

# 概要设计说明书

## 目录

- 1. 引言.....2
  - 1.1 编写目的..... 2
  - 1.2 背景.....2
  - 1.3 定义.....2
  - 1.4 参考资料..... 2
- 2. 总体设计.....2
  - 2.1 需求规定..... 2
  - 2.2 运行环境..... 2
  - 2.3 基本设计概念和处理流程.....2
  - 2.4 结构.....2
  - 2.5 功能需求与程序的关系..... 2
  - 2.6 人工处理过程..... 3
- 3. 接口设计.....3
  - 3.1 用户接口..... 3
  - 3.2 外部接口..... 3
  - 3.3 内部接口..... 3
- 4. 运行设计.....3
  - 4.1 运行模块组合..... 3
  - 4.2 运行控制..... 3
  - 4.3 运行时间..... 3
- 5. 系统数据结构设计..... 3
  - 5.1 逻辑结构设计要点..... 3
  - 5.2 物理结构设计要点..... 3
  - 5.3 数据结构与程序的关系..... 3
- 6. 系统出错处理.....3
  - 6.1 出错信息..... 4
  - 6.2 补救措施..... 4
  - 6.3 系统维护设计..... 4

# 1. 引言

## 1.1 编写目的

预期读者：开发人员；测试人员；运行维护人员

## 1.2 背景

项目名称：知福  
用户：福州大学在校师生  
平台：微信小程序、腾讯云

## 1.3 定义

UML：统一建模语言

## 1.4 参考资料

GB8567-88(国标)

# 2. 总体设计

## 2.1 需求规定

用户授权微信账户信息，系统使用此信息登录。  
用户输入所需活动的关键字进行查询，系统给出匹配的活动列表，并能按用户要求的方式排序。  
用户输入需要注册的活动信息，包括名称，起止时间，可参加人员，活动详情等，系统将此活动注册、保存并发布于活动查看中。  
用户进行个人信息查询操作，系统给出用户参与的活动历史、所获得的成绩等。

## 2.2 运行环境

用户端：运行移动端微信应用程序的移动设备。  
服务器端：可承载 Web 服务的云主机，运行 Linux 发行版或 Windows Server。

## 2.3 基本设计概念和处理流程

说明本系统的基本设计概念和处理流程，尽量使用图表的形式。

## 2.4 结构

用一览表及框图的形式说明本系统的系统元素（各层模块、子程序、公用程序等）的划分，扼要说明每个系统元素的标识符和功能，分层次地给出各元素之间的控制与被控制关系。

## 2.5 功能器求与程序的关系

本条用一张如下的矩阵图说明各项功能需求的实现同各块程序的分配关系：

	前端小程序	后端 Web 服务	微信 API 对接	后端数据库
用户登录	√	√	√	√
活动查询	√	√		√

活动注册	√	√		√
------	---	---	--	---

## 2.6 人工处理过程

对于新注册的活动，内容需要进行人工审核。人工审核无误后，在转换至系统审核。

## 3. 接口设计

### 3.1 用户接口

活动查询页

活动注册页

个人信息页

### 3.2 外部接口

微信用户信息 API

### 3.3 内部接口

说明本系统之内的各个系统元素之间的接口的安排。

## 4. 运行设计

### 4.1 运行模块组合

说明对系统施加不同的外界运行控制时所引起的各种不同的运行模块组合，说明每种运行所历经的内部模块和支持软件。

### 4.2 运行控制

说明每一种外界的运行控制的方式方法和操作步骤。

### 4.3 运行时间

说明每种运行模块组合将占用各种资源的时间。

## 5. 系统数据结构设计

### 5.1 逻辑结构设计要点

给出本系统内所使用的每个数据结构的名称、标识符以及它们之中每个数据项、记录、文卷和系的标识、定义、长度及它们之间的层次的或表格的相互关系。

### 5.2 物理结构设计要点

给出本系统内所使用的每个数据结构中的每个数据项的存储要求，访问方法、存取单位、存取的物理关系（索引、设备、存储区域）、设计考虑和保密条件。

### 5.3 数据结构与程序的关系

说明各个数据结构与访问这些数据结构的形式：

## 6. 系统出错处理

## 6.1 出错信息

用一览表的方式说明每种可能的出错或故障情况出现时，系统输出信息的形式、含意及处理方法。

## 6.2 补救措施

定期建立数据库备份，一旦服务器数据库被破坏，可以使用最近的一份数据库副本进行还原，

说明故障出现后可能采取的变通措施，包括：

- a. 后备技术说明准备采用的后备技术，当原始系统数据万一丢失时启用的副本的建立和启动的技术，例如周期性地把磁盘信息记录到磁带上就是对于磁盘媒体的一种后备技术；
- b. 降效技术说明准备采用的后备技术，使用另一个效率稍低的系统或方法来求得所需结果的某些部分，例如一个自动系统的降效技术可以是手工操作和数据的人工记录；
- c. 恢复及再启动技术说明将使用的恢复再启动技术，使软件从故障点恢复执行或使软件从头开始重新运行的方法。

## 6.3 系统维护设计

说明为了系统维护的方便而在程序内部设计中作出的安排，包括在程序中专门安排用于系统的检查与维护的检测点和专用模块。系统应用程序维护。系统的业务处理过程是通过应用程序的运行而实现的，一旦程序发生问题或业务发生变化，就必然地引起程序的修改和调整，因此系统维护的主要活动是对程序进行维护。