

**本科生毕业设计(论文)**

**游戏化个人管理Android App的设计与实现Design and Implementation of Android App for Gamified Personal Management**

|  |  |
| --- | --- |
| 学 院： | 计算机学院 |
| 专 业： | 软件工程 |
| 学生姓名： | 金晨 |
| 学 号： | 1120161943 |
| 指导教师： | 赵丰年 |

2020年 月 日

**原创性声明**

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下独立进行研究所取得的成果。除文中已经注明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

特此申明。

本人签名： 日 期： 年 月 日

**关于使用授权的声明**

本人完全了解北京理工大学有关保管、使用毕业设计（论文）的规定，其中包括：①学校有权保管、并向有关部门送交本毕业设计（论文）的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存本毕业设计（论文）；③学校可允许本毕业设计（论文）被查阅或借阅；④学校可以学术交流为目的,复制赠送和交换本毕业设计（论文）；⑤学校可以公布本毕业设计（论文）的全部或部分内容。

本人签名： 日 期： 年 月 日

指导老师签名： 日 期： 年 月 日

**游戏化个人管理Android App的设计与实现**

摘　要

游戏曾经是一个与效率完全相悖的名词，但随着互联网技术的发展，游戏化的概念逐渐普及并应用在方方面面。所谓游戏化，就是将游戏的各类设计元素应用于非游戏情境中，让用户能在其中完成特定的任务，体会到游戏的乐趣。而个人管理作为一个提升自我的有效途径，其枯燥与所需的自制力又往往让人难以坚持。将个人管理游戏化，是促使用户自主接受个人管理，提高效率的良策。

本文对于游戏化的主要元素与核心机制，及个人管理的主要内容进行了分析，对已有的个人管理app，如forest、pocketlist等进行研究，总结出游戏化与个人管理的设计模式。并基于此，在满足备忘、计划、每日待办等个人管理内容的前提下，加入积分、社交、勋章等游戏化元素，给予用户及时的反馈，设计了一个小型的个人管理app。本系统使用java、nodejs、mysql进行开发，并对用户的各项操作进行了测试。

**关键词：游戏化；个人管理；安卓；java**

**Design and Implementation of Android App for Gamified Personal Management**

Abstract

Games used to be a term completely contrary to efficiency, but with the development of Internet technology, the concept of gamification is gradually popularized and applied in all aspects. Gamification means applying various design elements of the game to non-game situations, so that users can complete specific tasks and experience the fun of the game. As an effective way to improve, personal management is often boring and requires self-control. Gamification of personal management is a good strategy to encourage users to independently accept personal management and improve efficiency.

This article analyzes the main elements and core mechanisms of gamification, as well as the main content of personal management. It studies existing personal management apps, such as forest and pocketlist, and summarizes the design patterns of gamification and personal management. Based on this, on the premise of satisfying personal management contents such as memos, plans, daily to-dos, etc., a small personal management app is designed by adding gamification elements such as points, social networking, medals, etc., to give users timely feedback. This system uses java, nodejs and mysql for development, and tests the user's various operations.

**Key Words: gamification；personal management；android；java**

目　录

[摘　要 I](#_Toc41877461)

[Abstract II](#_Toc41877462)

[第1章 引言 1](#_Toc41877463)

[1.1 概要叙述 1](#_Toc41877464)

[1.2 背景介绍 1](#_Toc41877465)

[1.3 研究目的 1](#_Toc41877466)

[1.3 论文结构 1](#_Toc41877467)

[第2章 内容 2](#_Toc41877468)

[第3章 个人管理app的需求分析 3](#_Toc41877469)

[3.1 用户模块需求分析 3](#_Toc41877470)

[3.2 好友模块 3](#_Toc41877471)

[3.3 备忘录模块 4](#_Toc41877472)

[3.4每日待办模块 4](#_Toc41877473)

[3.5 计划模块 5](#_Toc41877474)

[3.6 时间历程模块 6](#_Toc41877475)

[3.7 成就模块 7](#_Toc41877476)

[第4章 个人管理app的系统设计 8](#_Toc41877477)

[4.1 概要设计 8](#_Toc41877478)

[4.2 功能模块划分 8](#_Toc41877479)

[4.3 系统模块的详细设计 9](#_Toc41877480)

[4.3.1 用户模块详细设计 9](#_Toc41877481)

