**摘要**

在当今信息化社会，智能手机已成为人们生活中不可或缺的一部分。随着移动技术的快速发展，手机应用的种类和功能日益丰富，极大地便利了人们的生活。本大作业旨在开发一款集登录、任务管理、学习计时、休闲娱乐和个人信息管理于一体的多功能应用，以满足用户在日常生活和学习中的多样化需求。

随着移动互联网的普及，人们越来越依赖手机App来管理自己的生活和学习。然而，目前市场上的应用种类繁多，但功能单一，往往只能满足用户的某一特定需求。因此，开发一款集多种功能于一体的综合性应用，具有广阔的市场前景和实用价值。

本应用旨在为用户提供一种全新的生活方式和学习体验。通过登录功能，用户可以注册自己的账号，并享受个性化的服务；任务清单功能可以帮助用户更好地管理自己的任务和时间；番茄学习钟功能则可以提高用户的学习效率；休息区功能可以让用户在紧张的学习或工作之余得到放松；个人中心功能则可以让用户随时查看和修改自己的个人信息。

**summary**

In today's information society, smartphones have become an indispensable part of people's lives. With the rapid development of mobile technology, the types and functions of mobile phone applications are becoming more and more abundant, which greatly facilitates people's lives. The purpose of this project is to develop a multi-functional application that integrates login, task management, learning timing, leisure and entertainment, and personal information management to meet the diverse needs of users in their daily life and study.

With the popularization of mobile Internet, people are increasingly relying on mobile apps to manage their lives and studies. However, there are many kinds of applications on the market today, but they have a single function and often can only meet a specific need of the user. Therefore, the development of a comprehensive application that integrates multiple functions has broad market prospects and practical value.

This app is designed to provide users with a new way of living and learning experience. Through the login function, users can register their own accounts and enjoy personalized services; The task list feature can help users better manage their tasks and time; The Pomodoro learning clock function can improve the user's learning efficiency; The rest area function allows users to relax after intense study or work; The personal center function allows users to view and modify their personal information at any time.

目录

[第1章 绪论 2](#_Toc29018)

[1.1 课题研究的背景及意义 2](#_Toc11831)

[1.2 国内外研究现状 3](#_Toc19564)

[1.2.1国外研究现状 3](#_Toc25114)

[1.2.2国内研究现状 3](#_Toc17436)

[1.3 研究目的及内容 4](#_Toc6437)

[第2章 开发环境及相关技术 7](#_Toc15440)

[2.1 开发环境 7](#_Toc29100)

[2.1.1 Android Studio 7](#_Toc9196)

[2.1.2 SQLite数据库 8](#_Toc18786)

[第3章 系统分析 10](#_Toc19149)

[3.1 系统可行性分析 10](#_Toc15832)

[3.1.1 技术可行性 10](#_Toc22718)

[3.1.2 经济可行性 10](#_Toc24103)

[3.1.3 法律可行性 11](#_Toc29962)

[第4章 功能概述 12](#_Toc32443)

[4.1登录功能 12](#_Toc27534)

[4.2用户注册功能实现： 13](#_Toc26770)

[4.3任务清单功能 16](#_Toc17279)

[4.4番茄学习钟功能 18](#_Toc10103)

[4.5休息区功能 24](#_Toc7491)

[4.6个人中心功能 27](#_Toc5879)

[第5章 技术实现与挑战 32](#_Toc6144)

[第6章 总结与展望 33](#_Toc3112)

[参考文献 34](#_Toc17625)

# 第1章 绪论

## 1.1 课题研究的背景及意义

在当今信息化社会，智能手机已成为人们生活中不可或缺的一部分。随着移动技术的快速发展，手机应用的种类和功能日益丰富，极大地便利了人们的生活。本大作业旨在开发一款集登录、任务管理、学习计时、休闲娱乐和个人信息管理于一体的多功能应用，以满足用户在日常生活和学习中的多样化需求。

随着移动互联网的普及，人们越来越依赖手机App来管理自己的生活和学习。然而，目前市场上的应用种类繁多，但功能单一，往往只能满足用户的某一特定需求[1]。因此，开发一款集多种功能于一体的综合性应用，具有广阔的市场前景和实用价值。

理论意义：通过对番茄钟APP的研究，可以深入了解番茄工作法的原理和应用，为时间管理理论的研究提供新的视角和方法。同时，也可以为其他时间管理工具的开发和应用提供理论支持。

实践意义：番茄钟APP的研究可以为相关产品的设计和开发提供指导，帮助开发者更好地满足用户需求，提高产品的竞争力和市场占有率。此外，对于用户而言，了解番茄钟APP的使用方法和技巧，可以更好地管理自己的时间，提高学习和工作的效率。

本应用旨在为用户提供一种全新的生活方式和学习体验。通过登录功能，用户可以注册自己的账号，并享受个性化的服务；任务清单功能可以帮助用户更好地管理自己的任务和时间；番茄学习钟功能则可以提高用户的学习效率；休息区功能可以让用户在紧张的学习或工作之余得到放松；个人中心功能则可以让用户随时查看和修改自己的个人信息。

## 1.2 国内外研究现状

### 1.2.1国外研究现状

在国外，对于番茄钟APP的研究主要集中在如何提高其有效性和用户体验上。一些学者通过实证研究，探讨了番茄工作法在不同人群中的适用性，并分析了其对提高工作效率和减少拖延行为的影响。此外，还有一些研究关注于如何根据用户的个性化需求，优化番茄钟APP的功能设计。

Forest：这是一款备受欢迎的番茄钟APP，它通过让用户“种树”的方式，鼓励用户保持专注。在设定的时间内，如果用户不使用手机，树木就会茁壮成长；如果离开APP，树木就会枯萎。这种有趣的方式让用户更愿意保持专注[2]。

Pomodoro Timer：这是一款经典的番茄钟APP，它严格按照番茄工作法的原则设计。用户可以自定义“番茄”的时间和休息时间，并可以查看自己的工作统计数据。此外，该APP还支持多平台同步，方便用户在不同设备上使用。

在技术实现方面，国外研究者不断探索新的技术手段，如机器学习、大数据分析等，以实现对用户行为的精准预测和个性化推荐。例如，一些番茄钟APP已经能够根据用户的历史数据，智能调整“番茄”的时间和休息时间，以达到最佳的工作效果。

### 1.2.2国内研究现状

在国内，番茄钟APP的研究起步较晚，但发展迅速。近年来，随着移动互联网的普及和人们对时间管理需求的增加，越来越多的研究者开始关注番茄钟APP的研究。

在理论研究方面，国内学者主要借鉴国外的研究成果，结合中国的文化背景和用户需求，探讨番茄工作法在中国的适用性。一些学者还通过问卷调查、访谈等方式，收集用户对番茄钟APP的使用反馈和建议，为产品的优化和改进提供依据。

番茄ToDo：这是一款功能丰富的番茄钟APP，它结合了任务清单、数据统计等功能。用户可以将待办事项添加到任务清单中，并设置提醒时间和优先级。在设定的时间内，APP会锁定屏幕并阻止其他应用的干扰，帮助用户保持专注。此外，该APP还提供了丰富的统计数据和分析功能，让用户可以清晰地了解自己的工作效率和专注度。

番茄工作法的基本原理是将工作时间划分为多个番茄钟，一个番茄钟由 25  
分钟工作时间和 5 分钟休息时间组成①。番茄 ToDo 软件将其定义为“一种简单易行的时间管理方法，是由弗朗西斯科在 1992 年创立的一种微观时间管理方法”。在番茄 ToDo 软件中，番茄工作法的运用机制为设置一个待办事项，并且系统默认按照 25 分钟进行运行，自动进入学习状态，中途不允许退出，如果退出会被软件警告，如果在番茄时间内做了与任务无关的事情，则时间将会被作废。当工作时间结束后，便进入了 5 分钟的休息时间。番茄 ToDo 在番茄工作法的基础上还添加了其他功能，比如：“数据统计”“白噪音”“周期计划”等来搭配使用番茄工作法。  
 番茄 ToDo 主打“界面简单”“番茄时间”“无广告”“学霸模式”等特点，  
特别是其简单的操作方法让用户感到便捷，其在使用方法说明中写到：“如果您  
不需要用到其他高级功能，这样的简单操作能让你用得更加方便”。由于软件下  
载免费的原因，因此，其通过开通 VIP 服务来增加软件的盈利能力。主要为季  
度卡，每季度 12 元，VIP 服务则增加了“无限次习惯养成设定”“无限次添加正向计时待办”“多种白噪音”功能等。从产品上线以来，番茄 ToDo 一直保持着较高频率的功能更新。第一阶段是核心功能的探索时期，从 2017 年 9 月 1.6 版本到 2018 年 2 月 3.0 版本，主要在探索其核心功能，其中包括学霸模式的建立，加入放弃理由的分布，以及将计时模式修改为三种计时模式“番茄工作法”“正计时”“倒计时”，加入了习惯养成功能，这些功能主要都聚焦于待办页面上，这也是用户的主要使用功能。第二阶段是产品的优化期，从2018年2月的3.1版本到2018年8月的5.5版本，优化了学霸模式中的强制提醒模式，在计时模式中添加了“设置背景图”“打卡分享”等功能。第三阶段主要是社交功能的扩展，主要是加入了“自习室”功能，用户可以通过自习室查看伙伴的“专注时长”。第四阶段是至今，基本内容建设已经完善，产品功能不断优化，主要是增加了“自习室”加入密码、“自习室”排名、“许愿墙”等。

