**“HelpEachOther”**

**分析与设计**

**项目组成员：陈炳炜 11348005**

**林俊浩 11348076**

**彭婷婷 11348103**

**庄 杉 11348166**

**指导老师：衣杨（副教授）**

**版本号：V2.4**

**联系方式：陈炳炜 1085469299@qq.com**

**表1 文档版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本 | 描述 | 作者 |
| 2013/11/2 | V0.1 | 撰写用例图和用例规约 | 陈炳炜、彭婷婷、林俊浩、庄杉 |
| 2013/11/3 | V0.5 | 绘画用例活动图 | 陈炳炜、彭婷婷、林俊浩、庄杉 |
| 2013/11/4 | V0.8 | 撰写问题陈述 | 陈炳炜、彭婷婷、林俊浩、庄杉 |
| 2013/11/5 | V1.0 | 撰写补充规约、术语表 | 彭婷婷、庄杉 |
| 2013/11/6 | V1.1 | 撰写架构描述 | 彭婷婷、庄杉 |
| 2013/11/6 | V1.2 | 绘画类图 | 陈炳炜、林俊浩 |
| 2013/11/6 | V1.3 | 用例分析 | 陈炳炜、林俊浩 |
| 2013/11/13 | V1.4 | 修改用例规约 | 陈炳炜、彭婷婷、林俊浩、庄杉 |
| 2013/11/30 | V1.5 | 修改各用例类图 | 陈炳炜、彭婷婷、林俊浩、庄杉 |
| 2013/12/11 | V2.0 | 总类图 | 陈炳炜 |
| 2013/12/15 | V2.1 | 黑盒测试 | 林俊浩、彭婷婷 |
| 2013/12/15 | V2.2 | 白盒测试 | 陈炳炜、庄杉 |
| 2014/1/2 | V2.3 | 文档全部重新修订使前后统一、文字简练精准 | 庄杉、彭婷婷 |
| 2014/1/4 | V2.4 | 文档审查，提出建议 | 陈炳炜，林俊浩 |

[1. 问题陈述 4](#_Toc377721445)

[1.1 系统涉众 4](#_Toc377721446)

[1.2 功能描述 4](#_Toc377721447)

[2. 需求分析 5](#_Toc377721448)

[2.1 用例析取 5](#_Toc377721449)

[2.2 用例规约 6](#_Toc377721450)

[2.3 补充规约 11](#_Toc377721451)

[2.4 术语表 12](#_Toc377721452)

[2.5 测试用例的设计（注册新账户） 13](#_Toc377721453)

[3. 架构分析 14](#_Toc377721454)

[3.1 架构描述 14](#_Toc377721455)

[3.2 架构图 15](#_Toc377721456)

[3.3 关键抽象 15](#_Toc377721457)

[4. 用例分析 16](#_Toc377721458)

[4.1 注册用例类图 16](#_Toc377721459)

[4.2 登录用例类图 17](#_Toc377721460)

[4.3 任务用例类图 18](#_Toc377721461)

[4.4 总用例类图 19](#_Toc377721462)

[5. 软件测试 20](#_Toc377721463)

[5.1 用例规约(发布任务) 20](#_Toc377721464)

[5.2 黑盒测试 20](#_Toc377721465)

[5.3 白盒测试 21](#_Toc377721467)

[6. 附录 24](#_Toc377721468)

#### 问题陈述

当前大学生中普遍存在着一种现象，一部分人在某些情况下想要获得别人的帮助（例如，因为自己在上课而不能及时去取快递或希望别人顺路帮自己去图书馆还书），但没有合适的渠道来获得帮助；同时，也有一部分人，愿意去帮助他人，并希望从帮助别人的行为中获得回馈（例如：结交更多朋友、在必要的时候也能获得别人的帮助甚至自我价值的实现）。但是现在没有一种有效的方式能够解决以上问题，因此，HelpEachOther就基于这种需求

HelpEachOther（简称HEO）旨在帮助大学生有效解决实际问题，促进大学生的互帮互助行为以实现双赢，扩大加强人际关系，宣扬助人助己的良好品质，建立友好共处的和谐社会。HEO能够让需要帮助的人发布求助信息，让愿意帮助别人的人获取这些信息并提供帮助。此系统还会提供定期更新的榜单来表彰和鼓励那些愿意帮助他人的人。此外，HEO还会提供其他功能来更好地实现此目标。

