

**总体设计文档**

**关于科技资讯整合的**

**微信小程序Collect**



**G20小组·Grateful dead**

浙江大学城市学院

计算机与科学技术学院软件工程专业

**Release date : 2019年4月24日**

**Version : V0.3**

# 文档说明

## 文档信息

文档信息表

|  |  |
| --- | --- |
| **文档原作者** | 周磊 |
| **创建日期** | 2019年4月16日 |
| **当前版本** | V0.3 |
| **上次版本** | V0.2 |

## 变更记录

变更记录表

|  |  |  |  |
| --- | --- | --- | --- |
| **变更日期** | **变更人** | **版本** | **备注** |
| 2019.4.16 | 周磊 | V0.1 | 进行了初步的总体设计，进行了调研 |
| 2019.4.21 | 唐敏敏 | V0.2 | 对内容进行了修改和补充 |
| 2019.4.24 | 唐敏敏 | V0.3 | 进行了内容的修订 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 目录

[1](#_Toc6774379)

[**总体设计文档** 1](#_Toc6774380)

[**关于科技资讯整合的** 1](#_Toc6774381)

[**微信小程序Collect** 1](#_Toc6774382)

[1](#_Toc6774383)

[**G20小组·Grateful dead** 1](#_Toc6774384)

[浙江大学城市学院 1](#_Toc6774385)

[计算机与科学技术学院软件工程专业 1](#_Toc6774386)

[**Release date : 2019年4月21日** 1](#_Toc6774387)

[**Version : V0.2** 1](#_Toc6774388)

[文档说明 2](#_Toc6774389)

[文档信息 2](#_Toc6774390)

[变更记录 2](#_Toc6774391)

[目录 3](#_Toc6774392)

[1 引言 6](#_Toc6774393)

[编写目的 6](#_Toc6774394)

[在经过需求分析阶段的工作后，我们已经确定了做什么，而这份文档的主要目的便是确定系统该怎么去实现这些功能，这份文档会将各个阶段的工作划分为组成系统的物理元素---程序、文件、数据库、人工过程和文档等，但是每个物理元素又处于黑盒子级。 6](#_Toc6774395)

[确定系统中的每个程序是有哪些模块构成的，以及这些模块间的相互关系。 6](#_Toc6774396)

[系统概述 6](#_Toc6774397)

[1.1.1 该软件的用途 该软件是为了服务那些喜好阅读科技、IT行业、科研最新的资讯的人群。 6](#_Toc6774398)

[1.1.2 该软件的一般特性 开发、运行和维护的历史：该系统从2019/3/19开始正式开发，暂未正式投入使用和维护 投资方：该项目的由G20小组自行负担开发经费 需求方：在校学生、相关从业人员、对科技有兴趣的人群。 开发方：G20小组 支持机构：浙江大学城市学院 6](#_Toc6774399)

[1.1.1 基线 本项目的需求分析说明书 7](#_Toc6774400)

[2 参考文件 7](#_Toc6774401)

[国标GBT 8567-2006 7](#_Toc6774402)

[3 系统级设计决策 采用微信小程序 7](#_Toc6774403)

[4 系统体系结构设计 7](#_Toc6774404)

[系统总体设计 7](#_Toc6774405)

[4.1.1 概述 7](#_Toc6774406)

[4.1.2 设计思想 8](#_Toc6774407)

[4.1.3 基本处理流程 9](#_Toc6774408)

[4.1.4 系统体系结构 11](#_Toc6774409)

[4.1.5 功能需求与系统配置项的关系 12](#_Toc6774410)

[4.1.6 人工处理过程 暂无，除了定期查看数据库的信息是否有异常 12](#_Toc6774411)

[接口设计 12](#_Toc6774412)

[4.1.7 硬件接口 可以流畅运行微信客户端（手机平台），且开启小程序功能。【推荐手机配置：小米3或iPhone5及以上机型】 17](#_Toc6774413)

[4.1.8 软件接口 手机系统：Android系统，iOS系统 前端编程软件: 微信Web开发者工具 数据库工具：MongoDB 服务器工具：Node.js，Express 原型设计软件：墨刀、Axure RP 8 版本控制工具：Git 、 GitKareken 17](#_Toc6774414)

[5 运行设计 18](#_Toc6774415)

[5.1 系统初始化 该系统的初始化主要是通过重构数据库的数据来完成的 18](#_Toc6774416)

[5.2 运行控制 18](#_Toc6774417)

[5.2.1 用户点击进入小程序app 涉及到主界面展示模块和资讯展示模块、读取模块 此时需要调用云端服务器 18](#_Toc6774418)