滴答清单：虽然滴答清单不仅仅是一个番茄钟APP，但它也集成了番茄工作法的功能。用户可以在清单中创建任务，并为其设置番茄时间。在任务进行时，APP会启动倒计时并阻止其他应用的干扰。同时，滴答清单还提供了丰富的协作和分享功能，方便团队成员之间的沟通和协作。

在实践应用方面，国内的一些互联网公司积极推出自己的番茄钟APP产品，并不断进行迭代更新。这些产品不仅具有基本的时间管理功能，还融入了社交、游戏化等元素，以吸引更多的用户。同时，一些企业也开始将番茄工作法引入到员工的时间管理中，以提高工作效率和团队协作效果。

## 1.3 研究目的及内容

本文的总体目的是深入探究番茄钟APP的研究现状、发展趋势以及应用效果，为相关领域的研究和开发提供理论支持和实践指导。

梳理番茄钟APP的发展历程和现状，分析其发展趋势和前景。深入研究番茄钟APP的功能特点和设计原理，探讨其如何实现时间管理和提高专注力的目标。分析番茄钟APP的用户群体和使用场景，了解不同用户的需求和偏好。评估番茄钟APP的应用效果，探讨其在学习、工作等领域中的实际作用和价值。

番茄钟APP的起源可以追溯到20世纪80年代，由意大利人弗朗西斯科·西里洛提出的番茄工作法演变而来。随着移动互联网技术的发展和智能手机的普及，番茄钟APP逐渐成为了人们管理时间的重要工具之一。目前，市面上已经涌现出了众多优秀的番茄钟APP产品，如Forest、Pomodoro Timer、番茄ToDo等。

随着技术的不断进步和用户需求的不断变化[4]，番茄钟APP也在不断发展和创新。一方面，越来越多的产品开始注重个性化和智能化设计，通过引入机器学习、大数据分析等先进技术，实现对用户行为的精准预测和个性化推荐；另一方面，一些产品还开始探索与其他应用的融合和协作，如与任务清单、日程管理等应用的结合，为用户提供更加全面和便捷的时间管理服务。

番茄钟APP的功能特点主要体现在以下几个方面：一是支持自定义“番茄”时间和休息时间；二是提供多种提醒方式和白噪音背景音；三是支持数据统计和分析功能；四是具备跨平台同步和多设备支持功能。此外，一些产品还提供了社交分享、团队协作等特色功能，以满足不同用户的需求。

番茄钟APP的设计原理主要基于番茄工作法。该方法通过设定明确的工作时间和休息时间，帮助用户集中注意力、提高工作效率和专注力。在具体实现上，番茄钟APP采用了倒计时、提醒、锁屏等技术手段，确保用户在设定的时间内专注于当前任务。同时，通过提供多种提醒方式和白噪音背景音等辅助手段，帮助用户更好地进入工作状态并保持专注。

用户群体：番茄钟APP的用户群体广泛，包括学生、上班族、自由职业者等各类人群。其中，学生和上班族是使用番茄钟APP的主要群体。学生群体通过使用番茄钟APP可以更好地管理自己的学习时间，提高学习效率和成绩；上班族则可以通过使用番茄钟APP来管理自己的工作任务和进度，提高工作效率和生产力。

使用场景：番茄钟APP的使用场景多样，可以应用于学习、工作、阅读等多种场景。在学习场景中，用户可以将学习任务添加到任务清单中，并通过设置“番茄”时间和休息时间来完成学习任务；在工作场景中，用户可以将工作任务划分为多个小任务，并为每个小任务设置“番茄”时间，以确保工作的高效完成；在阅读场景中，用户可以通过设置“番茄”时间来提高阅读效率和专注力。

对于番茄钟APP的应用效果评估，可以采用问卷调查、访谈、实验等多种方法。其中，问卷调查可以收集大量用户的使用反馈和意见；访谈可以深入了解用户的需求和偏好；实验则可以通过对比使用前后的工作效率、专注力等指标来评估应用效果。根据已有的研究和实践经验，番茄钟APP在提高用户工作效率和专注力方面具有良好的应用效果。

# 第2章 开发环境及相关技术

本章节主旨在概述开发本安卓APP所采用的环境，这一系统环境对于开发团队来说是一个强大的主力。使用得当时，它可以明显提升设计人员和开发工程师的工作效率。我们的开发环境包括了android studio、安卓13虚拟机、调试设备、硬件环境的管理工具。

此外，为了确保系统的可行性，我们还从技术和经济的角度不断进行了综合评估。通过以上工具和技术的综合运用，我们能够构建一个既高效又稳定的开发环境，为系统的顺利开发提供了坚实的基础。

2.1 开发环境

### 2.1.1 Android Studio

Android Studio是Google推出的Android应用开发集成开发环境（IDE），它基于IntelliJ IDEA，并专门针对Android开发进行了优化。Android Studio提供了丰富的开发工具和功能，如代码编辑、调试、测试、版本控制等，帮助开发者更加高效地进行Android应用开发。

主要特点：

（1）强大的代码编辑功能：Android Studio支持多种编程语言的代码编辑[5]，包括Java、Kotlin等，并提供了代码高亮、自动补全、代码重构等强大的编辑功能，帮助开发者快速编写高质量的代码。

（2）丰富的调试和测试工具：Android Studio提供了强大的调试和测试工具，如模拟器、调试器、性能测试工具等，帮助开发者在开发过程中及时发现和修复问题，提高应用的质量和性能。

（3）便捷的版本控制：Android Studio支持多种版本控制系统，如Git、SVN等，并提供了可视化的版本控制界面，帮助开发者更好地管理代码版本和协作开发。

（4）丰富的插件和扩展：Android Studio支持丰富的插件和扩展，开发者可以根据自己的需求安装和使用各种插件和扩展，进一步提高开发效率和便利性。

### 2.1.2 SQLite数据库

SQLite是一个轻量级的关系型数据库管理系统，它占用资源少、处理速度快、可移植性强，并且支持标准SQL语法，因此在移动应用开发中得到了广泛应用。Android Studio自带了SQLite数据库，开发者可以在Android Studio中直接创建、管理和操作SQLite数据库。

数据库：SQLite数据库是一个文件，用于存储和管理数据。在Android应用中，SQLite数据库文件通常存储在应用的私有目录下（/data/data/your.package.name/databases/）。

表：表是数据库中的基本存储单元，用于存储数据。在SQLite中，每个表都有一个唯一的名称，并包含若干个字段（列），用于存储不同类型的数据。

字段：字段是表中的一列，用于存储某种类型的数据。每个字段都有一个唯一的名称和数据类型，用于定义该字段可以存储的数据类型和范围。

记录：记录是表中的一行，用于存储一组相关的数据。每个记录包含表中所有字段的值，用于描述一个具体的数据实体。

SQLite数据库的使用方法

（1）创建数据库和表：在Android Studio中，开发者可以通过编写SQL语句或使用SQLiteOpenHelper类来创建数据库和表。SQLiteOpenHelper是一个辅助类，它提供了创建和打开数据库的方法，并可以在数据库创建时执行一些初始化操作（如创建表、插入初始数据等）。

（2）插入数据：在SQLite中，可以使用INSERT INTO语句向表中插入数据。开发者可以编写SQL语句或使用SQLiteDatabase类的insert()方法将数据插入到表中。

