务的安全性；对用户的权限进行分级，限制不同用户群体之间的对系统的使用权限，从而有利用完善安全机制和激励机制的。

（3）在激励机制上，可以进一步开发积分商城功能来促使用户花费积分，给予用户实际的物质奖励；与相关政府部门合作，将积分转化为相应的志愿服务时间，给予用户政府认可的奖励。

（4）在任务推荐策略上，可以丰富推荐内容，促使推荐的多样化。目前的推荐策略主要是基于用户的历史记录来实现的，因此可以加大这方面的研究，采用混合式的推荐技术，综合基于内容和基于协同过滤推荐技术的特点，帮助用户发现新的兴趣点。此外，还可以丰富影响推荐的因素，例如判断用户的运动目的地等，为用户推荐不仅限于一公里内的任务，进一步提高推荐的效率。

# 参考文献

1. 李瑞昌, 李婧超. 国内外志愿者服务动机稳定性研究述评[J]. 复旦公共行政评论, 2013(2).
2. 中国残疾人联合会[EB/OL]. http://www.cdpf.org.cn/sjzx/cjrgk/201206 /t20120626\_387581.shtml
3. 谭建光. 中国志愿服务:从社区到社会[J]. 广东青年职业学院学报, 2010, 24(3):3-9.
4. 林敬平. 志愿者服务动机调查与激励机制设计[J]. 广东青年干部学院学报, 2008,22:15-17.
5. 刘欣然, 王次农, 刘鸣黄. 对我国高校体育志愿者管理活动特性的研究[J]. 内蒙古体育科技, 2007,21(1):16-18.
6. Bowles S, Gintis H. The evolution of strong reciprocity: cooperation in heterogeneous populations[J]. Theoretical population biology, 2004, 65(1): 17-28.
7. Robert E.Kraut and Paul esnick. Building Successful Online Communities[M]. The MIT Press, 2012.
8. 魏娜. 我国志愿服务发展:成就、问题与展望[J]. 中国行政管理, 2013(7).
9. Salamon L M, Sokolowski S W. Global civil society : dimensions of the nonprofit sector[M]. Kumarian Press, 2004.
10. Barad S. VolunteerMatch.org: balancing mission and earned‐revenue potential[J]. Strategy & Leadership, 2005, 33(2):17-23.
11. 李颖川. 北京2008年奥运会志愿者的组织管理模式与评价体系的研究[M]. 北京体育大学出版社, 2007.
12. Qian S, Wei D. Establishing and Improving the Incentive Mechanism for an Activity of "Youth Volunteers"in Colleges and Universities[J]. Journal of Guangxi Youth Leaders College, 2003.
13. Abowd G D, Atkeson C G, Hong J, et al. Cyberguide: A mobile context-aware tour guide[J]. Wireless networks, 1997, 3(5): 421-433.
14. Averjanova O, Ricci F, Nguyen Q N. Map-based interaction with a conversational mobile recommender system[C]//Mobile Ubiquitous Computing, Systems, Services and Technologies, 2008. UBICOMM'08. The Second International Conference on. IEEE, 2008: 212-218.
15. Ricci F, Nguyen Q N. Mobyrek: A conversational recommender system for on-the-move travelers[J]. Destination recommendation systems: Behavioural foundations and applications, 2006: 281-294.
16. Abowd G D, Atkeson C G, Hong J, et al. Cyberguide: A mobile context-aware tour guide[J]. Wireless networks, 1997, 3(5): 421-433.
17. Yuan J, Zheng Y, Xie X, et al. Driving with knowledge from the physical world[C]//Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011: 316-324.
18. Yuan J, Zheng Y, Zhang C, et al. T-drive: driving directions based on taxi trajectories[C]//Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems. ACM, 2010: 99-108.
19. Zheng Y, Liu Y, Yuan J, et al. Urban computing with taxicabs[C]//Proceedings of the 13th international conference on Ubiquitous computing. ACM, 2011: 89-98.
20. Bennett P N, Radlinski F, White R W, et al. Inferring and using location metadata to personalize web search[C]//Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. ACM, 2011: 135-144.
21. Church K, Smyth B. Who, what, where & when: a new approach to mobile search[C]//Proceedings of the 13th international conference on Intelligent user interfaces. ACM, 2008: 309-312.
22. Baldauf M, Dustdar S, Rosenberg F. A survey on context-aware systems[J]. International Journal of Ad Hoc and Ubiquitous Computing, 2007, 2(4): 263-277.
23. Weiser M. The computer for the 21 st, century[J]. Scientific American, 1991, 265(1):94-104.
24. Hong J, Suh E, Kim S J. Context-aware systems: A literature review and classification[J]. Expert Systems with Applications, 2009, 36(4): 8509-8522.
25. Want R, Hopper A, Falcao V, et al. The active badge location system[J]. ACM Transactions on Information Systems (TOIS), 1992, 10(1): 91-102.