[5.2.2 用户点击标栏的板块 涉及到资讯展示模块 18](#_Toc6774419)

[5.2.3 用户点击一条具体的新闻 涉及到具体的资讯展示模块 18](#_Toc6774420)

[5.2.4 用户用户点击评论或是收藏、分享按钮 涉及到资讯的收藏以及资讯的“我喜欢”模块、分享模块 此时需要在服务器端更新数据库 以及调用微信的朋友圈/朋友分享接口 18](#_Toc6774421)

[5.2.5 用户点击我的历史和我的收藏 涉及到时光机模块和资讯的收藏模块 将向云端服务器发送请求 18](#_Toc6774422)

[5.2.6 用户点击个人信息 涉及到用户个人信息的修改模块 19](#_Toc6774423)

[5.3 运行结束 用户点击可以通过两种方式退出该系统 1.点击“退出小程序”按钮，即可退出 2.直接退出微信也可以退出该系统 19](#_Toc6774424)

[6 系统出错处理设计 19](#_Toc6774425)

[6.1 出错信息表 19](#_Toc6774426)

[6.2 故障处理技术 故障处理技术： 1、事务故障的恢复：撤消事务（UNDO） 2、系统崩溃故障的恢复：撤消事务（UNDO）+重做事务（REDO） 3、磁盘（介质）故障的恢复 4、灾难故障的恢复：远程备份 19](#_Toc6774427)

[6.3 补救措施 回滚数据库 在清晨或是午夜不繁忙的时候，自动备份数据库和运行日志 发生错误时的时候，G20小组按照工作分工，对出错的模块进行分析，尽快使系统重新上线 19](#_Toc6774428)

[7 系统维护设计 20](#_Toc6774429)

[7.1 检测点的设计 采用日志模块记录系统运行过程状态以及出现的问题。 系统外部接口调用前后都要进行日志的详细记录，方便接口调试；系统中重要的状态信息的变化都要通过日志记录下来，方便查问题时还原现场，推断程序运行过程；系统入口与出口记录输入与输出，方便定位问题；任何业务异常都进行日志详细记录；日志记录要求清晰准确，能够通过日志准确定位系统问题 20](#_Toc6774430)

[7.2 检测专用模块的设计 待补充 20](#_Toc6774431)

[8 尚待解决的问题 在这份文档中我们还没有更加清晰的绘制出各个模块之前互相调用的关系以及完美符合需求的整体设计 20](#_Toc6774432)

[9 需求的可追踪性 我们会在每个里程碑前和长期用户确认需求是否达到他们的要求，同时对照着需求文档进行系统的修改和实现。 20](#_Toc6774433)

[10 注解 DBDD是数据库(顶层)设计说明 20](#_Toc6774434)

[**11** **附录** 20](#_Toc6774435)

[**接口设计说明** 20](#_Toc6774436)

[**数据库(顶层)设计说明(DBDD)** 20](#_Toc6774437)

[**12** **DBDD** 20](#_Toc6774438)

[**引言** 20](#_Toc6774439)

[**12.1.1** **标识** 20](#_Toc6774440)

[暂无，版本号v0.1，发行号v0.1 20](#_Toc6774441)

[**12.1.2** **数据库概述** 21](#_Toc6774442)

[本数据库主要用途是为了给爬虫程序提供不同资讯的存放的一个容器。同时允许用户在使用小程序的时候通过接口来调用数据库中的资讯，以在界面展示。 21](#_Toc6774443)

[本数据库采用的是非关系型数据库，采用此类型的数据库的原因是考虑到我们具体的需求。因为网页的爬取的信息并不是有规律的，不同的网页，图片和文字的布局都不一样。所以不能按照同一模板来存放不同字段的信息。所以我们采用了非关系型数据库，先把整个网页的文档以JSON格式存下来。在通过分类的程序，把关键字和页面的布局存下来。这样比Mysql等关系型数据库的存放更为灵活。 21](#_Toc6774444)

[维护方法我们的开发人员会每隔两天查看检查数据库是否有异常（字段的内容是否合理，有没有怪异的参数信息）还有服务器的运行的压力（暂时未讨论出明确的解决方案）。 21](#_Toc6774445)

[数据库搭建在Nginx + Node.js + Express的坏境下。选择的是非关系型数据库的热门数据库MongoDB。 21](#_Toc6774446)

[数据库部署在阿里云主机上，24小时不间断运行，同时我们会定期的创建快照，保证资料的安全和稳定。 21](#_Toc6774447)

[**12.1.3** **文档概述** 22](#_Toc6774448)