（3）查询数据：在SQLite中，可以使用SELECT语句从表中查询数据。开发者可以编写SQL语句或使用SQLiteDatabase类的query()方法来执行查询操作，并通过Cursor对象获取查询结果。

（4）更新和删除数据：在SQLite中，可以使用UPDATE语句和DELETE语句来更新和删除表中的数据。开发者可以编写SQL语句或使用SQLiteDatabase类的update()和delete()方法来执行更新和删除操作。

SQLite数据库的优势

（1）轻量级：SQLite数据库是一个轻量级的数据库管理系统，它占用资源少、处理速度快，非常适合在移动设备上使用。

（2）可移植性强：SQLite数据库是一个文件型数据库，它不需要独立的服务器进程或操作系统级别的权限，因此具有很好的可移植性。开发者可以将SQLite数据库文件与应用程序一起打包发布，并在不同的设备和操作系统上运行。

（3）支持标准SQL语法：SQLite支持标准的SQL语法，因此开发者可以使用熟悉的SQL语句来操作SQLite数据库。这使得开发者可以更加高效地编写和管理数据库代码。

（4）易于集成和使用：Android Studio自带了SQLite数据库，开发者可以直接在Android Studio中创建、管理和操作SQLite数据库。这使得开发者可以更加便捷地使用SQLite数据库来存储和管理应用中的数据。

第3章 系统分析

## 3.1 系统可行性分析

### 3.1.1 技术可行性

技术可行性主要评估的是当前的技术条件是否支持番茄钟APP的开发和运行。对于番茄钟APP来说，技术可行性主要体现在以下几个方面：

开发平台与工具：Android Studio作为目前广泛使用的Android应用开发平台，其提供了丰富的开发工具和框架，能够支持番茄钟APP的开发。同时，Android Studio还支持多种编程语言，如Java和Kotlin，这些语言都具有成熟的社区和文档支持，便于开发者进行学习和使用。

数据库技术：SQLite作为Android Studio自带的轻量级数据库，具有体积小、速度快、稳定性高等优点，能够满足番茄钟APP对数据存储和查询的需求。此外，SQLite还支持标准的SQL语法，使得开发者可以方便地进行数据库操作。

界面设计与用户体验：Android Studio提供了丰富的界面设计工具和组件，如Layout Editor、ConstraintLayout等，可以帮助开发者设计出美观、易用的用户界面。同时，通过合理的界面布局和交互设计，可以提高用户的使用体验，使得番茄钟APP更加易于被用户接受和使用。

系统兼容性：由于Android系统的广泛应用[6]，番茄钟APP需要能够在不同版本的Android系统上稳定运行。通过合理的开发和测试工作，可以确保番茄钟APP在不同版本的Android系统上的兼容性和稳定性。

### 3.1.2 经济可行性

经济可行性主要评估的是开发番茄钟APP所需的成本和收益是否合理。对于番茄钟APP来说，经济可行性主要体现在以下几个方面：

开发成本：开发番茄钟APP需要投入一定的人力、物力和时间成本。但是，由于Android Studio和SQLite等技术都是免费或开源的，因此可以大大降低开发成本。同时，通过合理的项目管理和团队协作，可以进一步提高开发效率，降低开发成本。

运营成本：番茄钟APP的运营成本主要包括服务器维护、数据备份、用户支持等方面的费用[7]。但是，由于番茄钟APP是一个轻量级的应用，其所需的服务器资源和用户支持相对较少，因此运营成本也相对较低。

收益来源：番茄钟APP可以通过多种方式实现收益，如内购、广告等。通过提供优质的用户体验和丰富的功能，可以吸引更多的用户使用番茄钟APP，从而增加收益来源。

### 3.1.3 法律可行性

法律可行性主要评估的是开发番茄钟APP是否符合相关法律法规和政策要求。对于番茄钟APP来说，法律可行性主要体现在以下几个方面：

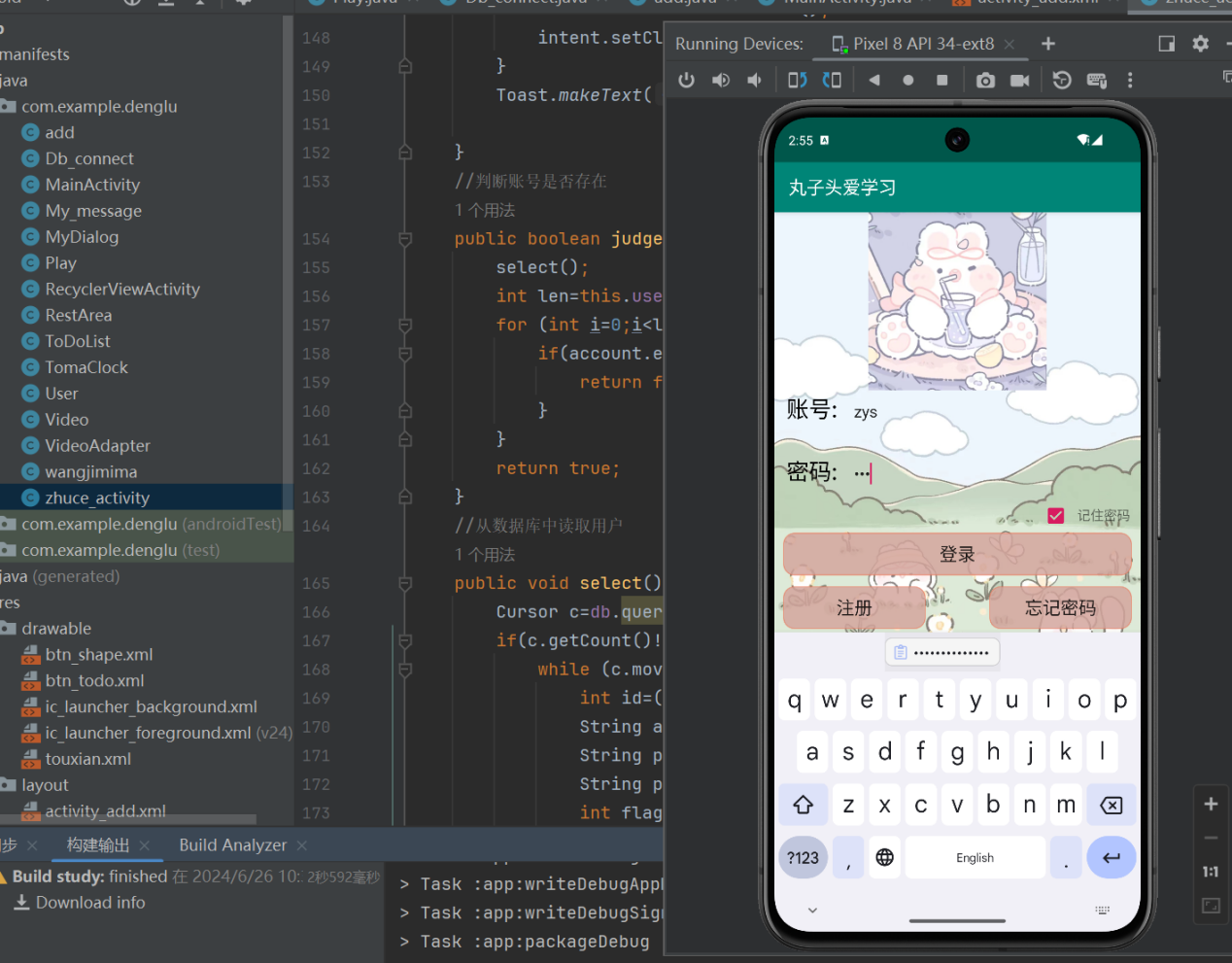
版权和知识产权：在开发番茄钟APP时，需要确保所使用的技术、工具、素材等都符合版权和知识产权的相关法律法规。同时，也需要尊重其他开发者的知识产权，避免侵犯他人的权益。

隐私保护：在收集和使用用户数据时，需要遵守相关的隐私保护法律法规，确保用户数据的安全和隐私[8]。同时，也需要向用户明确告知数据收集和使用的情况，并获得用户的授权和同意。

合规要求：在发布和应用番茄钟APP时，需要遵守相关的合规要求，如应用商店的审核标准、数据安全标准等[9]。通过确保应用的合规性，可以避免因违反相关法律法规而带来的风险。

