

# 武汉大学计算机学院

## 本科生课程设计报告

### 第 7 组需求规格说明

专 业 名 称： 计算机科学与技术

课 程 名 称： 软件工程

团 队 名 称： 2020HYSE07

指 导 教 师： 伍春香 副教授

团 队 成 员 一： 孔德昱（2020302191296）

团 队 成 员 二： 燕山楠（2020302191668）

团 队 成 员 三： 杜忠璠（2020302041100）

团 队 成 员 四： 王新喆（2020302142180）

团 队 成 员 五： 洪巍骏（2020302191852）

团 队 成 员 六： 苏睿懿（2020302191286）

二〇二三年四月

# 目 录

1 文档标准 .....	4
2 引言 .....	5
2.1 目的 .....	5
2.2 范围 .....	5
2.3 术语定义 .....	5
2.4 综述 .....	6
3 总体描述 .....	7
3.1 产品描述 .....	7
3.2 产品功能 .....	7
3.2.1 个人时间管理 .....	7
3.2.2 计划进度监控 .....	8
3.2.3 团队合作协调 .....	9
3.3 用户特点 .....	9
3.4 约束 .....	10
3.4.1 非技术性约束 .....	10
3.4.2 技术性约束 .....	10
3.5 假设和依赖关系 .....	11
3.5.1 项目假设 .....	11
3.5.2 依赖关系 .....	11
4 具体需求 .....	12
4.1 外部程序接口 .....	12
4.1.1 用户界面 .....	12
4.1.2 硬件接口 .....	14
4.1.3 软件接口 .....	14
4.2 功能需求 .....	14
4.2.1 整体分析 .....	14
4.2.2 个人时间管理 .....	20
4.2.3 计划进度监控 .....	26

4.2.4 团队合作协调 .....	27
4.3 性能需求 .....	28
4.3.1 处理能力 .....	28
4.3.2 响应时间 .....	28
4.4 交付与验收要求 .....	29

# 1 文档标准

本组选取 GB/T 9385-2008《计算机软件需求规格说明规范》(部分)作为本次小组实践的需求规格说明文档标准。

## 2 引言

### 2.1 目的

本文档的编写目的是详细介绍 Tomado 日常事务 APP 所解决的需求。本文档将通过文字叙述结合数据流图、E-R 图等传统建模方法以及状态转换图等面向对象建模方法向客户以及开发人员介绍本产品，使用户能够进一步确认产品的确切需求并使开发人员能够根据此文档进行设计编码。

### 2.2 范围

本文档的预期读者有客户代表、产品开发人员以及项目经理等。

### 2.3 术语定义

本需求规格说明文档中涉及到的术语如表 2.1 所示：

表 2.1 术语表

术语	具体解释
禁忌列表	用户希望约束自己的事情，如某个不良的生活习惯
番茄钟	番茄钟工作法是一种效率较高的时间管理方法。标准的番茄钟是“25+5”的形式，即 25 分钟专注在某件事情上，然后花 5 分钟休息。但是每个人可以根据自己的专注时间和待办实现所花费的时间来调整，比如“30+5”或“45+10”的形式。

## 2.4 综述

在本需求规格说明书中，每章的内容如下：

在第三章总体描述中，我们将对我们的产品进行概述，包括产品描述、产品功能、用户特点、约束、假设和依赖关系五个方面，此章节可以让文档读者对我们的产品有一个整体的了解。

在第四章具体需求中，我们将从外部程序接口、功能需求、性能需求和交付与验收需求四个方面介绍我们产品的具体需求。其中功能需求是最重要的一节，我们将在这一节中采用如 E-R 图的结构化建模方法和如状态转换图的面向对象建模方法等多种建模方法，并结合文字描述对产品的需求进行具体描述。

## 3 总体描述

### 3.1 产品描述

本产品（Tomato）是一款集个人时间管理、计划进度监控以及团队合作协调三大功能为一体的日常事务类软件，为学习、工作与生活中的计划管理等问题提供了可行性高、创新性强的解决方案。

### 3.2 产品功能

本产品深挖用户需求，检索集成后用于指导软件开发，以更好地贴合用户日常生活工作的需要。本产品主要有个人时间管理、计划进度监控、团队合作协调三大功能。

#### 3.2.1 个人时间管理

针对个人时间管理方面，产品推陈出新，实现了正反两个方向的时间管理与计划制定功能。

经调查发现，用户在使用个人计划管理软件时，其目的往往在于督促自身，实现对自己时间的合理把控，达成自我价值的提高。而针对这一点，市面上部分软件给出的解决方案，却仅仅只是提供某计划的制定与提醒，这使得用户在使用产品的过程中缺乏反馈，不能产生足够的成就感来促使其目标的达成；使其 app 逐渐被用户忽略，而在某一次软件清理过程中被卸载。原因在于这些产品与用户使用该类产品的初衷相去甚远。而小部分的产品虽然给予用户部分的反馈（如番茄钟的计划、Forest 专注森林的“种树”等），却只是在特定时间内起效果，缺乏对于某个短期计划针对性地给出相应的反馈。更进一步的，没有任何一款现有软件产品，能够直接实现用户自我约束的需求。如禁止某项娱乐活动、改掉某个不良习惯等。这片蓝海市场，为我们产品的加入提供了极好的市场环境 with 用户群体。

基于此，我们的产品设计了一个提供了正反两个方向的计划制定与监控功能的解决方案。

对于正向的计划监控，软件需要用户输入计划的名称、内容；计划执行的时间以及间隔；在每次计划执行的周期内，是否有特定的执行时间以及相应的时间区

间是什么；计划达成后对于自身的奖励等。对于短期计划而言，软件规定了计划设定的期限在一周至三个月之间，较为合理地时间区间，既不会让短期计划实现周期过长，导致用户产生厌烦心理，又不至于太短而导致计划无法实现；而计划最后强制用户设定的奖励，本质上强化了用户对于计划执行的动力，同时在计划达成时给予足够的成就感，提升用户的使用体验。

对于反向的自律性规约，产品提出了“禁忌列表”的概念。所谓“禁忌列表”，指的是用户希望约束自己的事情，如改掉某个不良的生活习惯，亦或者控制生活成本而禁止购买的物品等。我们的产品推出禁忌列表的初衷，在于帮助用户更好地规划自己的生活。如果说正向的计划制定，是从正面激励用户实现生活工作的管理，通过促进用户成就感、满足感的达成，激励用户实现自我价值的提升；那么禁忌列表则是从反方向规约用户的生活习惯，通过增强用户愧疚感、良心层面上谴责，来帮助用户更好地管理自己的生活。

具体而言，禁忌列表功能如下：设定禁忌列表事项，设定准则类似于正向短期计划的制定；计划后的奖励制定则换成奖惩制定，通过正反两方面的奖惩结合，增强用户的计划反馈，督促用户完成自己制定的目标；未完成的计划周期，将会反馈在黑色的未完成进度条上，而完成的计划则会显示金色的达成进度条（单位为该计划的计划周期，如以两天为一个单位）。具体的进度条功能将在下一节中介绍。

### **3.2.2 计划进度监控**

针对计划进度监控方面，产品不断创新，在短期计划与长期计划上均运用了进度条的提示方式，极大增强了用户的使用体验与反馈。

分析这类软件的使用需求可知，用户关注度最高的点，在于软件给出的反馈强弱，以及这样的反馈能否很好地督促自己完成计划。普通软件给出的解决方案，往往过于简洁而无法提供足够强大的反馈，亦或是反馈效果过强喧宾夺主，转移了用户的注意力，盖过了用户完成计划的初衷。

