**<我的挑战日志>**

系统设计说明书

**作 者：福州大学 2018 软件工程实践课程“第一组”**

**完成日期： 2020.11.06**

[目录 1.引言 3](#_Toc68397)

[1.1 编写目的 3](#_Toc68398)

[1.2 背景 3](#_Toc68399)

[1.3 定义 3](#_Toc68400)

[1.4 参考资料 3](#_Toc68401)

[2. 总体设计 3](#_Toc68402)

[2.1 需求规定 3](#_Toc68403)

[2.2 运行环境 4](#_Toc68404)

[2.3 基本设计概念和处理流程 5](#_Toc68405)

[2.4 结构 6](#_Toc68406)

[2.5 功能需求与程序的关系 6](#_Toc68407)

[2.6 人工处理过程 6](#_Toc68408)

[2.7 尚未问决的问题 7](#_Toc68409)

[3. 接口设计 7](#_Toc68410)

[3.1 用户接口 7](#_Toc68411)

[3.1.1 用户（组员和组长） 7](#_Toc68412)

[3.1.2 组长 7](#_Toc68413)

[3.1.3 管理员（助教和老师） 7](#_Toc68414)

[3.1.4 老师 8](#_Toc68415)

[3.2 外部接口 8](#_Toc68416)

[3.3 内部接口 9](#_Toc68417)

[3.3.1 约定 9](#_Toc68418)

[3.3.2 前台部分 9](#_Toc68419)

[3.3.3 后台部分 16](#_Toc68420)

[4 运行设计 26](#_Toc68421)

[4.1 运行模块组合 26](#_Toc68422)

[4.2 运行控制 26](#_Toc68423)

[4.3 运行时间 27](#_Toc68424)

[5. 系统数据结构设计 27](#_Toc68425)

[5.1 逻辑结构设计要点 27](#_Toc68426)

[5.2 物理结构设计要点 30](#_Toc68427)

[5.3 数据结构和程序的关系 30](#_Toc68428)

[6.系统出错处理设计 30](#_Toc68429)

[6.1 出错信息 30](#_Toc68430)

[6.2 补救措施 31](#_Toc68431)

[6.3 系统维护设计 31](#_Toc68432)

**1.引言**

## 1.1 编写目的

软件设计结构的具体任务是将一个复杂系统按功能划分，建立模块的层次结构及调用关系，确定模块间的接口及人机界面等。

此系统设计说明书是为了说明《我的挑战日志》系统的体系结构，以及需求用例的各个功能点在结构中的体现，为系统的详细设计人员进行详细设计时的输入参考文档。

## 1.2 背景

大学生群体总是面对着一个问题——拖延症。生活中有很多事情诱惑我们无法完成计划，比如打游戏，刷视频，看小说等等，于是就有了太多拖延的借口，没有时间、太累了、太难了……随着娱乐行业的发展，诱惑越来越多，借口就越来越丰富，以至于我们的时间利用率直线下降。可是生活学习的任务和压力却不会随之减少，这就让拖延症成为了大学生活的一大危机。

很多软件厂家看到了这个危机，于是市面上有了很多规划时间，制定计划的软件，可却出现了一个奇怪的现象：学生们往往雄心壮志兴致勃勃地下了软件，打算自此开始过上井井有条的生活，可是没过几天那个软件就被遗忘在不知哪个文件夹里吃灰了，或者即便偶尔能想起来制定个计划，也不觉得没有完成会怎么样。所以这些软件无法从根本上解决用户的问题。究其原因，还是因为这些软件的实现方式都是千篇一律的制定计划之后每隔一段时间用闹铃提醒，锁定手机也可以随时退出，虽然有奖惩体制但也不痛不痒，没有新意。

我们总结了市面上制定计划软件的优点和缺陷，以解决大学生拖延症并吸引大学生注意，产生长久坚持使用的兴趣为目的，设计了这款小程序。希望能够通过这款小程序帮助到大学生，丰富他们的学习生活，提高他们的学习质量。

小程序名称：我的挑战日志

项目开发者：福州大学 2018 软件工程实践课程“第一组”