# 第4章 功能概述

## 4.1登录功能

登录功能是本应用的基础功能之一。用户可以通过注册账号来创建自己的账户，并设置用户名和密码。在注册过程中，应用会检查用户名是否已被注册，以及手机号是否与账号名一致，以确保用户信息的唯一性和安全性。同时，应用还支持记住密码功能，方便用户下次登录时快速进入应用。

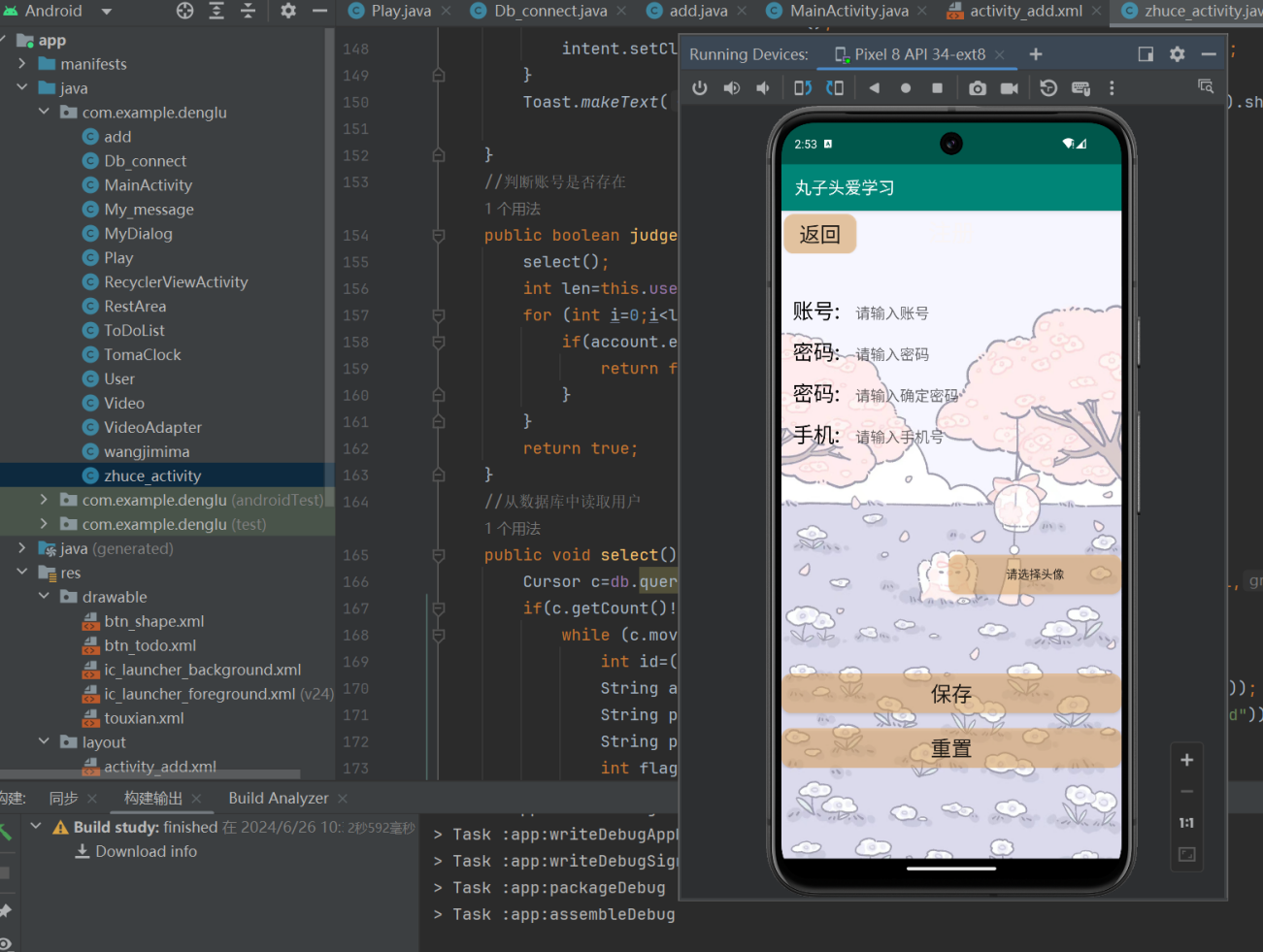
android studio虚拟机截图

登录用户判断：

public void zhaohui(View v){  
 String zhang=zhanghao.getText().toString();  
 String shou=shouji.getText().toString();  
 String message="";  
 if(zhang.length()!=0&&shou.length()!=0){  
 User user=new User(zhang,"",shou,0,0);  
 //判断是否存在  
 Boolean flag=this.users.contains(user);  
 if(flag==true){  
 String mima=this.getmima(zhang,shou);  
 Log.*i*("wangji", "Zhaohui"+mima);  
 message="用户密码为"+mima;  
 }else{  
 Log.*i*("wangji","不存在");  
 message="您输入的信息有误";  
 }  
 Toast.*makeText*(wangjimima.this,message,Toast.*LENGTH\_LONG*).show();  
 }else{  
 Toast.*makeText*(wangjimima.this,"账号和密码不能为空",Toast.*LENGTH\_LONG*).show();  
 }  
 }

