体系结构设计文档

161250049 金鑫

目录

1. 引言

1.1编制目的

1.2词汇表

1.3参考文献

1. 关注点

2.1功能需求、接口与约束

2.2质量属性

2.3项目环境

2.4商业目标

1. 体系结构需求定义

3.1体系结构需求与约束

3.2用例视图

3.3场景描述

1. 建立初始体系结构

4.1建立依据

4.2逻辑视图

4.3开发视图

4.4进程视图

4.5部署视图

1. 设计决策

5.1设计决策1

5.2设计决策2

1. 最终高层结构

6.1逻辑视图

6.2开发视图

6.3进程视图

6.4部署视图

1. 体系结构风格

7.1MVC风格

7.2管道过滤器风格

7.3隐式调用风格

7.4面对对象风格

1. 引言

1.1编制目的

本文档提供了同城互助系统的体系结构及高层次的设计方案。描述了关注点，体系结构需求定义，设计决策，高层结构等内容，指导后续的详细设计和开发工作，同时作为项目开发活动的一个重要依据。

1.2词汇表

|  |  |
| --- | --- |
| 缩略语或术语 | 解释 |
| FR | 功能需求 |
| QR | 质量属性 |
| C | 约束 |
| ENV | 项目环境 |
| BG | 商业目标 |
| AR | 体系结构需求 |
| DC | 设计约束 |
| S | 场景 |
| CS | 候选方案 |

1.3参考文献

《同城互助系统用例文档》

《同城互助系统需求规格说明文档》

1. 关注点

2.1功能需求、接口与约束

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 关注点 | 描述 | 灵活性 |
| FR1 | 个人信息管理 | 用户可以提交个人信息，注册用户，并且可以管理个人信息 | 基本无灵活性 |
| FR2 | 发布任务 | 用户可以在系统中发布任务 | 具有一定灵活性 |
| FR3 | 领取任务 | 用户在任务大厅浏览任务信息，符合任务要求的用户可以选择合适的任务并领取 | 要求筛选具有一定灵活性 |
| FR4 | 领取任务 | 用户可以领取相应的任务 | 基本无灵活性 |
| FR5 | 任务结算与评价 | 任务完成后，双方需确认完成，提交反馈评价 | 评价方法具有灵活性 |
| FR6 | 资质审核 | 用户上传专业技能资格证明应经过系统管理员审核 | 基本无灵活性 |
| FR7 | 投诉处理 | 管理员可以对投诉信息进行处理 | 投诉处理方法具有灵活性 |
| FR8 | 任务列表管理 | 管理可以对上传的违规任务进行删除，并对发布用户进行相应处理 | 具有一定灵活性 |
| FR9 | 智能推荐 | 系统根据用户信息预测适合用户的任务，使得用户可以快捷地找到适合自己的任务 | 具有灵活性 |
| C1 | 符合法律法规（约束） | 软件必须符合有关法律法规 | 不具备灵活性 |