[4.3.2好友模块详细设计 9](#_Toc41877482)

[4.3.3 备忘录模块 10](#_Toc41877483)

[4.4.4 每日待办模块 10](#_Toc41877484)

[4.4.5 计划模块 11](#_Toc41877485)

[4.4.6 成就模块 12](#_Toc41877486)

[4.4.7 时间历程模块 12](#_Toc41877487)

[4.4 数据库设计 13](#_Toc41877488)

[4.4.1 数据库总体设计 13](#_Toc41877489)

[4.4.2 user表设计 16](#_Toc41877490)

[4.4.3 note表设计 17](#_Toc41877491)

[4.4.4 todo表设计 18](#_Toc41877492)

[4.4.5 plan表设计 18](#_Toc41877493)

[4.4.6 chart表设计 18](#_Toc41877494)

[4.4.7 log表设计 19](#_Toc41877495)

[4.5 接口设计 20](#_Toc41877496)

[4.5.1 user 20](#_Toc41877497)

[4.5.2 friend 20](#_Toc41877498)

[4.5.6 note 20](#_Toc41877499)

[4.5.7 todo 20](#_Toc41877500)

[4.5.5 plan 20](#_Toc41877501)

[4.5.6 chart 20](#_Toc41877502)

[4.5.7 achieve 20](#_Toc41877503)

[4.5.8 log 20](#_Toc41877504)

[第5章 个人管理app的系统实现 21](#_Toc41877505)

[5.1 开发环境 21](#_Toc41877506)

[5.2用户模块（程序流图、页面截图） 21](#_Toc41877507)

[5.3好友模块 21](#_Toc41877508)

[5.4 备忘录模块 21](#_Toc41877509)

[5.5 每日待办模块 21](#_Toc41877510)

[5.6 计划模块 21](#_Toc41877511)

[5.7 成就模块 21](#_Toc41877512)

[5.8 时间历程模块 21](#_Toc41877513)

[第6章 个人管理app的测试 22](#_Toc41877514)

[6.1 测试环境 22](#_Toc41877515)

[6.2 测试方法？过程？内容？结果 22](#_Toc41877516)

[结　论 23](#_Toc41877517)

[参考文献 24](#_Toc41877518)

[附　录 25](#_Toc41877519)

[致　谢 26](#_Toc41877520)

# 第1章 引言

1.1 概要叙述

游戏化，即在非游戏情境下使用特定的游戏元素，从而激励用户获得特定价值。游戏化的特色在于将无趣的事趣味化，在各类严肃的课堂和教育中都有应用。而个人管理通常用于记录备忘、列举任务、进行计划等，其一丝不苟循规蹈矩往往让人难以坚持。因此在普遍的需求下，个人管理的游戏化应运而生。

游戏化的核心机制包括积分、社交、排名、勋章等，在个人管理上都有体现。如饱受好评的应用软件forest，将专注的时长量化为金币与树。当完成预定任务时，预设的树也会长成，但若途中使用手机白名单外的app，树便会枯萎，需要金币方可清除。同时树木的数量与枯树的数量都会作为排名依据，在好友排行榜中展示，以激励用户坚持专注。同时，达到一定目标还会获得相应勋章。无论是何种方式，在个人管理中加入游戏化元素，都让不可见的专注获得的益处量化，为用户提供及时的反馈以获取满足感，从而达到激励用户自我管理的目的。

1.2 背景介绍

游戏化的概念最早出现于1980年，Richard Bartle将其理解为“把不是游戏的东西（或工作）变成游戏”。

1.3 研究目的

1.3 论文结构

# 第2章 内容

# 第3章 个人管理app的需求分析

3.1 用户模块需求分析

用户模块的用例图如图3-1所示，包括登录、注册功能。

其中登录功能由系统自行识别该设备是否已注册，若已注册则从设备文件获取信息直接登录，否则需用户手动输入用户名，同时系统自动生成唯一id进行注册，注册成功后系统直接登录。换而言之，用户只需在首次使用时注册，此后自动登录。登录成功后用户可查看个人信息，复制由系统生成的随机id。



图3-1 用户模块用例图

3.2 好友模块

好友模块的用例图如图3-2所示，包括展示排行榜、添加好友、删除好友功能。

该模块自动加载好友列表，依据进行计划时长进行排序并标识，以排行榜的形式展示好友排名、昵称及进行时长。用户可以通过输入系统自动生成的用户id，将该用户直接添加好友，也可删除好友。



