

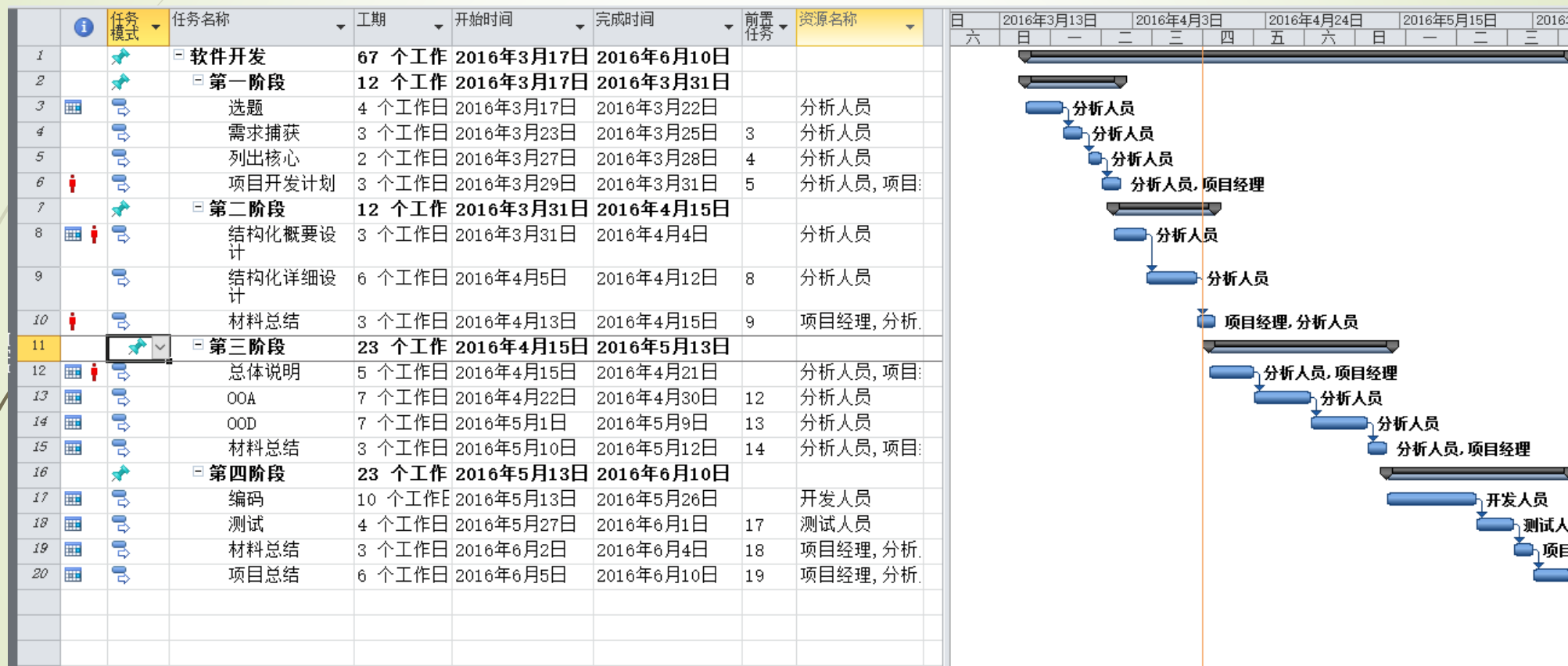
# 软件工程第二次课堂报告

恋机指数：手机实时统计分享系统

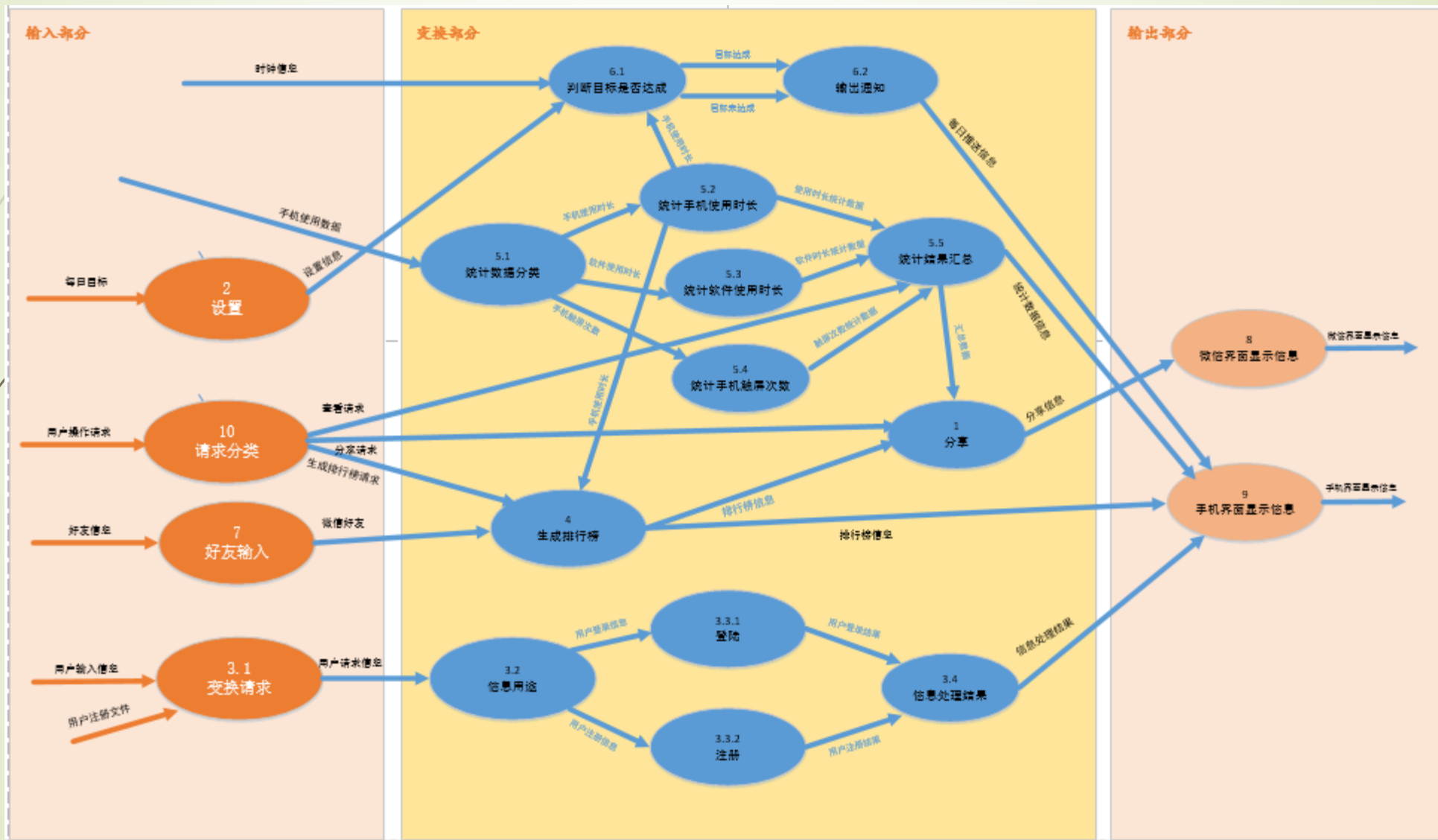
## 第二次实践分工

- 尤安升（项目经理）：参与概要设计，对整个实践过程负责，验收全部内容
- 段富尧（产品经理）：参与概要设计，课堂报告PPT的完成
- 陈瑞麟（架构师）：参与详细设计的模块伪码设计
- 黄挺：参与概要设计的体系结构设计，最终概要设计报告的书写
- 孔维：参与概要设计的接口设计和数据设计，MS Project文件的更新
- 赵宇航：参与详细设计，画出模块程序流程图以及N-S图
- 卜天童：参与详细设计，完成最后的详细设计报告

