

用户需求说明书

香港卫安运营服务平台

项 目 版 本 历 史

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2015-03-01 | 0.1 | 创建初稿 | 张雨舟 |
|  |  |  |  |
|  |  |  |  |

SEPG 版 本 历 史

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2011-5-13 | 1.0 | 新建 | CSST\_WHRD |
|  |  |  |  |

**目 录**

[1. 引言 3](#_Toc413051630)

[1.1 目的 3](#_Toc413051631)

[1.2 适用范围 3](#_Toc413051632)

[1.3 读者对象 3](#_Toc413051633)

[1.4 参考资料 4](#_Toc413051634)

[1.5 术语和缩略语 4](#_Toc413051635)

[2. 系统概述 4](#_Toc413051636)

[2.1 用户特征 4](#_Toc413051637)

[2.2 功能需求总表 5](#_Toc413051638)

[3. 功能性需求 5](#_Toc413051639)

[3.1 UR01\_电话拨号盘 6](#_Toc413051640)

[3.1.1 需求说明 6](#_Toc413051641)

[3.1.2 流程图 6](#_Toc413051642)

[3.1.3 数据处理 7](#_Toc413051643)

[3.1.4 原型图 7](#_Toc413051644)

[4. 非功能性需求 7](#_Toc413051645)

[5. 性能需求 7](#_Toc413051646)

[6. 接口需求 8](#_Toc413051647)

[7. 其他限制性需求 8](#_Toc413051648)

[7.1 设计约束 8](#_Toc413051649)

[7.2 软硬件环境需求 8](#_Toc413051650)

[7.2.1 硬件 8](#_Toc413051651)

[7.2.2 网络 9](#_Toc413051652)

[7.2.3 软件 9](#_Toc413051653)

[8. 其他需求 9](#_Toc413051654)

[附录A 需求确认 9](#_Toc413051655)

# 引言

## 目的

|  |  |
| --- | --- |
|  | **提示**： |
|  | *指出预期的读者，编写目的* |

本文档的目的是分析用户特点，记录用户对系统的期望，目前业务的问题，确定用户对系统的需求，供系统分析师、项目经理、开发人员及维护人员参考，有针对性的进行系统开发、测试、验收等工作。

* 项目需求阶段的主要交付物。
* 项目团队外人员进行沟通的外部接口，用于他们评审和理解系统。
* 设计人员进行系统设计的输入源。
* 开发人员对系统功能开发的依据。
* 测试人员编写系统测试计划，测试案例编写的输入源。
* 收集并记录所有的外部接口，以用于作为完成设计和实现系统的参考。
* 系统初验和终验的参考

## 适用范围

|  |  |
| --- | --- |
|  | **提示**： |
|  | *简要说明此需求说明书适用的范围，以及受到此文档影响的任何其他的事务。* |

在香港卫安运营服务平台整个开发周期，此文档作为中安消研究院和香港卫安之间的系统需求开发约定文档。

本文档涵盖：

* 项目的基本介绍：项目基础信息、项目范围等；
* 系统功能需求；
* 系统非功能性需求：用户界面需求、软硬件需求、扩展性需求等

系统开发功能依托于此文档，当此文档有改变时，会影响基于此文档为基础一系列后续软件开发流程。

## 读者对象

本文档读者包括：客户、需求人员、设计人员、编码人员、测试人员

客户，为客户确定需求提供依据

需求人员，可指导需求建模，需求文档编写；

设计人员，指定了各迭代版本需完成的功能，利于系统设计；

编码人员，利于开发人员理解需求，对需要实现的功能有较清楚的认识及理解。

测试人员，可指导测试用例的编写；

## 参考资料

|  |  |
| --- | --- |
| **资料名称** | **内容** |
|  |  |

## 术语和缩略语

|  |  |
| --- | --- |
| **术语、缩略语** | **解 释** |
|  |  |
|  |  |

# 系统概述

|  |  |
| --- | --- |
|  | **提示**： |
|  | *（1）介绍系统的开发背景。*  *（2）从实现系统功能的角度说明该系统的主要特征，实现什么功能* |