基于此，我们的软件在长期与短期计划上，都采用了进度条的解决方案，来解决此类问题。一方面，不同的进度条颜色，较为明显地区分出用户达成计划的周期数，在视觉上给用户带来了比较强烈的震撼，从而激发出用户自身的反馈；另一方面，不同于“种树”等显示方式，计划条的设计，不至于让用户沉浸于计划完成度本身的显示上，避免了软件本身的喧宾夺主、干扰用户的计划执行，很好地平



衡了计划显示的简洁感与力量感。

在短期计划上，进度条被设计成金灰黑三种颜色：金色的进度条标识着计划完成的周期数，而黑色的则表示目前计划错过的未完成的周期数，而剩余周期数则用灰色进度条展示。至于用死亡一般醒目的黑色，还是代表成功人生的金色，来填充未来的进度条，一切均由用户的行为所决定。

在长期计划上，进度条被设计成红绿灰三种颜色：绿色的进度条标识着计划完成的周期数，而红色进度条表示目前计划错过的未完成的周期数，而剩余周期数则用灰色进度条展示。与短期计划不同的是，长期计划未必需要有如金色与黑色一般极端的视觉效果来提醒用户：短暂的辉煌之金，可能导致用户一时的心高气傲，影响长期计划的执行；寂静的死亡之黑，也可能导致用户因为过于厚重的任务而破罐子破摔，放弃接下来的长期计划。相反地，三个月以上的长期计划，需要一些较为平和而又不失警醒的颜色，来督促用户执行计划。因此，倘若用户未完成今天的计划，那么进度条将会出现对应的警戒红来警示用户，若是在一段时间内都能很好地执行着长期计划，则一段象征着生气与希望的绿色将提醒用户继续前行。看似简单的设计，实际效用则颇为关键。

### **3.2.3 团队合作协调**

针对团队合作协调方面，产品在现有软件的基础上进行改进创新，提升了用户的交互程度与使用体验。

在实际生活中，大部分团队低效合作的原因，在于沟通效率的低下，以及团队成员之间互相拖延、推卸责任导致整体计划的延后。本质上，是由于没有一个较为完善的督促体系与责任追究体系导致的。

基于此，我们的软件设计出分段进度条的方式。对于每个团队成员而言，其它成员的进度条将会有选择地展示在界面上，以帮助其更好地执行自己的任务。对于一个团队的规划，每个阶段中，成员都能够查看到自己任务的前置任务，以及其成员进度条的完成程度，以更好帮助成员合理规划时间以完成进度，大大提高了团队的合作效率。

## **3.3 用户特点**

产品面向的用户，主要是具有以下特质的四类人群。

1.希望合理规划自己工作、学习的计划执行者。这一类人群的特点在于，其短期目标多样且繁杂，仅凭自身的力量很难记住所有工作上的事务，需要一定的机制提醒自己按时完成某个短期计划，从而需要一款合适的软件帮助自己实现短期计划的制定与执行。

2.希望严格规约自己，提升生活质量的苦行僧人士。这一类人群的特点在于，其通过自身的力量，很难改掉某些习惯与不好的习性，希望通过外界的方式帮助自己约束自身，从而提升自己的自律程度，获得生活体验的提升。

3.长期不懈致力于某个目标，却苦于缺乏动力的长征者，以及希望开启新征程的探险家。这一类人群的特点在于，其自身的矛盾性导致他们很难凭借自己的力量，长期追求于某个目标，需要外界提供一定的动力，激励其不断追求自我的提升。

4.希望提升团队合作效率的团队领袖与团队成员。这一类人群苦于自己的团队合作效率不高，难以在短时间内高效地合作完成任务，希望通过某个软件促进团队之间成员的沟通交流，使成员之间共享进度，让成员的任务行为更透明化，从而带来团队整体效率的提升。

### 3.4 约束

本项目的约束可以分为非技术性约束与技术性约束两类，具体如下：

#### 3.4.1 非技术性约束

1.开发时间约束。作为软件工程课程的大作业，本项目的开发时间有着明确的限制，因此在如此有限的时间内，开发出一款达到期望要求的产品具有一定难度。

2.开发资金约束。作为一款支持一定规模人群在线合作的项目，势必需要云上服务的支持，而该部分服务的支持需要一定的成本。

#### 3.4.2 技术性约束

1.平台兼容性约束。尽管 uniapp 的开发可以轻松移植至手机端的 app 或小程序，但基于该类项目开发的 ui 并不能非常好的适用于 PC 网页端。所以为了兼容 PC 端开发，平台兼容性约束问题亟待解决。

2.同步请求约束。基于数据库的独立性约束，相关原子数据不能被同时被请求修改，所以会在一定程度上影响多用户协同工作的效果。而分段式独立修改则会

影响数据库的存储结构，使存储复杂化，因此存在同步请求约束问题。

### 3.5 假设和依赖关系

#### 3.5.1 项目假设

- 1.项目组成员都具有一定的前后端分离式开发经验，可以独立完成部分功能的实现
- 2.现有的云资源可以提供充足的并发支持，以实现多用户在线同步协同日志更新
- 3.相关通信配置可以满足低延迟的要求
- 4.单元测试工作可以由部分成员进行有效完成
- 5.全部代码充分考量各种可能出现的异常，保证程序不会崩溃

#### 3.5.2 依赖关系

依赖可以分为项目的内部依赖与外部依赖两部分。就内部依赖而言，虽然诸如番茄钟、计划、禁忌以及协同开发部分的大致开发工作是并行进行的，而在这些内容开发前，需要首先进行相应的数据库、云资源配置以使其顺利运作，通过单元测试后，都需要等待相关后端 api 的完整开发才可以接入系统，而接入系统后，我们仍然需要等待前端 button 以及请求配置完毕后才可以进行实际测试。上述云资源与数据库属于外部依赖。示意图如图 3.1 所示：