图3-2 好友模块用例图

3.3 备忘录模块

备忘录模块的用例图如图3-3所示，包括展示给定日期范围内备忘录、添加备忘录、修改备忘录、删除备忘录功能。

系统自动根据用户信息获取备忘录信息，修改日期可查看不同时段的备忘录。同时，可在当前展示备忘录列表中对已有备忘录进行修改、删除操作，也可选择日期添加新的备忘录。

3.4每日待办模块

待办模块的用例图如图3-4所示，包括展示当日待办、添加待办、修改待办、删除待办、完成待办、重复当日待办功能。

系统自动根据用户信息获取当日待办信息，修改日期可查看历史待办，不可查询未来待办。对于当日待办的每一项任务，可进行添加、删除、修改、完成或取消完成操作，并可重复当日待办，即将当日所有任务清单复制至第二日。



图3-3 备忘录模块用例图



图3-4 每日待办模块用例图

3.5 计划模块

计划模块的用例图如图3-5所示，包括展示尚未完成计划、添加计划、修改计划、删除计划、进行计划、完成计划、统计进行计划时长并展示功能。

系统自动根据用户信息获取未完成计划信息，并对其进行添加、修改、删除、完成操作。对于未完成的计划，用户可选择一项计划进行，系统会进行计时并存储，统计的进行时长可依日期范围查看。



图3-5 计划模块用例图

3.6 时间历程模块

时间历程模块的用例图如图3-6所示，包括展示时间历程、记录历程功能。

用户完成当日待办、获得成就勋章或完成计划时，系统会自动创建时间历程。时间历程按创建时间倒序展示。



图3-6 时间历程模块用例图

3.7 成就模块

成就模块的用例图如图3-7所示，包括展示成就、获取成就功能。

用户完成当日所有待办项达到一定天数，或进行计划累计一定时长时会获得成就及勋章，用户可查看所获勋章。若完成当日待办，后又取消，仍计作当日待办已完成，成就仍然保留。



图3-7 成就模块用例图

# 第4章 个人管理app的系统设计

4.1 概要设计

个人管理app的系统架构图如图4-1所示。本系统采用MVC模式，将视图页面、数据库与控制器完全分离进行开发，以降低耦合度，增强系统的可扩展性、可维护性，便于测试。



图4-1 系统架构图

4.2 功能模块划分

系统功能模块的划分如图4-2所示，包括用户信息、好友列表、备忘录、每日待办、计划系统、时间历程、成就勋章七个模块，其中时间历程与成就勋章由系统直接生成，用户仅可查看，其余模块均有相应交互功能。

-

**图4-2 系统功能模块**

4.3 系统模块的详细设计

4.3.1 用户模块详细设计

用户模块业务流程图如图4-3所示。用户模块主要包括用户的登录注册与用户信息的展示。

进入登录页后，系统会自动读取本地文件以判定该设备是否是首次登录。若本地文件读取失败则说明未注册，系统跳转为注册，用户输入昵称，昵称不可与数据库已有重复。完成注册后系统自动生成唯一id，并跳转登录。若本地文件读取成功，系统直接从文件中读取用户id与昵称，并自动登录。登录成功后，用户可以查看个人信息，包括id与昵称。

4.3.2好友模块详细设计

好友模块主要包括用户好友信息的获取与排名、并可进行添加好友、删除好友操作。

进入页面后，系统获取好友列表，用户及其好友共同依据进行计划时长排序以排行榜形式展示，排行榜包括排名、用户头像（系统自动生成）、用户名与计划进行时长。信息展示依据排名和时长有细微区别。对于排行榜上除用户外已添加好友可选取删除，也可输入不在榜上的用户id添加好友。



**图4-3 用户模块流程图**

4.3.3 备忘录模块

备忘录模块包括展示备忘录、对备忘录进行添加、删除、修改的功能。

初始日期为当月内，系统获取展示当月所有备忘，以列表形式展示，同时在日历对应日期进行标识。展示内容包括用户自行设定的主题、内容、地点与截止日期时间，以及系统自动生成的图像。列表依据今日未办、当月未办、历史的顺序展示，同时对这三类的展示细节存在差异。用户可通过修改月份查询备忘录，选取日期后可添加备忘录。同时对于已有备忘录，用户可进行修改、删除操作。