本系统针对的客户为 。

本系统实现功能如下：

* 功能1
* 功能2

系统主要建设目标如下：

* 建设目标1
* 建设目标2

## 用户特征

|  |  |
| --- | --- |
| **用户** | **职 责** |
|  |  |
|  |  |
|  |  |
|  |  |

## 技术架构图



图表 1技术架构图

# 设计约束

|  |  |
| --- | --- |
|  | **提示**： |
|  | *体系结构设计人员从需求文档《软件需求规格说明书》中提取需求约束* |

## 遵循的标准规范

|  |  |
| --- | --- |
|  | **提示**： |
|  | *本系统应该遵循的标准或规范* |

## 软件/硬件环境约束

|  |  |
| --- | --- |
|  | **提示**： |
|  | *软件、硬件环境（包括运行环境和开发环境）的约束* |

### 软件环境约束

JDK1.8

MAVEN

Ubuntu

Docker

### 硬件环境约束

## 接口/协议约束

### 内部接口/协议约束

### 外部接口/协议约束

## 用户界面约束

## 软件质量约束

### 友好性

要求能方便的输入、输出数据，减少非法数据的输入，有固定分类的地方采用选择方式，替手工录入，减少出错率。

### 安全性

包括用户访问控制、信息加密、身份认证等，要求数据安全可靠，能够最大限度抵御外来攻击。

### 可维护性

增加软件能够被理解、校正、适应及增强功能的容易程度。使有编程经验的设计人员在理解算法、看懂源代码的基础上，能够对软件进行局部性修改。

### 可转移、转换性

J2EE的兼容性很高，在windows2003、windows 2008、windows 2012等操作系统都可以直接运行。

### 可复用性

流程引擎、消息引擎、财务管理等模块能够很好的支持复用

模块设计

## 其他约束

编码统一使用UTF-8编码

# 总体架构

## 设计原则

在设计中应遵循如下原则：

**规范性**

系统设计所采用的技术和设备应符合国家、地方的有关法规、行业标准以及工业标准；信息的分类及编码应严格执行现行的国家标准和行业标准。

**实用性**

在平台的建设中，要充分考虑应用系统对处理能力的需求，防止发生性能瓶颈，保证系统能够按时、按质和按量的交付使用。

**业务性原则**

紧密围绕联网报警的业务，系统应能适应业务的发展，遵从行业标准。

**易操作性原则**

遵从客户的需求和习惯，做到功能强大、界面友好、贴近实际、操作简单、使用方便。

**安全性和稳定性**

应用系统必须有高可靠性，并对使用信息进行严格的权限管理。在技术上，应采用数据库备份与恢复、身份认证和访问控制等相应的措施，保证数据库安全、应用软件运行、操作安全、系统的可靠和稳定等。

**可扩展性和可移植性**

系统建设必须考虑采用扩展性好的系统架构，保证能够适应将来的业务需求变化和公司战略发展的需要，预留扩展接口，适应业务需求变化，以利于系统的二次开发和升级。

**开放性**

系统总体方案设计在体系结构、硬件平台、软件平台的确定方面，从设备选型到设计、开发都要充分考虑“标准和开放”的原则。在应用系统的设计与开发方面，依据标准化和模块化的设计思想，在此基础上建立具有一定灵活性和可扩展性的应用平台，使系统不仅在体系结构上保持很大的开放性而且同时提供各种灵活可变的接口，系统内部也保持相当程度的可扩充性。

## 技术架构

系统技术架构从总体上分为五层，使用Spring管理对象。每个层在处理程序上都应该有一项明确的责任, 而不应该在功能上与其它层混合，并且每个层要与其它层分开的，但要给他们之间放一个通信接口。