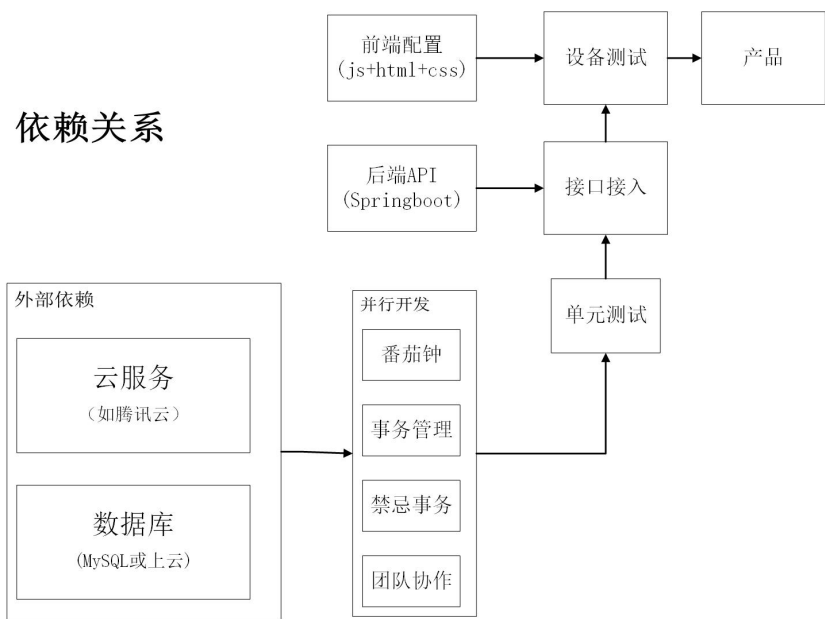


图 3.1 依赖关系

## 4 具体需求

### 4.1 外部程序接口

#### 4.1.1 用户界面

在《软件工程》课程中，日常事务 APP 的界面设计应满足以下需求：

- **易用性：**界面设计应简洁明了，用户能够快速理解和掌握 APP 的功能和操作方法。尽量避免过多的操作步骤和复杂的界面布局。
- **一致性：**界面设计应保持风格和操作逻辑的一致性，使用户在不同页面和功能之间能够自然过渡，提高使用体验。
- **可访问性：**考虑到不同用户的需求，如视力障碍、色盲等，界面设计应提供适当的文字大小、对比度和颜色搭配，以确保所有用户能够顺利使用 APP。
- **反馈性：**界面设计应提供充分的反馈，如按钮按下时的视觉变化、操作成功或失败的提示信息等，帮助用户了解当前操作状态和结果。
- **引导性：**对于新用户或复杂功能，界面设计可以提供引导，例如首次使用时的引导教程、功能介绍、帮助文档等，帮助用户快速上手。
- **适应性：**界面设计应具有良好的适应性，能够适应不同设备（如手机、平板等）、屏幕尺寸和分辨率，确保在各种环境下都能提供良好的使用体验。
- **个性化设置：**界面设计可以提供一定程度的个性化设置，如主题颜色、字体样式等，让用户根据自己的喜好进行调整，提高用户黏性。
- **功能分组：**将相关的功能模块进行分组，使界面布局更加有序，便于用户找到和使用相应功能。

根据以上需求，我们设计了用户界面，示例如图 4.1 所示。

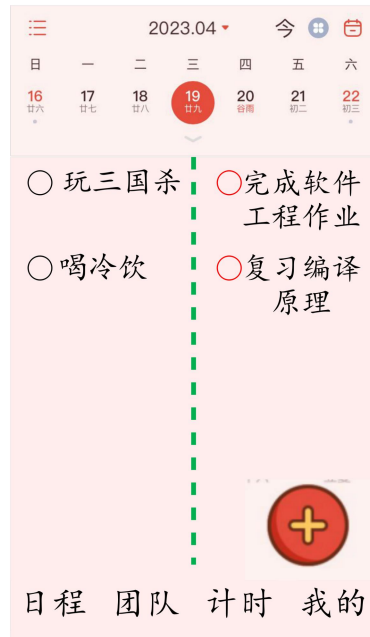


图 4.1 用户界面

点击图中的“+”号，可以将需要记录的事务记录下来。下方的空白处可分为左右两部分，左边列属于禁忌列表的事务，右边列计划要做的事务。

点击底端的“团队”按钮，可以查看团队其它成员的计划。

点击底端的“计时”按钮，可以切换到番茄钟功能界面，具体示例如图 4.2 所示：



图 4.2 番茄钟

该界面中，点击“开始”按钮，即可开始番茄钟计时。

长按一个条形框（如软件工程），可查看在该事项上投入的累计时间。

### 4.1.2 硬件接口

软件平台需要调用的硬件接口如下：

- 1.Wi-Fi 接口：用于连接到无线网络，实现设备之间的网络通信和数据同步。
- 2.扬声器接口：用于播放声音，例如语音提示、音乐播放等。
- 3.时钟接口：可以用于获取设备的系统时间，为用户提供准确的时间提示和管理功能。

### 4.1.3 软件接口

- 1.操作系统：Android 10 及以上、Microsoft Windows 10。
- 2.软件设备：VScode、IntelliJ IDEA、MySQL8.0。

