

**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。

本文对于游戏化与个人管理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，设计了一个基于android平台的个人管理app。系统基于Java、使用易用性强的开源项目进行前端开发，使用性能极佳的NodeJs中 express框架进行后端开发，数据库采用服务稳定的mysql。

**关键词：游戏化；个人管理；NodeJs；安卓**

**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. The system is based on Java and uses easy-to-use open source projects for design. The express framework in NodeJs with excellent performance is used for server development. The database uses stable mysql.

**Key Words: gamification；personal management；NodeJs；Android**

目　录

[摘　要 I](#_Toc42474478)

[Abstract II](#_Toc42474479)

[第1章 引言 1](#_Toc42474480)

[1.1 概要叙述 1](#_Toc42474481)

[1.2 背景介绍 1](#_Toc42474482)

[1.2.1 游戏化 1](#_Toc42474483)

[1.2.2 个人管理 2](#_Toc42474484)

[1.3 研究意义 2](#_Toc42474485)

[1.4 论文结构 3](#_Toc42474486)

[第2章 个人管理及游戏化的研究 4](#_Toc42474487)

[2.1 个人管理 4](#_Toc42474488)

[2.1.1 价值感 4](#_Toc42474489)

[2.1.2 自我效能感 4](#_Toc42474490)

[2.1.3 时间监控 5](#_Toc42474491)

[2.1.4 研究结果 5](#_Toc42474492)

[2.2 游戏化 5](#_Toc42474493)

[2.2.1 积分 5](#_Toc42474494)

[2.2.2 勋章 6](#_Toc42474495)

[2.2.3 排行榜 6](#_Toc42474496)

[2.2.4 研究结果 6](#_Toc42474497)

[第3章 个人管理app的需求分析 8](#_Toc42474498)

[3.1系统总体需求分析 8](#_Toc42474499)

[3.2 用户模块 8](#_Toc42474500)

[3.2.1 用例图 8](#_Toc42474501)

[3.2.2 功能需求分析 8](#_Toc42474502)

[3.3 好友模块 9](#_Toc42474503)

[3.3.1 用例图 9](#_Toc42474504)

[3.3.2 功能需求分析 10](#_Toc42474505)

[3.4 备忘录模块 10](#_Toc42474506)

[3.4.1 用例图 10](#_Toc42474507)

[3.4.2 功能需求分析 10](#_Toc42474508)

[3.5 每日待办模块 11](#_Toc42474509)

[3.5.1 用例图 11](#_Toc42474510)

[3.5.2 功能需求分析 11](#_Toc42474511)

[3.6 计划模块 11](#_Toc42474512)

[3.6.1 用例图 11](#_Toc42474513)

[3.6.2 功能需求分析 12](#_Toc42474514)

[3.7 时间历程模块 12](#_Toc42474515)

[3.7.1 用例图 12](#_Toc42474516)

[3.7.2 功能需求分析 12](#_Toc42474517)

[3.8 成就模块 13](#_Toc42474518)

[3.8.1 用例图 13](#_Toc42474519)

[3.8.2 功能需求分析 14](#_Toc42474520)

[第4章 个人管理app的设计 15](#_Toc42474521)

[4.1 概要设计 15](#_Toc42474522)

[4.2 系统模块的详细设计 15](#_Toc42474523)

[4.2.1 用户模块详细设计 15](#_Toc42474524)

[4.2.2 好友模块详细设计 16](#_Toc42474525)

[4.2.3 备忘录模块 17](#_Toc42474526)

[4.2.4 每日待办模块 17](#_Toc42474527)

[4.2.5 计划模块 18](#_Toc42474528)

[4.2.6 成就模块 18](#_Toc42474529)

[4.2.7 时间历程模块 19](#_Toc42474530)

[4.3 数据库设计 20](#_Toc42474531)

[4.3.1 数据库总体设计 20](#_Toc42474532)

[4.3.2 user表设计 22](#_Toc42474533)

[4.3.3 note表设计 23](#_Toc42474534)

[4.3.4 todo表设计 24](#_Toc42474535)

[4.3.5 plan表设计 24](#_Toc42474536)

[4.3.6 chart表设计 24](#_Toc42474537)

[4.3.7 log表设计 25](#_Toc42474538)

[4.4 接口设计 26](#_Toc42474539)

[4.4.1 user 26](#_Toc42474540)

[4.4.2 friend 26](#_Toc42474541)

[4.4.3 note 28](#_Toc42474542)

[4.4.4 todo 28](#_Toc42474543)

[4.4.5 plan 30](#_Toc42474544)

[4.4.6 achieve 32](#_Toc42474545)

[4.4.7 chart 32](#_Toc42474546)

[4.4.8 log 32](#_Toc42474547)

[第5章 个人管理app的实现 35](#_Toc42474548)

[5.1 开发环境 35](#_Toc42474549)

[5.2 用户模块 35](#_Toc42474550)

[5.2.1 页面截图 35](#_Toc42474551)

[5.2.2 技术细节 35](#_Toc42474552)

[5.2.3 程序流图 38](#_Toc42474553)

[5.3 好友模块 38](#_Toc42474554)

[5.3.1 页面截图 38](#_Toc42474555)

[5.3.2 技术细节 39](#_Toc42474556)

[5.4 备忘录模块 40](#_Toc42474557)

[5.4.1 页面截图 40](#_Toc42474558)

[5.4.2 技术细节 41](#_Toc42474559)

[5.5 每日待办模块 42](#_Toc42474560)

[5.5.1 页面截图 42](#_Toc42474561)

[5.5.2 技术细节 43](#_Toc42474562)

[5.5.3 程序流图 43](#_Toc42474563)

[5.6 计划模块 44](#_Toc42474564)

[5.6.1 页面截图 44](#_Toc42474565)

[5.6.2 技术细节 46](#_Toc42474566)

[5.6.3 程序流图 48](#_Toc42474567)

[5.7 成就模块 49](#_Toc42474568)

[5.7.1 页面截图 49](#_Toc42474569)

[5.7.2 技术细节 49](#_Toc42474570)

[5.8 时间历程模块 50](#_Toc42474571)

[5.9 接口实现 51](#_Toc42474572)

[5.9.1 语言框架 51](#_Toc42474573)

[5.9.2 跨域 51](#_Toc42474574)

[5.9.3 断线重连 52](#_Toc42474575)

[第6章 个人管理app的测试 53](#_Toc42474576)

[6.1 页面跳转测试 53](#_Toc42474577)

[6.2 功能测试 53](#_Toc42474578)

[6.2.1用户模块 53](#_Toc42474579)

[6.2.2 好友模块 54](#_Toc42474580)

[6.2.3 备忘录模块 55](#_Toc42474581)

[6.2.4 每日待办模块 55](#_Toc42474582)

[6.2.5 计划模块 56](#_Toc42474583)

[6.2.6 成就模块 58](#_Toc42474584)

[6.2.7 时间历程模块 58](#_Toc42474585)

[6.3 接口测试 59](#_Toc42474586)

[结　论 60](#_Toc42474587)

[参考文献 61](#_Toc42474588)

[致　谢 63](#_Toc42474589)

# 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[1]，从而激励用户获得特定价值。游戏化的特色在于将无趣的情景趣味化，在各类严肃的课堂和教育中都有应用。而个人的时间管理对个人的学习工作效率、成果都有着显著的影响。而时间管理往往因其对于自制力的高要求让人难以坚持，从而影响导致拖延行为愈加普遍。时间管理能力越低，个体焦虑、压力、紧张、抑郁等消极情绪体验越多，个体时间管理能力的增强，有助于个体积极情绪体验的增加[2]。因此，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，在世面常见的个人管理类app中都有体现。如饱受好评的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

1.2.1 游戏化

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”。 Richard Bartle和Roy Trubshaw开发了一个地下城堡游戏，将非游戏的文本系统变成了游戏。直至2010年第一届游戏化峰会的举办,游戏化才开始广泛流行起来[3]。