表示层

系统的前端是表示层，我们将使用AngularJS框架。

下面是AngularJS所负责的：

1. 数据、逻辑、视图的分离

2. 数据和视图间的自动绑定

3. UI输入验证

Rest服务

系统的前端与后台的交互通过调用Rest服务实现，Rest服务使用SpringMVC实现。

下面是Rest服务所负责的：

1. 管理Rest的请求，做出相应的响应

2. 提供一个Controller，委派调用业务逻辑和其它上层处理

3. 处理异常

4. 数据校验

以下条款，不该在Rest服务的编码中出现。 它们与Rest服务无关的

1. 直接的与数据库通信，例如JDBC调用

2. 与你应用程序相关联的业务逻辑以及校验

3. 事务管理

在Rest服务层引入这些代码，则会带来高偶合和麻烦的维护。

业务逻辑层

系统的中间部分是业务逻辑层。

下面是业务逻辑层所负责的：

1. 处理应用程序的业务逻辑和业务校验

2. 管理事务

3. 允许与其它层相互作用的接口

4. 管理业务层级别的对象的依赖

5. 在Rest服务和持久层之间增加了一个灵活的机制，使得他们不直接的联系在一起

6. 管理程序的执行（从业务逻辑层到持久层，从业务逻辑层到服务层）

服务引擎层

系统的业务逻辑依赖于服务引擎层。服务引擎层提供工作流、消息服务、业务规则等底层接口，将业务逻辑与服务解耦，使业务逻辑指关注自身的业务，降低了模块直接的依赖。

持久层

系统的末端是持久层。我们将使用SpringJDBC和QueryDSL来进行持久化的数据库交互系统。Spring对JDBC的连接有很好的支持，可以对结果集进行对象映射。同时JDBC是最灵活的持久化方式。QueryDSL提供linq的查询方式，主要为SpringJDBC的查询方法生成SQL语句

# 系统架构

## 子系统介绍

|  |  |  |
| --- | --- | --- |
| 子系统编号 | 子系统名称 | 功能说明 |
| 01 | API网关 | 系统的唯一入口，负责服务请求路由、组合及协议转换 |
| 02 | 任务服务 | 任务调度的rest服务，提供任务创建、分配、执行等接口 |
| 03 | 用户服务 | 用户管理的rest服务，提供用户创建，修改、状态更新等接口 |
| 04 | 结算服务 | 任务佣金的rest服务，提供钱包管理、佣金管理、支付等接口 |
| 05 | GPS服务 | GPS的rest服务，通过消息接收用户的GPS坐标，提供历史轨迹查询接口 |
| 06 | 权限服务 | 角色管理的rest服务，提供角色管理、权限管理等接口 |
| 07 | 意见服务 | 意见反馈管理的rest服务，提供意见管理等接口 |
| 08 | 消息推送服务 | 手机推送的rest服务，通过消息接收推送的消息，提供消息查询接口，使用百度云推送服务 |
| 09 | 短信服务 | 手机短信的rest服务，通过消息接收推送的短信，提供短信查询接口，使用云服务功能 |

## 子系统关系图

|  |  |
| --- | --- |
|  | **提示**： |
|  | *描述子系统之间的依赖关系* |



## 子系统接口说明

|  |  |
| --- | --- |
|  | **提示**： |
|  | *描述子系统之间相互依赖的接口说明* |

所有服务直接的交互都使用rest接口或消息服务器通信，由于rest接口在开发过程中会时常变化，因此本文档并不详细描述各子系统的接口说明，具体接口参考各服务的/swagger-ui.html页面。基于消息服务器通信的消息格式如下：

**用户GPS变更事件GpsChangedEvent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 说明 | 类型 | 必需 | 备注 |
| userId | 用户ID |  | 是 |  |
| longitude | 经度 |  | 是 |  |
| latitude | 纬度 |  | 是 |  |