##### 系统涉众

* **发布者(Poser)：**提交任务的用户
* **接收者(Receiver)：**接收任务的用户
* **管理员(Admin）：**能够注册并登陆系统，拥有高于发布者和接受者的权限，主要是管理系统，处理用户意见的特殊用户
* **用户(User)：**发布者、接收者和管理员的概化
* **系统监控者(SysMonitor)：**对系统进行监控，处理一些简单的事务（例如：回复反馈、发送邮件和任务监控）的特殊用户（一般独立于系统而存在，被一个计算机程序所控制）

##### 功能描述

* **注册(Register)：**用户填写相关信息后注册获得系统合法的账号
* **登陆(Login)：**用户填写用户名和密码来登录系统
* **发布任务(PoseTask)：**发布者发布需要帮助的信息到任务栏，等待接收者进行回应
* **接收任务(ReceiveTask)：**接收者接收已发布的任务
* **查看任务(ViewTask)：**用户查看已有的任务
* **提交任务(SubmitTask)：**接收者提交已完成的任务
* **更改任务(ChangeTask)：**发布者删除任务，取消发布任务者；接收者取消接收任务
* **私信(Message)：**用户和用户可通过私信功能进行交流
* **管理任务(AdminTask)：**管理员管理已发表的任务、处理投稿和举报
* **查看榜单(ViewRankingList)：**用户查看好人点排行榜
* **投稿(Submiss)：**用户提交好人好事通讯稿
* **举报(Prosecute)：**用户向系统举报某个任务或某个用户
* **反馈(Feedback)：**用户可以将系统的不足反馈给开发人员

#### 需求分析

##### 2.1 用例析取

对该系统的用例进行析取，得到如下所示的用例图：

****

图1HelpEachOther用例图

##### 2.2 用例规约

* **注册用例规约**

**主事件流：**

1. Poser/Receiver/Admin选择注册
2. Poser/Receiver/Admin提供个人信息：账户（Account）、昵称（Nickname）、密码（Password）
3. 完成注册

**备用事件流：**

3a. 昵称（Nickname）已被注册，Poser/Receiver/Admin更换Nickname直到未被注册

3b. 账户（Account）已被注册

1. Poser/Receiver/Admin直接登录（进入登陆规约）

2. 更换账户（Account）继续注册



图2 注册用例活动图

* **登陆用例规约**
* **发布任务用例规约**

**主事件流：**

1. 发布者执行发布任务

2. 发布者提供任务详情[**注：**填写任务类型(TaskType)、任务标题(TaskTitle)、任务描述(TaskDescription)、有效时间(AvaliableTime)、完成时间(AccomplishTime)，选择接收者的约束条件(ReveiverConstrain)，填写支付好人点(good person point)的数量和备注(Note)]

3. 发布者成功提交任务

4. 系统将任务添加至任务列表，并将任务状态标记为“新发布”。

**备选事件流：**

3a. 发布者所提供的任务标题、有效时间、完成时间为空，系统提示发布者任务发布失败

3b. 发布者所提交的有效时间早于系统时间，系统提示发布者任务发布失败

3c. 发布者所提交的完成时间早于有效时间，系统提示发布者任务发布失败

3d. 发布者所提供的好人点小于发布者所拥有的好人点，系统提示发布者任务发布失败



图3 发布任务用例活动图

* **接收任务用例规约**

**主事件流：**

1. 接收者选择接收任务

2. 系统给发布者发出任务被成功接收的提示

3. 系统给接收者发出无接收成功的提示

4. 系统将任务标记为“已接收”

****

图4接收任务用例活动图

* **查看任务用例规约**

**主事件流：**

1. 用户选择查看任务功能

2. 系统显示按时间排序的任务列表

3. 用户选择筛选功能

4. 系统显示筛选输入页面

5. 用户输入任务类型(TaskType)、任务关键词（TaskKey）、任务状态（TaskStatus）

6. 系统按照筛选条件显示任务列表

7. 用户选择查看某一任务信息

8. 系统显示该任务的详细信息，包括发布者的信息，任务的有效时间(AvaliableTime)、作务的完成时间（AccomplishTime）、任务悬赏GPP(Good Person Point)、任务状态（TaskStatus）