游戏化作为提高用户参与度和黏着度的有效途径，在教育等严肃行业广受推崇，被学者认为是教育革新的有效手段。在游戏化研究的学科中，教育相关的研究也名列前茅[4]。显然，游戏化的正确使用是能促进用户对于严肃情景的参与度与忠诚度的，这一点在管理、营销、健康等各个方面的成果都可以作为佐证。但在有些情况下，游戏化的失败设计导致游戏化无法对用户的使用起到激励效果，80%的用现有游戏化产品都无法达成预期目标[5]。无论是对于游戏化元素的错误理解，抑或是对于游戏化元素的过度看重而削弱了用户的原本动机，使用户无法得到应有的提升，游戏化应用之路并非表面上的一帆风顺。

1.2.2 个人管理

个人管理的核心是时间管理，时间管理最初应用于工厂，用于提升生产效率。随着个人对于自我能力的要求不断提高，时间管理也作为衡量个人能力的标准而受到广泛关注。

时间管理的发展经历了四个阶段[6]。最初是备忘录形式，仅以一张张纸条记录待办事项，缺少时间安排，也没有严谨的结构。第二阶段注重于规划，以日程表的形式事先将时间进行规划，以便让时间安排更有条理。第三阶段则更强调效率，以自身价值观为导向，为不同阶段定制目标做出规划，强调目标和行动转化的价值。第四阶段则是迄今为止的最受认可的，注重个人的思想而非行动，以任务的轻重缓急和价值为中心确定目标，更关注效能和个人的自我管理[7]。个人管理早已成为一套发展完成的管理学体系，对个人的提升和发展有着显著影响。

1.3 研究意义

随着应用开发成本的降低，作为市场热门的游戏化元素在各个应用中都可见其身影。游戏化作为一种激励手段，其根本目的始终在于激发用户的内部动机。通过游戏化的奖励机制和使用的乐趣，用户对于原本或许枯燥的任务产生乐趣，是游戏化使用的最好场景。

然而并非所有场景都能融入游戏化元素，也并非所有的游戏化元素都能对其本质目的产生积极的影响。就个人管理而言，是极需要自制力而又对于个人能产生切实影响的严肃情境。传统的形式不利于个人管理的推广和接受，而过于游戏化又往往在增加用户粘性的同时削弱了个人管理的严肃意义，本末倒置。个人管理的游戏化，绝非简单的将广为人知、广为人用的积分元素照搬至个人管理中。对于影响个人管理的各种因素、以及游戏化的各种元素在其中相互作用结果的研究，都是很有必要的。基于此，设计并开发出一款满足要求的app，可以更好地展示游戏化与个人管理的内在意义。

1.4 论文结构

本论文紧紧围绕“游戏化个人管理Android App的设计与实现”这一课题，对于主流的个人管理app，如Jiffy，Forest，番茄todo，Pocket Lists等使用的游戏化元素进行研究，结合影响用户进行时间管理的各种因素，设计出一个添加游戏化元素的小型个人管理app，并使用Java、NodeJs进行开发。论文结构如下：

第 1 章 引言，对于游戏化和个人管理进行简要介绍，并阐述了课题研究的意义。

第 2 章 游戏化及个人管理元素的研究，研究适用于个人管理app的游戏化元素，及个人管理的影响因素。

第 3 章 个人管理app的需求分析，介绍了本文实现的个人管理android app的功能需求。

第 4 章 个人管理app的设计，介绍了系统的概要设计、各功能模块详细设计、数据库设计和接口设计。

第 5 章 个人管理app的实现，介绍了系统各功能模块的实现方法。

第 6 章 个人管理app的测试，对于系统进行功能测试，确保系统各功能模块正常运行。

# 第2章 个人管理及游戏化的研究

2.1 个人管理

2.1.1 价值感

价值感，即对于所付出的获得成果的价值衡量。传统学习往往让人感觉枯燥，其原因就在于付出的努力得不到及时的价值体现，一段时间的努力往往要经过很长时间才能得到价值的反馈。而游戏则恰恰相反，每一个动作都能够得到及时的反馈，这也是其趣味性之一。

因此在个人管理中，以清单形式将待办任务罗列，完成后将其与未完成进行标记区分。随着完成标记的逐渐增多，直至所有的任务都标记完成，成就感也逐渐累积，获得自我价值的认知。如Pocket Lists在所有任务完成时飘下的彩带，就是为用户获得价值的喝彩。这种形式不仅让待办任务以更清晰的形式展现，更激励着用户去依次消灭所有未完成标记，以及时获得成就感。其在过程中为完成任务付出的努力，也及时展现着价值。

2.1.2 自我效能感

20世纪70年代，美国著名心理学家班杜拉于在其著作《思想和行为的社会基础》中提出了自我效能感的概念，即面对一项任务，依据对自我能力的判定，对于自己是否有完成的能力的推测。曹操“望梅止渴”的成功，也就在于将艰巨的任务细化，提高了兵士的自我效能感，从而对于完成任务也充满了积极的情绪。

边玉芳依据班杜拉的“三元交互作用论”和自我效感理论，将自我效能感分为能力感、努力感、环境感和控制感[8]。其中的环境感，就是学习环境对个人判定造成的影响。Deadline作为大多数人头顶的达摩克里斯之剑，若其过于艰巨让人对于自己完成可能性的估计降低，即使看着其慢慢逼近也难以坚持进行。在此种情况下，加上一些参照物，如已完成的任务，往往会影响用户的环境感，从而对用户的自我效能感造成影响。同样，清单上已完成的待办任务，也对用户关于未完成任务的自我效能感产生着积极的影响。

2.1.3 时间监控

时间监控常包括设置目标、制定计划、设定事件优先级、分配时间、计划反馈性五个部分[9]。对于时间有着良好的把握能力，也是个人管理的重要一环。将任务依次细分，按部就班完成即可。

时间监控映射在个人管理上的计划，就是复杂版的待办任务。包括了多个事件的优先级、时间的分配，而不单只关注一个任务的完成与否。此外，计划更多的是进行中的连续状态，而非待办任务的是否两种情况。计划随时可以暂停，根据优先级插入新的计划，更具有灵活性，也让时间管理显得更为游刃有余。

2.1.4 研究结果

价值感、自我效能感与时间监控是自我管理的重要要素。周永红，吕翠芳，杨于岑[10]的研究结果表明，三者对于个人管理的自主性都起到积极的正向作用。因此，一个能吸引用户、激励用户的个人管理系统，应能提升用户的自我效能感与价值感，为用户提供进行时间监控的功能。

2.2 游戏化

游戏化的吸引人之处归根究底是其核心：乐趣。Nicole将游戏的乐趣分为四类：轻松乐趣、艰巨乐趣、人际乐趣和严肃乐趣[11]。顾名思义，轻松乐趣让人放松，艰巨乐趣让用户从获得征服挑战的快感，人际乐趣为用户提供竞争或合作的社交乐趣，严肃乐趣则是为了解决现实问题。基于此，个人管理的游戏化核心就在于严肃乐趣，社交乐趣也是可选项。

依据肖亦奇[12]对于玩家动机类型的总结，包括积累性成就、竞争性拯救、团队形成就、社交归属与深度关系、探索与自由度、叙事性的角色扮演、个性表达、逃避现实八种类型。显然个人管理类属于第一种，积累性成就也就是用户使用本系统的核心追求，即内部动机。激励理论中，将动机分为内部动机和外部动机两种[13]。而作为推进内在动机的手段，外部动机主要包括PBL三大元素：积分、勋章和排行榜[14]。

2.2.1 积分

积分是获得勋章及进行排行的基础，可分为价值型、经验型、混合型三种类型，代表着不同的价值导向[15]。其中价值型积分模式倾向于物质奖励，积分存在着一定汇率，可用于消费；而经验型积分模式旨在于鼓励，积分仅涉及等级相关；混合型模式则是二者兼有。

作为个人管理工具，本系统自然采用经验型积分模式。本系统中，采用个人管理的核心功能——时间监控计算积分。将用户分配于各个计划的时间量化为积分，为用户提供了足够及时的反馈。同时，进行时间监控也是用户的内在动机，也让对于积分追求的外在动机与内部动机保持了一致。