**任务完成事件TaskCompletedEvent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 说明 | 类型 | 必需 | 备注 |
| userId | 用户ID |  | 是 |  |
| taskId | 任务ID |  | 是 |  |
| commission | 佣金 |  | 是 |  |
| completeOn | 完成时间 |  | 是 |  |

**推送到单台设备MsgPushEvent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 说明 | 类型 | 必需 | 备注 |
| userId | 用户ID |  | 是 |  |
| channelId | 通道ID |  | 是 |  |
| subject | 摘要 |  | 是 |  |
| text | 正文 |  | 是 |  |
| expires | 消息过期时间 |  | 是 |  |

**广播消息MsgPushAllEvent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 说明 | 类型 | 必需 | 备注 |
| subject | 摘要 |  | 是 |  |
| text | 正文 |  | 是 |  |
| expires | 消息过期时间 |  | 是 |  |

**批量推送到多台设备MsgPushBatchEvent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 说明 | 类型 | 必需 | 备注 |
| userIds | 用户ID的集合 |  | 是 |  |
| channelIds | 通道ID的集合 |  | 是 |  |
| subject | 摘要 |  | 是 |  |
| text | 正文 |  | 是 |  |
| expires | 消息过期时间 |  | 是 |  |

**推送到单个号码SmsPushEvent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 说明 | 类型 | 必需 | 备注 |
| userId | 用户ID |  | 是 |  |
| tel | 短信号码 |  | 是 |  |
| content | 正文 |  | 是 |  |
| expires | 消息过期时间 |  | 是 |  |

**批量推送到多个号码SmsPushBatchEvent**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 名称 | 说明 | 类型 | 必需 | 备注 |
| userIds | 用户ID的集合 |  | 是 |  |
| tels | 短信号码的集合 |  | 是 |  |
| content | 正文 |  | 是 |  |
| expires | 消息过期时间 |  | 是 |  |

## API网关

### 服务介绍

API网关是一个服务器，是系统的唯一入口。API网关封装了系统内部架构，为每个客户端提供一个定制的API。API网关负责服务请求路由、组合及协议转换。客户端的所有请求都首先经过API网关，然后由它将请求路由到合适的微服务。API网管经常会通过调用多个微服务并合并结果来处理一个请求。它可以在Web协议（如HTTP与WebSocket）与内部使用的非Web友好协议之间转换。它还可以具有其它职责，如身份验证、监控、负载均衡、缓存、静态响应处理等。