4.4.4 每日待办模块

每日待办模块业务流程图如图4-4所示。该模块主要包括展示待办任务，查看待办，和在当日添加待办项，对已有待办项进行删除、修改、完成、取消完成操作。对于当日待办，可直接重复至第二日。

初始日期为当日，系统展示当日待办项。对于当日待办，用户可添加、修改、删除、完成或取消完成单个任务项。对于当日所有任务，用户可选择重复当日，所有待办项添加至明日，重复操作会覆盖之前的重复。用户可修改日期查看历史待办，历史待办仅可查看和进行重复。未来待办不可查看。每完成当日所有待办，用户的对应信息会自动更新。



**图4-4 待办模块流程图**

4.4.5 计划模块

计划模块业务流程图如图4-5所示。该模块主要包括展示计划，进行或暂停计划，查看未完成计划，及对其进行修改、删除，此外还可添加计划。

进入页面时仅展示未完成计划，展示顺序依创建时间而定，展示内容包括主题、备注和以颜色表示的紧急度。对于展示的计划，用户可对其进行修改、删除和完成操作，删除或完成操作后该计划都会从列表移除。用户可在展示的计划列表中选取一项进行或暂停，进行中的计划会自动计时，用户进行删除、完成、修改操作时会先自动暂停进行。暂停时会保存已进行时长，用户的对应信息会自动更新。



**图4-5 计划模块流程图**

4.4.6 成就模块

成就模块业务流程图如图4-6所示。该模块主要包括展示成就和获取成就。

用户获得的成就会按照成就的含金量由高到低依序排列，仅可查看。成就包括两种：计划进行时长和待办累积完成天数。待办完成指完成当日所有代办，多次完成当日所有代办仅记录一次，若完成一次后进行删除、添加、取消完成等操作导致最后未完成不做记录。日期用户的两项信息更新时系统会进行判定，若达到预定标准则会更新成就信息，即获取成就。

4.4.7 时间历程模块

时间历程模块业务流程图如图4-7所示。该模块主要包括展示历程和记录历程。

用户经历的历程会按照记录时间由近及远依序排列，展示内容包括历程内容和创建该历程的时间，仅可查看。历程包括三种：完成当日所有待办、完成计划、获得成就。完成待办时系统会根据已有内容添加或更新日程，计划和成就日程在完成时会直接记录。



**图4-6 成就模块流程图**



**图4-7 时间历程模块流程图**

4.4 数据库设计

4.4.1 数据库总体设计

系统使用MySQL数据库，采用utf-8编码，包含user、note、todo、plan、chart、log共6个数据表。

系统的总体E-R图如图4-8所示。各实体属性如图4-9至4-14所示。



**图4-8 系统E-R图**



**图4-9 用户实体的属性**



**图4-10 备忘录实体的属性**



**图4-11 待办实体的属性**



**图4-12 计划实体的属性**



**图4-13 图表实体的属性**



**图4-14 历程实体的属性**

4.4.2 user表设计

用户注册时，系统会自动向表中插入数据。用户进行计划、完成待办、获得计划或待办成就、添加或删除好友时，表内数据会自动更新。

user表的详细设计如表4-1所示。其中id为自增主键；user\_id为系统自动生成的唯一id；name为用户昵称，且昵称之间不重复；todo\_day为用户累积完成全部待办天数，同一天内重复完成全部待办只计作1次；plan\_time为用户累积进行计划时长;friend为好友user\_id的合集，以“,”分隔，添加数据时列表内默认添加用户id；prize\_plan为获得计划成就所至标识，均为偶数；prize\_todo为获得待办成就所至标识，均为奇数。

表4-1 user表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| id | int | 否 | 无 | 主键 |
| user\_id | varchar(20) | 否 | 无 |  |
| name | varchar(50) | 否 | 无 |  |
| todo\_day | int | 否 | 0 |  |
| plan\_time | varchar(200) | 否 | 0m |  |
| friend | varchar(2000) | 否 | 无 |  |
| prize\_plan | int | 否 | 0 |  |
| prize\_todo | int | 否 | -1 |  |