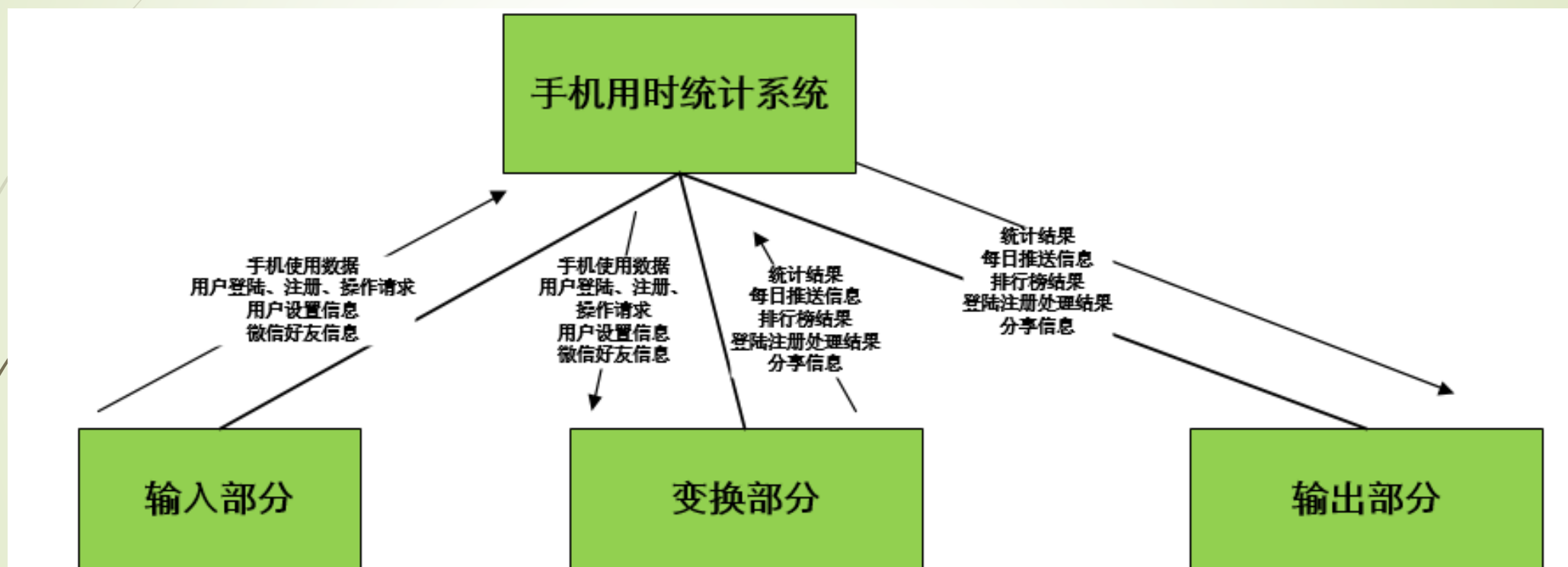
# 项目执行情况的MS project文件



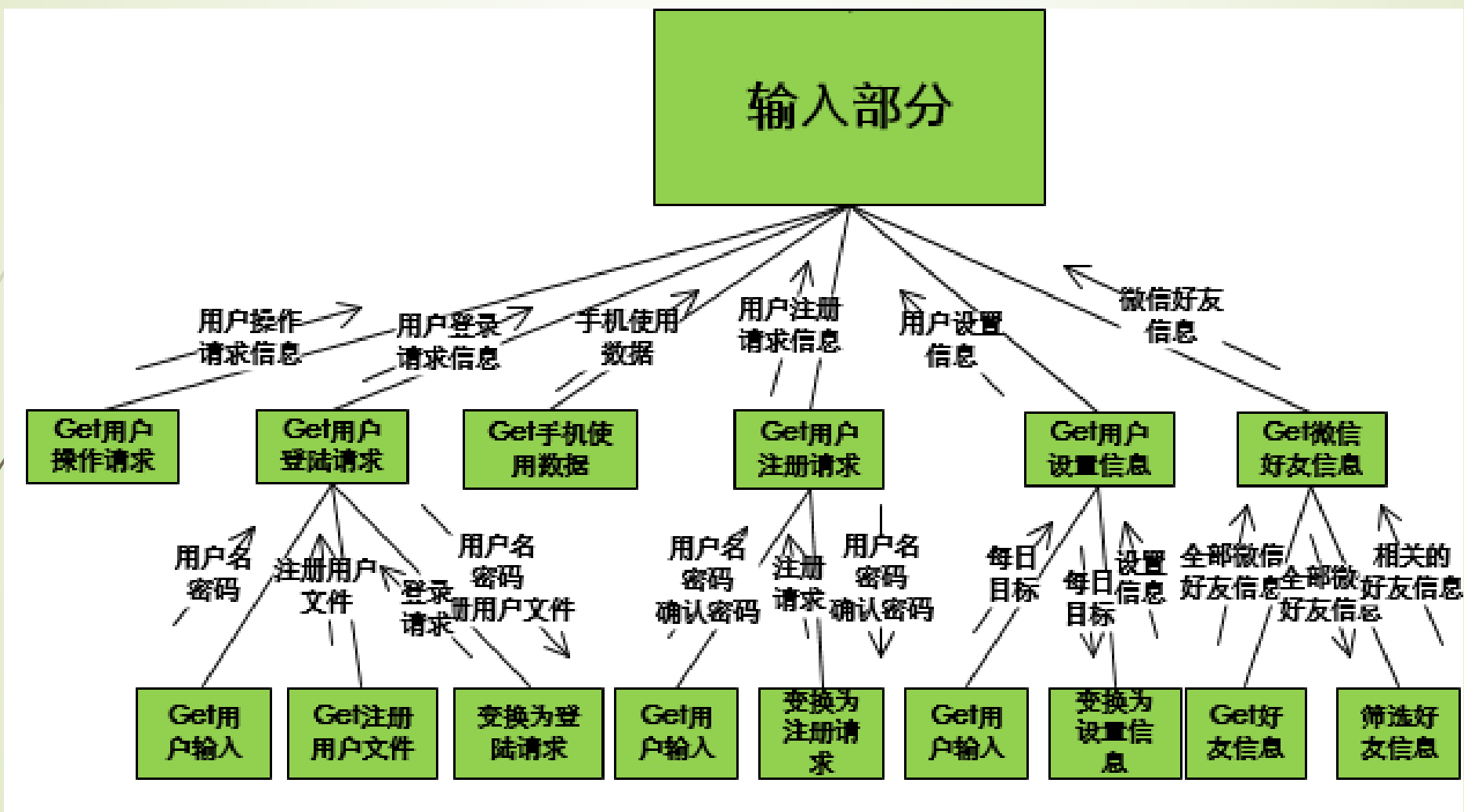
# 概要设计（DFD）



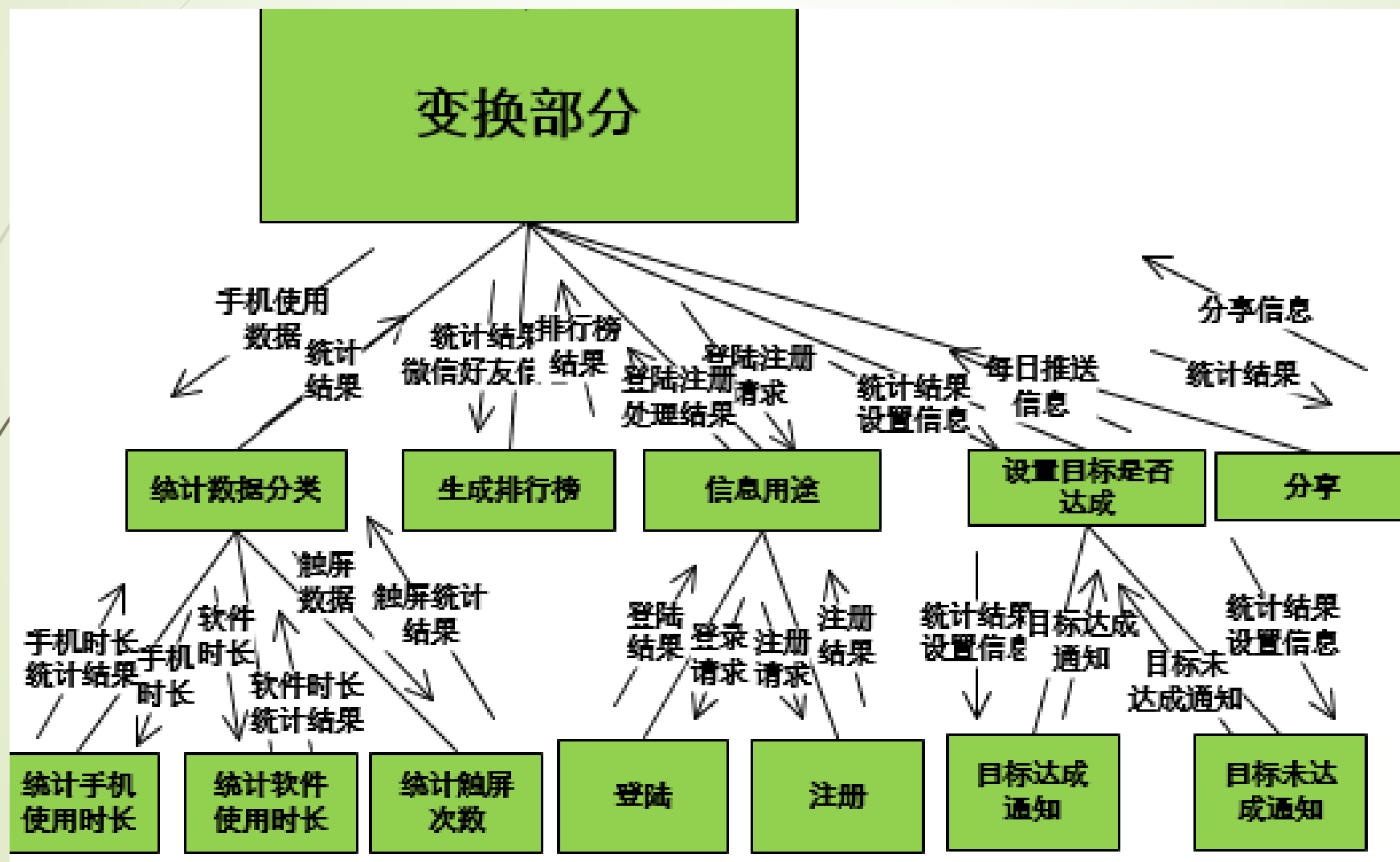
## 概要设计（初始MSD）



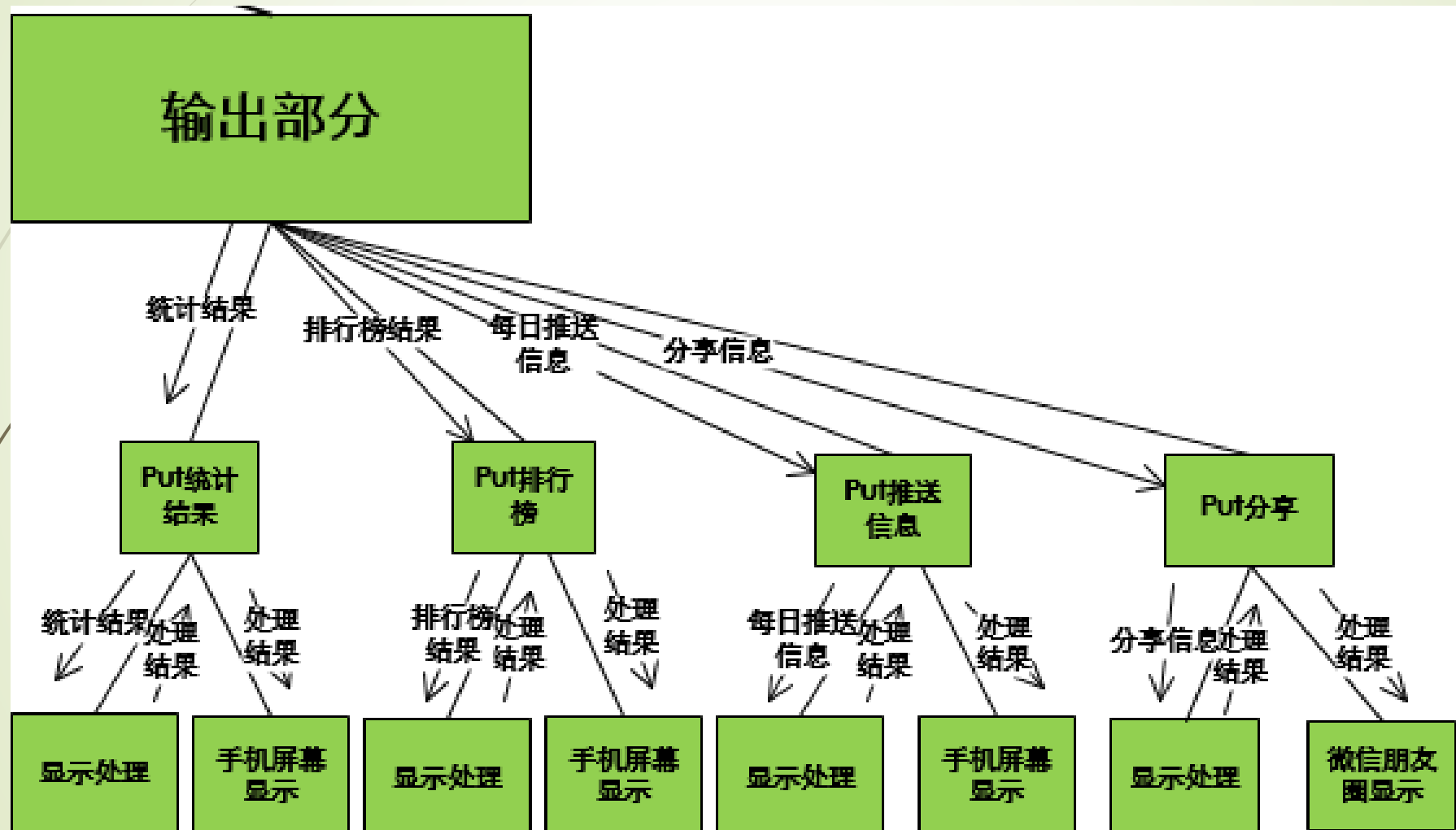
## 概要设计（初始MSD）



## 概要设计（初始MSD）

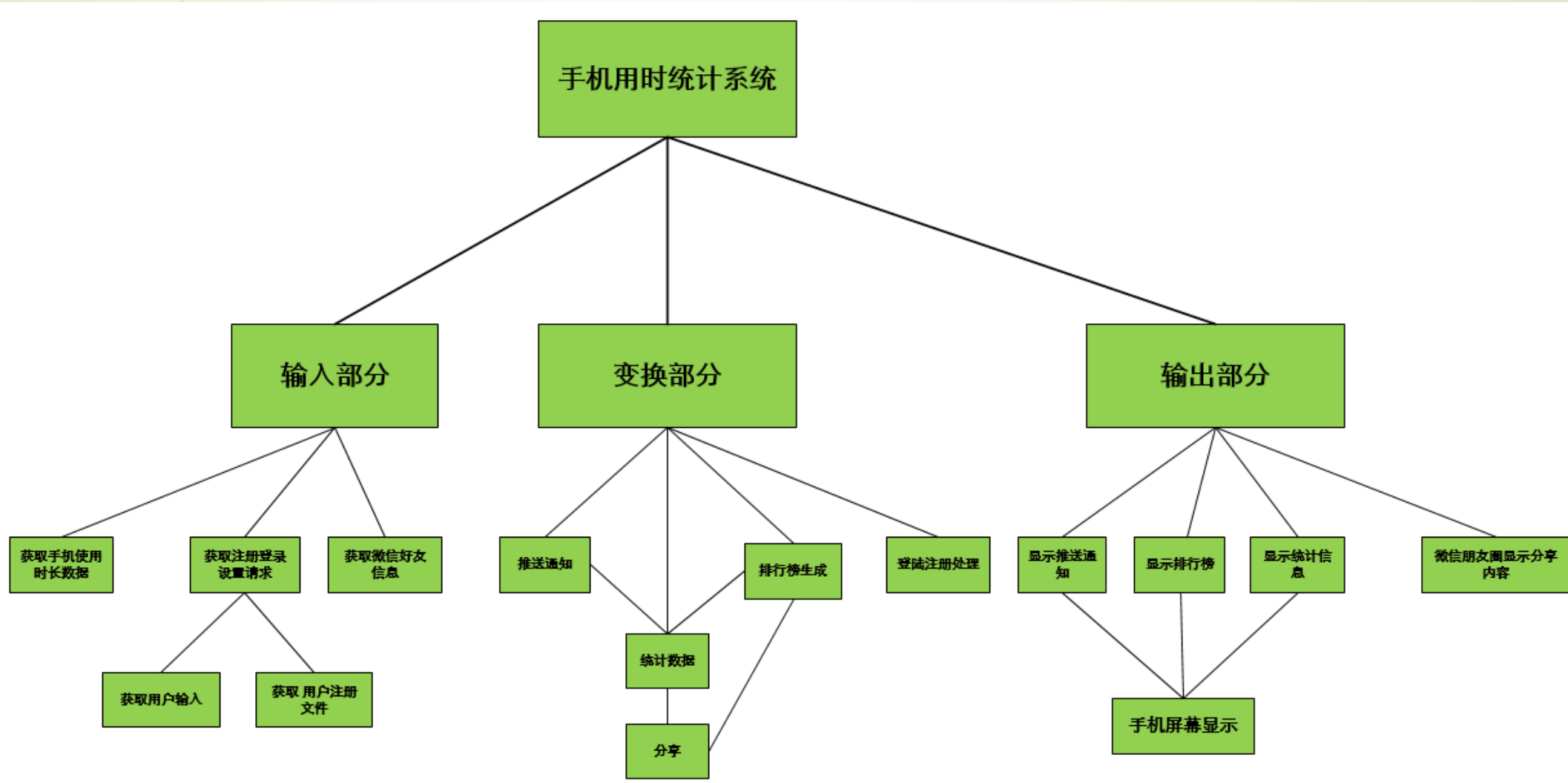


## 概要设计（初始MSD）





## 概要设计（精化后的 MSD）



# 接口设计:

## 外部接口:

### 与微信:

- 好友信息: `WeiChatFriendInfo getFriendInfo()`

### 与手机底层:

- 时间信息: `TimeInfo getTime()`