2.2.2 勋章

勋章是对用户特定行为奖励的具象化，可分为连续奖励、固定比率的奖励、固定时间的奖励和不定奖励四种[16]。勋章的收集也是满足感和成就感的一大来源。在综合的个人管理app中，用户付出了多种类型的努力，包括每日任务的完成和对计划时间的分配，对此设定两种勋章，用于奖励和激励用户的坚持使用。

然而，心理学研究表明，过分强调外部动机往往会削弱内部动机[17]。奖励并非全然是一件好事，甚至过于轻易的奖励往往更容易让人厌弃。因此，本系统仅设置两种类型各六种，且获取难度逐渐升高。

2.2.3 排行榜

排行榜作为社交乐趣的重要来源，体现着用户之间的竞争。竞争从来都能激励特定行为的发生。在排行榜上将不同等级的用户区分化，更有利于激励用户对于超越的积极性。此外，本系统的排行榜由坚持计划时长决定，颇受用户使用系统时间长短影响。故将竞争范围划分在好友内部，增加了排名的含金量。

2.2.4 研究结果

外部动机是对于系统使用的吸引之处，游戏化的过程也是使用的新奇感所在。但随着新奇感逐渐消退，往往对于系统使用的坚持也会随之消失。游戏化相较于传统改变的是表达的方式，其内核是并且必须始终如一。外部动机促使用户对系统的使用，而内部动机则是用户使用系统的意义所在。外部动机应与内部动机进行融合，在外部动机的激励中，内部动机逐渐增强甚至形成的自律，才是用户对于系统忠诚度的保证。

就个人管理系统而言，其意义在于用户对于“个人管理”这一行为的坚持，从而进行自我的提升。个人管理系统始终只是一个协助工具，系统进行时，用户的注意力始终是在具体任务的进行上的，游戏化的体现也应忽视固有的架构，而是以用户体验为核心。因此，个人管理系统应保持简洁以免喧宾夺主，在外部动机中融入对于内部动机的提升，才是合理的游戏化。

# 第3章 个人管理app的需求分析

3.1系统总体需求分析

系统总体需求如图3-1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3-1 系统功能模块**

3.2 用户模块

3.2.1 用例图

用户模块的用例图如图3-2所示，包括登录、注册功能。

3.2.2 功能需求分析

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3-2 用户模块用例图

3.3 好友模块

3.3.1 用例图

好友模块的用例图如图3-3所示，包括展示排行榜、添加好友、删除好友功能。



图3-2 好友模块用例图

3.3.2 功能需求分析

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。

3.4 备忘录模块

3.4.1 用例图

备忘录模块的用例图如图3-4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。



图3-4 备忘录模块用例图

3.4.2 功能需求分析

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。

3.5 每日待办模块

3.5.1 用例图

待办模块的用例图如图3-5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。



图3-5 每日待办模块用例图

3.5.2 功能需求分析

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。

3.6 计划模块

3.6.1 用例图

计划模块的用例图如图3-6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。



图3-6 计划模块用例图

3.6.2 功能需求分析

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。

3.7 时间历程模块

3.7.1 用例图

时间历程模块的用例图如图3-7所示，包括展示时间历程、记录历程功能。

3.7.2 功能需求分析

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3-7 时间历程模块用例图

3.8 成就模块

3.8.1 用例图

成就模块的用例图如图3-8所示，包括展示成就、获取成就功能。



图3-8 成就模块用例图

3.8.2 功能需求分析

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。

# 第4章 个人管理app的设计

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 系统模块的详细设计

4.2.1 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。

为减少操作负荷，使系统尽可能地简洁从而提升用户的使用体验，用户注册时仅需输入用户名，登录时无需操作。进入系统后，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。

4.2.2 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。

好友作为社交元素的重要体现，应有能引发用户社交乐趣的环节，排行榜的竞争机制就是其中重要一环。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**

4.2.3 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。

4.2.4 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**

4.2.5 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**

4.2.6 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。

4.2.7 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历程，计划和成就历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**

4.3 数据库设计

4.3.1 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**

4.3.2 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(32) | 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 |  |

4.3.3 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(32) | 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 |  |

4.3.4 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(32) | 否 | 无 |  |
| title | varchar(100) | 否 | 无 |  |
| date | date | 否 | 无 |  |
| isChecked | int | 否 | 无 |  |

4.3.5 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。

4.3.6 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(32) | 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(32) | 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 |  |

4.3.7 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(32) | 否 | 无 |  |
| type | int | 否 | 无 |  |
| name | varchar(100) | 否 | 无 |  |
| date | date | 否 | 无 |  |

4.4 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。

4.4.1 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。

4.4.2 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"} |

4.4.3 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。

4.4.4 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,  "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"} |
| 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,  "title": "清单1",  "date": "2020-05-30",  "isChecked": 1}]} |
| 删除待办 | post | /delete | {todo\_id:38} | {"code": 200,  "msg": "删除成功"} |
| 删除当日全部待办 | post | /deleteAll | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date:'2020-05-30’} | {"code": 200,  "msg": "删除成功"} |
| 添加待办 | post | /add | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  title: '清单2',  date: '2020-05-30'} | {"code": 200,  "msg": "添加成功"  "data":[{"todo\_id":39}]} |
| 编辑待办 | post | /edit | {todo\_id:39,  title: '学习'} | {"code": 200,  "msg": "修改成功"} |
| 选取是否完成 | post | /check | {todo\_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}]} |

4.4.5 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,  "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: 1} | {"code": 200,  "msg": "添加成功"} |
| 修改计划 | post | /update | {plan\_id'9,  title: '计划1',  content: "复习",  urgency: 0} | {"code": 200,  "msg": "修改成功"} |
| 进行计划 | post | /pass | {plan\_id:9,  pass: '1m'} | {"code": 200,  "msg": "修改成功"} |
| 完成计划 | post | /finish | {plan\_id:9} | {"code": 200,  "msg": "修改成功"} |

4.4.6 achieve

achieve部分接口详细设计如表4-12所示。以下请求路径均已省略http://ip: 8080 /achieve。其中参数对应user表中字段。

表4-12 achieve接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取成就信息 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4} | {"code": 200,  "msg": "查询成功",  "data":[{"prize\_plan": 2,  "prize\_todo": 1}]} |
| 更新用户待办成就 | post | /todo | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  prize\_todo: 3} | {"code": 200,  "msg": "更新成功"} |
| 更新用户计划成就 | post | /plan | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  prize\_plan: 4} | {"code": 200,  "msg": "更新成功"} |

4.4.7 chart

chart部分接口详细设计如表4-13所示。以下请求路径均已省略http://ip: 8080 /chart。其中参数type表示查询范围，0表示查询该周数据（按周一开头计算），1表示查询该月数据，2表示查询概念数据，其余参数与chart表字段一一对应。

4.4.8 log

log部分接口详细设计如表4-14所示。以下请求路径均已省略http://ip: 8080 / log。其中参数与log表字段一一对应。

表4-13 chart接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取图表信息 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date: '2020-05-27',  type:0} | {"code": 200,  "msg": "查询成功",  "data":[{"date": "2020-05-25",  "pass\_red": 0,  "pass\_yellow": 10,  "pass\_green": 70,  "pass\_blue": 0}]} |
| 该日图表是否存在 | post | /isExist | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date: '2020-05-25'} | {"code": 200,  "msg": "存在",  "data":[{"chart\_id": 12,  "user\_id": "jc",  "date": "2020-05-25",  "pass\_red": 0,  "pass\_yellow": 10,  "pass\_green": 70,  "pass\_blue": 0}]}  或{"code": 201,  "msg": "不存在"} |
| 添加图表信息 | post | /add | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date: '2020-05-26',  pass\_red: 0,  pass\_yellow: 10,  pass\_green: 0,  pass\_blue: 0} | {"code": 200,  "msg": "添加成功"} |