4.4.3 note表设计

用户创建备忘录时，系统会自动向表中插入数据。用户修改、删除备忘录时，表内数据会自动更新。

note表的详细设计如表4-2所示。其中note\_id为自增主键；user\_id表示该备忘录所属，该字段与user中的user\_id字段完全一致；title为任意备忘录标题；content为任意备忘录内容；place为备忘录地点；date为备忘录截止日期；time为备忘录截止时间。

表4-2 note表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| note\_id | int | 否 | 无 | 主键 |
| user\_id | varchar(20) | 否 | 无 |  |
| title | varchar(100) | 否 | 无 |  |
| content | varchar(400) | 否 | 无 |  |
| place | varchar(100) | 否 | 无 |  |
| date | date | 否 | 无 |  |
| time | time | 否 | 无 |  |

4.4.4 todo表设计

用户添加待办项、重复某日待办内容时，系统会自动向表中插入数据。用户修改、删除、完成或取消完成待办任务时，表内数据会自动更新。

todo表的详细设计如表4-3所示。其中todo\_id为自增主键；user\_id表示该待办项所属，该字段与user中的user\_id字段完全一致；title为任意主题；date为备忘录创建日期；isChecked标示该项任务是否完成，1代表已完成，0代表未完成。

表4-3 todo表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| todo\_id | int | 否 | 无 | 主键 |
| user\_id | varchar(20) | 否 | 无 |  |
| title | varchar(100) | 否 | 无 |  |
| date | date | 否 | 无 |  |
| isChecked | int | 否 | 无 |  |

4.4.5 plan表设计

用户新建计划时，系统会自动向表中插入数据。用户修改、删除、暂停进行中计划、选取完成时，表内数据会自动更新。

plan表的详细设计如表4-4所示。其中plan\_id为自增主键；user\_id表示该计划所属，该字段与user中的user\_id字段完全一致；title为任意标题；content为任意内容；urgency为该项计划紧急程度，分为0、1、2、3四个等级；pass代表该计划已进行时间，与user表中plan\_time同步更新；finish代表该计划是否完成，1为已完成，0为未完成。

4.4.6 chart表设计

用户记录每日首次进行计划时，系统会自动向表中插入数据。该日继续进行计划，表内数据会自动更新。

chart表的详细设计如表4-5所示。其中chart\_id为自增主键；user\_id表示该待办项所属，该字段与user中的user\_id字段完全一致；date为当日日期；pass\_red、pass\_yellow、pass\_green、pass\_blue分别为该日进行紧急度为0、1、2、3的计划时长，以分钟为单位。

表4-4 plan表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| plan\_id | int | 否 | 无 | 主键 |
| user\_id | varchar(20) | 否 | 无 |  |
| title | varchar(100) | 否 | 无 |  |
| content | varchar(400) | 否 | 无 |  |
| urgency | int | 否 | 无 |  |
| pass | varchar(50) | 否 | 无 |  |
| finish | int | 否 | 无 |  |

表4-5 chart表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| chart\_id | int | 否 | 无 | 主键 |
| user\_id | varchar(20) | 否 | 无 |  |
| date | date | 否 | 无 |  |
| pass\_red | int | 否 | 0 |  |
| pass\_yellow | int | 否 | 0 |  |
| pass\_green | int | 否 | 0 |  |
| pass\_blue | int | 否 | 0 |  |

4.4.7 log表设计

用户完成计划、获得成就或在当日首次完成待办项时，系统会自动向表中插入数据。在当日重复完成待办项时，表内数据自动更新。

log表的详细设计如表4-2所示。其中log\_id为主键；user\_id表示该待办项所属，该字段与user中的user\_id字段完全一致;type表示该历程对应类型，0为待办类，1位计划类，2位成就类；name为该历程内容，包括完成当日一定数量待办、获得某成就、经历一定时间完成某计划三种可能，与type对应。

表4-6 log表

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 数据类型 | 空 | 默认 | 主外键 |
| log\_id | int | 否 | 无 | 主键 |
| user\_id | varchar(20) | 否 | 无 |  |
| type | int | 否 | 无 |  |
| name | varchar(100) | 否 | 无 |  |
| date | date | 否 | 无 |  |