[**本文档用于向开发人员和客户介绍本项目采用的数据库的数据结构、定义的接口和交互的逻辑，并尽量简洁明了的介绍数据库所起到的作用和数据的流向以及处理过程，相信这在对后期的测试和数据库的开发是一个指南。** 22](#_Toc6774449)

[**参考文件** 22](#_Toc6774450)

[**数据库级设计决策** 22](#_Toc6774451)

[ 关于该数据库应接受的查询或其他输入和它应产生的输出(显示、报告、消息、响应等)的设计决策————通过Node.js结合HTTP的GET和POST，调用接口 22](#_Toc6774452)

[ 有关响应每次输入或查询的数据库行为的设计决策，包括动作、响应时间和其他性能特性、所选择的方程式/算法/规则、配置和对不允许的输入的处理。——————每次输入，【1】如果是请求搜索相关ID和标签的资讯的集合，将通过数据库语言在数据库中搜索（提前会建立好索引并定期更新），搜索并返回对应的信息。【2】如果是爬虫请求插入信息，先检测字段有无异常，无异常则插入，并返回成功结果。响应时间每条不应超过5秒 22](#_Toc6774453)

[ 有关数据库/数据文件如何呈现给用户的设计决策————通过前端的设计以一个整洁的布局将资讯展示给用户 23](#_Toc6774454)

[ 有关数据库要提供的可用性、保密性、私密性和运行连续性的层次与类型的设计决策——————数据库的登录信息只有组内开发人员知道，并且使用前需在小组内申明理由并获得全员同意。同时给服务器增加一些安全的保障（购买额外的服务），同时每天给服务器的数据库记录进行备份，选择的阿里云的主机运行较为稳定且广受好评 23](#_Toc6774455)

[ 有关备份与恢复的设计决策，包括数据与处理分布策略、备份与恢复期间所允许的动作、对例如音像等新技术或非标准技术的特殊考虑——————**暂无思路** 23](#_Toc6774456)

[**数据库详细设计** 23](#_Toc6774457)

[**12.1.4** **暂无，详见详细设计** 24](#_Toc6774458)

[**用于数据库访问或操纵的软件配置项的详细设计** 24](#_Toc6774459)

[**12.1.5** **详见4.1软件配置项** 24](#_Toc6774460)

[**需求的可追踪性** 24](#_Toc6774461)

[**暂无，我们需要和固定用户不断地联系，并且以可以运行的原型展示，测试用户指定的需求，若满足，则数据库设计达到要求。（压力测试？）** 24](#_Toc6774462)

[**注解** 24](#_Toc6774463)

[**暂无** 24](#_Toc6774464)

[**附录** 24](#_Toc6774465)

[**暂无** 24](#_Toc6774466)

# 引言

## 编写目的

## 在经过需求分析阶段的工作后，我们已经确定了做什么，而这份文档的主要目的便是确定系统该怎么去实现这些功能，这份文档会将各个阶段的工作划分为组成系统的物理元素---程序、文件、数据库、人工过程和文档等，但是每个物理元素又处于黑盒子级。

## 确定系统中的每个程序是有哪些模块构成的，以及这些模块间的相互关系。

## 系统概述

### 该软件的用途 该软件是为了服务那些喜好阅读科技、IT行业、科研最新的资讯的人群。

### 该软件的一般特性 开发、运行和维护的历史：该系统从2019/3/19开始正式开发，暂未正式投入使用和维护 投资方：该项目的由G20小组自行负担开发经费 需求方：在校学生、相关从业人员、对科技有兴趣的人群。 开发方：G20小组 支持机构：浙江大学城市学院

### 基线 本项目的需求分析说明书

# 参考文件

# 国标GBT 8567-2006

# 系统级设计决策 采用微信小程序

# 系统体系结构设计

## 系统总体设计

### 概述

#### 功能描述 1、用户可通过该软件进行登陆注册 2、用户拥有对新闻进行收藏的功能。

#### 3、用户收到的信息都是及时的最新内容。

#### 4、用户可以分享资讯。

#### 运行环境 能够流畅的运行的微信的移动设备

### 设计思想

#### 系统构思 我们的项目由三大部分组成。第一大部分是运行在服务器的爬虫程序，爬虫定期到各大资讯网站上去爬取制定分类的文章。提取出标题、作者、来源和日期信息和标签。然后通过调用数据库的连接接口，将一个资讯作为一条字段加入到数据库中。 在服务器上搭建的后端，需要有相应程序调用的请求的接口【包括相应爬虫程序的插入还有小程序前端调用的请求】，要有当数据出错的警告或者安全措施 在小程序前端设计上，以微信开发者文档为指导进行开发，要有用户注册的模块和界面布局设计（包括与服务器的数据库的调用）。相关分类的资讯的展示模块和界面布局设计（包括与服务器的数据库的调用） 。具体资讯的展示的模块和界面布局设计（括与服务器的数据库的调用） . 具体资讯的喜欢、分享、收藏的功能模块（包括与服务器的数据库的调用和微信接口的调用） 用户信息的修改（包括与服务器的数据库的调用和微信接口的调用 。时光机功能（展示全部收藏信息、指定时间段）