表4-14 log接口详细设计

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 接口功能 | 请求方法 | 请求路径 | 参数 | 返回值 |
| 获取历程信息 | post | /query | {user\_id:'413192b4ed8744c380fdc68c5c6804c4'} | {"code": 200,  "msg": "查询成功",  "data":[{"log\_id": 23,  "type": 2,  "name": "达成成就“浅尝辄止”",  "date": "2020-05-30"},  {"log\_id": 22,  "type": 0,  "name": "完成10项待办",  "date": "2020-05-30"}]} |
| 该历程是否存在 | post | /isExist | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  date: '2020-05-25',  type:2} | {"code": 200,  "msg": "存在",  "data":[{"log\_id": 22}]}  或{"code": 201,  "msg": "不存在"} |
| 添加历程信息 | post | /set | {user\_id:'413192b4ed8744c380fdc68c5c6804c4',  type: 1,  name: "经过3小时5分钟，完成计划“复习”",  date: "2020-05-31"} | {"code": 200,  "msg": "添加成功"} |
| 更新历程 | post | /update | {log\_id:22,  name: "完成5项待办"} | {"code": 200,  "msg": "更新成功"} |

# 第5章 个人管理app的实现

5.1 开发环境

前端开发语言：Java

前端开发工具：Android Studio

后端开发语言：NodeJs

后端开发工具：WebStorm

数据库：MySQL

数据库管理工具：Navicat

服务器：腾讯云

服务器管理工具：FinalShell

项目管理工具：Github

5.2 用户模块

5.2.1 页面截图

由于虚拟机无法输入中文，以下所有模块页面截图均为真机运行效果。

注册页如图5-1所示，若无账号则需先注册，输入不重复、50字符内的用户名；若该设备已注册过账号则系统自动获取信息。系统获取用户名和id后自动登录，故登录页不展示。个人信息页如图5-2所示，展示用户名，用户id隐藏，按文字点击可直接复制id至剪贴板。

5.2.2 技术细节

系统首先对于是否首次注册进行判定，使用getUser()函数获取用户id与用户名。函数中使用FileInputStream流接收文件，使用BufferReader读取数据流。若文件不存在，读取出错，返回false；若文件存在，读取到文件内容，由于文件内容按“用户id;用户名”格式存储，使用String的分割函数即可获取用户id和用户名，函数返回值为真。以下为判定代码：

private boolean getUser(){  
 try {  
 File file = new File(this.getFilesDir()+"/user.txt");//根据地址读取文件  
 FileInputStream inputStream = new FileInputStream(file);  
 BufferedReader bufferedReader = new BufferedReader(new InputStreamReader (inputStream));  
 String info[] = bufferedReader.readLine().split(";");//存储内容为user\_id;用户名  
 ……  
 return true;  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 return false;  
}

若本地文件读取失败，则跳转至注册页面，用户手动进行注册。先判定该输入是否合法，当输入内容合法时方才对数据库进行插入操作，同时也存储为本地文件。以下为将个人信息存储至本地代码：

private void setUser(){

UUID id = UUID.randomUUID();//生成唯一用户id

String FILENAME = "user.txt";

user\_id = id.toString().replace("-","");//将id转成纯数字字母组合

FileOutputStream fos = null;

try {

fos = openFileOutput(FILENAME, Context.MODE\_PRIVATE);

fos.write((user\_id+";"+name).getBytes());//写入信息

fos.close();

} catch (IOException e) {

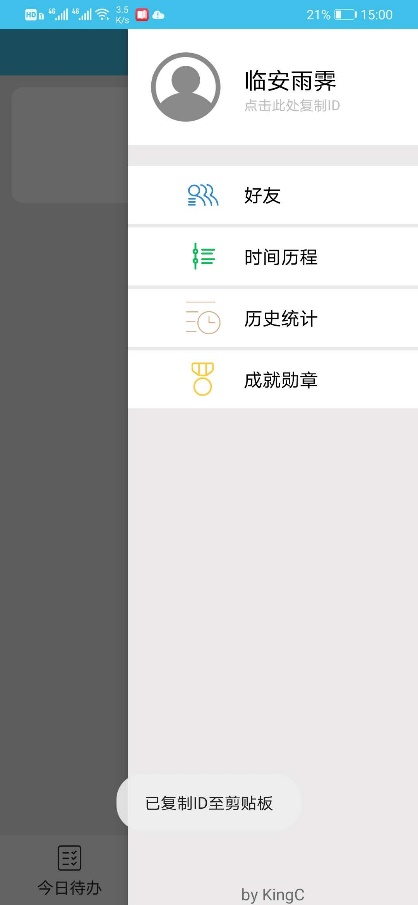
e.printStackTrace();

}

}



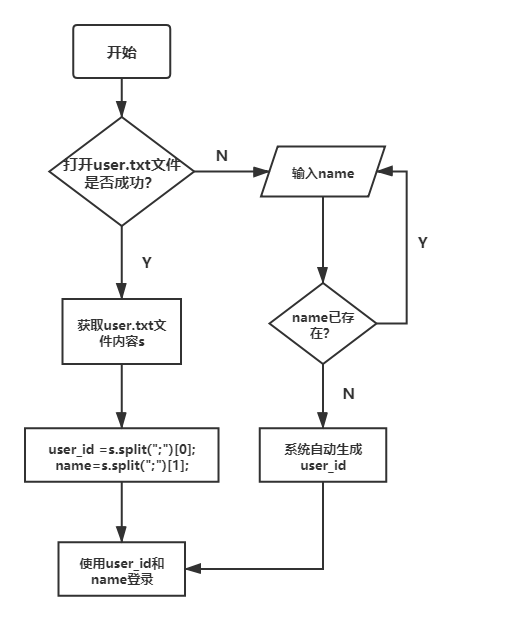
**图5-1 注册页页面截图**



**图5-2 个人信息截图**

5.2.3 程序流图

图5-3为用户进入系统、注册直至登录的程序流图。



**图5-3 登录功能程序流图**

5.3 好友模块

5.3.1 页面截图

好友排行榜页面截图如图5-4所示。以进行计划时长作为排名标准，根据不同时长生成不同颜色背景的头像。前三名排名及进行时长分别对应金、银、铜三种字体颜色，其后颜色为普通黑色。在底部输入框可输入用户id以添加好友，已在列表好友不可添加，输入正确id可添加成功。点击好友可选择删除，不可删除用户自己。



**图5-4 好友页页面截图**

5.3.2 技术细节

获取数据后积分以“xhxm”字符串形式展示，在Global类中创建countTime()方法，将此类字符串转换成分钟的数值，代码如下：

public int countTime(String time) {

int min = 0;

time= time.substring(0, time.length() - 1);

if (time.indexOf('h') != -1) {//字符串为xm形式

String s[]=time.split("h");

min = 60\*Integer.parseInt(s[0]);

time = s[1];

}

min+=Integer.parseInt(time);

return min;

}

FriendActivity中排序代码如下：

Collections.sort(friend\_list, new Comparator<Friend>() {

@Override

public int compare(Friend o1, Friend o2) {

return global.countTime(o2.getTime())-global.countTime(o1.getTime());

}

});

5.4 备忘录模块

5.4.1 页面截图

备忘录页面截图如图5-5所示。该页分为上下两部分，上部分为日历，可展示当月或收缩为当周，下半部分为备忘录列表，点击左上角日期可修改日期。日历部分以日期右上角标展示备忘，与列表中备忘图像内容一致。列表部分按时间顺序展示该备忘的详细信息。若是当日已经历备忘，文字内容为浅灰色；若是当日未经历备忘，文字内容为红色；若是未来备忘，文字内容为正常黑色。选取日期后点击添加按钮可跳转详情页添加，点击列表中单项备忘可选择删除，点击单项备忘右侧修改图标可进行修改。