为了减少系统的复杂性，尽量避免分布式事务，对于需要调用多个服务的POST、PUT、DELETE请求，API网关只调用一个服务的rest接口，并由rest接口通过消息服务器将数据广播到其他服务中。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | 用户登录 | POST /login | 用户登录，必须使用HTTPS访问 |
| 02 | 忘记密码 | POST /forgotpwd | 忘记密码，通过短信服务发送 |
| 03 | 修改密码 | PUT /users/{userId}/changepwd |  |
| 04 | 用户上线 | PUT /users/{userId}/on |  |
| 05 | 用户下线 | PUT /users/{userId}/off |  |
| 06 | 用户GPS跟踪 | PUT /users/{userId}/gps |  |
| 07 | 获取用户 | GET /users/${userId} |  |
| 08 | 获取用户钱包 | GET /users/{userId}/wallet | 必须使用HTTPS访问 |
| 09 | 提现 | PUT /users/{userId}/withdraw | 必须使用HTTPS访问 |
| 10 | 用户列表 | GET /users |  |
| 11 | 分页获取用户列表 | GET /users/page |  |
| 12 | 新增外勤用户 | POST /users/outdoor |  |
| 13 | 新增后台用户 | POST /users/backer |  |
| 14 | 锁定用户 | PUT /users/{userId}/lock |  |
| 15 | 解锁用户 | PUT /users/{userId}/unlock |  |
| 16 | 设置个人资料 | PUT /users/{userId}/profile |  |
| 17 | 设置钱包 | PUT /users/{userId}/wallet | 调用结算服务的rest接口 |
| 18 | 查询任务列表 | GET /users/{userId}/tasks |  |
| 19 | 查询提现列表 | GET /users/${userId}/withdraw |  |
| 20 | 新任务 | POST /tasks |  |
| 21 | 接单 | PUT /tasks/${taskId}/accept |  |
| 22 | 到达现场 | PUT /tasks/${taskId}/reach |  |
| 23 | 放弃任务 | PUT /tasks/${taskId}/abandon |  |
| 24 | 完成任务 | PUT /tasks/${taskId}/complete |  |
| 25 | 获取任务 | GET /tasks/${taskId} |  |
| 26 | 任务列表 | GET /tasks |  |
| 27 | 分页获取任务 | GET /tasks/page |  |
| 28 | 分页查询提现 | GET /withdraws/page |  |
| 29 | 支付 | PUT /withdraws/{withdrawId} |  |
| 30 | 查看提现记录 | GET /withdraws/${ withdrawId } |  |
| 31 | 新增角色 | POST /roles |  |
| 32 | 修改角色 | PUT /roles/${roleId} |  |
| 33 | 查询角色列表 | GET /roles |  |
| 34 | 角色授权 | POST /roles/${roleId}/permission |  |
| 35 | 查询角色授权 | GET /roles/${roleId}/permission |  |
| 36 | 分页获取意见 | GET /suggest /page |  |
| 37 | 查看意见详情 | GET /suggest/{suggestId} |  |
| 38 | 新增意见 | POST /suggest |  |
| 39 | 消息列表 | GET /msg |  |
| 40 | 分页查询消息记录 | GET /msg/page |  |
| 41 | 查看消息详情 | GET /msg/{msgId} |  |
| 42 | 短信列表 | GET /sms |  |
| 43 | 分页查询短信记录 | GET /sms/page |  |
| 44 | 查看短信详情 | GET /sms/{smsId} |  |
| 45 | GPS轨迹查询 | GET /gpstrack | GPS轨迹列表 |
| 46 | 获取新版本 | GET /version/lastest |  |

## 用户服务

### 服务介绍

用户服务主要用来维护用户数据。服务对外提供用户的维护和查询的接口。在用户的GPS坐标更新后需要发送GpsChangedEvent消息。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | 修改密码 | PUT /users/{userId}/changepwd | 修改密码 |
| 02 | 用户上线 | PUT /users/{userId}/on | 用户上线 |
| 03 | 用户下线 | PUT /users/{userId}/off | 用户下线 |
| 04 | 用户GPS跟踪 | PUT /users/{userId}/gps | 用户GPS跟踪 |
| 05 | 获取用户 | GET /users/${userId} | 获取用户 |
| 06 | 用户列表 | GET /users | 用户列表 |
| 07 | 分页获取用户列表 | GET /users/page | 分页获取用户列表 |
| 08 | 新增外勤用户 | POST /users/outdoor | 新增外勤用户 |
| 09 | 新增后台用户 | POST /users/backer | 新增后台用户 |
| 10 | 锁定用户 | PUT /users/{userId}/lock | 锁定用户 |
| 11 | 解锁用户 | PUT /users/{userId}/unlock | 解锁用户 |
| 12 | 设置个人资料 | PUT /users/{userId}/profile | 设置个人资料 |

### 核心逻辑

**用户的状态变化图**



**用户状态**，任务有2种状态：

1-活动：初始化状态

2-锁定：被锁定的用户禁止登陆系统

**用户类型**，用户有两种类型：1-外勤 2-后台用户

**私有**：如果是自有外勤出警，私有为true

**权限：**目前有两个系统 1-出警APP 2-后台，如果权限值为0，表示该用户不允许访问1和2；如果权限值为1，表示只允许访问1；如果权限值该值为2，表示只允许访问2；如果权限值为3，表示允许访问1和2 。如果增加新的系统，按照1、2、4、8、16的规则增加系统的权限值