#### 关键技术与算法

#### 爬虫相关技术：我们将采用scarpy+multiprocessing的技术多线程爬取数据，编程语言采用python来实现。 数据的存储方面，我们将使用MongoDB和scrapy\_redis来搭建分布式爬虫。

#### 新闻分类相关技术： 利用python语言构建朴素贝叶斯分类器，通过网络上已经公开的词库和人工的审核添加，进行模型的训练达到分类新闻的效果。

#### 服务器后端技术和算法：采用的是Nginx + Node.js + Express的框架，让开发更高效和容易上手。重要的是和服务器调用（Require(数据库表)），GET和POST（相关HTTP知识）

#### 数据库的技术：采用MongoDB，具体详见数据库设计文档，存入的数据为了方便查找应该建立索引

#### 前端技术：因为微信小程序主要基于JavaScript的语言，CSS+HTML5，所以我们界面设计我们主要是参考已有的界面美观的小程序的源码，进行符合我们的界面要求的修改

#### 关键数据结构 见数据字典

### 基本处理流程

#### 系统流程图

#### 

#### 

#### 数据流程图

### 系统体系结构

#### 系统配置项

#### 系统层次结构

#### 系统配置项设计 详见1.1.3.1

### 功能需求与系统配置项的关系

### 人工处理过程 暂无，除了定期查看数据库的信息是否有异常

## 接口设计

用户接口  
给用户的接口  
  
登陆界面：  
  
  
用户个人数据管理接口：  
   
标签分类选择与删除接口：  
  
资讯评论与分享接口；  
  
收藏和历史记录查看接口  


### 硬件接口 可以流畅运行微信客户端（手机平台），且开启小程序功能。【推荐手机配置：小米3或iPhone5及以上机型】

### 软件接口 手机系统：Android系统，iOS系统 前端编程软件: 微信Web开发者工具 数据库工具：MongoDB 服务器工具：Node.js，Express 原型设计软件：墨刀、Axure RP 8 版本控制工具：Git 、 GitKareken

# 运行设计

# 系统初始化 该系统的初始化主要是通过重构数据库的数据来完成的

# 运行控制

### 用户点击进入小程序app 涉及到主界面展示模块和资讯展示模块、读取模块 此时需要调用云端服务器

### 用户点击标栏的板块 涉及到资讯展示模块

### 用户点击一条具体的新闻 涉及到具体的资讯展示模块

### 用户用户点击评论或是收藏、分享按钮 涉及到资讯的收藏以及资讯的“我喜欢”模块、分享模块 此时需要在服务器端更新数据库 以及调用微信的朋友圈/朋友分享接口

### 用户点击我的历史和我的收藏 涉及到时光机模块和资讯的收藏模块 将向云端服务器发送请求

### 用户点击个人信息 涉及到用户个人信息的修改模块

# 运行结束 用户点击可以通过两种方式退出该系统 1.点击“退出小程序”按钮，即可退出 2.直接退出微信也可以退出该系统

# 系统出错处理设计

# 出错信息表

|  |  |  |  |
| --- | --- | --- | --- |
| 微信小程序 | 错误类型 | 出错模块 | 处理方式 |
| 网络错误 | 新闻读取模块 | 提示错误，停止页面读取 |
| 操作失败 | 获取个人收藏  获取个人标签信息  获取历史记录  分享到微信  评论 | 告知用户 建议重试 |

# 故障处理技术 故障处理技术： 1、事务故障的恢复：撤消事务（UNDO） 2、系统崩溃故障的恢复：撤消事务（UNDO）+重做事务（REDO） 3、磁盘（介质）故障的恢复 4、灾难故障的恢复：远程备份

# 补救措施 回滚数据库 在清晨或是午夜不繁忙的时候，自动备份数据库和运行日志 发生错误时的时候，G20小组按照工作分工，对出错的模块进行分析，尽快使系统重新上线

# 系统维护设计

# 检测点的设计 采用日志模块记录系统运行过程状态以及出现的问题。 系统外部接口调用前后都要进行日志的详细记录，方便接口调试；系统中重要的状态信息的变化都要通过日志记录下来，方便查问题时还原现场，推断程序运行过程；系统入口与出口记录输入与输出，方便定位问题；任何业务异常都进行日志详细记录；日志记录要求清晰准确，能够通过日志准确定位系统问题