## 4.2 功能需求

### 4.2.1 整体分析

系统的用例图如图 4.3 所示：

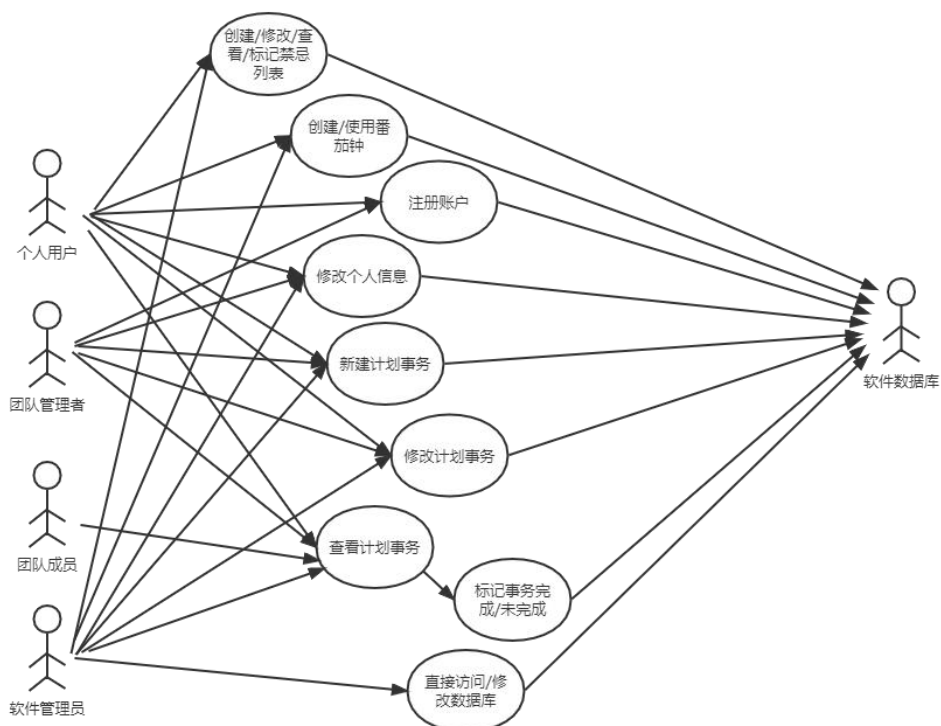


图 4.3 用例图

### 用例图场景描述：

**注册账户：**用户在使用此 APP 之前，需要通过输入账号、密码等信息注册新的账户，之后才能通过登录使用此 APP。

**修改个人信息：**用户在进行登录后，可以修改昵称、年龄、密码等个人信息。

**新建计划事务：**用户可以使用该 APP 通过输入事务的开始时间、结束时间、进度描述、奖惩规则等内容新建一个计划事务。

**修改计划事务：**用户在使用该 APP 时可以随时修改事务的开始时间、结束时间、进度描述、奖惩规则等内容。

**查看计划事务：**用户可以查看目前所有的计划事务情况，例如事务的计划周期数，完成周期数，剩余周期数等。

**标记事务完成/未完成：**当用户在预定时间内完成某个事务时，可以标记此事务已经完成，并按照奖惩规则获得奖励。当用户在预定时间内未完成某个事务时，此事务被标记为未完成事务，用户应按照奖惩规则受到惩罚。

**创建/修改/查看/标记禁忌列表：**禁忌列表的相关操作与计划事务类似，此处不再赘述。

### 系统的状态转移图

考虑完了该系统的用例场景描述，接着，我们可以展示系统的整体状态转移图。系统的整体状态转移图如图 4.4 所示：

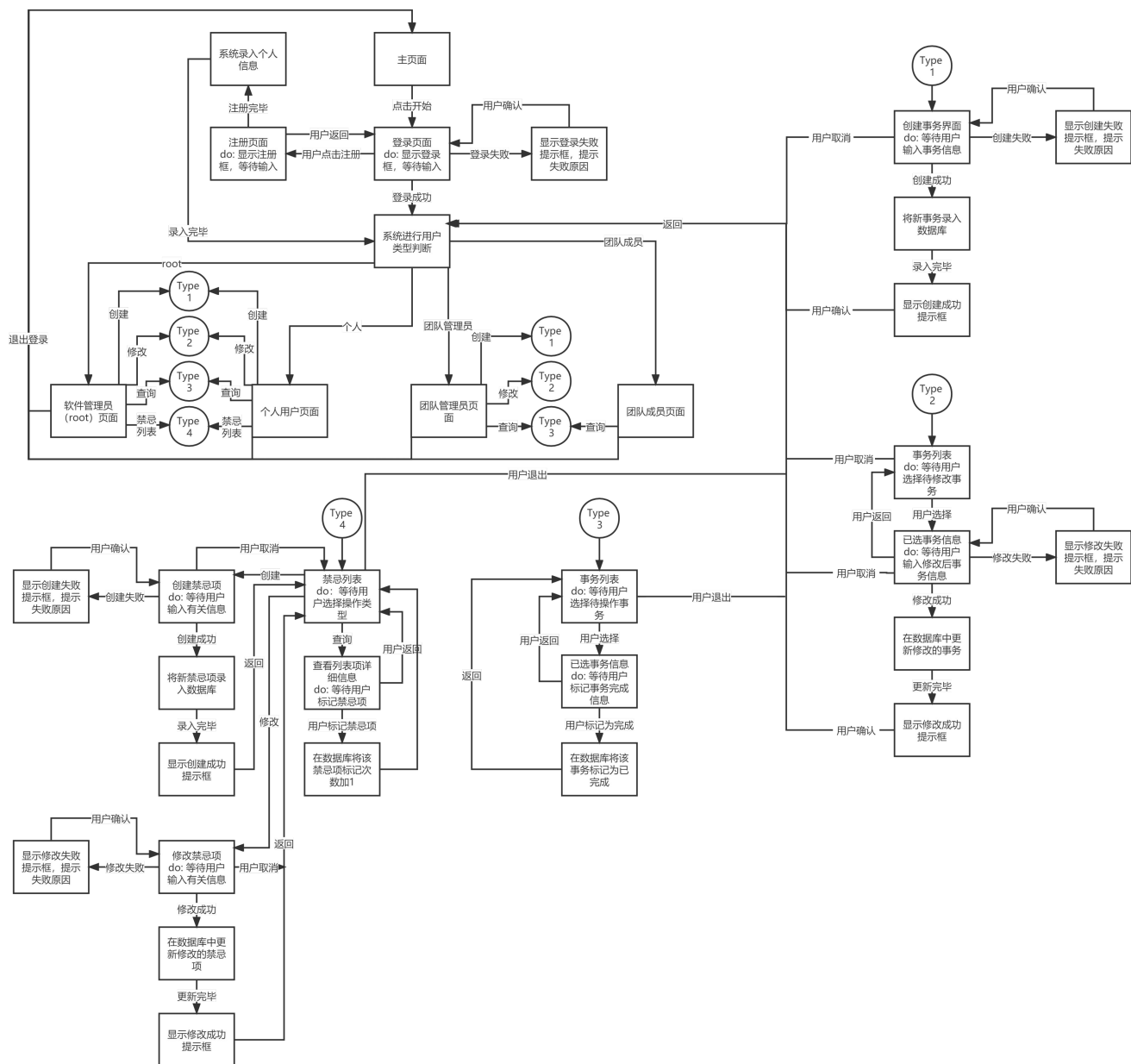


图 4.4 整体状态转移图

各部分状态转移图分别为：

### 1.主界面

主界面部分的状态转移图如图 4.5 所示。用户进入主界面时需先登录，未注册的用户先注册，注册并登录后系统将根据登录账号类型进入不同的用户界面，在各自对应的用户界面中，用户可以使用不同的功能，图中 Type1 表示创建新事务，Type2 表示修改事务，Type3 表示查询/标件事务，Type4 表示禁忌列表。所有功能使用完成后均将返回用户类型对应的主界面。



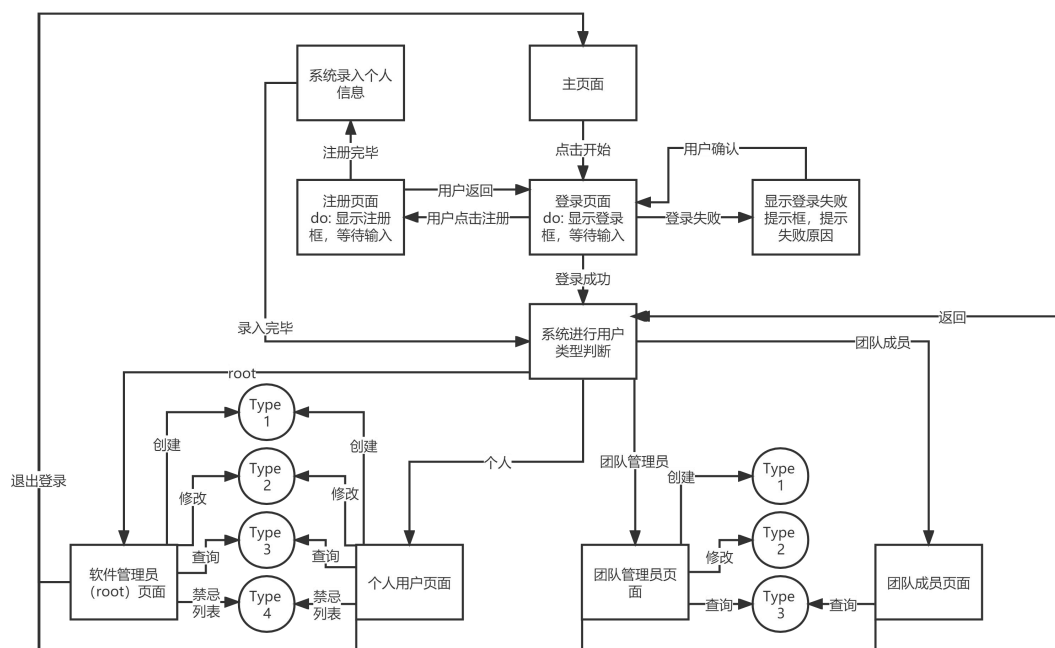


图 4.5 主界面状态转移图

## 2. 创建新事务

创建新事务的状态转移图如图 4.6 所示。创建新事务时，用户需要输入该新事务的信息，若信息无误则创建成功，新事务将录入数据库并提示用户创建成功，若信息有误则弹出创建失败提示框并提示用户重新输入。创建完成后将返回主页面。

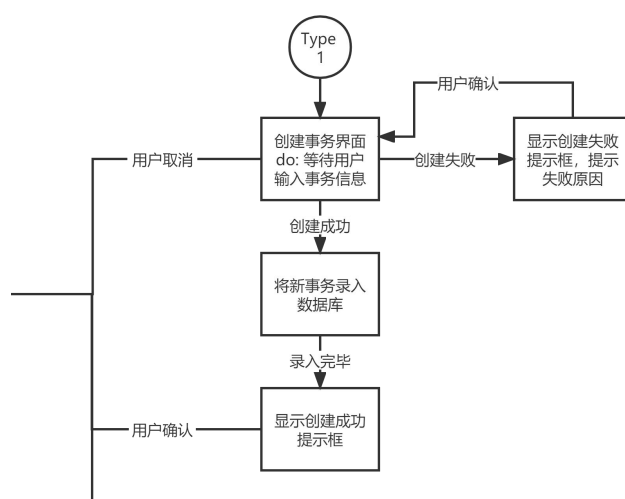


图 4.6 创建新事务状态转移图

### 3.修改事务

修改事务的状态转移图如图 4.7 所示。修改事务时，用户需从事务列表中选出需要修改的事务，用户输入修改的信息，若信息无误则修改成功，修改后的事务将录入数据库并提示用户修改成功，若信息有误则弹出修改失败提示框并提示用户重新输入。修改完成后将返回主页面。