- 手机用时信息: `PhoneUsingTime getAppTime()`

### 与用户:

- 查询请求(含排行榜, 历史数据, 分享): `QueryRequest Request( int queryType, string userInfo )`
- 登录请求: `LoginInfo Login(string username, string password)`
- 每日目标: `DailyGoal setDailyGoal(float time)`
- 排行榜信息: `RankingInfo* postRankingInfo()`
- 历史数据: `PastData postPastData()`
- 分享信息: `ShareInfo postSharingInfo()`
- 每日推送信息: `PushInfo postDailyInfo()`

# 接口设计:

- 内部接口:

- 设置文件:

- 仅有一个DailyGoal结构, 可存于本地setting.data文件中

- 历史信息:

- PastData结构记录所有需要的历史信息, 在本地可存于past.data中

- 用户信息:

- UserInfo结构, 在本地存于userinfo.data文件中

# 接口设计:

## 内部接口:

### ➤ 输入部分:

- Get 手机使用时长数据: `PhoneUsingTime getAppTime()`
- Get 注册登录设置请求:  
`LoginInfo Login(string username, string password) // 用户登录、注册`  
`DailyGoal setDailyGoal(float time) // 用户设置每日目标`
- Get 用户输入:  
`QueryRequest Request( int queryType, string userInfo ) // 用户请求`  