2.2质量属性

2.3项目环境

2.4商业目标

1. 体系结构需求定义

3.1体系结构需求与约束

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 需求ID | 体系结构需求描述 | 优先级 | 设计约束定义 | | 相关约束 |
| ID | 约束内容 |
| AR1 | 个人信息管理 | 5 | DC1 | 用户初次使用应用时，需要对用户进行短暂的操作示例教程 |  |
| DC2 | 系统采用移动端（ iOS 版本和 Android 版本）开发 |  |
| AR2 | 发布任务 | 5 | DC3 | 在1000人并发使用时，系统不能崩溃 |  |
| AR3 | 领取任务 | 5 | DC4 | 在进行数据的下载和上传中，如果网络出现故障，系统不能出现故障 |  |
| AR4 | 领取任务 | 3 | DC5 | 广告的植入可以有多种方式 |  |
| AR5 | 任务结算与评价 | 4 | DC6 | 要方便系统管理员进行查看各购物网站的信用信息，方便管理 |  |
| AR6 | 资质审核 | 5 | DC7 | 系统发生异常要在10分钟内通知系统管理员 |  |
| DC8 | 系统管理员进行人工存储要能够在1小时内完成 |  |
| AR7 | 投诉处理 | 4 | DC9 | 投诉处理结果要反馈给投诉人和被投诉人 |  |
| AR8 | 任务列表管理 | 3 | DC10 | 使用网络爬虫技术，每天自动从网上更新任务信息 |  |
| AR9 | 智能推荐 | 2 | DC11 | 推荐结果根据用户习惯得出 |  |
| AR12 | 可靠性 | 5 | DC17 | 存储出现故障要能够快速恢复，恢复应在半个小时内完成 |  |
| DC18 | 采取RAID机制进行数据备份 | 促进DC17 |
| DC19 | 故障频率每月不能超过3次 |  |
| DC20 | 服务器出现故障时要能够及时发现并通知系统管理员，并且不能影响网站访问 | 促进DC17 |
| AR13 | 兼容性 | 2 | DC21 | 兼容各种浏览器，如firefox, chrome, internet explorer |  |
| DC22 | 兼容各种价格单位和商品别名(建立词汇表) |  |
| AR14 | 容量 | 3 | DC23 | 最大要能够存储1千万个商品的相关信息 | 阻碍DC31 |
| DC24 | 系统数据分布式存储在多台设备上 | 阻碍DC17 |
| AR15 | 并发性 | 3 | DC25 | 在高峰期时间（例如节假日），要允许5百万用户同时在线使用 |  |
| AR16 | 安全性 | 5 | DC26 | 实现短时间内频繁访问的IP（即插件扫描攻击）并屏蔽 |  |
| DC27 | 发现特定IP地址（例如搜索引擎Agent）的访问并屏蔽 |  |
| AR17 | 可修改性 | 3 | DC28 | 支持增加新的入侵方式侦测 |  |
| DC29 | 支持设计新的审查条件，并在满足条件的情况下通知相应人员 |  |
| DC30 | 增加一种新的侦测算法或功能所需时间不能超过2.5个人月 |  |
| AR18 | 效率 | 4 | DC31 | 系统响应时间不能超过1.5s |  |
| AR18 | 准确性 | 5 | DC32 | 系统出错概率不能超过%0.01 |  |
| AR19 | 快速开发 | 3 | DC33 | 第1个版本2个月内上线  第2个版本4个月后上线 |  |

3.2用例视图

3.3场景描述

1. 建立初始体系结构

4.1建立依据

4.2逻辑视图

4.3开发视图

4.4进程视图

4.5部署视图

1. 设计决策

5.1设计决策1

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR1 个人信息管理  DC1 用户初次使用应用时，需要对用户进行短暂的操作示例教程  DC2 系统采用移动端（ iOS 版本和 Android 版本）开发 | |
| 候选方案 | CS1.1 | 个人信息设置在本机APP上，通过登陆验证后显示在信息管理页面，可自由修改 |
| CS1.2 | 通过第三方api将信息与微信或QQ或手机号绑定，通过第三方验证后方可查看并自由修改。 |
| 选择选项 | CS1.2 | |
| 选择理由 | 加入第三方绑定可以增强数据安全性，同时便于第一次给用户引导教程。 | |
| 影响 | 开发视图 | |
| 详细设计约束 | 1.需要增加第三方数据接口以及相应的数据交换实现方法  2.采用适配器模式，将相应数据转换格式后进行交互。 | |
|  | | |

5.2设计决策2

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR2 发布任务  DC3 在1000人并发使用时，系统不能崩溃 | |
| 候选方案 | CS2.1 | 为正在发布中的任务以HTTP请求来传递消息，需要手动刷新或等待系统自动更新来触发信息同步。 |
| CS2.2 | 通过创建会话来传递数据，保证实时数据通讯与同步。 |
| CS2.3 | 仅对部分内容使用长连接（任务被接取的消息提醒），其余部分保留为基础HTTP请求。 |
| 选择选项 | CS2.3 | |
| 选择理由 | 保证效率的同时能够提高并发性能，减轻服务器压力 | |
| 影响 | 逻辑视图、开发视图 | |
| 详细设计约束 | 1.采用webSocket双向通信 | |
|  | | |

5.3设计决策3

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR3 领取任务  DC4 在进行数据的下载和上传中，如果网络出现故障，系统不能出现故障 | |
| 候选方案 | CS3.1 | 在用户界面设置限制，当系统欠响应时短时间内不允许连续领取任务 |
| CS3.2 | 通过后台增加限制，同域请求同一时间只受理一个请求。 |
| 选择选项 | CS3.2 | |
| 选择理由 | 保证系统故障率下降，尽管可能因为网络问题丢包，但一定程度上可以抵御恶意攻击 | |
| 影响 | 进程视图、逻辑视图 | |
| 详细设计约束 | 1.增加过滤器，拦截未验证他域请求  2.给respose方法增加逻辑约束，使得每次至多只有一个同域请求占据系统性能 | |
|  | | |