26. Sumi Y, Etani T, Fels S, et al. C-map: Building a context-aware mobile assistant for exhibition tours[M]//Community computing and support systems. Springer Berlin Heidelberg, 1998: 137-154.
27. Cheverst K, Davies N, Mitchell K, et al. Developing a context-aware electronic tourist guide: some issues and experiences[C]//Proceedings of the SIGCHI conference on Human Factors in Computing Systems. ACM, 2000: 17-24.
28. Wan K, Alagar V. Context-Aware Mobile Tour Guide[C]//Computing in the Global Information Technology, 2006. ICCGI'06. International Multi-Conference on. IEEE, 2006: 49-49.
29. Gulliver S R, Ghinea G, Patel M, et al. A context-aware Tour Guide: User implications[J]. Mobile Information Systems, 2007, 3(2): 71-88.
30. Schilit B N, Theimer M M. Disseminating active map information to mobile hosts[J]. Network, IEEE, 1994, 8(5): 22-32.
31. Ryan N, Pascoe J, Morse D. Enhanced reality fieldwork: the context aware archaeological assistant[J]. Bar International Series, 1999, 750: 269-274.
32. Dey A K. Context-aware computing: The CyberDesk project[C]//Proceedings of the AAAI 1998 Spring Symposium on Intelligent Environments. 1998: 51-54.
33. Hull R, Neaves P, Bedford-Roberts J. Towards situated computing[C]//Wearable Computers, 1997. Digest of Papers., First International Symposium on. IEEE, 1997: 146-153.
34. Brown P J. The stick-e document: a framework for creating context-aware applications[J]. ELECTRONIC PUBLISHING-CHICHESTER-, 1995, 8: 259-272.
35. Prekop P, Burnett M. Activities, context and ubiquitous computing[J]. Computer Communications, 2003, 26(11): 1168-1176.
36. Gustavsen R M. Condor–an application framework for mobility-based context-aware applications[C]//Proceedings of the workshop on concepts and models for ubiquitous computing. 2002, 39.
37. Hofer T, Schwinger W, Pichler M, et al. Context-awareness on mobile devices-the hydrogen approach[C]//System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on. IEEE, 2003: 10 pp.
38. Budzik J, Hammond K J. User interactions with everyday applications as context for just-in-time information access[C]//Proceedings of the 5th international conference on intelligent user interfaces. ACM, 2000: 44-51.
39. Finkelstein L, Gabrilovich E, Matias Y, et al. Placing search in context: The concept revisited[C]//Proceedings of the 10th international conference on World Wide Web. ACM, 2001: 406-414.
40. Rani R, Kumar-Lenka S. Motivation and Work Motivation: Concepts, Theories & Researches[J]. International Journal of Research in IT & Management, 2012, 8(2): 12-22.
41. 汪锐, 张艳阳. 良好的激励催生优秀的业绩[J]. 现代企业教育, 2003(10):45-46.
42. Mooney R J, Roy L. Content-based book recommending using learning for text categorization[C]//Proceedings of the fifth ACM conference on Digital libraries. ACM, 2000: 195-204.
43. Su X, Khoshgoftaar T M. A survey of collaborative filtering techniques[J]. Advances in artificial intelligence, 2009, 2009: 4.
44. Burke R. Hybrid web recommender systems[M]//The adaptive web. Springer Berlin Heidelberg, 2007: 377-408.
45. Zanker M, Jessenitschnig M. Case-studies on exploiting explicit customer requirements in recommender systems[J]. User Modeling and User-Adapted Interaction, 2009, 19(1-2): 133-166.
46. 晴天晒雪. APP产品运营：做好消息推送，你必须知道的三件事[EB/OL]. http://www.baijingapp.com/article/4329
47. 158180. 百度地图API[DB/OL]. 百度百科. http://baike.baidu.com/view/3521171.htm
48. Chuntuk. Great-circle distance [DB/OL]. Wikipedia. https://en.wikipedia.org/wiki/Great