**备选事件流：**

6a. 用户的输入不符合要求，系统提示输入不符合要求



图5 查看任务用例活动图

* **提交任务用例规约**

**主事件流：**

1. 接收者提交任务

2. 系统向发布者发出消息询问任务是否完成

3. 发布者向系统确认任务已被完成

4. 系统把发布者任务悬赏好人点扣至接收者帐号中

5. 系统标记任务状态为“已完成”

**备选事件流：**

3a. 发布者选择任务未完成，本用例结束

3b. 发布者在10分钟无回应，系统默认任务已被完成



图6 提交完成的任务用例活动图

* **更改任务用例规约**
* **私信用例规约**
* **管理任务用例规约**
* **查看榜单用例规约**
* **投稿用例规约**
* **举报用例规约**
* **反馈用例规约**

##### 补充规约

* **简介：**该补充规约的目的是列出系统的非功能性需求，结合用例规约反映系统的整体需求。
* **功能性：**无
* **可用性：**用户界面与各种内核的浏览器实现兼容
* **可靠性：**本系统在每周七天，每天二十四小时内均可使用，宕机时间应少于1%
* **性能**

1. 在任意既定时刻，系统最多可支持 2000 名用户同时使用中央数据库，并在任意时刻最多可支持 500 名用户同时使用本地服务器；

2．系统必须能够在 3秒钟内对用户的请求作出反应。

* **可支持性：**无
* **设计约束：**系统的用户界面应基于主流的网页浏览器（如IE、Firefox、Chrome）进行设计开发。
* **安全性**

1. 系统必须能防止Poser/Receiver查看或修改他人的注册信息；

2. 系统只允许管理员（Admin）处理用户任务。

##### 术语表

**表2术语表**

|  |  |  |
| --- | --- | --- |
| **术语** | **英语名词** | **定义和信息** |
| 发布者 | Poser | 提交任务的用户 |
| 接受者 | Receiver | 接受任务的用户 |
| 管理员 | Admin | 注册并登陆系统，拥有高于发布者和接受者的权限，主要是管理系统，处理用户意见的特殊用户 |
| 用户 | User | 发布者、接收者和管理员的概化 |
| 系统监控者 | SysMonitor | 对系统进行监控，处理一些简单的事务（例如：回复反馈、发送邮件和任务监控）的特殊用户（一般独立于系统而存在，被一个计算机程序所控制） |
| 任务 | Task | 用户提交的需求，包含详细描述 |
| 有效时间 | Available Time | 任务能被接受的最后期限 |
| 完成时间 | Accomplished Time | 任务必须被完成的最后期限 |
| 任务状态 | Status | 包含：  1新发布：任务没有Receiver，时间为超过任务的有效时间  2已接受：任务有Receiver，未完成，时间未超过任务的有效时间  3已完成：任务有Receiver，任务已完成，时间未超过任务的完成时间  4已过期：任务没有Receiver，时间超过任务的有效时间  5超期未完成：任务有Receiver，任务未完成，时间超过任务的完成时间 |
| 好人点 | GPP(Good Person Point) | 用户在系统中使用的虚拟货币，可当作任务的悬赏和其他用途 |

##### 测试用例的设计（注册新账户）

1. 前置条件：Poser/Receiver/Admin选择注册选项
2. 设计
3. 步骤---选择注册
4. 检验点---注册页面正确显示
5. 步骤---提供个人信息：账户（Account）、昵称（Nickname）、密码（Password）
6. 检验点---个人信息符合要求
7. 步骤---完成注册
8. 检验点---用户个人信息被正确添加进系统数据库
9. 后置条件---用户成功注册用例之后，能够登陆系统
10. 可接受标准---所有检验点必须成功

#### 架构分析

##### 架构描述

本系统采用ThinkPHP框架，ThinkPHP是一个快速、简单的基于MVC和面向对象的轻量级PHP开发框架，遵循Apache2开源协议发布，从诞生以来一直秉承简洁实用的设计原则，在保持出色的性能和至简的代码的同时，尤其注重开发体验和易用性，并且拥有众多的原创功能和特性，为WEB应用开发提供了强有力的支持。