5.4设计决策4

5.5设计决策5

5.6设计决策6

5.7设计决策7

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR7 投诉处理  DC9 投诉处理结果要反馈给投诉人和被投诉人 | |
| 候选方案 | CS7.1 | 开发一个反馈模块，每次处理自动调用该模块 |
| 选择选项 | CS7.1 | |
| 选择理由 | 成本低，保障了系统效率 | |
| 影响 | 逻辑视图，进程视图 | |
| 详细设计约束 | 1.使用隐式调用的方法  2.保证审查策略可灵活变更，具备可扩展性 | |
|  | | |

5.8设计决策8

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR8 任务列表管理  DC10 使用网络爬虫技术，每天自动从网上更新任务信息 | |
| 候选方案 | CS8.1 | 为了满足DC13,可以使用适配器模式对爬虫爬取的不同格式数据进行处理 |
| CS8.2 | 可以将爬虫封装成一个独立模块，里面有网页提取器（可以自己定制网页解析规则），爬虫任务生产器（生产URL）等，可以自己定制爬取策略和配置文件，爬虫策略由解析配置文件决定 |
| 选择选项 | CS8.2结合CS8.1 | |
| 选择理由 | CS8.2能够保证实现对数据的规范化处理  CS8.1可以实现爬取策略内容的灵活可变更 | |
| 影响 | 逻辑视图、开发视图、进程视图、部署视图 | |
| 详细设计约束 | 1.需要按一定规则写配置文件（例如xml），配置文件描述爬取策略的接口  2.适配器中定制数据统一规则  3.模块内部爬虫技术使用策略模式 | |
|  | | |

5.9设计决策9

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR9 智能推荐  DC11 推荐结果根据用户习惯得出 | |
| 候选方案 | CS9.1 | 对任务添加多个属性描述其特性 |
| CS9.2 | 建立过滤器类，利用策略模式实现 |
| 选择选项 | CS9.2结合CS9.1 | |
| 选择理由 | 1.和设计决策1的Sorter类似，可以为设计提供经验  2.增加新的屏蔽策略不必修改以前的代码，具有更好的灵活性和可扩展性 | |
| 影响 | 逻辑视图、开发视图 | |
| 详细设计约束 | 1..封装屏蔽规则，使用策略模式 | |
|  | | |

5.10设计决策10

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR12 可靠性  DC17 存储出现故障要能够快速恢复，恢复应在半个小时内完成  DC18 采取RAID机制进行数据备份  DC19 故障频率每月不能超过3次  DC20 服务器出现故障时要能够及时发现并通知系统管理员，并且不能影响网站访问  AR14 容量  DC23 最大要能够存储1千万个任务的相关信息  DC24 系统数据分布式存储在多台设备上  AR15 并发性  DC25 在高峰期时间（例如节假日），要允许5百万用户同时在线使用 | |
| 候选方案 | CS5.1 | 使用分布式数据存储，对数据进行热备份，可以提供数据查询存储效率，并且一个物理节点故障，系统可以使用备用节点。可以通过其他数据库进行数据恢复，不会影响操作。 |
| CS5.2 | 为数据恢复和数据备份建立独立的模块 |
| CS5.3 | 对数据库进行一个备份，故障时人工切换备份 |
| 选择选项 | CS5.1结合CS5.2 | |
| 选择理由 | CS5.1可以一举多得的解决对于系统可靠性、并发性、容量的需求，可以显著提高数据库性能  CS5.3需要产生额外的人工费用，且容错度很低 | |
| 影响 | 逻辑视图、开发视图、进程视图、部署视图 | |
| 详细设计约束 | 1. 要建立数据访问的分配规则，并且提供统一的访问接口 2. 使用分布式数据存储，屏蔽物理节点差异 3. 使用多个备用节点，保证数据可恢复 4. dataBackup和dataRevovery内部实现策略模式 | |
|  | | |