## 4.2用户注册功能实现：

账号名、密码、手机号（后面可以改密码）的输入，选择用户头像

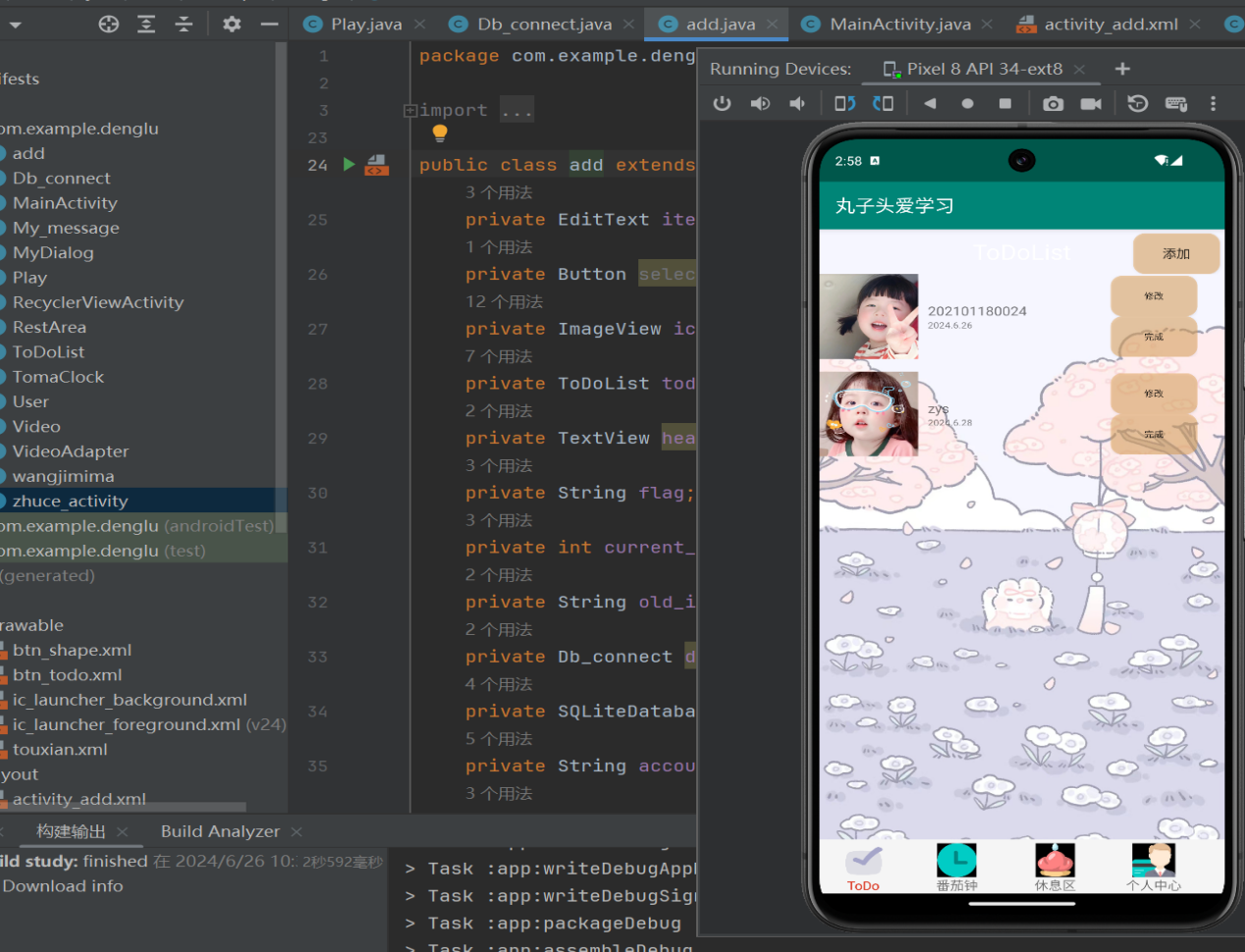


实现注册的代码：

if(this.content.length()==0){  
 this.content="注册成功，请返回登录";  
 //插入数据  
 insert(new User(zhang,mima\_o,ph,1,0));  
 //判断账号是否存在  
 public boolean judge\_account(String account){  
 select();  
 int len=this.users.size();  
 for (int i=0;i<len;i++){  
 if(account.equals(this.users.get(i).getZhanghao())){  
 return false;  
 }  
 }  
 return true;  
 }  
 //从数据库中读取用户  
 public void select(){  
 Cursor c=db.query("user",null,null,null,null,null,null);  
 if(c.getCount()!=0){  
 while (c.moveToNext()){  
 int id=(int)c.getInt(c.getColumnIndex("\_id"));  
 String account=c.getString(c.getColumnIndex("account"));  
 String password=c.getString(c.getColumnIndex("password"));  
 String phone=c.getString(c.getColumnIndex("phone"));  
 int flag=c.getInt(c.getColumnIndex("flag"));  
 this.users.add(new User(account,password,phone,id,flag));//保存  
 }  
 }  
 }  
 //插入数据  
 public void insert(User user){  
 ContentValues values=new ContentValues();  
 values.put("account",user.getZhanghao());  
 values.put("password",user.getMima());  
 values.put("phone",user.getPhone());  
 values.put("flag",0);//1表示记住密码，0表示不记住密码  
 values.put("icon",this.current\_icon);  
 long result=db.insert("user",null,values);  
 System.out.println("插入数据成功,当前第"+result+"条数据");

## 4.3任务清单功能

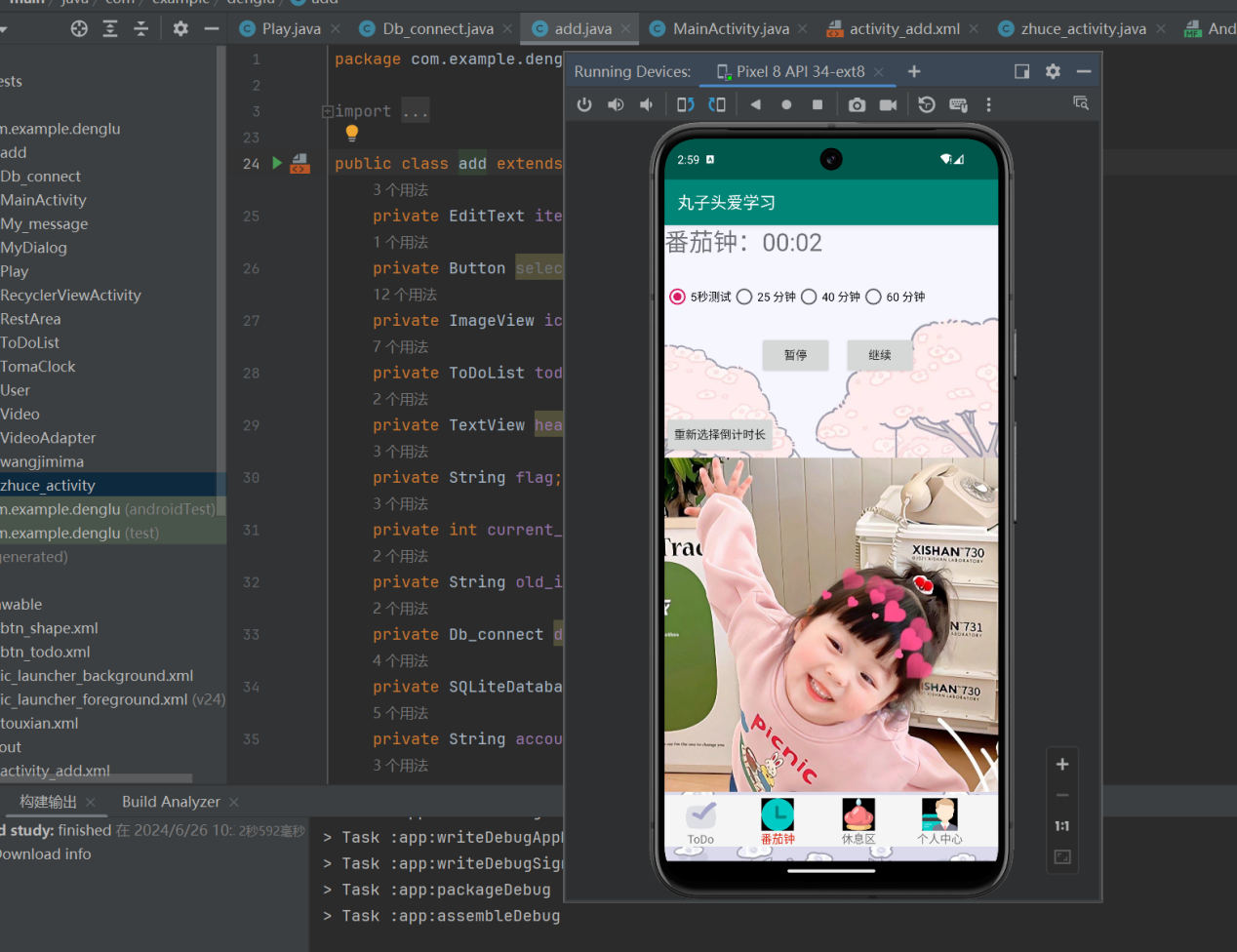
任务清单功能可以帮助用户更好地管理自己的任务和时间。用户可以在应用中添加、删除任务，并设置任务的完成时间和封面。通过任务清单，用户可以清晰地了解自己的任务进度和完成情况，从而更好地规划自己的时间和资源。



实现代码：  
public class add extends AppCompatActivity {  
 private EditText item\_name,deadline;  
 private Button select\_icon,save,back;  
 private ImageView icon;  
 private ToDoList todolist;  
 private TextView head;  
 private String flag;  
 private int current\_icon;//当前图片  
 private String old\_item;//老名字  
 private Db\_connect db\_connect;  
 private SQLiteDatabase db;  
 private String account;//账号  
 private ArrayList<ToDoList>todolists=new ArrayList<ToDoList>();  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_add);  
 item\_name=(EditText)findViewById(R.id.item\_name);  
 deadline=(EditText)findViewById(R.id.deadline);  
 select\_icon=(Button)findViewById(R.id.select\_icon);  
 save=(Button)findViewById(R.id.save);  
 back=(Button)findViewById(R.id.back);  
 icon=(ImageView)findViewById(R.id.icon);  
 head=(TextView)findViewById(R.id.head);  
 create\_db();//初始化数据库  
 Intent intent=new Intent();  
 this.todolist=(ToDoList) getIntent().getSerializableExtra("todolist");  
 this.flag=(String)getIntent().getStringExtra("flag");  
 this.account=getIntent().getStringExtra("name");  
 Log.i("account",this.account);  
 if(this.flag.equals("1")){  
 item\_name.setText(this.todolist.getItem\_name());  
 this.old\_item=this.todolist.getItem\_name();  
 this.current\_icon=this.todolist.getIcon();  
 deadline.setText(this.todolist.getDeadline());  
 icon.setImageResource(this.todolist.getIcon());  
 head.setText("编辑任务项");  
 }  
 }