ThinkPHP框架的目录结构为：

├─ThinkPHP.php 框架入口文件

├─Common 框架公共文件

├─Conf 框架配置文件

├─Extend 框架扩展目录

├─Lang 核心语言包目录

├─Lib 核心类库目录

│├─Behavior 核心行为类库

│├─Core 核心基类库

│├─Driver 内置驱动

││├─Cache 内置缓存驱动

││├─Db 内置数据库驱动

││├─TagLib 内置标签驱动

││└─Template 内置模板引擎驱动

│└─Template 内置模板引擎

└─Tpl 系统模板目录

* **视图层：**视图层能够实现数据有目的的显示。在视图中一般没有程序上的逻辑。为了实现视图上的刷新功能，需要调用控制层来获取数据。在本系统中，视图层指的是在浏览器客户端所看到的网页。
* **控制层：**控制器起到不同层面间的组织作用，用于控制整个应用的流程。它处理事件并作出响应。“事件”包括用户的行为和数据模型上的改变。在本系统中，控制层指的是后台起到控制作用的php程序。
* **实体层：**实体层存储着用户和系统的信息，在本系统中，实体层由用户信息(UserInfo)、任务信息(TaskInfo)、私信信息(MessageInfo)等组成。

##### 架构图

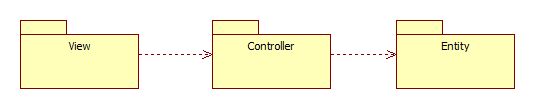
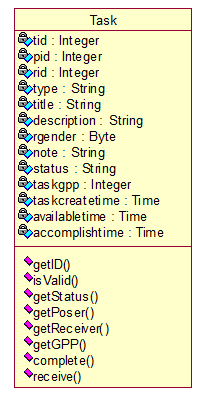
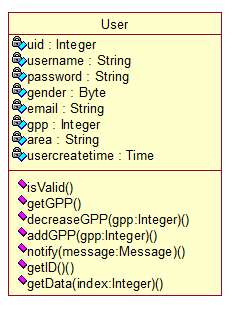


图7架构图

##### 关键抽象



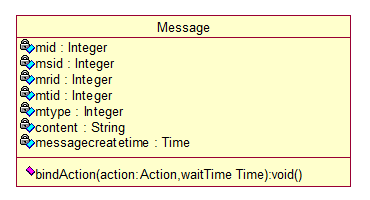


图8实体类

#### 用例分析

##### 注册用例类图

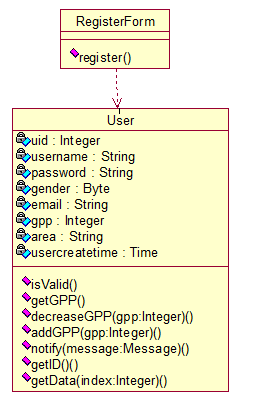


图9注册用例类图

##### 登录用例类图

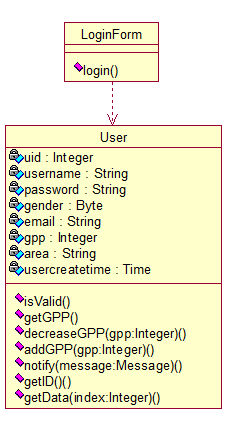


图10登录用例类图

##### 任务用例类图



图11 任务用例类图

##### 总用例类图

****

图12 总用例类图

#### 软件测试

##### 用例规约(发布任务)

**主事件流：**

1. 发布者执行发布任务

2. 发布者提供任务详情[注：填写任务类型(TaskType)、任务标题(TaskTitle)、任务描述(TaskDescription)、有效时间(AvaliableTime)、完成时间(AccomplishTime)，选择接收者的约束条件(ReveiverConstrain)，填写支付好人点(good person point)的数量和备注(Note)]