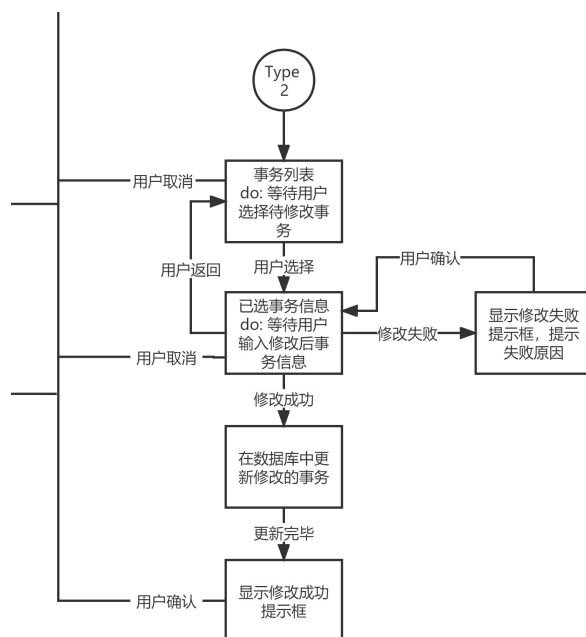


图 4.7 修改事务状态转移图

### 4.查询/标记事务

查询/标记事务的状态转移图如图 4.8 所示。查询/标记事务时，用户需从事务列表中选出需要查询/标记的事务，点击后用户可查看事务，同时若用户对正在查看的事务进行标记，标记后的事务将录入数据库并返回事务列表。操作完成后，用户可退出并返回主页面。

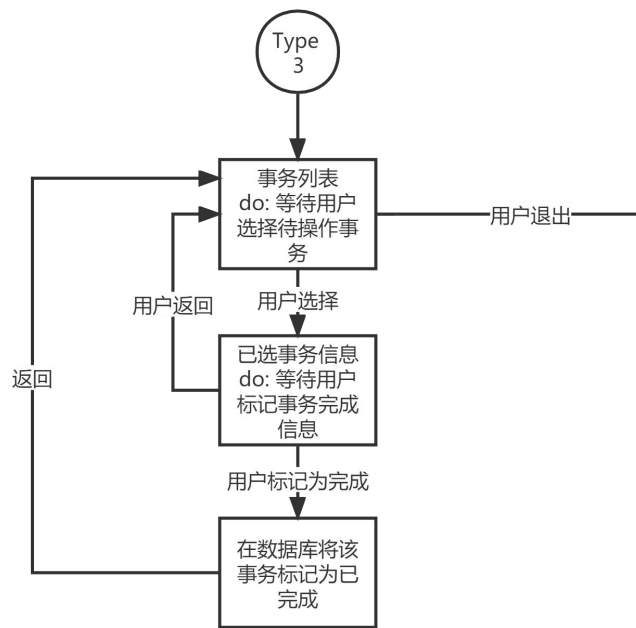


图 4.8 查询/标记事务状态转移图

## 5. 禁忌列表功能

禁忌列表的状态转移图如图 4.9 所示。用户在进入禁忌列表后，可以与普通事务的修改/创建/查询/标记一样对禁忌列表进行操作。用户主动退出后将返回主界面。

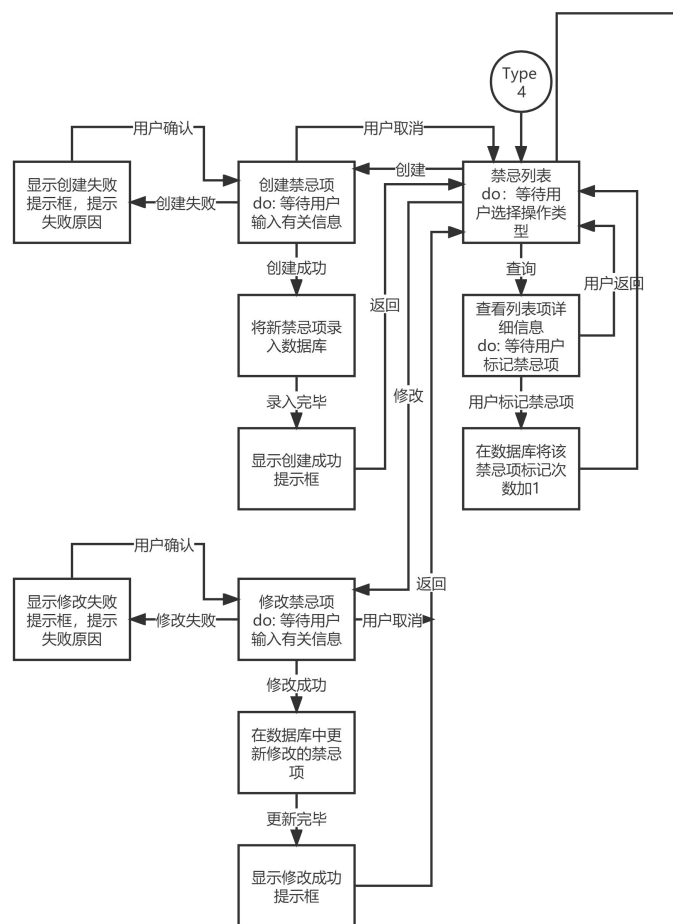


图 4.9 查询/标记事务状态转移图

#### 4.2.2 个人时间管理

##### 数据流图：

事项记录功能仅需要用户与 APP 直接交互即可，即用户将计划信息以及执行情况的数据传给 APP，APP 对这些数据进行变化处理后将奖惩情况的数据返回给用户。其顶层数据流图如图 4.10 所示：

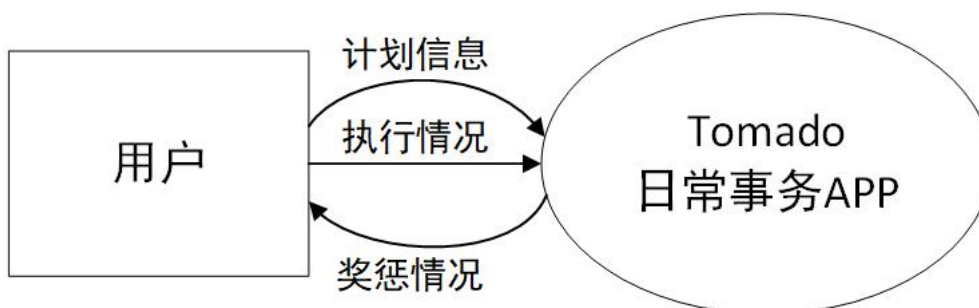


图 4.10 顶层数据流图

进一步地，我们对该对功能级数据流图中描绘的系统主要功能进一步细化。根据第 3 章 3.2 节中关于产品功能的描述，得到如图 4.11 所示的第 1 级数据流图：

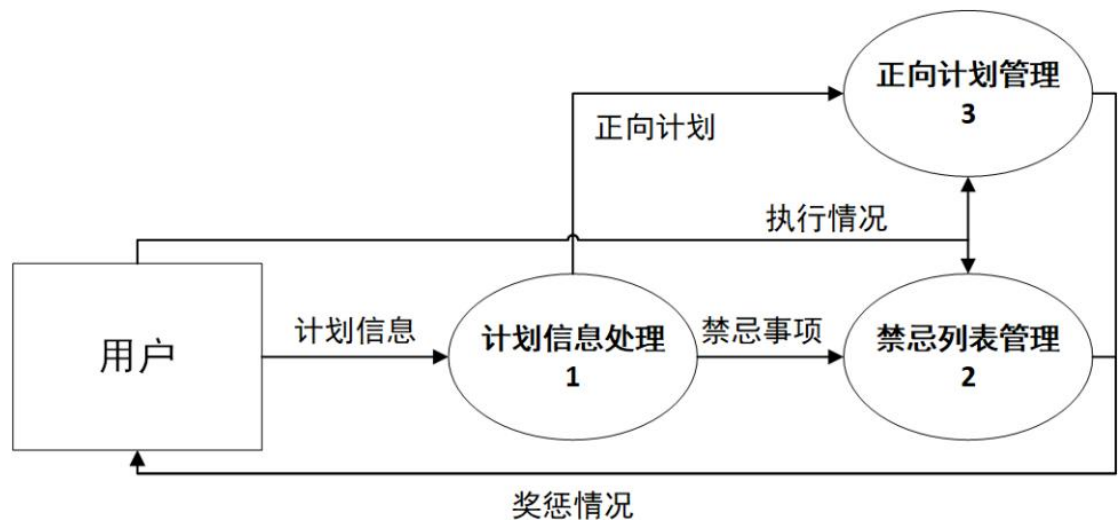


图 4.11 第 1 级数据流图

通过第 1 级数据流图可知，该 APP 首先对用户传来的计划信息数据进行识别，分为正向计划和禁忌列表两类，并将执行情况的数据传给这两个数据处理模块，最后，这两个数据处理模块将奖惩情况的数据传回给用户。