5.11设计决策11

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR15 并发性  DC25 在高峰期时间（例如节假日），要允许5百万用户同时在线使用  AR16 安全性  DC26 实现短时间内频繁访问的IP（即插件扫描攻击）并屏蔽  DC27 发现特定IP地址（例如搜索引擎Agent）的访问并屏蔽 | |
| 候选方案 | CS6.1 | 采取分布式存储数据的方法，将数据保存在多台服务器，提高系统并发性能 |
| CS6.2 | 实现负载均衡，合理安排访问请求，保证服务器资源能够有效利用，提高访问效率 |
| CS6.3 | 由于负载均衡系统需要将请求分派，所以会随时得到各个服务器的状态，则依据此使用Message通信机制进行故障检测。 |
| 选择选项 | CS6.1结合CS6.2，CS6.3 | |
| 选择理由 | 1.能够提供服务器访问性能，多台服务器可以避免服务器故障的影响。  2.分布式处理能够协调对任务的处理  3.增加服务器等方式来应对高峰吞吐量。 | |
| 影响 | 逻辑视图、开发视图、进程视图、部署视图 | |
| 详细设计约束 | 1. 封装负载均衡，屏蔽实现细节 2. 分布式存储数据 3. 多台服务器 | |
|  | | |

5.12设计决策12

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR12 可靠性  DC17 存储出现故障要能够快速恢复，恢复应在半个小时内完成  DC18 采取RAID机制进行数据备份  DC19 故障频率每月不能超过3次  DC20 服务器出现故障时要能够及时发现并通知系统管理员，并且不能影响网站访问 | |
| 候选方案 | CS7.1 | 使用ping/echo验证服务器状态，进行故障检测 |
| CS7.2 | 使用心跳机制，即各个服务器定期发送心跳，进行故障检测 |
| CS7.3 | 由于负载均衡系统需要将请求分派，所以会随时得到各个服务器的状态，则依据此使用Message通信机制进行故障检测。 |
| CS7.4 | 开发一个Monnitor模块，通过message检测ServerStatus |
| 选择选项 | CS7.1结合CS7.2,CS7.3 | |
| 选择理由 | 这样选择可以帮助ServerControl中负载均衡的处理，CS7.1，CS7.2方式成本低，且能够迅速检测服务器故障 | |
| 影响 | 逻辑视图，开发视图 | |
| 详细设计约束 | 1. 服务器端要实现健壮的代码，多使用错误处理机制 2. 建立日志模块，可以查询服务器端错误 3. 建立一个服务器管理来实现ping/echo,心跳机制 | |
|  | | |

5.13设计决策13

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR16 安全性  DC26 实现短时间内频繁访问的IP（即插件扫描攻击）并屏蔽  DC27 发现特定IP地址（例如搜索引擎Agent）的访问并屏蔽 | |
|  | CS8.1 | 封装一个攻击检测模块，用来检测攻击，用事件实现隐式调用 |
| CS8.2 | 实现一个联系人列表，用于侦测异常后通知 |
| CS8.3 | 实现一个防火墙模块，对于攻击 过滤和防御 |
| 选择选项 | CS8.1结合CS8.2 CS8.3 | |
| 选择理由 | 1.封装成独立模块具有可扩展性  2.使用隐式调用的方式通知相关人员 | |
| 影响 | 逻辑视图、开发视图、进程视图、部署视图 | |
| 详细设计约束 | 1.使用隐式调用的方式通知相关人员  2.将攻击防御和攻击检测分离  3.模块内部使用策略模式 | |
|  | | |

5.14设计决策14

|  |  |  |
| --- | --- | --- |
| 需求&约束 | AR13 兼容性  DC21 兼容各种浏览器，如firefox, chrome, internet explorer | |
| 候选方案 | CS9.1 | 采用适配器模式对于不同的格式进行处理 |
| CS9.2 | 使用bootstrap之类的框架实现，调用固有的api |
| 选择选项 | CS9.2结合CS9.1 | |
| 选择理由 | 利用bootstrap框架能实现大部分兼容。对于不兼容的部分自己实现兼容性 | |
| 影响 | 逻辑视图、开发视图 | |
| 详细设计约束 | 1使用bootstrap等前端框架实现浏览器兼容  2.对于不兼容部分自己实现兼容性 | |
| @JF$YMX6Z]~FDCD_VQ)ANPX | | |

1. 最终高层结构

6.1逻辑视图

6.2开发视图

6.3进程视图

6.4部署视图

1. 体系结构风格

7.1MVC风格

7.2管道过滤器风格

7.3隐式调用风格

7.4面向对象风格