`LoginInfo UserLogin(string username, string password) // 用户登录`
- Get 用户注册文件: `LoginInfo UserLogin(string username, string password) // 用户注册`
- Get 微信好友信息: `WeiChatFriendInfo getFriendInfo()`

### ➤ 输出部分:

- 显示统计信息: `PastData postPastData()`
- 显示推送信息: `PushInfo postDailyInfo()`
- 显示排行榜: `RankingInfo* postRankingInfo()`
- 微信朋友圈显示分享内容榜: `ShareInfo postSharingInfo()`

# 接口设计:

## 内部接口:

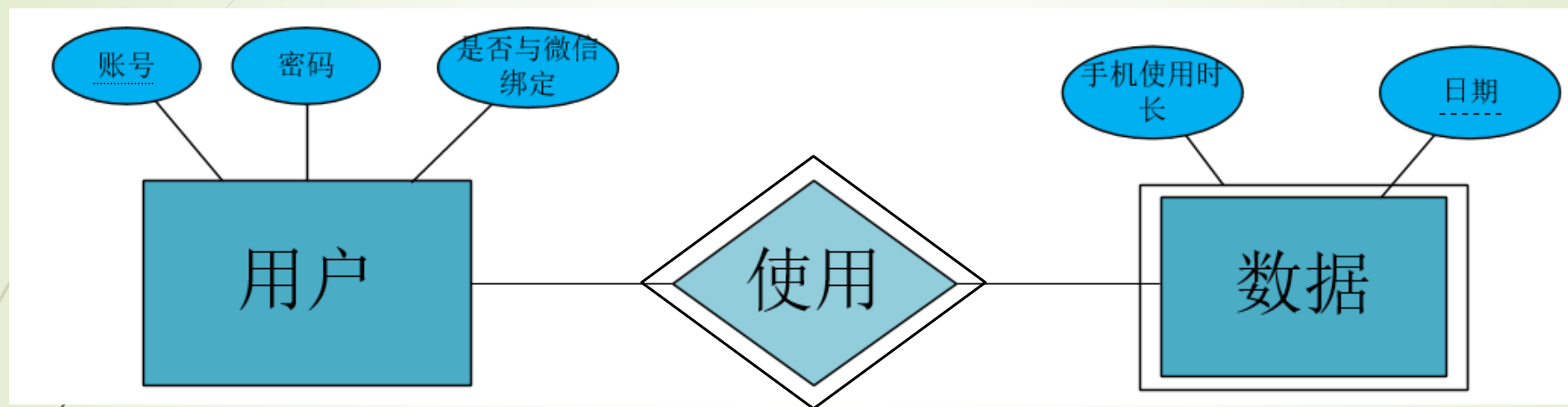
### 变换部分:

- 登录注册处理: `void DoSomethingAboutLoginAndRegister(string username, string password)`
- 排行榜生成: `RankingInfo* Ranking(UserInfo userinfo)`
- 推送: `PushInfo TongJiAndTuiSong(UserInfo userinfo, PhoneUsingTime phonetime, DailyGoal usergoal)`
- 分享: `ShareInfo Share(UserInfo userinfo)`
- 统计数据与推送: `PushInfo TongJiData(UserInfo uesrinfo, PhoneUsingTime phonetime, DailyGoal usergoal)`  
`Void HandleData(UserInfo userinfo, PhoneUsingTime phonetime)`
- 排行榜生成: `RankingInfo* calcRanking(FriendInfo* usersInfo)`
- 分享: `ShareInfo calcShare(UserInfo userinfo)`

## 数据设计：

数据	数据结构
APP 用时：	struct AppTime { string name; float time; }
查询请求：	struct QueryRequest { int queryType; UserInfo userinfo; }
用户信息：	struct UserInfo { string username; }
历史数据：	struct PastData{ List* DailyTime; List* AppTime; }
排行榜信息：	struct RankingInfo{ int order; string username }
分享信息：	struct ShareInfo { float UsedTime; RankingInfo rank; }
时间信息：	struct TimeInfo { float SystemTime; }
手机用时信息：	struct PhoneUsingTime { float TotalTime; List* AppTime; }
每日推送：	struct PushInfo { bool finished; }
登录请求：	struct LoginInfo { string username; string password; string others; }
每日目标：	struct DailyGoal { float goal; }
好友信息：	struct FriendInfo { string friendname; float time; }
微信好友：	struct WeiChatFriendInfo { FriendInfo* friendlist; }

## 数据库设计：



### 用户表

用户名	密码	是否与微信绑定
dfy1995dfy	*****	True

### 数据表

用户名	日期	手机使用时长
dfy1995dfy	2016/4/14	15364.1000s



# 文件设计

## 文件名称:

history.data

Setup.data

## history.data

日期	手机用时时长	软件1用时时长	软件2用时时长	触屏次数
2016/4/14	12345.1000s	1234.1000s	2356.3456s	150

## SetUp.data

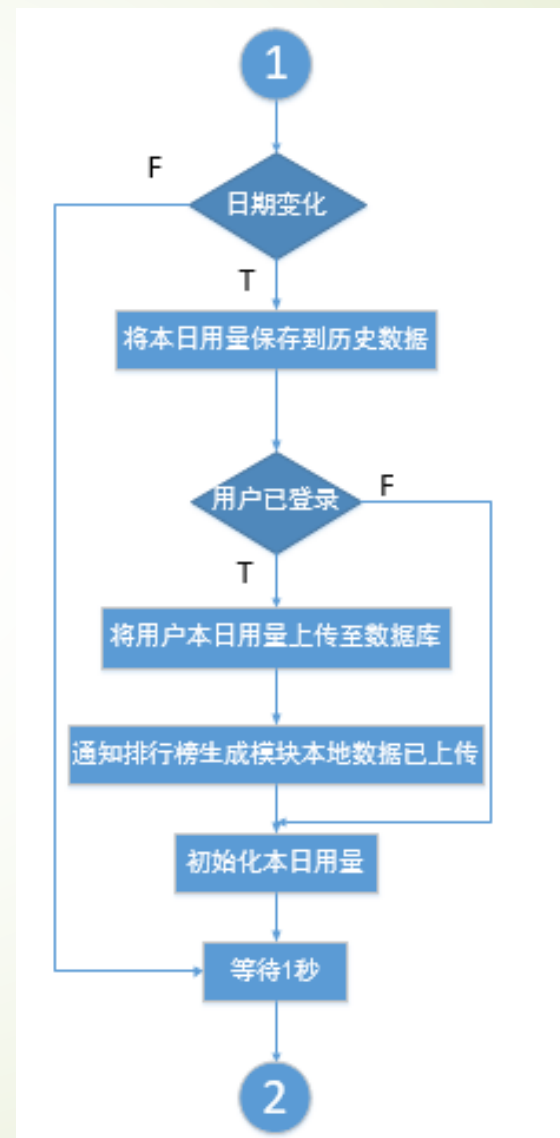
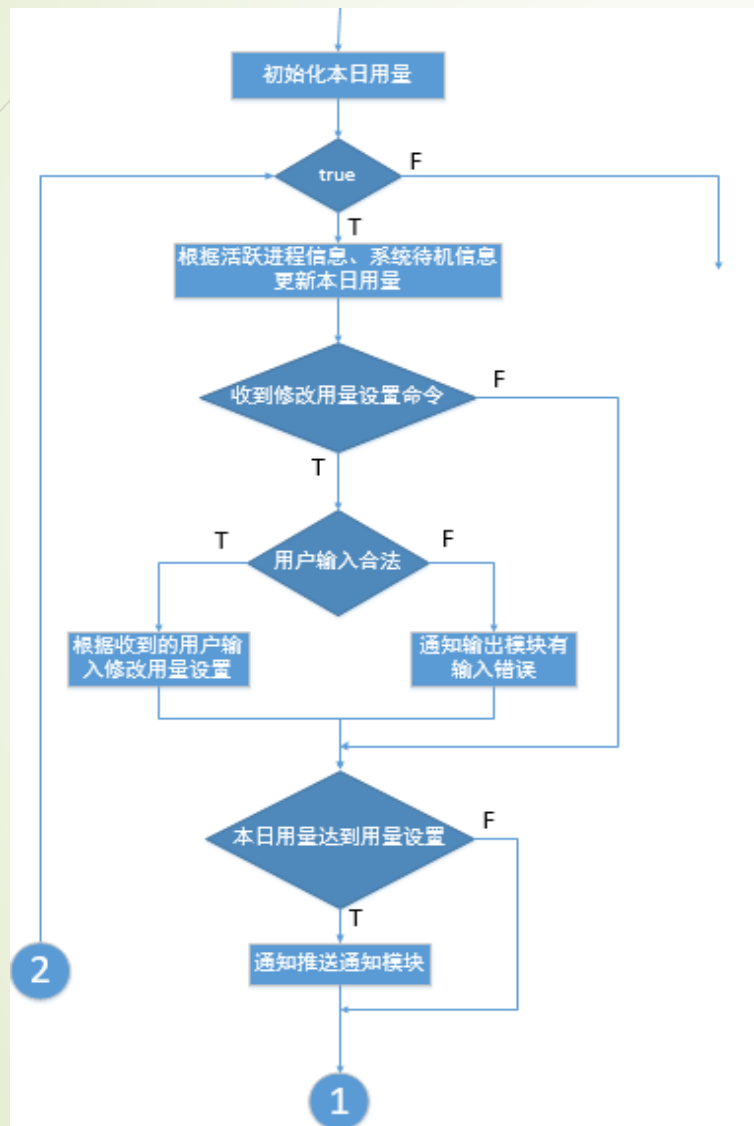
日期	目标时间
2016/4/14	14400.0000s