**图5-5 备忘录页面截图**

图5-6为备忘录详情页截图，日期为选定日期。若是添加备忘录，则所有输入框为空，时间默认当时时间；若为修改备忘录，系统自动加载修改前内容。主题为空时不可提交。



**图5-6 备忘录详情页页面截图**

5.4.2 技术细节

本模块使用了黄海彬的开源项目CalendarView，使用canvas绘制，速度快内存低，同时支持自主绘制。在对于日历控件中的备忘标记时，采取HashMap进行存储，在数据量较大时也不影响遍历性能提高性能。绘制备忘标记代码如下：

private void initScheme() {

String date = "";

Map<String, Calendar> map = new HashMap<>();

for (int i = 0; i < note\_list.size(); i++) {

String[] d = note\_list.get(i).getDate().split("-");

if(date.equals(note\_list.get(i).getDate())) continue; //只绘制当日首个备忘

……

map.put(getSchemeCalendar(year, month, day, colors[(year + month + day) % 6], txt).toString(),

getSchemeCalendar(year, month, day, colors[(year + month + day) % 6], txt));

date = note\_list.get(i).getDate();

}

mCalendarView.setSchemeDate(map);

}

5.5 每日待办模块

5.5.1 页面截图

当日待办页面截图如图5-7所示。修改以往日期会展示历史待办，修改未来日期不展示内容。点击重复图标可重复该日所有待办项内容至当日的第二日。对于单个待办项，可点击进行删除，或点击左侧修改按钮进行修改。点击选择框可选择完成，选取完成后该待办项文字加删除线，不可再修改，取消选取后可正常操作。在输入框中可添加待办项，待办内容可与已有待办重复，但不得为空。对于历史待办，对于单个待办项不可进行任何操作，也不能添加新待办。



**图5-7 待办页页面截图**

5.5.2 技术细节

对于获取待办和每个添加、删除、选取或取消选取操作，未完成的待办数量都会更新。对于删除、选取等减少未完成待办数量的操作，都对是否完成进行判定。若未完成待办数量为0，且当页还存在其他待办，则对于历程数据进行插入或更新操作。若是当日首次完成，用户信息同时更新，并对能否获得成就进行判定，若获得成就，添加获得成就历程。相关代码如下：

if(notfinish==0){//若不存在未完成待办

……

int num = getNum(date); //获取当前存在待办数

if(num>0){//当前存在待办事项

int log\_id = isExist(0,date);

if(log\_id==-1){//首次完成所有待办，历程未创建

int day = getTododay()+1;//获得用户完成呆半天书

setTododay(day);//用户信息更新完成待办天数

global.setLog(0, "完成" + num + "项待办", date);//创建完成待办历程

……

if(pos!=prize.length){//完成待办天数达到获取成就标准

setAchieve(2\*pos+1);//更新用户成就信息

……

global.setLog(2, content,date);//创建获得成就历程

}

}else {//非首次完成

updateLog(log\_id,"完成" + num + "项待办");//更新历程

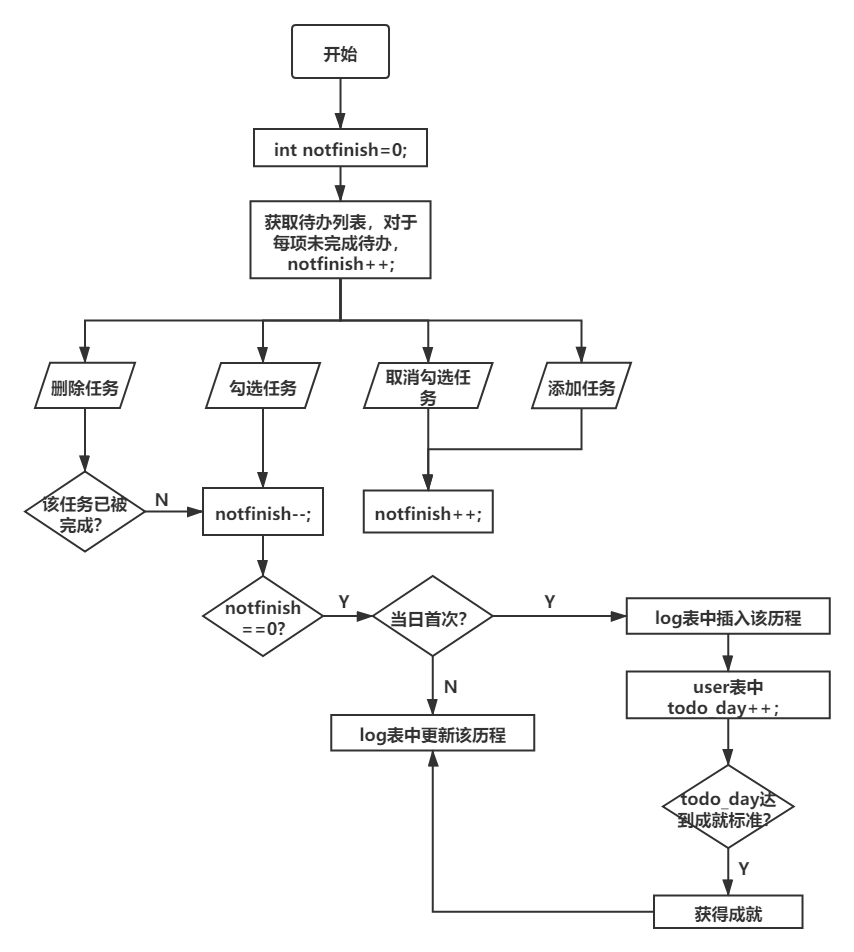
}

}

}

5.5.3 程序流图

图5-8为对待办进行操作后用户信息、历程、成就相关更新的程序流图。



**图5-8 勾选待办程序流图**

5.6 计划模块

5.6.1 页面截图

计划页面截图如图5-9所示，图中的列表为计划列表，按紧急度优先、创建时间其次的顺序排序。点击右下添加按钮可跳转详情页添加，点击单项计划右侧功能键可对其进行编辑、删除、完成操作。点击单项计划可选取计划以进行，该计划展示在上方计时部分。计时部分右侧为计时器，按秒展示。中部为暂停、进行按钮，若在一项任务进行时选取其他任务，或删除、完成该项任务，系统预先自动暂停进行数据更新后操作。



**图5-9 计划页页面截图**

计划详情页面截图如图5-10所示，若是添加则内容全部为空，若为修改自动加载修改前内容。紧急度为四个选择框，以颜色标识不同紧急度。主题为空或紧急度未选择时不可提交。



**图5-10 计划详情页页面截图**

图5-13为计划图表信息页面截图，仅可查看。该页根据选取日期查看方式，以图表形式展示该周、月、年内完成四种不同紧急度计划的时长，点击每段柱状图会显示对应数据。查看时，系统自动调至可查看范围，按月查看时可左右滑动图表。该页面仅可从计划页进入。



**图5-13 计划图表信息页面截图**

5.6.2 技术细节

计划完成时，历程信息更新。计划暂停时，用户进行计划时长更新，chart表中创建或更新数据。若旧时长与新时长之间包含成就标准，则更新成就信息，同时创建获取成就的历程信息。相关代码如下：

int user\_pass\_before = global.countTime(getTime());//旧时长

int pass\_add = global.countTime(now\_t)-global.countTime(plan\_now.getPass());//新增时长

……

int pos = 0;

for(;pos<prize.length;pos++) {

if (user\_pass\_before < prize[pos]\*60 && prize[pos]\*60<=user\_pass\_now)

break;

}

if(pos!=prize.length) {

setAchieve(2 \* pos);//更新成就信息

String content = "达成成就”" + global.getAchieve().get(2 \* pos+1).getName() + "”";

global.setLog(2, content, date);//新建历程

Toast.makeText(getContext(), content, Toast.LENGTH\_SHORT).show();

}

计划的图表信息使用了开源框架hellochart，提供了丰富的可定制性。根据用户选定的类型设置x轴长度，再从列表读取数据并记录最早存在数据的柱子位置。柱状图的子柱按四种颜色累加展示，y轴上限为24。

int subColumns=4;//红、黄、蓝、绿四个子柱

//初始化x轴

switch (type){

case 0:numColumns=7;break;

case 1:numColumns=getMonthDay();break;//获得当月天数

case 2:numColumns=12;break;

