概要设计说明书

# 引言

**本文档指在说明本系统的概要设计的过程和结构化设计中概要设计的结果。预期读者为本软件的使用者和对本系统感兴趣的各界人士。参考资料包括软件工程课程的主课课件**

# 总体设计

## 需求规定

**详见需求规格说明书中对需求的详细规约。**

## 运行环境

**本系统要求在android手机平台上运行，暂时没有android系统的版本要求。为了实现分享的功能还需要微信公众号的支持。**

## 基本设计概念和处理流程

**本系统按照变换型设计划分为输入模块、变换模块、输出模块这三个主要模块。我们通过对数据流图DFD的划分，得到如下结果（有些模糊，清晰图详见附加的文件中的图）**



## 结构

**分别画出三个大模块输入部分，输出部分，变换部分的MSD，和这三个部分的关联。**



|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **模块名称** | **功能描述** | **程序标识符** |
| 1 | Get用户操作请求 | 获得用户操作请求信息 | Request |
| 2 | Get用户登录请求 | 获取用户登录请求信息 | Login |
| 3 | Get手机使用数据 | 获取手机使用数据 | getAppTime |
| 4 | Get用户注册请求 | 获取用注册请求信息 | Login |
| 5 | Get用户设置信息 | 获取用户设置信息 | serDailyGoal |
| 6 | Get微信好友信息 | 获取微信好友信息 | getFriendInfo |
| 7 | Get用户输入 | 获取用户输入信息 | UserLogin |
| 8 | Get注册用户文件 | 获取这测用户文件 | UserLogin |
| 9 | 变换为登陆请求 | 将用户名、密码和注册用户文件变化为登录请求 |  |
| 10 | 变换为注册请求 | 将用户名、密码、确认密码变换为注册请求 |  |
| 11 | 变换为设置信息 | 将每日目标变换为设置信息 |  |
| 12 | Get好友信息 | 获取全部微信好友信息 |  |
| 13 | 筛选好友信息 | 筛选出使用了本系统的微信好友信息 |  |



|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **模块名称** | **功能描述** | **程序标识符** |
| 14 | 统计数据分类 | 对于得到的手使用数据分类统计 | handleData |
| 15 | 生成排行榜 | 根据好友信息和使用时长生成排行榜 | Ranking |
| 16 | 信息用途 | 处理登陆注册请求 |  |
| 17 | 设置目标是否达成 | 根据每日目标和使用时长来得到每日推送的信息 |  |
| 18 | 分享 | 将统计结果封装成药分享的信息 | calcShare |
| 19 | 统计手机使用时长 | 统计手机使用时长 |  |
| 20 | 统计软件使用时长 | 统计软件使用时长 |  |
| 21 | 统计触屏次数 | 统计触屏次数 |  |
| 22 | 登陆 | 检验用户输入的用户名是否匹配，并返回对应的结果 | doSomethingAbout-  LoginAndRegister |
| 23 | 注册 | 检验用户名是否已经被使用，检验密码和确认密码是否一致，并返回对应的结果 | doSomethingAbout-  LoginAndRegister |
| 24 | 目标达成通知 | 根据目标达成通知的标准来生成推送通知 | pushData |
| 25 | 目标未达成通知 | 根据目标未达成通知的标准来生成推送通知 | pushData |



|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **模块名称** | **功能描述** | **程序标识符** |
| 26 | Put统计结果 | 显示统计结果 | postPastData |
| 27 | Put排行榜 | 显示排行榜的结果 | postRankingInfo |
| 28 | Put推送信息 | 显示推送通知 | postDailyInfo |
| 29 | Put分享 | 显示分享的内容 | postSharingInfo |
| 30 | 显示处理 | 将要显示的内容转化为可以显示在手机或微信朋友圈的格式的内容 |  |
| 31 | 手机屏幕显示 | 把经过显示处理的结果显示在手机屏幕上 |  |
| 32 | 微信朋友圈显示 | 把经过显示处理的结果显示微信朋友圈中 |  |

## 功能需求与程序的关系

|  |  |
| --- | --- |
| **功能需求（根据用况图得到）** | **关联模块** |
| 1.查看统计数据 | 3、14、19、20、21、26、30、31 |
| 2.设置 | 5、7、11 |
| 3.登陆 | 2、7、8、9、16、22 |
| 4.注册 | 4、7、10、16、23 |
| 5.通知提示 | 5、7、11、17、24、25、28、30、31 |
| 6.分享 | 18、29、30、32 |
| 7.查看排行榜 | 15、27、30、31 |