## 4.4番茄学习钟功能

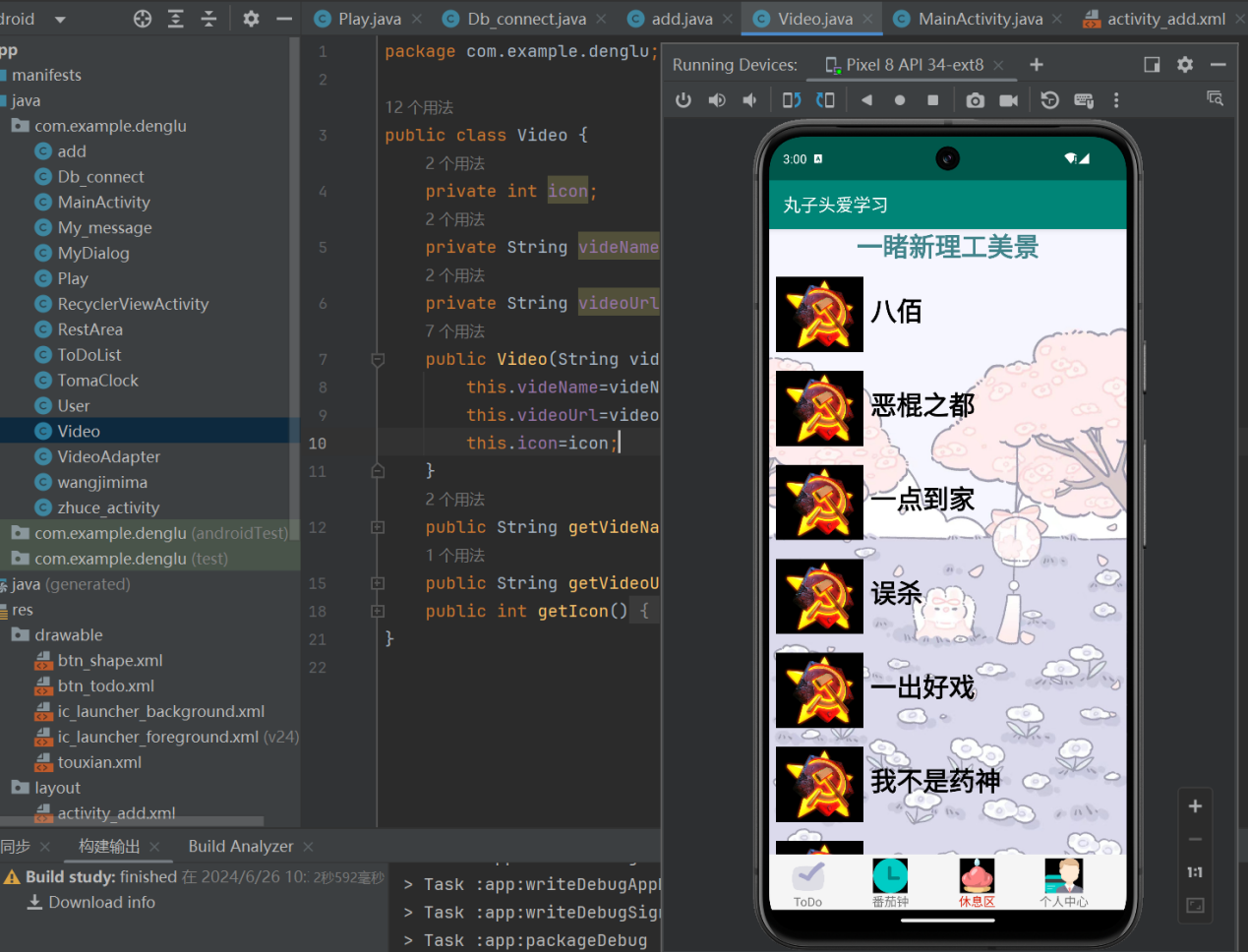
番茄学习钟功能是本应用的一大特色。它采用了番茄工作法的原理，通过设定25分钟的工作时间和5分钟的休息时间，帮助用户提高学习效率。在计时过程中，应用会以弹窗形式提醒用户时间到，并在完成任务后给出统计报告，让用户了解自己的学习效率和进度。此外，应用还支持设置不同时长的倒计时，如40分钟、60分钟等，以适应不同用户的需求。

功能实现

package com.example.denglu;  
import android.content.Intent;  
import android.os.Bundle;  
import android.os.CountDownTimer;  
import android.view.View;  
import android.widget.Button;  
import android.widget.RadioGroup;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
import java.util.Locale;  
public class TomaClock extends AppCompatActivity {  
 private CountDownTimer countDownTimer;  
 private String timerName = "番茄钟";  
 private boolean isCountingDown = false; // 是否正在倒计时中  
 private long remainingMillis; // 剩余毫秒数  
 private String username;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.tomaclock);  
 // 找到倒计时长度选择的 RadioGroup，添加监听器  
 final RadioGroup radioGroup = findViewById(R.id.radio\_group\_duration);  
 radioGroup.setOnCheckedChangeListener(new RadioGroup.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(RadioGroup group, int checkedId) {  
 // 如果计时器当前正在运行，则先暂停  
 if (isCountingDown) {  
 pauseCountdown();  
 }  
 switch (checkedId) {  
 case R.id.radio\_5:  
 remainingMillis = 5\* 1000;  
 break;  
 case R.id.radio\_25:  
 remainingMillis = 25 \* 60 \* 1000;  
 break;  
 case R.id.radio\_40:  
 remainingMillis = 40 \* 60 \* 1000;  
 break;  
 case R.id.radio\_60:  
 remainingMillis = 60 \* 60 \* 1000;  
 break;  
 }  
 // 更新显示和按钮状态  
 isCountingDown = false;  
 TextView textView = findViewById(R.id.text\_countdown);  
 textView.setText(timerName + "：" + formatMillis(remainingMillis));  
 showStartButton();  
 }  
 });  
 // 找到开始/暂停/继续按钮并添加点击事件  
 Button buttonStart = findViewById(R.id.button\_start);  
 Button buttonPause = findViewById(R.id.button\_pause);  
 Button buttonResume = findViewById(R.id.button\_resume);  
 buttonStart.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 startCountdown(remainingMillis); // 开始倒计时  
 }  
 });  
 buttonPause.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 pauseCountdown(); // 暂停倒计时  
 }  
 });  
 buttonResume.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 resumeCountdown(); // 继续倒计时  
 }  
 });  
 // 找到重置按钮并添加点击事件  
 Button buttonReset = findViewById(R.id.button\_reset);  
 buttonReset.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 resetCountdown(); // 重置倒计时  
 }  
 });  
 }  
 // 更新状态和按钮状态  
 isCountingDown = true;  
 hideResumeButton();  
 showPauseButton();  
 }  
 private void resetCountdown() {  
 // 取消计时器  
 if (countDownTimer != null) {  
 countDownTimer.cancel();  
 }  
 // 重置剩余毫秒数、状态和显示，并更新按钮状态  
 remainingMillis = 25 \* 60 \* 1000;  
 isCountingDown = false;  
 TextView textView = findViewById(R.id.text\_countdown);  
 textView.setText(timerName + "：" + formatMillis(remainingMillis));  
 hidePauseButton();  
 hideResumeButton();  
 showStartButton();  
 Button buttonReset = findViewById(R.id.button\_reset);  
 buttonReset.setVisibility(View.GONE);  
 }  
 private void updateCountdown(long millisUntilFinished) {  
 remainingMillis = millisUntilFinished; // 保存剩余毫秒数  
 isCountingDown = true; // 将状态设置为正在倒计时中  
 TextView textView = findViewById(R.id.text\_countdown);  
 textView.setText(timerName + "：" + formatMillis(millisUntilFinished));  
 }  
 private void onCountDownFinished() {  
 isCountingDown = false; // 将状态设置为未在倒计时中  
 showResetButton();  
 Toast.makeText(this, timerName + " 时间到！（为了您的学习，未响铃）", Toast.LENGTH\_SHORT).show();  
 }

## 4.5休息区功能

休息区功能旨在帮助用户在紧张的学习或工作之余得到放松。通过调用网上公开视频Url，应用可以播放各种休闲娱乐视频，如音乐、电影、动画等。用户可以根据自己的喜好选择视频进行播放，以缓解压力和疲劳。