}

//设置每个柱

for (int i = 0; i < numColumns; ++i) {

values = new ArrayList<SubcolumnValue>();

for(int j=0;j<subColumns;j++) {

//对应四种颜色

values.add(new SubcolumnValue(hour[i][j], Color.parseColor(clr[j])));

}

columns.add(new Column(values).setHasLabelsOnlyForSelected(true));

}

data = new ColumnChartData(columns);//设置数据

data.setStacked(true);//子柱状累加展示

……

data.setFillRatio(0.75F);

col\_chart.moveTo(beg, 0);//从有数据开始展示

Viewport v = col\_chart.getCurrentViewport();

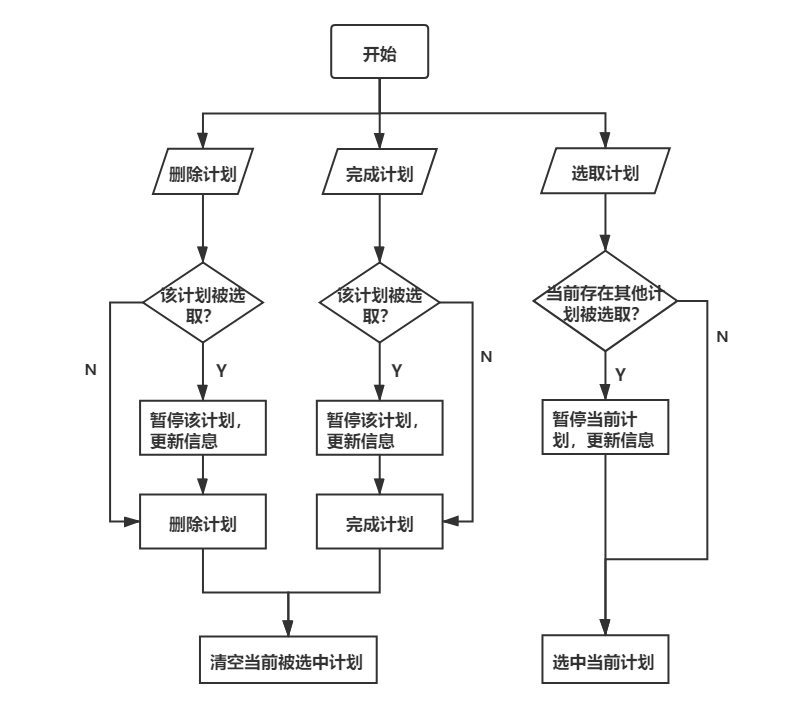
v.top = 25;

col\_chart.setMaximumViewport(v);//设置y轴范围

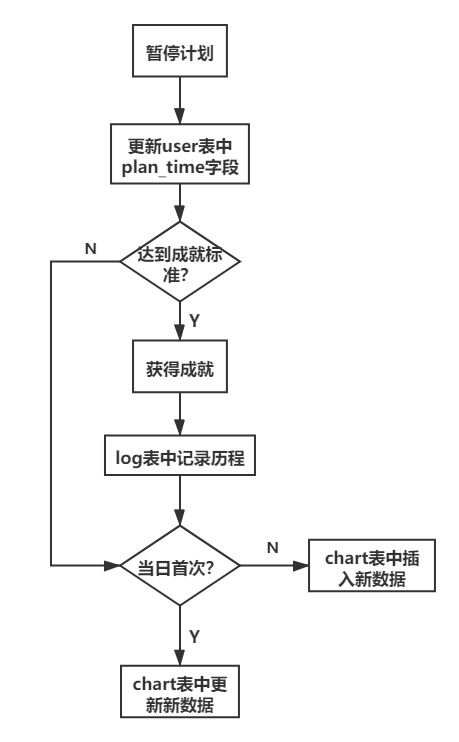
col\_chart.setCurrentViewport(v);

5.6.3 程序流图

图5-11为计划模块总体程序流图，图5-12为暂停当前计划后用户信息、图表信息、历程、成就相关更新的程序流图。



**图5-11 计划页**程序流图



**图5-12 计划暂停后数据更新**程序流图

5.7 成就模块

5.7.1 页面截图

图5-14为成就页页面截图，该页仅可查看。奖杯中计划或待办图标代表该项成就类型，金、银、铜颜色代表成就等级。成就包括完成当日全部待办日期数和累积进行计划时长两种类型，每个类型六个成就，包括一个金色成就、两个银色成就和三个铜色成就，两个类型的六种成就一一对应。成就按照获得难易程度降序排列，计划成就展示在前。点击奖杯时会弹出该项成就达成条件。

5.7.2 技术细节

成就设置代码如下：

private ArrayList<Achieve> achieve = new ArrayList<Achieve>(Arrays.asList(

new Achieve(1,5,"浅尝辄止","累计完成待办1天"),

new Achieve(2,2,"走马观花","累计进行计划1小时"),

new Achieve(3,5,"学贵有恒","累计完成待办7天"),

new Achieve(4,2,"目不转睛","累计进行计划24小时"),

new Achieve(5,5,"锲而不舍","累计完成待办30天"),

new Achieve(6,2,"聚精会神","累计进行计划50小时"),

new Achieve(7,4,"持之以恒","累计完成待办100天"),

new Achieve(8,1,"全神贯注","累计进行计划100小时"),

new Achieve(9,4,"细水长流","累计完成待办500天"),

new Achieve(10,1,"心无旁骛","累计进行计划500小时"),

new Achieve(11,3,"聚沙成塔","累计完成待办1000天"),

new Achieve(12,0,"废寝忘食","累计进行计划1000小时")

));



**图5-14** 成就**页页面截图**

5.8 时间历程模块

图5-15为时间历程页页面截图，该页仅可查看，时间历程按时间顺序由近及远展示。每项历程对应表示类型的图标、完成该历程的日期及历程详细内容，类型包括获得成就类、完成待办数量类和完成计划类。



**图5-15 时间历程页页面截图**

****5.9 接口实现****

****5.9.1 语言框架****

后端最常用的开发语言是Java，对于复杂计算的稳定性让其在后端开发中屹立不倒，但新兴的NodeJs因其轻量和开发的便捷也饱受好评。Js完全支持非阻塞的IO，使用单线程处理请求，在等待时IO时仍能高效地使用服务器，具有极强的高并发处理能力。而Java的优点在于对复杂计算的支持，更适用于大型项目。故在本系统中，使用NodeJs的express框架进行后端开发。部署服务器后使用pm2管理。

****5.9.2 跨域****

由于服务器的访问端口和数据库的数据传输端口不同，存在跨域问题。故在开始文件app.js中加入允许跨域的代码：

//设置跨域访问

app.all("\*", function (req, res, next) {

res.header("Access-Control-Allow-Origin", "\*");

res.header("Access-Control-Allow-Headers", "\*");

res.header("Access-Control-Allow-Methods", "\*");

res.header("Content-Type", "application/json;charset=utf-8");//数据形式为json

res.header("Access-Control-Allow-Credentials",true);//携带cookie跨域请求

req.method.toUpperCase() === "OPTIONS" ? res.sendStatus(200) : next();//防止在预请求阶段就响应接口

});

5.9.3 断线重连

为适应高并发，系统使用连接池进行数据库连接的管理，并实现断线重连机制。接口调用mysql数据库时直接使用pool.getConnection()即可。断线重连机制实现代码如下：

function repool() {

pool = mysql.createPool({// 创建连接池

...option,//option为数据库配置

waitForConnections: true, //当无连接池可用时，等待（true）还是抛错（false）

connectionLimit: 100, //连接数限制

queueLimit: 0 //最大连接等待数，0为不限制

})

pool.on('error', err => {

err.code === 'PROTOCOL\_CONNECTION\_LOST' && setTimeout(repool, 2000)

})

app.all('\*', (\_,\_\_, next) => {

pool.getConnection( err => {err && setTimeout(repool, 2000) || next()})

})