3. 发布者成功提交任务

4. 系统将任务添加至任务列表，并将任务状态标记为“新发布”。

**备选事件流：**

3a. 发布者所提供的任务标题、有效时间、完成时间为空，系统提示发布者任务发布失败

3b. 发布者所提交的有效时间早于系统时间，系统提示发布者任务发布失败

3c. 发布者所提交的完成时间早于有效时间，系统提示发布者任务发布失败

3d. 发布者所提供的好人点小于发布者所拥有的好人点，系统提示发布者任务发布失败

##### 黑盒测试

## 等价分类法

**表3 等价分类表**

|  |  |  |
| --- | --- | --- |
| 输入数据 | 有效等价类 | 无效等价类 |
| 任务标题 | 1. 任意文本 除纯空格外 | 1. 纯空格,如: “” |
| 任务详情 | 1. 任意文本 |  |
| 任务的有效时间 | 1. 大于当前时间的任意时间, 格式为:2013-12-10 | 1. 非法输入,如:ab 2. 格式不对,如2013.12.10 3. 小于当前时间 |
| 任务的确认时间 | 1. 大于任务的有效时间,格式如上. | 同5,6   1. 小于任务的有效时间 |
| 指定Receiver的限定范围  (1为东校,2为南校,3为北校) | 1. 1,2或3 | 1. 非法输入,如:a 2. 越界输入,如:5 |
| GPP | 1. 大于等于0, 小于等于 Poser当前GPP,的整数 | 1. 非法输入,如:a 2. 越界输入,如:-2,1000000 |

##### 边界值分析法

**图4 边界值分析表**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入等价类 | 测试用例说明 | 测试数据 | 期望结果 | 选择理由 |
| GPP | 1. 非数字 2. 小数 3. 负数 4. 大于Poser的GPP 5. 恰为Poser的GPP 6. 0 | 1. a 2. 0.1 3. -1 4. Poser.GPP+1 5. Poser.GPP 6. 0 | 1. 无效 2. 无效 3. 无效 4. 无效 5. 输入有效 6. 输入有效 | 1. 第一个小写字母 2. 接近0的小数 3. 最大负整数 4. 恰好大于Poser.GPP的整数 5. 最大有效输入 6. 最小有效输入 |

## 错误猜测法

1. 在标题或详情上输入含html标记的文本,会不会显示出问题,如: ><a href="">hello</a>< 或 “><a href="">hello</a><” (含引号)
2. GPP输入的值过大,如4294967296 (2^32)

##### 5.3白盒测试

* **路径测试法**

****

图13 程序流程图

将程序流程图转换为程序图如下：

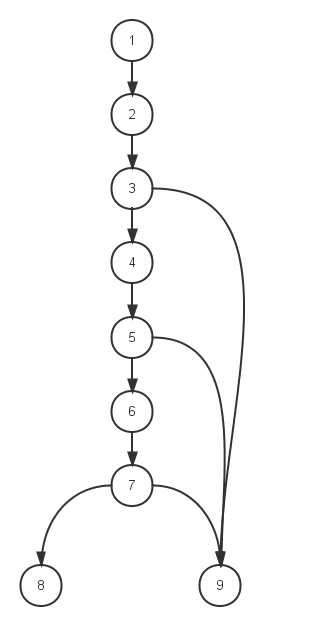
v

图14程序图

1. 圈复杂度

圈复杂度，即程序图中区域的数量，可根据图4-2知圈复杂度为3。

1. 导出测试用例

根据图2-2可得出四个独立的路径：

路径1：1-2-3-4-5-6-7-8

路径2：1-2-3-9

路径3：1-2-3-4-5-9

准备测试用例

测试数据格式=[现在的时间剩余GPP availabletime accomplishtime 支付GPP]

①路径1：1-2-3-4-5-6-7-8

输入：value=[2013-11-25 21:30,20,2013-11-25 22:00,2013-11-25 23:00,15]

预期结果：提交成功

②路径2：1-2-3-9

输入：value=[2013-11-25 21:30,20,2013-11-25 21:00,2013-11-25 23:00,15]

预期结果：提交失败

③路径3：1-2-3-4-5-9

输入：value=[2013-11-25 21:30,20,2013-11-25 22:00,2013-11-25 21:45,15]

预期结果：提交失败

#### 附录

* **工程源码**（**注：**本项目已通过版本控制工具git上传到代码仓库github，可以访问github上saucebing账号下的heo\_test项目并从最新的分支中获取最新的源码）
* **HelpEachOther项目成果展示视频**
* **HelpEachOther项目感想**