主要用户：需要制定计划，摆脱“拖延症”的大学生

## 1.3 定义

系统：《我的挑战日志》，即该产品。

SQL 语言：结构化查询语言(Structured Query Language)简称 SQL，是一种特殊目的的编程语言，是一种数据库查询和程序设计语言，用于存取数据以及查询、更新和管理关系数据库系统。

## 1.4 参考资料

[1]《概要设计说明书》（Gb8567-88）

[2]《需求分析报告》（<我的挑战日志>）

[3]《数据库设计报告》（<我的挑战日志>）

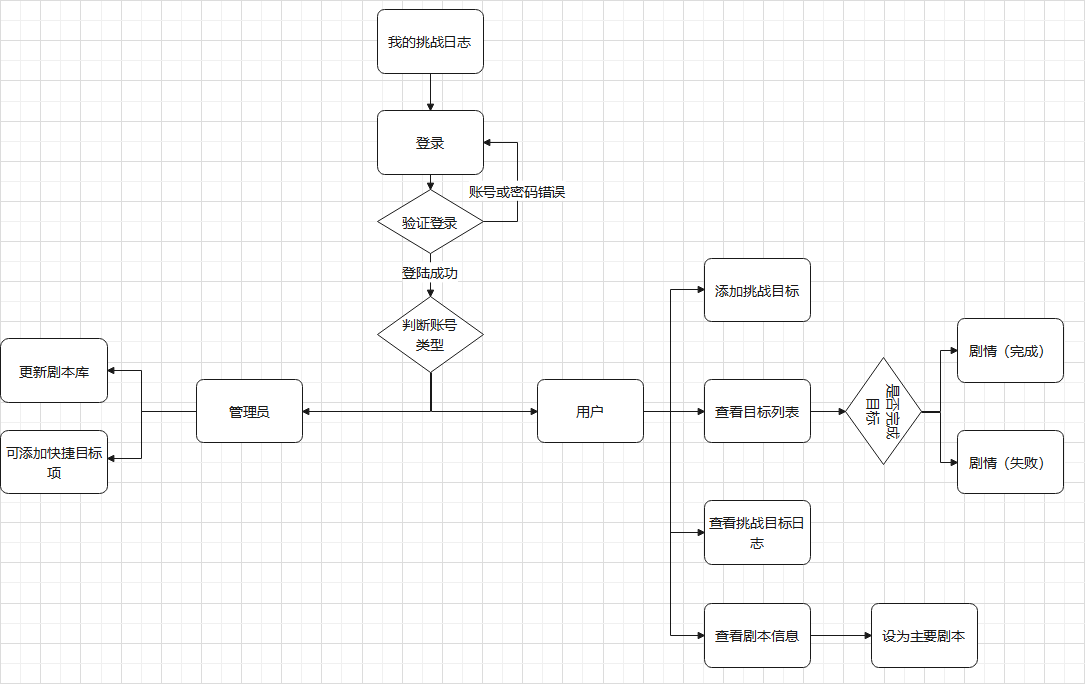
# 2. 总体设计

## 2.1 需求规定

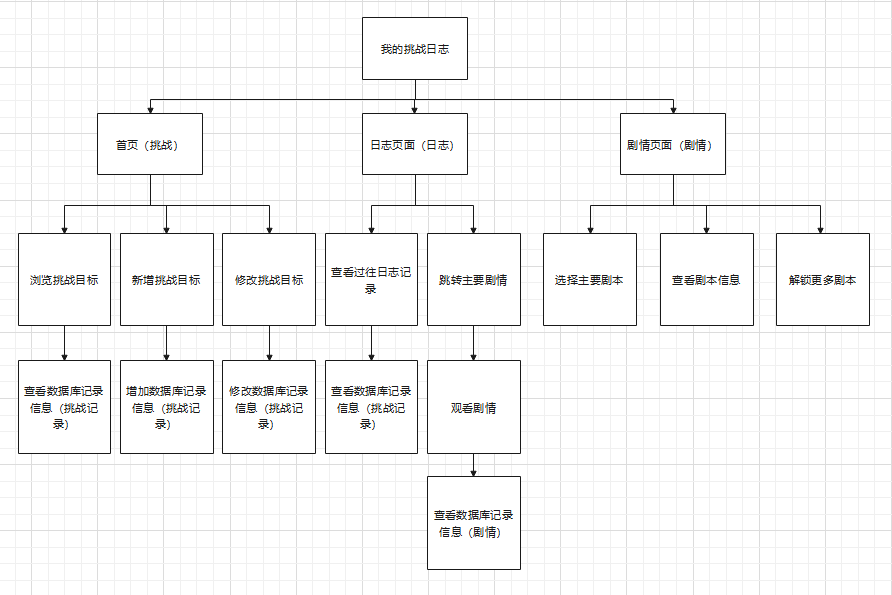
## IMG_2562.2 运行环境

软件系统运行环境：微信7.0 及以上

## 2.3 基本设计概念和处理流程



## 2.4 结构



## 2.5 功能需求与程序的关系

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 权限 | 注册 | 登录 | 修改个人信息 | 新增目标 | 确认目标 | 确认目标超时 | 查看往期目标 | 查看剧情日志 | 目标确认后查看剧情 | 添加新剧情到日志 | 删除目标 | 选择剧本 | 新增剧本 | 用户管理 |
| 用户 | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ | √ |  |  |
| 管理员 |  | √ |  |  |  |  |  |  |  |  |  |  | √ | √ |

## 2.6 人工处理过程

管理账户有些功能需要人工修改数据库。

## 2.7 尚未问决的问题

暂无

# 3. 接口设计

## 3.1 用户接口

### 3.1.1 用户（组员和组长）

|  |  |  |
| --- | --- | --- |