4.5 接口设计

接口作为Controller，向视图页提供修改数据库内容的通道，也是数据传输最为核心的部分。在此依据视图页各功能不同，将接口分为user、friend、note、todo、plan、chart、achieve、log八个部分。user提供用户的注册及各项字段的获取与查询方法；friend可获取好友信息、添加和删除好友；note、todo、plan提供添加、删除及各个字段的修改方法；chart、log提供添加、更新接口； achieve提供新建与获取接口。

以下所有接口调用出错时，返回值均为{"code": 300, "msg": "获取失败","err": "ER\_BAD\_FIELD\_ERROR"}。其中err的值为该错误具体内容。

4.5.1 user

user部分接口详细设计如表4-8所示。以下请求路径均已省略http://ip:8080/user。其中参数与user表字段一一对应。

4.5.2 friend

friend部分接口详细设计如表4-8所示。以下请求路径均已省略http://ip: 8080 /friend。其中参数ids为传入id合集，以“,”分隔。ids与返回值中的friend意义相同，其余参数与user表字段一一对应。

表4-7 user接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 注册 | post | /register | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "注册成功"} |
| 判断用户名是否存在 | post | /isNameExist | {name:'jc'} | {"code": 201,  "msg": "存在"}  或{"code": 200,  "msg": "不存在"} |
| 判断id是否存在 | post | /isIdExist | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "存在"}  或{"code": 201,  "msg": "不存在"} |
| 获取用户完成待办天数 | post | /getDay | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功",  "data":[{"todo\_day": 1}]} |
| 设置用户完成待办天数 | post | /setDay | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  todo\_day: 1} | {"code": 200,  "msg": "更新成功"} |
| 获取用户进行计划时长 | post | /getTime | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功",  "data":[{"plan\_time":"1m"}]} |
| 设置用户进行计划时长 | post | /setTime | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  todo\_day: '1h1m'} | {"code": 200,  "msg": "更新成功"} |

表4-8 friend接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取好友列表 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功",  "data":[{"friend":"'413192b4ed8744c380fdc68c5c6804c4','1d6c2fd7c1cb40b09999a9b4de75f523',"}] } |
| 删除或添加好友 | post | /update | {user\_id:"'413192b4ed8744c380fdc68c5c6804c4',  ids:"'413192b4ed8744c380fdc68c5c6804c4'"} | {"code": 200,  "msg": "查询成功",  "data":[{"user\_id":"413192b4ed8744c380fdc68c5c6804c4",  "name": "jc",  "plan\_time":"1h1m"}]} |
| 获取用户信息 | post | /getInfo | {ids:"'413192b4ed8744c380fdc68c5c6804c4',"} | {"code": 200,  "msg": "存在"}  或{"code": 201,  "msg": "不存在"} |

4.5.6 note

note部分接口详细设计如表4-9所示。以下请求路径均已省略http://ip: 8080 /note。其中参数与note表字段一一对应。

4.5.7 todo

todo部分接口详细设计如表4-10所示。以下请求路径均已省略http://ip: 8080 /todo。其中参数与todo表字段一一对应。

表4-9 note接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取当月备忘录 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date:'2020-05-04’} | {"code": 200,  "msg": "查询成功",  "data":[{"note\_id": 25,  "user\_id":"413192b4ed8744c380fdc68c5c6804c4",  "title": "毕设",  "content": "会议",  "place": "1-101",  "date": "2020-05-01",  "time": "12:13:00"}]} |
| 删除备忘 | post | /delete | {note\_id:25} | {"code": 200,  "msg": "删除成功"} |
| 添加备忘 | post | /add | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  title: '考试',  content: '',  place: '',  date: '2020-05-30',  time: '11:00:00'} | {"code": 200,  "msg": "添加成功"  "data":[{"note\_id":28}]} |
| 编辑备忘 | post | /update | {note\_id:28,  title: '考试',  content: '',  place: '',  date: '2020-05-30',  time: '13:00:00'} | {"code": 200,  "msg": "修改成功"} |