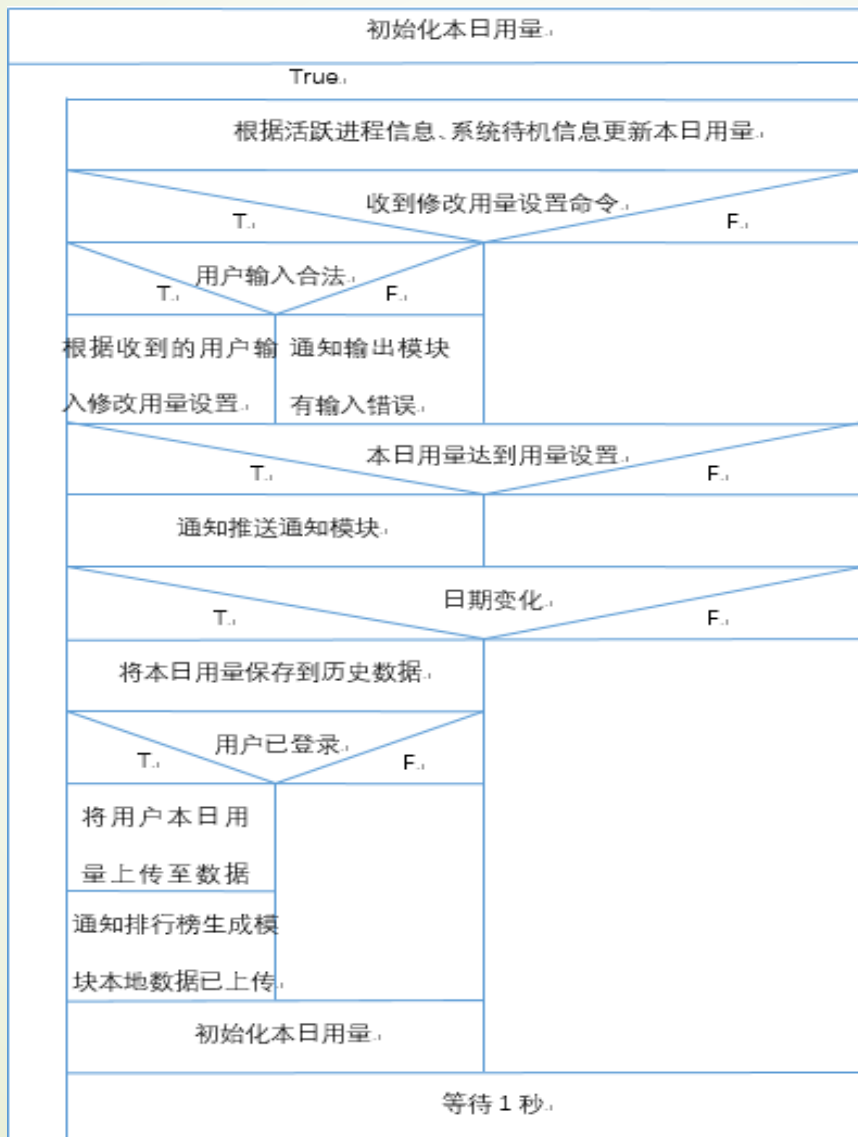
## 详细设计（伪码实例：数据统计）

```
begin
    初始化本日用量
    while true begin
        根据活跃进程信息、系统待机信息更新本日用量
        if 收到修改用量设置命令 then
            if 用户输入合法 then 根据收到的用户输入修改用量设置 else 通知输出模块有输入错误
        if 本日用量达到用量设置 then 通知推送通知模块
        if 日期变化 then begin
            将本日用量保存到历史数据
            if 用户已登录 then begin
                将用户本日用量上传至数据库
                通知排行榜生成模块本地数据已上传
            end
            初始化本日用量
        end
        等待1秒
    end
end
```