| 命令 | 语法 | 作用/消息提醒 |
| 注册 | 点击 | 注册成功/注册失败信息提醒 |
| 登录 | 点击 | 登录成功/登录失败信息提醒 |
| 个人信息 | 点击 | 进入个人信息页面 |
| 修改个人信息 | 点击 | 进入修改个人信息页面 |
| 退出 | 点击 | 退出登录 |
| 新增目标 | 点击 | 进入新增目标页面 |
| 确认目标 | 点击 | 完成/未完成目标提醒 |
| 删除目标 | 点击 | 删除设定的目标 |
| 更新目标 | 进入页面自动更新 | 刷新当前页面目标状态 |
| 查看历史目标 | 点击 | 进入历史目标页面 |
| 查看剧情日志 | 点击 | 进入当前剧本页面 |
| 选择新剧本 | 点击 | 修改当前使用的剧本 |

## 3.2 外部接口

无

## 3.3 内部接口

### 3.3.1 约定

/\*

1.用户注册

请求路径:/user/register

提交参数：

\*/

{

"user\_id":"abc1234"//用户id（微信号）

}

/\*返回结果\*/

{

"status":true,//注册成功时

"msg":"注册成功"//返回的消息

}

{

"status":false,//注册失败时

"msg":"提示词"//失败原因

}

/\*

2.用户登录

请求路径:/user/login

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"password":"111111"//用户密码（视具体情况修改参数格式）

}

/\*返回结果\*/

{

"status":true,//登录成功时

"msg":"登录成功"//返回的消息

}

{

"status":false,//登录失败时

"msg":"提示词"//失败原因

}

/\*

3.修改个人信息

请求路径:/user/update

提交参数:

\*/

{

"user\_id":"abc1234"//用户id（微信号）

"data":{

"user\_name":"abc1234",//用户昵称

//其他数据

}

}

/\*返回结果\*/

{

"status":true,//修改个人信息成功时

"msg":"修改成功"//返回消息

}

{

"status":false,//修改个人信息失败时

"msg":"提示词"//失败原因

}

/\*

4.新增目标

请求路径:user/target/add

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"target":{

"title":"MyTarget",//目标标题

"start\_time":{

//开始日期

"year":"2020",//年

"month":"11",//月

"day":"11"//日

},

"end\_time":{

//结束日期

"year":"2020",//年

"month":"11",//月

"day":"11"//日

},

"content":"TargetContent"//目标具体内容

}

}

/\*返回结果\*/

{

"status":true,//新增目标成功时

"msg":"新增目标成功"//返回消息

}

{

"status":false,//新增目标失败时

"msg":"提示词"//失败原因

}

/\*

5.完成（确认）目标

请求路径:/user/target/conplete

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"target\_id":"114514"//目标id

}

/\*返回结果\*/

{

"status":true,//完成目标成功时

"msg":"目标完成成功"//返回消息

}

{

"status":false,//完成目标失败时

"msg":"提示词"//失败原因

}

/\*

7.超时未确认目标（更新目标状态）

请求路径:/user/target/fail

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"target\_id":"114514",//目标id

"update\_time":{

//更新时间

"year":"2020",//年

"month":"11",//月

"day":"11",//日

}

}

/\*返回结果\*/

{

"status":true,//更新目标状态成功时

"msg":"超时未确认目标"//返回消息

}

{

"status":false,//更新目标状态失败

"msg":"提示词"//失败原因

}

/\*

7.查看往期目标

请求路径:/user/target/history

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"target\_id":"114514",//目标id

}

/\*返回结果\*/

{

"status":true,//查找成功时

"target":{//返回目标类

"title":"MyTarget",//标题

"start\_time":{//开始日期

"year":"2020",//年

"month":"11",//月

"day":"11"//日

},

"end\_time":{//结束日期

"year":"2020",//年

"month":"11",//月

"day":"11",//日

},

"content":"TargetContent"//目标具体内容

},

"msg":"查找成功"//返回消息

}

{

"status":false,//查找失败时

"msg":"提示词"//失败原因

}

/\*

8.查看剧情日志

请求路径:/user/line/story/find

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"line\_id":"1",//剧本id

"story\_id":"1919810"//故事编号

}

/\*返回结果\*/

{

"status":true,//查找成功时

"story":{

"story\_id":"1919810",//故事编号

"story\_title":"Title",//故事标题

"story\_content":"StoryContent",//故事内容

}

"msg":"查找成功"//返回消息

}

{

"status":false,//查找失败时

"msg":"提示词"//失败原因

}

/\*

9.目标确认后查看（解锁）剧情

请求路径:/user/target/complete/unlock

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"line\_id":"1",//剧本编号

"story\_progress":"24",//故事编号

}

/\*返回结果\*/

{

"status":true,//解锁成功时

"story":{

"story\_id":"1919810",//故事编号

"story\_title":"Title",//故事标题

"story\_content":"StoryContent",//故事内容

},

"msg":"解锁成功"//返回消息

}

{

"status":false,//解锁失败时

"msg":"提示词"//失败原因

}

/\*

10.添加新剧情到日志

请求路径:/user/line/story/add

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"story\_id":"1919810",//故事id

"target\_id":"114514"//目标id

}

/\*返回结果\*/

{

"status":true,//添加成功时

"story":{

"story\_id":"1919810",//故事编号

"story\_title":"Title",//故事标题

"story\_content":"StoryContent",//故事内容

}

"msg":"添加成功"//返回消息

}

{

"status":false,//添加失败时

"msg":"提示词"//失败原因

}

/\*

11.删除目标

请求路径:/user/target/delete

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"target\_id":"114514",//目标id

}

/\*返回结果\*/

{

"status":true,//删除成功时

"msg":"删除成功"//返回消息

}

{

"status":false,//删除失败时

"msg":"提示词"//失败原因

}

/\*

12.选择剧本

请求路径:/user/line/story/choose

提交参数:

\*/

{

"user\_id":"abc1234",//用户id（微信号）

"line\_id":"1",//剧本id

}

/\*返回结果\*/

{

"status":true,//选择剧本成功时

"msg":"选择剧本成功"//返回消息

}

{

"status":false,//选择剧本失败时

"msg":"提示词"//失败原因

}

# 4 运行设计

## 4.1 运行模块组合

管理员账户：权限管理、用户的添加和删除、剧本增删查改、基本数据设定、注册、修改密码等涉及到了数据库的查询和修改。

普通用户：浏览挑战目标、新增挑战目标、修改挑战目标、查看过往日志、查看剧情、选择剧本、查看剧本、解锁剧本

## 4.2 运行控制

1.用户登录：用户登录时通过微信获得用户信息，无需用户名密码

2.权限管理：管理员用户登录后，可在后台修改用户信息与剧本信息，权限可更改为学生/组长/助教。

5.发布评价：学生用户登录后，可以进行组内或组间的评价。

## 4.3 运行时间

每个操作（点击）反应时间不超过 0.5 秒

# 5. 系统数据结构设计

## 5.1 逻辑结构设计要点

1. 用户表(user)：记录id,open\_id,昵称，头像，性别，创建时间。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 外键 | 描述 |
| id | int | 11 | 是 | Not null |  | 唯一id |
| open\_id | varchar | 255 |  | Not null |  | 微信openid |
| nick\_name | varchar | 50 |  | Not null |  | 昵称 |
| avatar\_url | varchar | 255 |  | Not null |  | 头像地址 |
| gender | varchar | 2 |  | Not null |  | 性别 |
| create\_time | datetime |  |  | Not null |  | 创建时间 |

（2）目标表(challenge)：记录目标的id，内容，起始时间，截止时间，是/否完成，备注，是/否提醒。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 外键 | 描述 |
| id | int | 11 | 是 | Not null |  | 唯一id |
| user\_id | int | 11 |  | Not null | user.id | 外键和用户表id属性对应 |
| challenge\_id | int | 10 |  | Not null |  | 目标编号 |
| contents | varchar | 100 |  | Not null |  | 内容 |
| begin\_time | datetime |  |  | Not null |  | 起始时间 |
| deadline | datetime |  |  | Not null |  | 截止时间 |
| status | bool | 1 |  | Not null |  | 是/否完成 |
| tags | varchar | 100 |  |  |  | 备注 |
| reminder | bool | 1 |  | Not null |  | 是/否提醒 |

(3)日志表(log)：记录剧本编号，日志编号，时间，剧情。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 外键 | 描述 |
| id | int | 11 | 是 | Not null |  | 唯一id |
| user\_id | int | 11 |  | Not null | user.id | 外键和用户表id属性对应 |
| story\_id | int | 10 |  | Not null | story.story\_id | 剧本编号 |
| log\_id | int | 10 |  | Not null |  | 日志编号 |
| time | datetime |  |  | Not null |  | 时间 |
| plot | varchar | 255 |  | Not null |  | 剧情 |

（4）剧本表(story)：记录剧本编号，剧本名，封面，解锁状态，章节数，进度。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 长度 | 主键 | 非空 | 外键 | 描述 |
| id | int | 11 | 是 | Not null |  | 唯一id |
| user\_id | int | 11 |  | Not null | user.id | 外键和用户表id属性对应 |
| story\_id | int | 10 | 是 | Not null |  | 剧本编号 |
| story\_name | varchar | 30 |  | Not null |  | 剧本名 |
| lock | bool | 1 |  | Not null |  | 是/否解锁 |
| cover\_url | varchar | 255 |  | Not null |  | 封面地址 |
| plot\_num | int | 10 |  | Not null |  | 章节数 |
| current\_plot | varchar | 30 |  |  |  | 剧情进度 |

## 5.2 物理结构设计要点

前端访问接口，后端与数据库进行交互，

## 5.3 数据结构和程序的关系

# 6.系统出错处理设计

## 6.1 出错信息

|  |  |
| --- | --- |
| 错误类型 | 错误提示 |
| 输入信息错误 | 信息输入错误，请重新输入 |

## 6.2 补救措施

采用后备技术，当原始系统数据丢失时启动副本建立和启动技术，由于数据库中已经有备份，故在系统出错后可以依靠数据库的恢复功能

## 6.3 系统维护设计

由于系统较小没有外加维护模块，所以维护工作比较简单，仅对数据库做一些基本维护即可