如果用户的私有属性为true，那么用户在上线时必须设置待命点

如果用户是手机用户，用户名与手机号码保持一致；

如果must\_change\_passwd=true，用户在第一次登录时必须修改密码

## 任务服务

### 服务介绍

任务服务主要用来创建、分配、执行任务。服务对外提供任务维护和查询的接口。在任务完成之后需要发送TaskCompletedEvent消息。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | 新任务 | POST /tasks | 创建新任务 |
| 02 | 接单 | PUT /tasks/${taskId}/accept | 接单 |
| 03 | 到达现场 | PUT /tasks/${taskId}/reach | 到达现场 |
| 04 | 放弃任务 | PUT /tasks/${taskId}/abandon | 放弃任务 |
| 05 | 完成任务 | PUT /tasks/${taskId}/complete | 完成任务 |
| 06 | 获取任务 | GET /tasks/${taskId} | 获取任务 |
| 07 | 任务列表 | GET /tasks | 任务列表 |
| 08 | 分页获取任务 | GET /tasks/page | 分页获取任务 |

### 核心逻辑

**任务的状态变化图**



**任务状态**，任务有8种状态：

1-新建：初始化状态，更新任务发布时间

2-已指派：查询到合适的外勤人员，并发送消息推送事件，更新任务指派时间

3-已接单：外勤人员已接单，更新任务接单时间

4-处理中：到达现场，更新任务处理时间

5-完成：处理完成，更新任务完成时间，记录耗时

6-放弃：外勤人员再接单后因为某些原因无法到达现场处理，更新任务关闭时间

7-关闭：任务完成结算和评价，更新任务关闭时间

8-无人接单：在规定的时间内没有人接单，更新任务关闭时间

**任务类型**，任务只有一种类型：1-出警任务

**耗时**：任务从指派到完成所花费的时间，精确到秒

**等级**：1、2、4、8、16、32，当任务的状态变为放弃、无人接单时，需要以此任务为基础，新建一个升级后的订单。当等级超过阙值后需要采取后续流程，32为最高级别，等级达到32的任务，不再进入自动分配流程，改由后台人工处理。

**等待分配时间**：规定多长时间内任务必须有人接单。如果超过时间未接单，将状态设为无人接单，任务关闭。

**私有**：如果是自有外勤出警，私有为true

**锁定**：如果任务已经被接单，锁定为true

**关闭**：任务状态为异常、关闭、无人接单时，关闭状态为true

**处理结果**：无异常，实警

任务的佣金和人员搜索半径与等级存在对应关系，保存在task\_level表中

与任务有关的警情细节保存在task\_detail表中

与任务有关的活动细节保存在task\_activity表中

任务关闭状态为true时，需要将任务转移到历史任务表中

任务升级后需要将两个任务之间的关系保存在task-relation表中，目前只有一种关系类型：1-复制

## 结算服务

### 服务介绍

结算服务用于计算任务的佣金，管理用户钱包。服务对外提供钱包查询、支付的接口。佣金的数据来源是通过读取消息服务器的TaskCompletedEvent来新增数据。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | 查询用户钱包 | GET /wallets/${userId} | 查询用户钱包 |
| 02 | 提现 | POST /withdraws | 创建提现请求 |
| 03 | 查询提现列表 | GET /withdraws | 查询提现列表 |
| 04 | 分页查询提现 | GET /withdraws/page | 分页查询提现列表 |
| 05 | 支付 | PUT /withdraws/${withdrawId} | 支付 |
| 06 | 新增用户钱包 | POST /wallets | 新增用户钱包 |
| 07 | 查看提现记录 | GET /withdraws/${ withdrawId } |  |