-circle\_distance

# 致谢

随着毕业论文的逐步完成，我的研究生学习阶段也马上要结束了。两年多的时间一晃而过，好似昨日才刚刚来到复旦大学一般。当年我还是一个刚刚大学毕业的学生，对学术的向往之心让我选择了加入协同信息与系统实验室，在研究生期间，经历过刚加入实验室的迷茫，走过不少的弯路，但是也收获了许多。在此，我要感谢所有帮助过我的老师和朋友们！

首先，我要感谢我的指导老师丁向华副教授在这段时间一直以来的对我的指导和关心。在进行研究生期间的学术研究和论文撰写过程中，丁老师为我指明了正确的研究方向，经常与我交流并帮助不断改进和完善论文的结构和内容，如果没有丁老师的指导，我是很难完成毕业论文撰写的。丁老师的工作作风潜移默化的影响着我，对我今后的工作和生活有着巨大的帮助，值此机会，我要向丁老师对我的悉心栽培报以最诚挚的谢意！

其次，本次毕业设计能够顺利完成也离不开顾宁教授的帮助。是顾老师让我拥有充足的项目实践机会，并且能够在优越的实验室环境下进行研究、学习和生活，此外，顾老师也对我的毕业论文提供了宝贵的建议。同时，我也要感谢张亮教授、卢暾副教授以及李银胜副教授，感谢他们在论文的开题、中期、预答辩等过程中提出的意见。再次，我还要感谢这两年多里教授过我的所有老师，正是因为他们的辛勤耕耘和耐心培养，才为我本次毕业设计打下坚实的理论基础。

感谢我的朋友们在这段时间给予我的帮助，是你们给予我的关心与支持；感谢实验室所有的师兄弟和同学，是你们让我度过了这美好的研究生时光，因为你们，我的生活充满着快乐，有你们的陪伴，才有我今天的成绩，谢谢你们！

感谢我的家人，在两年多的研究生生活中给我最大限度的鼓励的支持，作为我最坚强的后盾，一直给予我最无私的帮助和鼓励，谢谢你们！在今后的生活中，我将更加努力，刻苦奋斗，绝不辜负你们的期望。

最后，请容许我再一次向所有关心、帮助和支持我的人表示衷心的感谢！

**复旦大学**

**学位论文独创性声明**

本人郑重声明：所呈交的学位论文，是本人在导师的指导下，独立进行研究工作所取得的成果。论文中除特别标注的内容外，不包含任何其他个人或机构已经发表或撰写过的研究成果。对本研究做出重要贡献的个人和集体，均已在论文中作了明确的声明并表示了谢意。本声明的法律结果由本人承担。

作者签名： 日期：

**复旦大学**

**学位论文使用授权声明**

本人完全了解复旦大学有关收藏和利用博士、硕士学位论文的规定，即：学校有权收藏、使用并向国家有关部门或机构送交论文的印刷本和电子版本；允许论文被查阅和借阅；学校可以公布论文的全部或部分内容，可以采用影印、缩印或其它复制手段保存论文。涉密学位论文在解密后遵守此规定。

作者签名： 　 导师签名： 日期：