由上述分析可知，我们的计划信息处理模块仅仅用来区分正向计划与禁忌事项，因此没有必要继续展开。进一步地，我们精化“禁忌列表管理”，得到如图 4.12 所示的第 2 级数据流图：

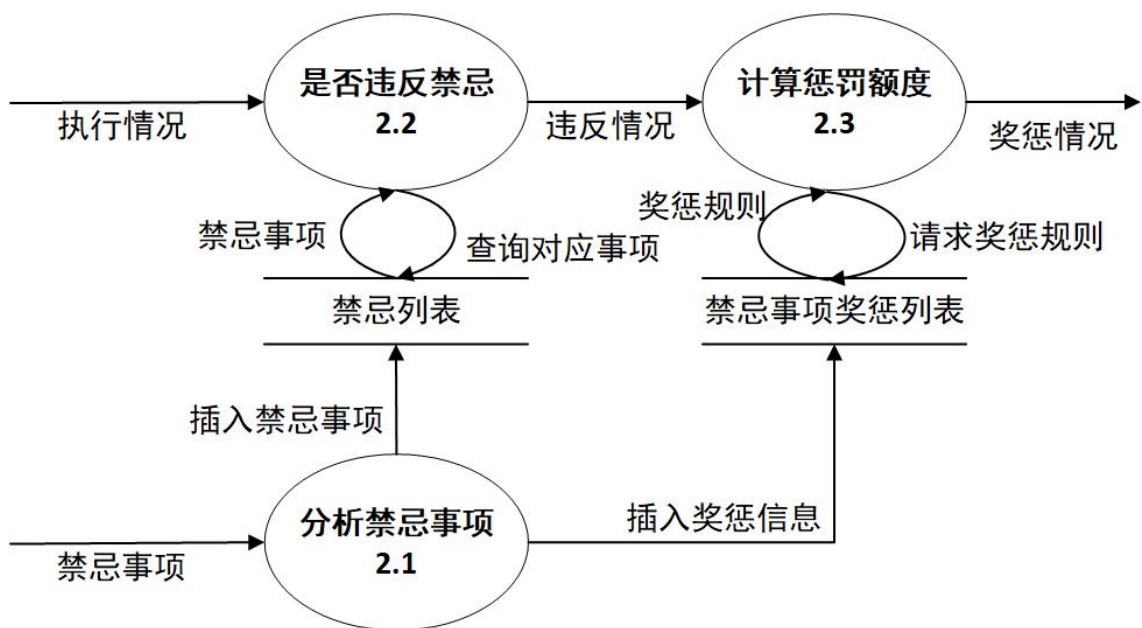


图 4.12 “禁忌列表管理”第 2 级数据流图

在该数据流图中，分析禁忌事项仅仅是将事项从程序数据转换为 SQL 命令修改对应的数据库，无需继续划分。而判断是否违反禁忌与计算惩罚额度同样都可以直接进行，无需细分。用户将执行情况的数据传送给 2.2 模块后，该模块首先从禁忌列表数据库中查询对应事项，然后判断是否违反禁忌，接下来将违反情况传送给 2.3 模块，该模块从禁忌事项奖惩列表数据库中请求奖惩规则信息，然后根据此信息将奖惩情况返回给用户。

接着是“正向计划管理”的数据流图，如图 4.13 所示：

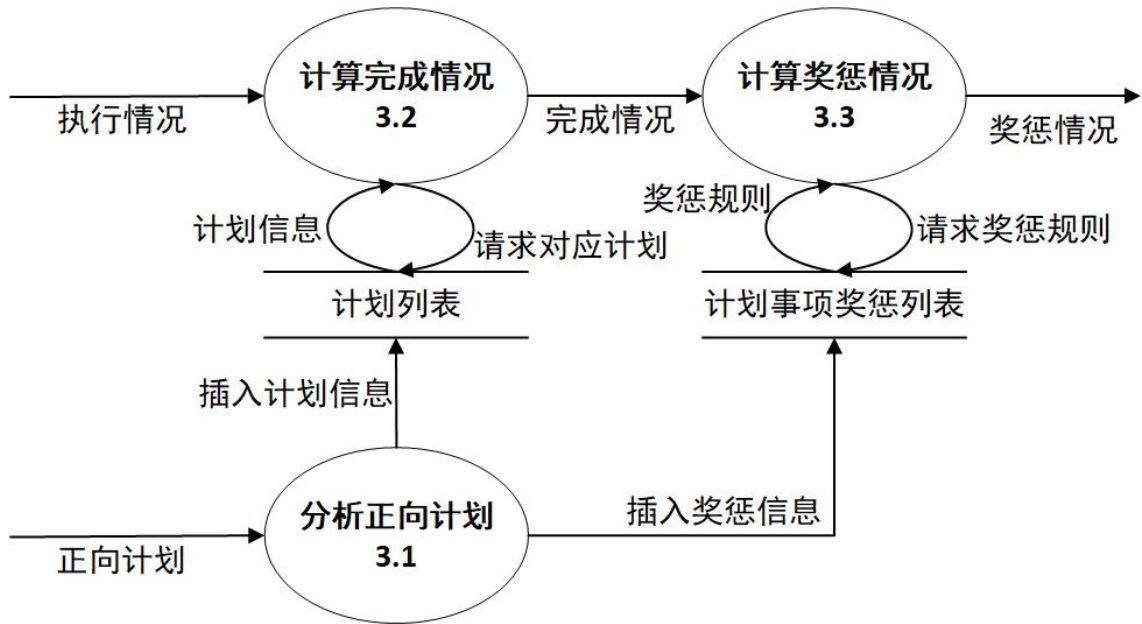


图 4.13 “正向计划管理”第 2 级数据流图

正向计划管理与禁忌列表管理类似，只不过将禁忌事项转变为了计划信息，这里不再进行赘述。

接着，我们描述一下其数据字典：

表 4.1 数据字典——数据流条目 1

名称	计划信息
使用地点和方式	用户将新建的计划信息上传给 APP 软件
内容（组成）描述	计划信息=计划类型+开始时间+目标时间+计划内容+奖惩规定
数据流来源	用户上传
数据流去向	计划信息处理模块（数据加工）

**表 4.2 数据字典——数据流条目 2**

名称	执行情况
使用地点和方式	用户实时上传自己的最新执行情况
内容（组成）描述	执行情况=[事项 id, 计划 id]+当前时间+执行进度
数据流来源	用户上传
数据流去向	用于计算计划完成情况