}

# 第6章 个人管理app的测试

本次测试包括页面跳转测试、功能性测试、接口测试，测试方法采用黑盒测试、边界测试法。

6.1 页面跳转测试

页面跳转测试包括导航栏的跳转和不包含数据获取判定跳转的测试，涉及功能判定的页面跳转见相关功能模块测试。测试用例如表6-1所示。

表6-1 页面跳转测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 说明 | 预计输出 | 实际输出 | 结果 |
| 侧导航中点击“好友” |  | 跳转好友页 | 跳转好友页 | 通过 |
| 侧导航中点击“时间历程” |  | 跳转时间历程页 | 跳转时间历程页 | 通过 |
| 侧导航中点击“历史统计” | 仅计划页存在该导航 | 跳转计划图表页 | 跳转计划图表页 | 通过 |
| 侧导航中点击“成就勋章” |  | 跳转成就页 | 跳转成就页 | 通过 |
| 底部导航点击“待办” |  | 跳转待办页 | 跳转待办页 | 通过 |
| 底部导航点击“计划” |  | 跳转计划页 | 跳转计划页 | 通过 |
| 底部导航点击“备忘录” |  | 跳转计划备忘录页 | 跳转计划备忘录页 | 通过 |
| 添加/修改计划页点击返回 |  | 跳转计划页 | 跳转计划页 | 通过 |
| 添加/修改备忘页点击返回 |  | 跳转计划备忘录页 | 跳转计划备忘录页 | 通过 |

6.2 功能测试

6.2.1用户模块

用户模块的功能测试包括各种可能情景下的登录注册功能和用户信息相关功能的测试。测试用例如表6-1所示。

表6-2 用户模块功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 说明 | 预计输出 | 实际输出 | 结果 |
| 点击注册 | 首次注册出现该页面 | 显示输入框 | 显示输入框 | 通过 |
| 不输入用户名 |  | 提示用户名为空 | 提示用户名为空 | 通过 |
| 输入已存在用户名 |  | 提示该用户名已存在 | 提示该用户名已存在 | 通过 |
| 输入超过50字符用户名 |  | 提示用户名长度应在50字符以内 | 提示用户名长度应在50字符以内 | 通过 |
| 输入1-50字符不存在用户名 |  | 注册成功，跳转主页，个人信息可查看 | 注册成功，跳转主页，个人信息可查看 | 通过 |
| 个人信息页点击复制id |  | id复制至剪贴板 | id复制至剪贴板 | 通过 |

6.2.2 好友模块

好友模块功能测试包括添加和删除好友功能的测试，测试用例如表6-2所示。

表6-3 好友模块功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 说明 | 预计输出 | 实际输出 | 结果 |
| 不输入好友id |  | 提示好友id为空 | 提示好友id为空 | 通过 |
| 输入不存在id |  | 提示该id不存在 | 提示该id不存在 | 通过 |
| 输入好友id |  | 提示该好友已存在 | 提示该好友已存在 | 通过 |
| 输入未添加用户id |  | 添加成功，展示好友信息 | 添加成功，展示好友信息 | 通过 |
| 删除好友 |  | 删除成功，移除个人信息 | 删除成功，移除个人信息 | 通过 |
| 删除用户自己 |  | 不可删除 | 不可删除 | 通过 |

6.2.3 备忘录模块

备忘录模块功能测试包括对于不同日期备忘增删改查的测试。测试用例如表6-4所示。

表6-4 备忘录模块功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 说明 | 预计输出 | 实际输出 | 结果 |
| 点击添加按钮 |  | 跳转备忘详情页，日期为选取日期，时间为当时时间 | 跳转备忘详情页，日期为选取日期，时间为当时时间 | 通过 |
| 点击修改按钮 | 备忘已过期 | 跳转备忘详情页，所有内容为待修改备忘信息 | 跳转备忘详情页，所有内容为待修改备忘信息 | 通过 |
| 点击修改按钮 | 备忘未过期 | 跳转备忘详情页，所有内容为待修改备忘信息 | 跳转备忘详情页，所有内容为待修改备忘信息 | 通过 |
| 详情页不输入主题 |  | 提示主题为空 | 提示主题为空 | 通过 |
| 主题长度超过100字符 |  | 提示主题长度不得超过100字符 | 提示主题长度不得超过100字符 | 通过 |
| 内容长度超过400字符 |  | 提示内容长度不得超过400字符 | 提示内容长度不得超过400字符 | 通过 |
| 地点长度超过100字符 |  | 提示地点长度不得超过100字符 | 提示地点长度不得超过100字符 | 通过 |
| 备忘详情页输入合法，点击完成 |  | 跳转备忘页，更新数据 | 跳转备忘页，更新数据 | 通过 |
| 修改日期 |  | 日历信息和备忘信息更新 | 日历信息和备忘信息更新 | 通过 |
| 删除备忘 | 备忘已过期 | 备忘删除，页面更新 | 备忘删除，页面更新 | 通过 |
| 删除备忘 | 备忘未过期 | 备忘删除，页面更新 | 备忘删除，页面更新 | 通过 |

6.2.4 每日待办模块

每日待办模块功能测试包括对于不同情境下待办任务的增删改查、勾选或取消勾选、以及重复当日待办功能的测试。测试用例如表6-5所示。

表6-5 每日待办模块功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 说明 | 预计输出 | 实际输出 | 结果 |
| 不输入添加内容，选择完成 |  | 提示添加内容为空 | 提示添加内容为空 | 通过 |
| 输入重复内容 |  | 添加成功，数据更新 | 添加成功，数据更新 | 通过 |
| 添加长度超过100字符内容 |  | 提示内容长度不得超过100字符 | 提示内容长度不得超过100字符 | 通过 |
| 输入不重复内容 |  | 添加成功，数据更新 | 添加成功，数据更新 | 通过 |
| 删除历史待办 |  | 无法删除 | 无法删除 | 通过 |
| 删除今日待办 |  | 删除成功 | 删除成功 | 通过 |
| 修改历史待办 |  | 无法修改 | 无法修改 | 通过 |
| 修改今日待办 |  | 修改成功 | 修改成功 | 通过 |
| 完成/取消完成历史待办 |  | 无法修改 | 无法修改 | 通过 |
| 完成/取消完成今日待办 |  | 修改成功 | 修改成功 | 通过 |
| 点击重复按钮 | 日期为今日 | 重复今日待办至明日 | 重复今日待办至明日 | 通过 |
| 点击重复按钮 | 日期为以前 | 重复该日待办至明日 | 重复该日待办至明日 | 通过 |
| 修改日期为以前 |  | 展示历史待办 | 展示历史待办 | 通过 |
| 修改日期为今日 |  | 展示今日待办 | 展示今日待办 | 通过 |
| 修改日期为未来 |  | 不展示待办 | 不展示待办 | 通过 |
| 修改日期 |  | 日历信息和备忘信息更新 | 日历信息和备忘信息更新 | 通过 |

6.2.5 计划模块

计划模块功能测试包括对于计划的增删改查、进行、完成及查询历史统计操作的测试。测试用例如表6-6所示。

表6-6 计划模块功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 说明 | 预计输出 | 实际输出 | 结果 |
| 点击添加按钮 |  | 跳转计划详情页，所有内容为空 | 跳转计划详情页，所有内容为空 | 通过 |
| 点击修改按钮 |  | 跳转计划详情页，所有内容为待修改计划信息 | 跳转计划详情页，所有内容为待修改计划信息 | 通过 |
| 详情页不输入主题 |  | 提示主题为空 | 提示主题为空 | 通过 |
| 主题长度超过100字符 |  | 提示主题长度不得超过100字符 | 提示主题长度不得超过100字符 | 通过 |
| 内容长度超过400字符 |  | 提示内容长度不得超过400字符 | 提示内容长度不得超过400字符 | 通过 |
| 不选择紧急度 |  | 提示未选择紧急度 | 提示未选择紧急度 | 通过 |
| 详情页输入合法 |  | 跳转计划页，更新数据 | 跳转计划页，更新数据 | 通过 |
| 删除计划 | 该计划非进行中 | 计划删除，页面更新 | 计划删除，页面更新 | 通过 |
| 删除计划 | 该计划在进行中 | 暂停计划存储数据，删除计划，页面更新 | 暂停计划存储数据，删除计划，页面更新 | 通过 |
| 完成计划 | 该计划非进行中 | 计划完成，页面更新 | 计划完成，页面更新 | 通过 |
| 完成计划 | 该计划在进行中 | 暂停计划存储数据，完成计划，页面更新 | 暂停计划存储数据，完成计划，页面更新 | 通过 |
| 选取计划 | 无计划正在进行 | 选取计划至计时框 | 选取计划至计时框 | 通过 |
| 选取计划 | 有计划正在进行 | 暂停进行中的计划，存储数据，选取计划至计时框 | 暂停进行中的计划，存储数据，选取计划至计时框 | 通过 |
| 进行计划 | 未选取计划 | 提示未选择计划 | 提示未选择计划 | 通过 |
| 进行计划 | 已选取计划 | 计划进行 | 计划进行 | 通过 |
| 暂停计划 |  | 计时暂停，数据更新 | 计时暂停，数据更新 | 通过 |