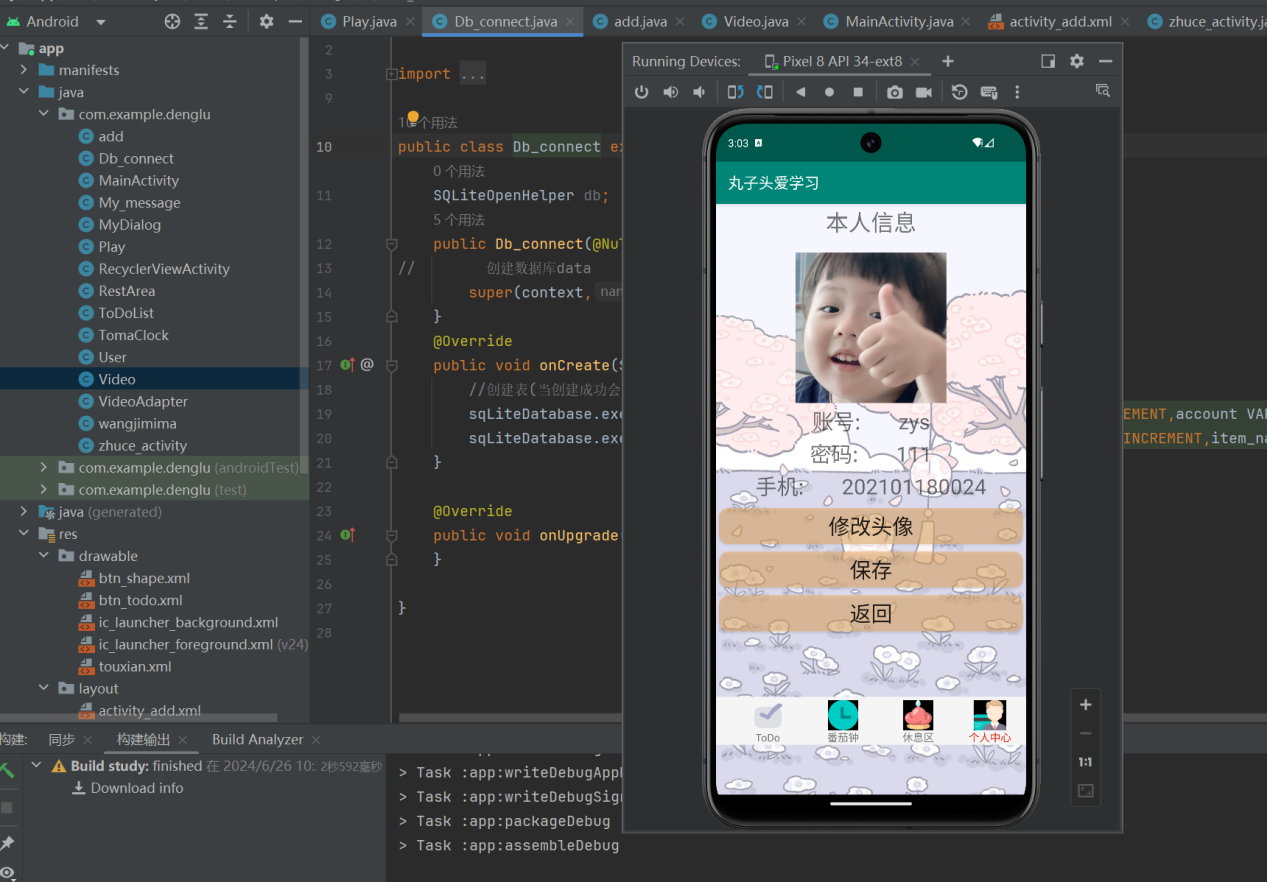
（网上视频播放卡顿，延迟高）

代码实现：

private void initView(){  
 recyclerView=findViewById(R.id.recyclerview);  
 }  
 private void initData(){  
 List<Video> videlist=new ArrayList<>();  
 videlist.add(new Video("八佰","https://vt1.doubanio.com/202011121422/402a7cf332c7a8872b26f34f1da66f7e/view/movie/M/402670047.mp4",R.mipmap.icon\_1));  
 videlist.add(new Video("恶棍之都","https://vt1.doubanio.com/202011121427/bb698fcd0365ee8949c40989187cdcd7/view/movie/M/402660947.mp4",R.mipmap.icon\_1));  
 videlist.add(new Video("一点到家","https://vt1.doubanio.com/202011121429/3b7edd7e79d1abac319c5d364c1a4571/view/movie/M/402670105.mp4",R.mipmap.icon\_1));  
 videlist.add(new Video("误杀","https://vt1.doubanio.com/202011121431/6e7c24efddca20768d54a1a3fbc58cec/view/movie/M/402560730.mp4",R.mipmap.icon\_1));  
 videlist.add(new Video("一出好戏","https://vt1.doubanio.com/202011121435/0fcf83d2b7fc80124c1713431929be02/view/movie/M/402340740.mp4",R.mipmap.icon\_1));  
 videlist.add(new Video("我不是药神","https://vt1.doubanio.com/202011121748/c5e407af3a99d15062b7b389c364e204/view/movie/M/402330315.mp4",R.mipmap.icon\_1));  
 videlist.add(new Video("无邪","https://vt1.doubanio.com/202011121949/8b9f4e93c9865a8584e3bfc9b00c9673/view/movie/M/402680124.mp4",R.mipmap.icon\_1));  
  
 //适配器  
 VideoAdapter adapter=new VideoAdapter(videlist,this.username);  
 StaggeredGridLayoutManager layoutManager=new StaggeredGridLayoutManager(1,StaggeredGridLayoutManager.VERTICAL);  
 recyclerView.setLayoutManager(layoutManager);  
 recyclerView.setAdapter(adapter);  
 }  
}