表4-10 todo接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取该日待办项 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date:'2020-05-30’} | {"code": 200,  "msg": "查询成功",  "data":[{"todo\_id":38,  "user\_id":"413192b4ed8744c380fdc68c5c6804c4",  "title": "清单1",  "date": "2020-05-30",  "isChecked": 1}]} |
| 删除备忘 | post | /delete | {todo\_id:38} | {"code": 200,  "msg": "删除成功"} |
| 删除当日全部备忘 | post | /deleteAll | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date:'2020-05-30’} | {"code": 200,  "msg": "删除成功"} |
| 添加备忘 | post | /add | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  title: '清单2',  date: '2020-05-30'} | {"code": 200,  "msg": "添加成功"  "data":[{"todo\_id":39}]} |
| 编辑备忘 | post | /edit | {todo\_id\_id:39,  title: '学习'} | {"code": 200,  "msg": "修改成功"} |
| 选取是否完成 | post | /check | {todo\_id\_id:39,  isChecked: 1} | {"code": 200,  "msg": "修改成功"} |
| 获取该日总事项数 | post | /number | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date:'2020-05-30’} | {"code": 200,  "msg": "查询成功", "data":[{"count(\*)": 11}]} |

4.5.5 plan

plan部分接口详细设计如表4-11所示。以下请求路径均已省略http://ip: 8080 /plan。其中参数与plan表字段一一对应。

表4-11 plan接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取所有未完成计划 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功",  "data":[{"plan\_id": 8,  "user\_id":"413192b4ed8744c380fdc68c5c6804c4",  "title": "毕设",  "content": "论文",  "urgency": 0,  "pass": "0m",  "finish": 0}]} |
| 删除计划 | post | /delete | {plan\_id:8} | {"code": 200,  "msg": "删除成功"} |
| 添加计划 | post | /add | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  title: '计划1',  content: "复习",  urgency: 0,} | {"code": 200,  "msg": "添加成功"} |
| 修改计划 | post | /update | {todo\_id\_id:39,  title: '学习'} | {"code": 200,  "msg": "修改成功"} |
| 进行计划 | post | /pass | {todo\_id\_id:39,  isChecked: 1} | {"code": 200,  "msg": "修改成功"} |
| 完成计划 | post | /finish | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date:'2020-05-30’} | {"code": 200,  "msg": "查询成功", "data":[{"count(\*)": 11}]} |

4.5.6 chart

todo部分接口详细设计如表4-10所示。以下请求路径均已省略http://ip: 8080 /todo。其中参数与todo表字段一一对应。

表4-7 user接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取好友列表 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功",  "data":[{"friend":"'413192b4ed8744c380fdc68c5c6804c4','1d6c2fd7c1cb40b09999a9b4de75f523',"}]} |
| 删除或添加好友 | post | /update | {user\_id:"'413192b4ed8744c380fdc68c5c6804c4',  ids:"'413192b4ed8744c380fdc68c5c6804c4'"} | {"code": 200,  "msg": "查询成功",  "data":[{"user\_id": "413192b4ed8744c380fdc68c5c6804c4",  "name": "jc",  "plan\_time":"1h1m"},{"user\_id":"1d6c2fd7c1cb40b09999a9b4de75f523",  "name": "test",  "plan\_time": "0m"}]} |
| 获取用户信息 | post | /getInfo | {ids:"'413192b4ed8744c380fdc68c5c6804c4','1d6c2fd7c1cb40b09999a9b4de75f523',"} | {"code": 200,  "msg": "存在"}  或{"code": 201,  "msg": "不存在"} |
| 获取用户完成待办天数 | post | /getDay | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功", "data":[{"todo\_day": 1}]} |
| 设置用户完成待办天数 | post | /setDay | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  todo\_day: 1} | {"code": 200,  "msg": "更新成功"} |
| 获取用户进行计划时长 | post | /getTime | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功", "data":[{"plan\_time":"1m"}]} |
| 设置用户进行计划时长 | post | /setTime | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  todo\_day: '1m'} | {"code": 200,  "msg": "更新成功"} |