### 核心逻辑

**金额**：用户可以自由支配的金额

**锁定金额**：用户发起提现，但是并未支付的金额

每次发起提现请求，需要同时修改金额和锁定金额

支付完成后，需要更新锁定金额，并将数据转移到历史支付表中

## 权限服务

### 服务介绍

权限服务维护用户的角色，授权等数据，服务对外提供角色管理、授权管理的接口。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | 新增角色 | POST /roles | 新增角色 |
| 02 | 修改角色 | PUT /roles/${roleId} | 修改角色 |
| 03 | 查询角色列表 | GET /roles | 查询角色列表 |
| 04 | 角色授权 | POST /roles/${roleId}/permission | 角色授权 |
| 05 | 查询角色授权 | POST /roles/${roleId}/permission | 查询角色的授权 |

### 核心逻辑

待定

## 意见服务

### 服务介绍

意见服务维护用户的意见反馈数据，服务对外提供意见反馈的新增和查询接口。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | 分页获取意见反馈 | GET /suggest /page | 分页查询意见列表 |
| 02 | 查看意见详情 | GET /suggest/{suggestId} | 查看单条意见的详情 |
| 03 | 新增意见详情 | POST /suggest | 新增意见反馈 |

## 消息推送服务

### 服务介绍

消息推送服务接收消息推送请求，并通过百度云服务将消息发送给指定的设备。服务的数据来源是通过读取消息服务器的MsgPushEvent、MsgPushBatchEvent和MsgPushAllEvent来新增数据，服务对外提供用户的消息记录的查询接口。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | 消息列表 | GET /msg/${userId} | 消息列表 |
| 02 | 分页查询消息记录 | GET /msg/page | 分页查询消息记录 |
| 03 | 查看消息详情 | GET /msg/{msgId} | 查看单条消息的详情 |

### 核心逻辑

**消息类型：**

1-单播：消息只发给一个用户

2-批量单播：消息发送给一组用户

**消息状态：**

1-创建：初始状态

2-已推送：已调用百度云推送服务

3-已发送：

4-未发送

5-正在发送

6-失败

7-过期：在规定时间内没有调用百度云推送服务

已发送、失败、过期的消息应该转移到历史表

如果超过一定时间，状态仍然为创建，将消息设为过期

## 短信服务

### 服务介绍

短信服务接收短信发送请求，并通过短信云服务将短信发送给指定的号码。服务的数据来源是通过读取消息服务器的SmsPushEvent和SmsPushBatchEvent来新增数据，服务对外提供短信的查询接口。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | 短信列表 | GET /sms | 短信列表 |
| 02 | 分页查询短信记录 | GET /sms/page | 分页查询短信记录 |
| 03 | 查看短信详情 | GET /sms/{smsId} | 查看单条短信的详情 |

### 核心逻辑

待定

## GPS服务

### 服务介绍

GPS服务用来记录用户的GPS轨迹。服务的数据来源是通过读取消息服务器的用户GIS变更事件GpsChangedEvent来新增数据，服务对外提供用户的GPS轨迹查询接口。

### 服务Rest接口

|  |  |  |  |
| --- | --- | --- | --- |
| 接口编号 | 接口名称 | URL | 接口说明 |
| 01 | GPS轨迹查询 | GET /gpstrack | GPS轨迹列表 |

### 核心逻辑

待定

# 部署模型

|  |  |
| --- | --- |
|  | **提示**： |
|  | *描述了如何把构件（可执行应用程序、应用服务）映射到底层平台、计算机节点上。*  *最终系统运行的环境。*  *说明用来部署和运行本产品的一种或多种物理网络（硬件）配置。它是部署模型的视图。对于每种配置，它至少应该指出用来执行该软件的物理节点（计算机、CPU）及其互连情况（总线连接、LAN 连接、点到点连接等）。另外还要包括各个功能模块到物理节点的映射。* |