# 检测专用模块的设计 待补充

# 尚待解决的问题 在这份文档中我们还没有更加清晰的绘制出各个模块之前互相调用的关系以及完美符合需求的整体设计

# 需求的可追踪性 我们会在每个里程碑前和长期用户确认需求是否达到他们的要求，同时对照着需求文档进行系统的修改和实现。

# 注解 DBDD是数据库(顶层)设计说明

1. **附录**

**接口设计说明**

**数据库(顶层)设计说明(DBDD)**

1. **DBDD**

**引言**

* + 1. **标识**

暂无，版本号v0.1，发行号v0.1

* + 1. **数据库概述**

本数据库主要用途是为了给爬虫程序提供不同资讯的存放的一个容器。同时允许用户在使用小程序的时候通过接口来调用数据库中的资讯，以在界面展示。

本数据库采用的是非关系型数据库，采用此类型的数据库的原因是考虑到我们具体的需求。因为网页的爬取的信息并不是有规律的，不同的网页，图片和文字的布局都不一样。所以不能按照同一模板来存放不同字段的信息。所以我们采用了非关系型数据库，先把整个网页的文档以JSON格式存下来。在通过分类的程序，把关键字和页面的布局存下来。这样比Mysql等关系型数据库的存放更为灵活。

维护方法我们的开发人员会每隔两天查看检查数据库是否有异常（字段的内容是否合理，有没有怪异的参数信息）还有服务器的运行的压力（暂时未讨论出明确的解决方案）。

数据库搭建在Nginx + Node.js + Express的坏境下。选择的是非关系型数据库的热门数据库MongoDB。

数据库部署在阿里云主机上，24小时不间断运行，同时我们会定期的创建快照，保证资料的安全和稳定。

* + 1. **文档概述**

**本文档用于向开发人员和客户介绍本项目采用的数据库的数据结构、定义的接口和交互的逻辑，并尽量简洁明了的介绍数据库所起到的作用和数据的流向以及处理过程，相信这在对后期的测试和数据库的开发是一个指南。**

**参考文件**

* **GB/T 8567-2006 《中华人民共和国国家标准-计算机软件文档编制规范》 之** *7.14数据库(顶层)设计说明(DBDD)*
* **《软件工程导论（第六版）》 张海藩 牟永敏 编著 ISBN 978-7-302-33098-1**
* **《PMBOK》项目管理计划书**

**数据库级设计决策**

* 关于该数据库应接受的查询或其他输入和它应产生的输出(显示、报告、消息、响应等)的设计决策————通过Node.js结合HTTP的GET和POST，调用接口
* 有关响应每次输入或查询的数据库行为的设计决策，包括动作、响应时间和其他性能特性、所选择的方程式/算法/规则、配置和对不允许的输入的处理。——————每次输入，【1】如果是请求搜索相关ID和标签的资讯的集合，将通过数据库语言在数据库中搜索（提前会建立好索引并定期更新），搜索并返回对应的信息。【2】如果是爬虫请求插入信息，先检测字段有无异常，无异常则插入，并返回成功结果。响应时间每条不应超过5秒
* 有关数据库/数据文件如何呈现给用户的设计决策————通过前端的设计以一个整洁的布局将资讯展示给用户
* 有关数据库要提供的可用性、保密性、私密性和运行连续性的层次与类型的设计决策——————数据库的登录信息只有组内开发人员知道，并且使用前需在小组内申明理由并获得全员同意。同时给服务器增加一些安全的保障（购买额外的服务），同时每天给服务器的数据库记录进行备份，选择的阿里云的主机运行较为稳定且广受好评
* 有关备份与恢复的设计决策，包括数据与处理分布策略、备份与恢复期间所允许的动作、对例如音像等新技术或非标准技术的特殊考虑——————**暂无思路**

**数据库详细设计**

* + 1. 暂无，详见详细设计

用于数据库访问或操纵的软件配置项的详细设计

* + 1. 详见4.1软件配置项

**需求的可追踪性**

暂无，我们需要和固定用户不断地联系，并且以可以运行的原型展示，测试用户指定的需求，若满足，则数据库设计达到要求。（压力测试？）

**注解**

暂无

**附录**

暂无