4.5.7 achieve

todo部分接口详细设计如表4-10所示。以下请求路径均已省略http://ip: 8080 /todo。其中参数与todo表字段一一对应。

表4-7 user接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取好友列表 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功",  "data":[{"friend":"'413192b4ed8744c380fdc68c5c6804c4','1d6c2fd7c1cb40b09999a9b4de75f523',"}]} |
| 删除或添加好友 | post | /update | {user\_id:"'413192b4ed8744c380fdc68c5c6804c4',  ids:"'413192b4ed8744c380fdc68c5c6804c4'"} | {"code": 200,  "msg": "查询成功",  "data":[{"user\_id": "413192b4ed8744c380fdc68c5c6804c4",  "name": "jc",  "plan\_time":"1h1m"},{"user\_id":"1d6c2fd7c1cb40b09999a9b4de75f523",  "name": "test",  "plan\_time": "0m"}]} |
| 获取用户信息 | post | /getInfo | {ids:"'413192b4ed8744c380fdc68c5c6804c4','1d6c2fd7c1cb40b09999a9b4de75f523',"} | {"code": 200,  "msg": "存在"}  或{"code": 201,  "msg": "不存在"} |
| 获取用户完成待办天数 | post | /getDay | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功", "data":[{"todo\_day": 1}]} |
| 设置用户完成待办天数 | post | /setDay | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  todo\_day: 1} | {"code": 200,  "msg": "更新成功"} |
| 获取用户进行计划时长 | post | /getTime | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功", "data":[{"plan\_time":"1m"}]} |
| 设置用户进行计划时长 | post | /setTime | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  todo\_day: '1m'} | {"code": 200,  "msg": "更新成功"} |

4.5.8 log

todo部分接口详细设计如表4-10所示。以下请求路径均已省略http://ip: 8080 /todo。其中参数与todo表字段一一对应。

表4-7 user接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取好友列表 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功",  "data":[{"friend":"'413192b4ed8744c380fdc68c5c6804c4','1d6c2fd7c1cb40b09999a9b4de75f523',"}]} |
| 删除或添加好友 | post | /update | {user\_id:"'413192b4ed8744c380fdc68c5c6804c4',  ids:"'413192b4ed8744c380fdc68c5c6804c4'"} | {"code": 200,  "msg": "查询成功",  "data":[{"user\_id": "413192b4ed8744c380fdc68c5c6804c4",  "name": "jc",  "plan\_time":"1h1m"},{"user\_id":"1d6c2fd7c1cb40b09999a9b4de75f523",  "name": "test",  "plan\_time": "0m"}]} |
| 获取用户信息 | post | /getInfo | {ids:"'413192b4ed8744c380fdc68c5c6804c4','1d6c2fd7c1cb40b09999a9b4de75f523',"} | {"code": 200,  "msg": "存在"}  或{"code": 201,  "msg": "不存在"} |
| 获取用户完成待办天数 | post | /getDay | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功", "data":[{"todo\_day": 1}]} |
| 设置用户完成待办天数 | post | /setDay | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  todo\_day: 1} | {"code": 200,  "msg": "更新成功"} |
| 获取用户进行计划时长 | post | /getTime | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功", "data":[{"plan\_time":"1m"}]} |
| 设置用户进行计划时长 | post | /setTime | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  todo\_day: '1m'} | {"code": 200,  "msg": "更新成功"} |

# 第5章 个人管理app的系统实现

5.1 开发环境

5.2用户模块（程序流图、页面截图）

5.3好友模块

5.4 备忘录模块

5.5 每日待办模块

5.6 计划模块

5.7 成就模块

5.8 时间历程模块

# 第6章 个人管理app的测试

6.1 测试环境

6.2 测试方法？过程？内容？结果

结　论

本文结论……。

结论作为毕业设计（论文）正文的最后部分单独排写，但不加章号。结论是对整个论文主要结果的总结。在结论中应明确指出本研究的创新点，对其应用前景和社会、经济价值等加以预测和评价，并指出今后进一步在本研究方向进行研究工作的展望与设想。结论部分的撰写应简明扼要，突出创新性。阅后删除此段。

结论正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。阅后删除此段。

参考文献

[1]潘晶. 浅谈游戏化思维方式在体验性展示设计中的应用[J]. 美术研究,2020,(02):116-118.

附　录

附录相关内容…

附录是毕业设计（论文）主体的补充项目，为了体现整篇文章的完整性，写入正文又可能有损于论文的条理性、逻辑性和精炼性，这些材料可以写入附录段，但对于每一篇文章并不是必须的。附录依次用大写正体英文字母A、B、C……编序号，如附录A、附录B。阅后删除此段。

附录正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。阅后删除此段。

注：附录和致谢页脚内容为页码，宋体、五号，居中排列。阅后删除此文本框。

注：此处无需更改。阅后删除此文本框。

致　谢

值此论文完成之际，首先向我的导师……

致谢正文样式与文章正文相同：宋体、小四；行距：22磅；间距段前段后均为0行。阅后删除此段。