**表 4.3 数据字典——数据流条目 3**

名称	奖惩情况
使用地点和方式	系统将经过分析的得出的具体奖惩情况反馈给用户
内容（组成）描述	执行情况=当前时间+执行进度
数据流来源	系统经过计算出的奖惩额度
数据流去向	反馈给前端的用户界面，用于提供反馈

**表 4.4 数据字典——数据流条目 4**

名称	正向计划
使用地点和方式	系统处理用户计划信息，区分出正向计划与禁忌事项
内容（组成）描述	正向计划=开始时间+目标时间+计划内容+奖惩规定
数据流来源	系统的计划信息处理模块
数据流去向	发送到正向计划管理模块，并用于更改数据库内容

**表 4.5 数据字典——数据流条目 5**

名称	禁忌事项
使用地点和方式	系统处理用户计划信息，区分出的禁忌事项
内容（组成）描述	禁忌列表=开始时间+目标时间+事项内容+奖惩规定
数据流来源	系统的计划信息处理模块
数据流去向	发送到禁忌列表管理模块，并用于更改数据库内容

**表 4.6 数据字典——数据流条目 6**

名称	插入禁忌事项
使用地点和方式	系统经过对于禁忌事项的分析，区分出事项本身和对应的奖惩信息。
内容（组成）描述	禁忌事项=事项 id+开始时间+目标时间+事项内容

数据流来源	分析禁忌事项模块
数据流去向	用于更改禁忌列表

**表 4.7 数据字典——数据流条目 7**

名称	查询禁忌事项
使用地点和方式	为了计算是否违反禁忌，我们通过执行情况中的事项 id 进行查询，请求得到该事项的详细设置。
内容（组成）描述	查询禁忌事项=事项 id
数据流来源	是否违反禁忌模块
数据流去向	向数据库发送查询请求

**表 4.8 数据字典——数据流条目 8**

名称	禁忌事项
使用地点和方式	数据库返回的禁忌事项
内容（组成）描述	禁忌事项=事项 id+开始时间+目标时间+事项内容
数据流来源	分析禁忌事项模块
数据流去向	用于更改禁忌列表

**表 4.9 数据字典——数据流条目 9**

名称	违反情况
使用地点和方式	用于计算惩罚额度
内容（组成）描述	违反情况=事项 id+[true, false]+当前时间
数据流来源	系统判读是否违反禁忌事项
数据流去向	用于计算对应惩罚

**表 4.10 数据字典——数据流条目 10**

名称	请求奖惩规则
使用地点和方式	用于查询用户设置的奖惩规则
内容（组成）描述	请求奖惩规则=事项 id
数据流来源	计划惩罚情况模块
数据流去向	用于查询禁忌事项奖惩列表



表 4.11 数据字典——数据流条目 11

名称	奖惩规则
使用地点和方式	返回给计算惩罚情况的奖惩规则
内容（组成）描述	奖惩规则=[目标时间+具体规则]
数据流来源	数据库查询结果
数据流去向	用于计算奖惩情况