6.2.6 成就模块

成就模块测试包括通过各种情景下完成当日所有待办获得成就、和进行计划获得成就、查看成就的测试，测试用例如表6-7所示。

表6-7 成就模块功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 说明 | 预计输出 | 实际输出 | 结果 |
| 首次完成当日所有待办 | 至今日待办日期达到成就标准 | 获得新成就 | 获得新成就 | 通过 |
| 首次完成当日所有待办 | 至今日待办日期未达到成就标准 | 不获得成就 | 不获得成就 | 通过 |
| 非首次完成当日所有待办 | 至今日待办日期达到成就标准 | 获得新成就 | 获得新成就 | 通过 |
| 完成待办后待办置空 |  | 不获得成就 | 不获得成就 | 通过 |
| 非首次完成当日所有待办 | 至今日待办日期未达到成就标准 | 不获得成就 | 不获得成就 | 通过 |
| 完成计划 | 进行计划时长达到成就标准 | 获得新成就 | 获得新成就 | 通过 |
| 完成计划 | 进行计划时长达未到成就标准 | 不获得成就 | 不获得成就 | 通过 |
| 查看成就 | 未获得成就 | 展示空白页 | 展示空白页 | 通过 |
| 查看成就 | 已获得成就 | 成就按序展示 | 成就按序展示 | 通过 |

6.2.7 时间历程模块

时间历程模块功能测试包括完成当日所有待办、获得成就、完成计划创建历程和查看历程的测试。测试用例如表6-8所示。

表6-8 时间历程模块功能测试用例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试用例 | 说明 | 预计输出 | 实际输出 | 结果 |
| 首次完成当日所有待办 |  | 创建历程 | 创建历程 | 通过 |
| 完成待办后待办置空 |  | 不更新历程 | 不更新历程 | 通过 |
| 非首次完成当日所有待办 |  | 更新历程 | 更新历程 | 通过 |
| 完成计划 |  | 创建历程 | 创建历程 | 通过 |
| 获得成就 |  | 创建历程 | 创建历程 | 通过 |
| 查看历程 | 历程为空 | 展示空白页 | 展示空白页 | 通过 |
| 查看历程 | 历程不为空 | 时间历程按序展示 | 时间历程按序展示 | 通过 |

6.3 接口测试

接口测试使用postman进行，测试用例与表4-7至表4-14基本一致，在此不再赘述。测试用例均已通过。

结　论

本文对于游戏化的三大元素、个人管理的核心机制和现状误区进行了研究、分析和总结，并依此设计、实现了一个游戏化的个人管理android app。系统前端使用用Java进行开发，使用多个开源项目增加项目易用性。使用NodeJs的express框架进行后端开发，提高了系统在计算复杂度较小时面对高并发的稳定性。数据库采用mysql，已部署至服务器，使用pm2进行进程管理，可直接使用app进行相关数据传输。经测试系统已可以稳定运行。

由于时间问题许多预想的功能，如多人计划、以及子计划等功能尚未实现。游戏化的部分也相对浮于表面，对于将用户的外部动机转化为内部动机的激励尚显不足。不论是项目开发，还是游戏化的核心，都仍需继续探究。

参考文献

[1] Deterding S, Dixon D, Khaled R, et al. From game design elements to gamefulness: defining gamification[C]. Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments. ACM, 2011. 9-15.

[2] 张志杰,黄希庭,凤四海,邓麟. 青少年时间管理倾向相关因素的研究[J].心理科学,2001(06):649-653+764.

[3] JuhoHamari, JonnaKoivisto, HarriSarsa. Does Gamification Work? — A Literature Review of Empirical Studies on Gamification.2014 47th Hawaii International Conference on System Science[C]. 2014. 3025~3034.

[4] 鲍雪莹,赵宇翔. 游戏化学习的研究进展及展望[J].电化教育研究,2015,36(08):45-52.

[5] 孙凯,左美云,孔栋. 游戏化有用吗——教育领域实证研究的比较分析[J].电化教育研究,2016,37(11):85-92.

[6] 马张鹤. 别把时间给了浪费[M].北京:经济管理出版社,2004.

[7] 贾慢慢. 大学生时间管理现状调查研究[D].河南大学,2016.

[8] 边玉芳. 学习自我效能感量表的编制与应用[D].华东师范大学,2003.

[9] 黄希庭,张志杰. 论个人的时间管理倾向[J].心理科学,2001(05):516-518+636.

[10] 黄希庭,张志杰. 青少年时间管理倾向量表的编制[J].心理学报,2001(04):338-343.

[11] The 4 Keys to Fun. [EB/OL].http://www.xeodesign.com/the-4-keys-to-fun,2015-02-10.

[12]肖亦奇. 基于玩家动机类型的游戏化概念设计方法[J].设计艺术研究,2020,10(02):54-60.

[13] Motivation. [DB/OL]. http://en.wikipedia.org/wiki/Motivation, 2015-02-10.

[14] 丹·亨特，凯文·伟巴赫． 游戏化思维: 改变未来商业的新力量[M]. 杭州: 浙江人民出版社.2014.70-79.

[15] 李青,李莹莹. 移动学习应用中积分激励机制设计[J].北京邮电大学学报(社会科学版),2018,20(03):104-112.

[16] 刘俊,祝智庭. 游戏化——让乐趣促进学习成为教育技术的新追求[J].电化教育研究,2015,36(10):69-76+91.

[17] Lepper, M.P., Greene, D., Nisbett, R. E.. Undermining Children's Intrinsic Interest with Extrinsic Reward: A Test of the "Overjustification" Hypothesis[J]. Journal of Personality and Social Psychology, 1973,28 (1): 129~137

[18] Glanz, Lewis，Rimers. 健康行为与健康教育:理论、研究与实践[M]. 加利福尼亚州:约瑟-巴斯, 1990. 17。

致　谢

值此论文完成之际，首先要感谢我的母校北京理工大学。四年的时光扩展了我的视野，让我见识到了更广阔的世界。感谢我的毕设导师赵丰年老师。从开题开始，文献翻译、参考资料、系统设计，赵老师一直给了我许多指导性的意见和建议，让我对于课题有了全面的了解，也让我在设计时没有囿于自己单纯的学生视角做出过于片面的设计。此外，赵老师还给了我很多鼓励，在毕设之外，对于我的职业规划也给出了作为师长的前瞻性建议。

其次，感谢众多在游戏化、个人管理、安卓应用开发上孜孜以求的前辈们。各种理论研究、实际调研、开源项目，才让我能够在自身知识微薄的情况下产生更深刻的认知，让我在巨人的肩膀上前行而无需从头开始，在无数个错误的中看到前方的光。

最后，感谢在这段特殊的时期为了我们的安全奋战前线的医护人员，让我能有安稳的环境进行研究，也感谢小伙伴们一直以来给我的鼓励，和陪我共同度过的那些敲键盘的不眠之夜。一愿世清平，二愿身强健。三愿临老头,数与君相见。