## 系统拓扑图

## 节点说明

|  |  |
| --- | --- |
|  | **提示**： |
|  | *说明本系统应当在什么样的环境下运行，有什么强制要求和建议？*  *说明部署模型中的每个节点的情况（计算机硬件，网络通信，OS，软件，包括中间件，应用，外部系统等）。*  *在备注中可以描述节点的标准配置、最低配置等信息。。* |

|  |  |  |
| --- | --- | --- |
| **节点名称** | **节点描述** | **备注** |
| 数据库（主库） | 部署数据库服务器 | SQL Server 2012 |
| 数据库（从库） | 部署备份数据库，与主库实现主主备份 | SQL Server 2012 |
| 后台管理 | 不是后台管理服务器，与主数据库连接 | Windows Server 2012 |
| REST API | 部署REST API服务，与后台管理相连，不需要连接数据库，必须使用HTTPS访问 | Windows Server 2012 |
| 硬件防火墙 | 硬件防火墙 |  |

# 系统机制设计

|  |  |
| --- | --- |
|  | **提示**： |
|  | *描述系统中各种机制（即“复杂问题”的设计）的解决方法。*  *通常机制包括：持久化、通讯、安全性、错误处理、系统日志、分布式、接口、事务管理、性能等。*  *机制设计并不局限于以下列出的标题，可以根据实际项目情况增加相应的机制设计。* |

## 前端设计

系统通过AngularJS实现前端页面的渲染。AngularJS根据路由配置，调用后台管理的Rest服务实现单页面应用

## 持久化设计

## 系统运行维护设计

### 系统提示信息设计

系统提示信息通常包括以下三种级别的信息提示：

操作提示：为用户展示操作结果

警告信息：引起用户注意。

错误提示：解释出错原因。

尝试这种方法：给出可行解决方案建议，为用户展示问题所在。

错误消息的可视化设计属性（其颜色、字体大小、位置）可以清晰地将它与系统用户界面的其他部分区分开来，引起用户的注意。如图

图：操作提示

图：警告信息

图：错误提示

### 系统补救措施设计

使用数据库的主主备份保证数据在数据库中已经有备份，在系统出错后可以依靠数据库的恢复功能使系统再启动，就算系统崩溃用户数据也不会丢失或遭到破坏。当系统数据万一丢失时，可以立即启用备份副本。

### 系统维护设计

系统提供系统问题日志模块，考虑了系统维护的可控性，记录系统问题发生可能的原因，提供维护人员跟踪问题的途径和依据，通过系统问题日志，可以有效的减少系统维护的工作量。

系统提供用户管理模块，维护人员可以自行创建用户及角色，分配用户的使用权限，满足一般的用户管理要求。

系统提供系统问题日志模块，考虑了系统维护的可控性，记录系统问题发生可能的原因，提供维护人员跟踪问题的途径和依据，通过系统问题日志，可以有效的减少系统维护的工作量。

系统提供用户管理模块，维护人员可以自行创建用户及角色，分配用户的使用权限，满足一般的用户管理要求。

系统提供系统运行参数设置模块，操作简便，对人员要求不高，可以依据需要由客户或维护人员调整运行参数。

系统提供数据管理模块，在系统发生灾难性故障时，可以对系统数据进行初始化或恢复，及时恢复系统的正常运行状态

### 系统信息备份

系统不允许匿名用户访问，只允许授权用户访问。

系统对存储的用户密码等安全信息采用MD5+盐值并多次迭代的方式加密存储。

授权用户按照系统管理员分配的权限访问系统模块，对于一些重要的模块及数据，系统设置只有高权限的人员方可读取或是操作，系统安全保密性较高。

系统所有访问均使用HTTPS访问，禁止HTTP请求

## 系统安全设计

系统不允许匿名用户访问，只允许授权用户访问。

系统对存储的用户密码等安全信息采用MD5+盐值并多次迭代的方式加密存储。

授权用户按照系统管理员分配的权限访问系统模块，对于一些重要的模块及数据，系统设置只有高权限的人员方可读取或是操作，系统安全保密性较高。

重要数据的交互必须使用HTTPS访问，禁止使用HTTP访问

## 节点说明

# 通信协议

# 功能性需求