我们将该部分的类图绘制如图 4.14 所示。

此图中存在的类有：用户、正向计划、禁忌事项、奖励规则、惩罚规则、奖励情况和惩罚情况。类中的属性以及方法在数据字典中已经介绍过，这里不再赘述。

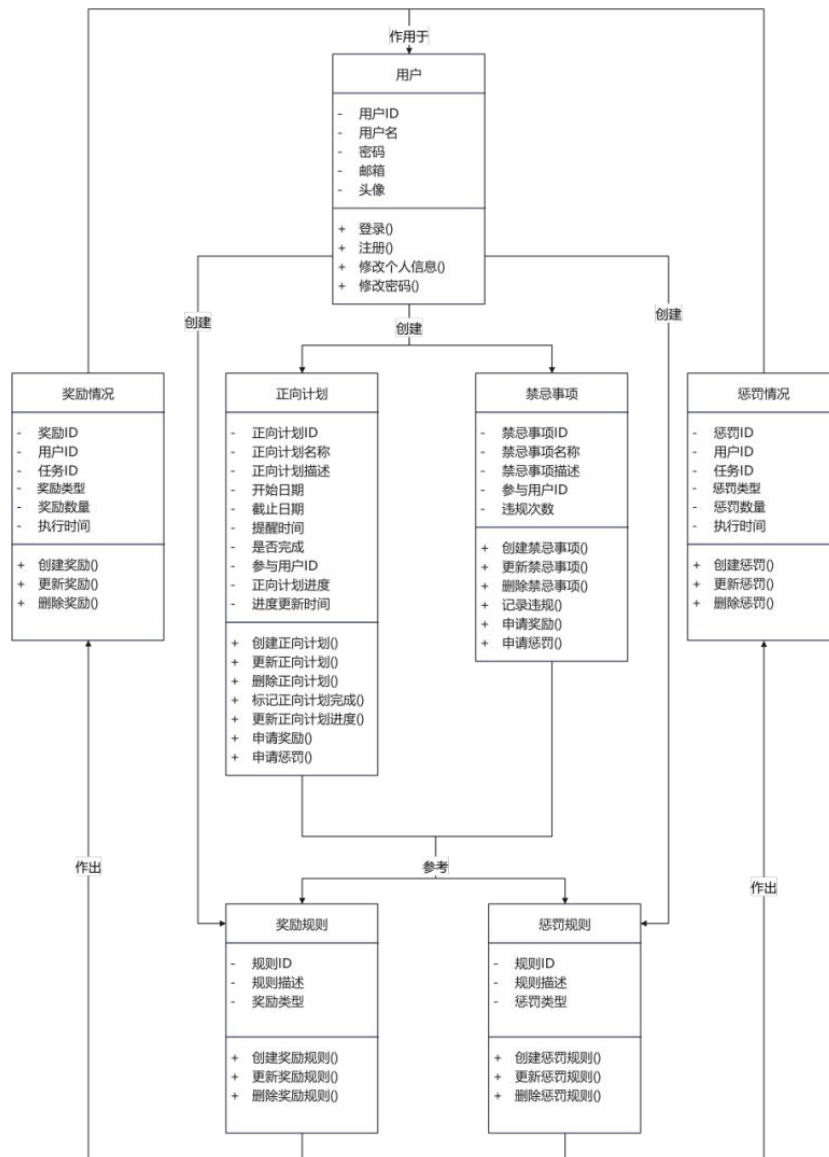


图 4.14 类图

同时，我们画出其对应的 E-R 图，如图 4.15 所示：

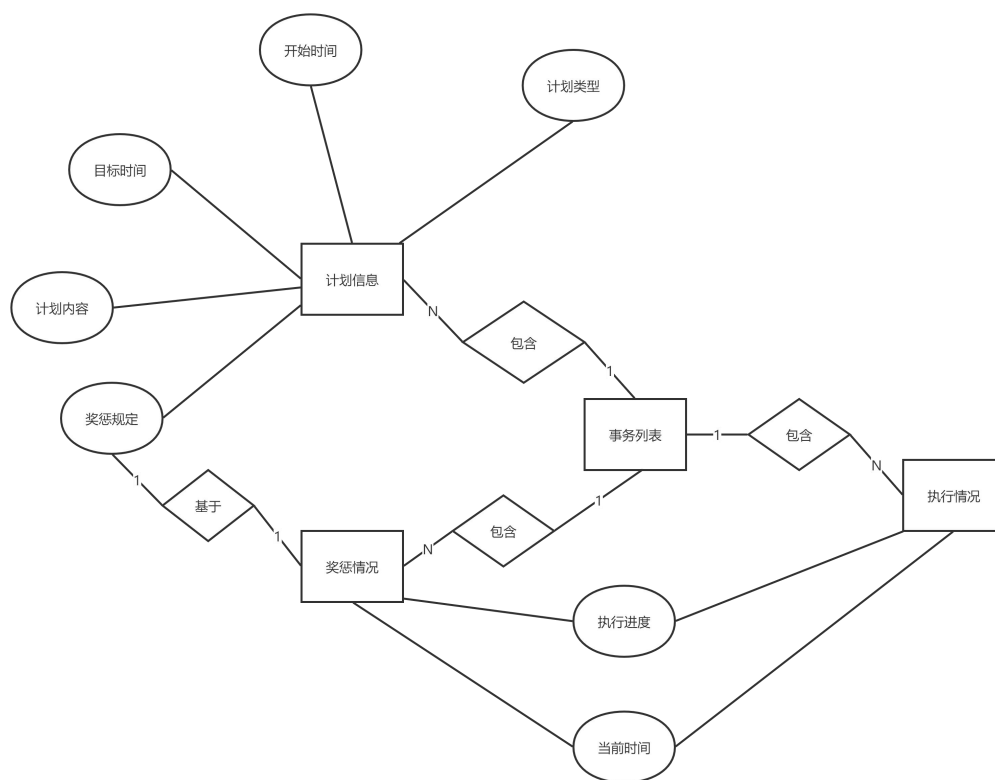
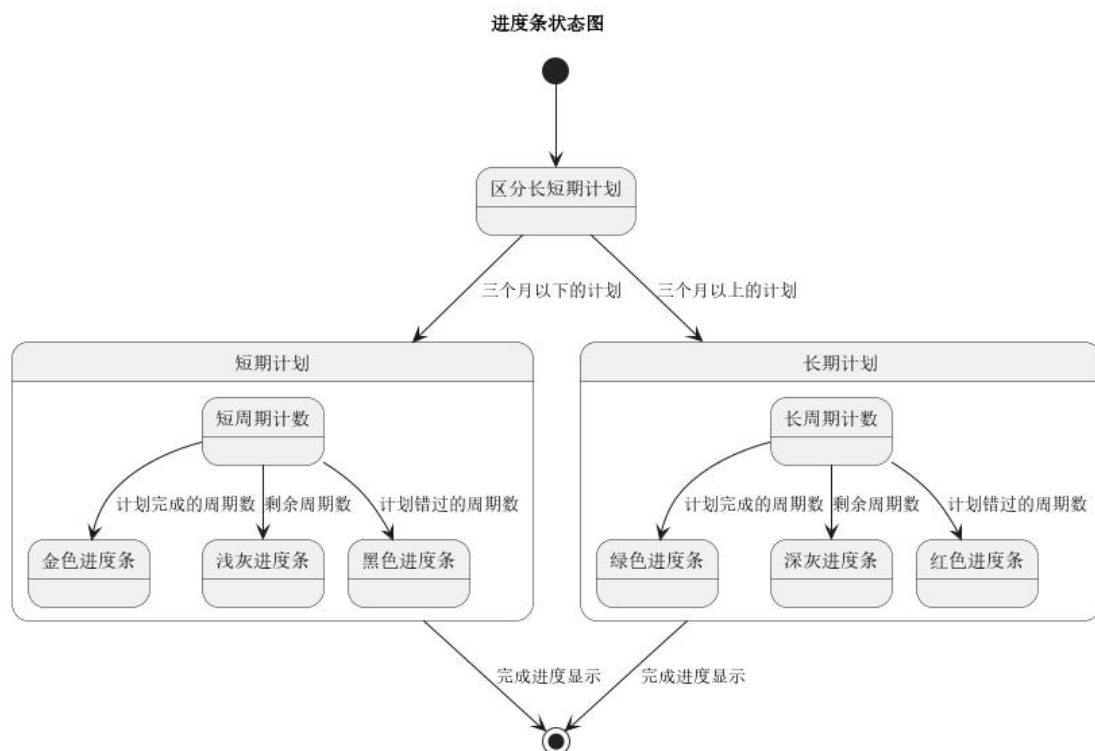


图 4.15 E-R 图

### 4.2.3 计划进度监控

在计划进度控制这一模块中，由于我们具有较多的状态，而没有什么数据的流动。因此，我们使用状态转移图进行建模。状态转移图如图 4.16 所示：

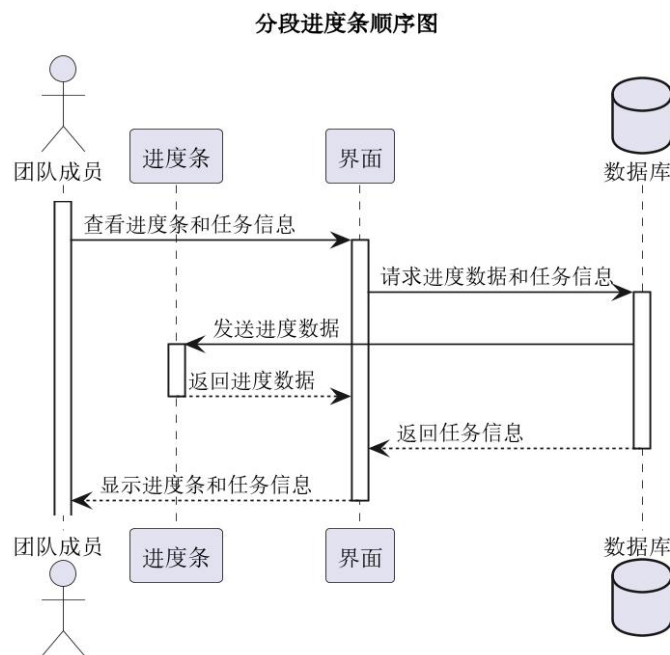


**图 4.16 状态转移图**

通过上面的进度条状态转移图可以知道，我们将计划分为长期计划和短期计划两种类型，当判断为短期计划时，进度条转移到短期计划状态，分别按照金色、浅灰色、黑色显示计划完成的周期数、剩余周期数以及计划错过的周期数；当判断为长期计划时，进度条转移到长期计划状态，分别按照绿色、深灰色、红色显示计划完成的周期数、剩余周期数以及计划错过的周期数。

#### **4.2.4 团队合作协调**

为了实现团队的协作，我们需要我们的 APP 能够实现不同团队成员都能够查看进度条与任务信息，因此，我们绘制出如图 4.17 所示的顺序图：



**图 4.17 顺序图**

通过分析上面的顺序图我们可以知道，在团队合作协调功能中，团队成员可以在界面中发出查看进度条和任务信息请求，APP 收到此请求后调取数据库中的数据然后发送到进度条，进度条将进度数据显示在界面上，同时数据库返回的任务信息也会显示在界面上，之后团队成员可以在界面中查看进度条和任务信息。

## 4.3 性能需求

### 4.3.1 处理能力

该系统的处理能力主要考虑系统能承载的最大并发用户数。按照实际情况的规划，在当前情况下系统至少能承载的最大并发用户数要求达到 500 人，之后会随服务器容量而定。

### 4.3.2 响应时间

为了能够使用户的体验良好，系统应该能够快速响应客户端的请求。由于是联网应用，用户最终得到结果的响应时间除了与系统响应速度有关外，还与网络状况有关，因此对 Web 服务器端需要较高的要求。我们希望即便在高峰工作负载条件下，用户操作后至多 1 秒时间内能够完成响应。

## 4.4 交付与验收要求

交付时间：本软件于 2023 年 6 月 17 日前交付。

性能要求：本软件确保最多 500 个用户并发操作，单个操作的响应时间最多 1s。

质量要求：软件应具有可靠性、可维护性、可扩展性、安全性等。

文件要求：软件交付应包括必要的文档和文件，如用户手册、技术文档、代码文档、测试文档等。

环境要求：软件应能在 PC 端 Windows 10 以及移动端 Android 10 及以上的版本中运行。