## 详细设计（程序框图实例：数据统计）



## 详细设计（N-S图实例：数据统计）



## 详细设计（伪码实例：生成排行榜）

```
begin
```

```
  while true
```

```
    if 本地数据已上传
```

```
    then
```

```
      begin
```

根据微信好友信息从数据库获取昨日用量信息，并通知他们好友数据已更新

将好友用量排序得到排行榜

输出排行榜

```
      end
```

```
    else if 好友数据已更新 then begin
```

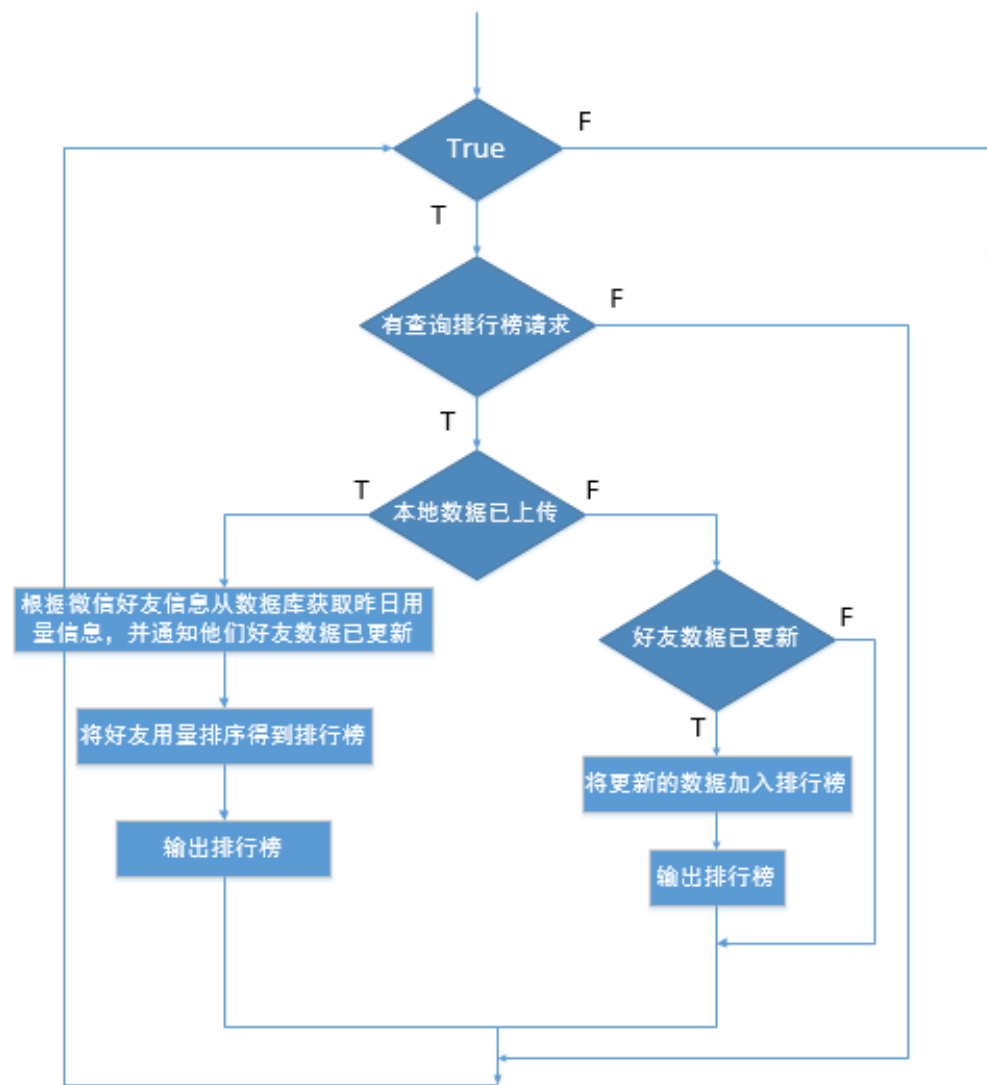
将更新的数据加入排行榜

输出排行榜

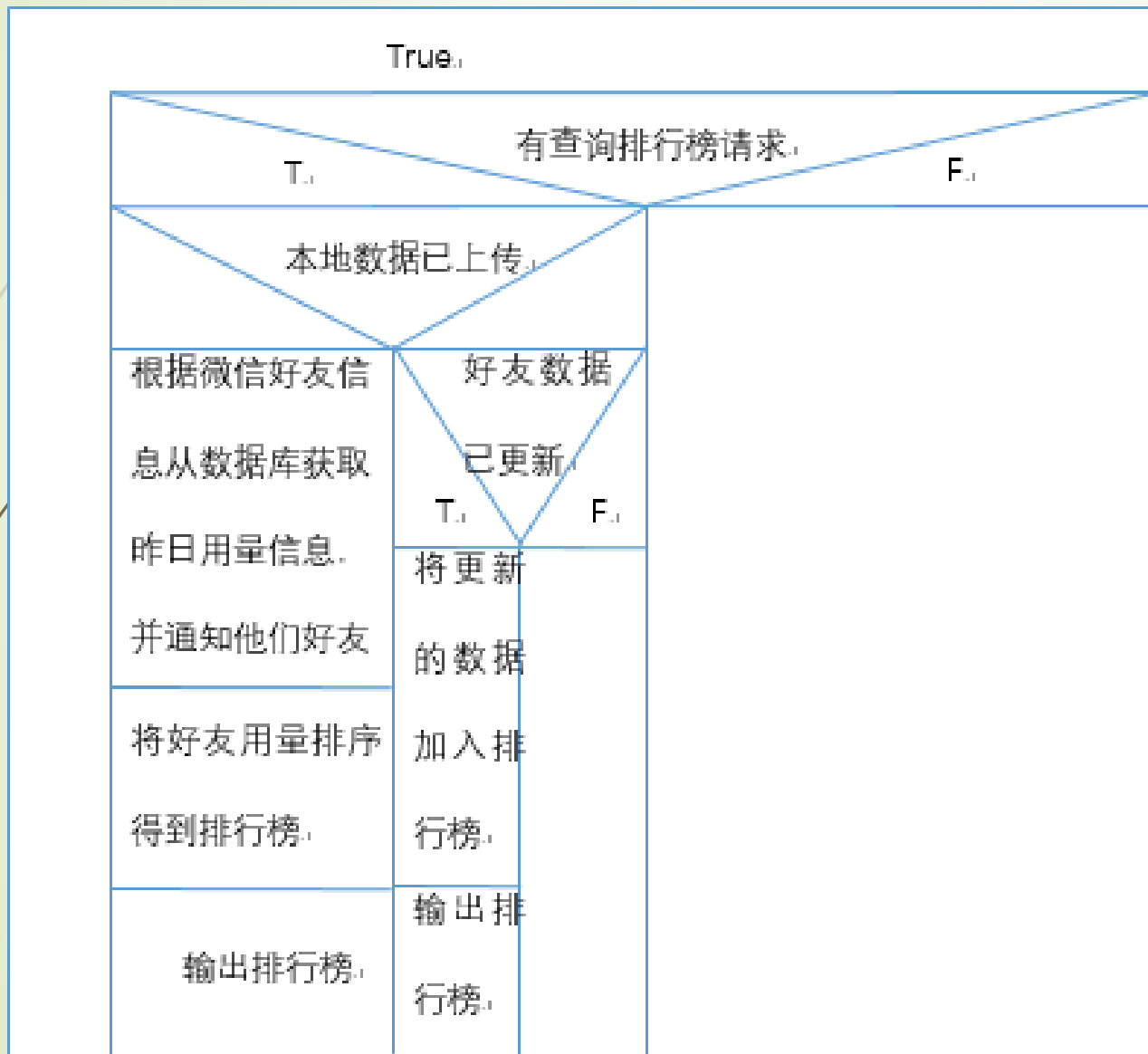
```
    end
```

```
end
```

## 详细设计（程序框图实例：生成排行榜）



## 详细设计（N-S图实例：生成排行榜）





*Thank You*