# 接口设计

## 用户接口

暂无

## 外部接口

|  |  |  |  |
| --- | --- | --- | --- |
| 外界 | 目的 | 接口 | 备注 |
| 手机硬件 | 获取时间信息 | TimeInfo getTime() |  |
| 获取各app用时 | PhoneUsingTime getAppTime() |  |
| 微信 | 获取好友信息 | WeiChatFriendInfo getFriendInfo() |  |
| 用户 | 登陆请求 | LoginInfo Login(string user, string pwd) |  |
| 查询请求（排行榜，历史数据，分享） | QueryRequest Request(int queryType, string userinfo) |  |
| 设置每日目标 | DailyGoal setDailyGoal(float time) |  |
| 获取排行榜 | RankingInfo postRankingInfo() |  |
| 获取历史数据 | PastData postPastData() |  |
| 分享信息 | ShareInfo postSharingInfo() |  |
| 每日推送信息 | PushInfo postDailyInfo() |  |

## 内部接口

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **编号** | **模块名** | **接口名** | **接口说明** | **参数** | **返回值** | **备注** |
| 输入部分 | 获取手机使用时长数据 | getAppTime() |  | void | PhoneUsingTime |  |
| 获取登陆注册设置请求 | Login() | 登陆注册 | String:user  String:pwd | LoginInfo |  |
| serDailyGoal() | 设置每日目标 | Float:time | DailyGoal |  |
| 获取用户输入 | Request() | （排行榜，分享，每日目标）请求 | Int:queryType  String:userinfo | QueryRequest |  |
| UserLogin() | 用户登录 | String:user  String:pwd | LoginInfo |  |
| 获取用户注册文件 | UserLogin() | 用户注册 | String:user  String:pwd | LoginInfo |  |
| 获取微信好友信息 | getFriendInfo() |  | void | WeiChatFriendInfo |  |
| 输出部分 | 显示统计数据 | postPastData() |  | void | PastData |  |
| 显示推送信息 | postDailyInfo() |  | void | PushInfo |  |
| 显示排行榜 | postRankingInfo() |  | void | RankingInfo |  |
| 朋友圈分享 | postSharingInfo() |  | void | ShareInfo |  |
| 变换部分 | 登陆注册处理 | doSomethingAbout-  LoginAndRegister() | 处理用户登陆注册请求 | String:user  String:pwd | void |  |
| 排行榜生成 | Ranking() |  | UserInfo:usrinfo | RankingInfo |  |
| 分享 | calcShare() |  | UserInfo:usrinfo | ShareInfo |  |
| 推送 | pushData”() |  | UserInfo:usrinfo  PhoneUsingTime:pt  DailyGoal:dailygoal | PushInfo |  |
| 统计数据 | handleData() |  | UserInfo:usrinfo  PhoneUsingTime:pt | void |  |

# 系统数据结构设计

## 逻辑结构设计要点

|  |  |
| --- | --- |
| 数据 | 数据结构 |
| APP 用时： | structAppTime{ string name; float time;} |
| 查询请求： | structQueryRequest{ intqueryType; UserInfouserinfo; } |
| 用户信息： | structUserInfo{ string usename; } |
| 历史数据： | structPastData{ List\* DailyTime; List\* AppTime; } |
| 排行榜信息： | structRankingInfo{ intorder; string username} |
| 分享信息： | structShareInfo{ float UsedTime; RankingInforank; } |
| 时间信息： | structTimeInfo{ float SystemTime; } |
| 手机用时信息： | structPhoneUsingTime{ float TotalTime; List\* AppTime; } |
| 每日推送： | structPushInfo{ bool finished; } |
| 登录请求： | structLoginInfo{ string username; string password; string others; } |
| 每日目标： | structDailyGoal{ float goal; } |
| 好友信息： | structFriendInfo{ string friendname; float time; } |
| 微信好友： | structWeiChatFriendInfo{ FriendInfo\* friendlist; } |

使用的数据库信息：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **编号** | **字段名称** | **类型** | **长度** | **是否可为空** | **主键** | **说明** |
| 1 | 用户 | UserInfo | 20byte | 否 | 账号 | 用户使用数据 |
| 2 | 数据 | PastData | 1000byte | 否 | 账号 |  |

## 物理结构设计要点

本系统存储与用户本地的文件只有三个：设置文件，历史信息，用户信息。

|  |  |  |
| --- | --- | --- |
| 文件 | 对应数据 | 对应本地文件 |
| 设置文件 | DailyGoal | Setting.data |
| 历史信息 | PastData | Past.data |
| 用户信息 | UserInfo | Userinfo.data |

# 运行设计

**本部分需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。在课程实践中本部分可不必考虑。**

# 系统出错处理设计

**本部分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，故障出现后可能采取的变通措施，以及为了系统维护的方便而在程序内部设计中作出的安排。在课程实践中本部分可不必考虑。**