## 4.6个人中心功能

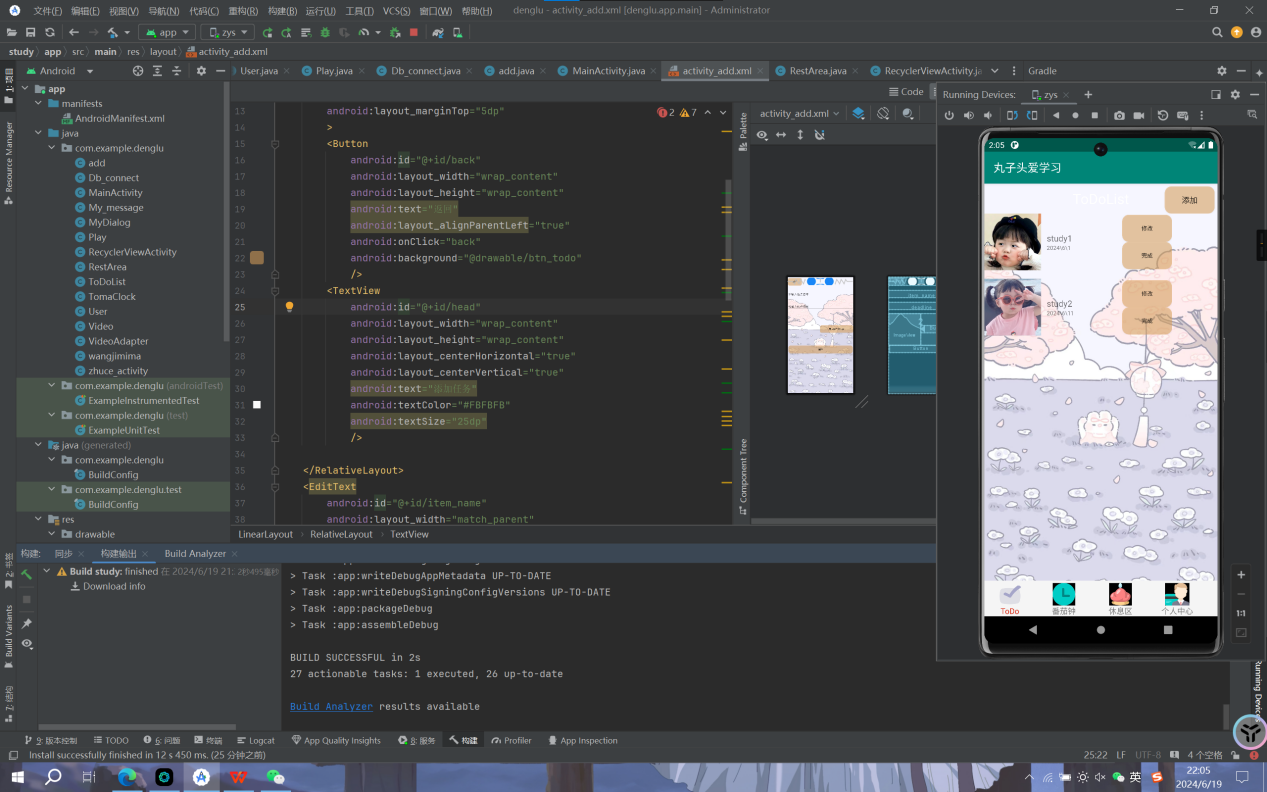
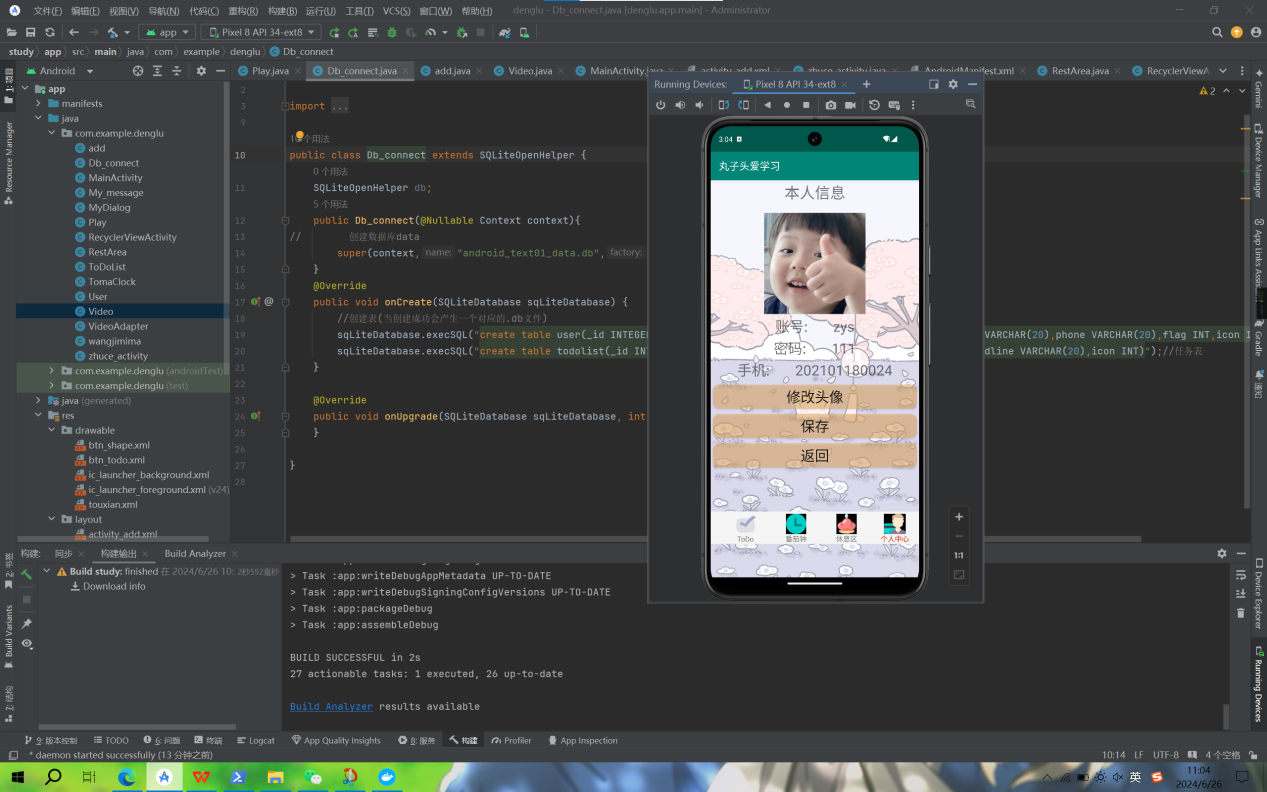
个人中心功能是用户管理自己个人信息的重要入口。用户可以在这里修改头像、查看密码、修改手机号等操作。为了保护用户隐私，应用采用了多种安全措施，如密码加密存储、敏感信息脱敏处理等。同时，应用还支持在个人中心查看自己的任务清单、学习统计等信息，方便用户随时了解自己的学习和生活状态。



忘记密码的实现：

package com.example.denglu;  
import androidx.appcompat.app.AlertDialog;  
import androidx.appcompat.app.AppCompatActivity;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.util.Log;  
import android.widget.Button;  
import android.view.View;  
import android.widget.EditText;  
import android.widget.TextView;  
import android.widget.Toast;  
  
import java.io.Serializable;  
import java.util.ArrayList;  
  
public class wangjimima extends AppCompatActivity {  
 private Button back\_two,zhaohui;  
 private EditText shouji,zhanghao;  
  
 private ArrayList<User> users=new ArrayList<User>();  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_wangjimima);  
 back\_two=(Button)findViewById(R.id.back\_two);  
 zhaohui=(Button)findViewById(R.id.zhaohui);  
 shouji=(EditText)findViewById(R.id.shouji);  
 zhanghao=(EditText)findViewById(R.id.zhanghao);  
  
  
 Intent intent=new Intent();  
 this.users=(ArrayList<User>) getIntent().getSerializableExtra("users");  
}  
 public void back\_two(View v){  
 Intent intent=new Intent();  
 intent.setClass(wangjimima.this,MainActivity.class);  
 startActivity(intent);  
 }  
 public void zhaohui(View v){  
 String zhang=zhanghao.getText().toString();  
 String shou=shouji.getText().toString();  
 String message="";  
 if(zhang.length()!=0&&shou.length()!=0){  
 User user=new User(zhang,"",shou,0,0);  
 //判断是否存在  
 Boolean flag=this.users.contains(user);  
 if(flag==true){  
 String mima=this.getmima(zhang,shou);  
 Log.i("wangji", "Zhaohui"+mima);  
 message="用户密码为"+mima;  
 }else{  
 Log.i("wangji","不存在");  
 message="您输入的信息有误";  
 }  
 Toast.makeText(wangjimima.this,message,Toast.LENGTH\_LONG).show();  
  
 }else{  
 Toast.makeText(wangjimima.this,"账号和密码不能为空",Toast.LENGTH\_LONG).show();  
 }  
 }  
 public String getmima(String zhang,String phone){  
 String mima="";  
 for (int i=0;i<this.users.size();i++){  
 if(zhang.equals(this.users.get(i).getZhanghao())&&phone.equals(this.users.get(i).getPhone())){  
 mima=this.users.get(i).getMima();  
 }  
 }  
 return mima;  
 }  
}

Android Studio开发截图



# 第5章 技术实现与挑战

在开发过程中，我们采用了Android Studio作为开发工具，利用Java语言进行编程。同时，我们还使用了SQLite数据库来存储用户数据，以确保数据的持久化和安全性。

然而，在开发过程中我们也遇到了一些挑战。例如，在登录功能中，我们需要实现用户名和密码的验证功能，并确保用户信息的唯一性和安全性。在任务清单功能中，我们需要设计合理的任务添加和删除流程，并考虑如何设置任务时间和封面。在番茄学习钟功能中，我们需要实现精确的计时和提醒功能，并考虑如何适应不同用户的需求。在休息区功能中，我们需要解决视频加载时间长、无法播放等问题，以提高用户体验。在个人中心功能中，我们需要保护用户隐私，防止密码泄露等安全问题。

为了克服这些挑战，我们不断学习和探索新的技术和方法。我们研究了Android开发文档和社区资源，学习了数据库设计和优化技术，并参考了其他成功应用的开发经验。通过不断尝试和改进，我们最终实现了应用的各项功能，并提高了应用的稳定性和用户体验。

# 第6章 总结与展望

本大作业旨在开发一款集多种功能于一体的综合性应用，以满足用户在日常生活和学习中的多样化需求。通过登录、任务清单、番茄学习钟、休息区和个人中心等功能的设计和实现，我们为用户提供了一种全新的生活方式和学习体验。然而，我们也意识到应用中还存在一些缺点和问题，如登录功能中用户名和密码不支持回车下一步功能、任务清单中设置任务时间没有调用选取时间组件等。在未来的工作中，我们将继续改进和优化应用的功能和性能，提高用户体验和满意度。同时，我们也将探索更多的应用场景和功能扩展，以满足用户不断增长的需求和期望。

# 参考文献

1. IEEE, University of Virginia, Charlattesville,2010:23-25.
2. Darcey L, Conder S. Android Wireless Application Development Volume I: Android Essentials[J].
3. Chicago,Addison-WesleyProfessional,2012,8(8):1294-1305.
4. 差沙, 地狱男爵. 用 Android 开发手机应用[J]. 程序员,2008(1):56-61.
5. 杜艳绥. 基于 Android 平台架构应用程序开发研究[J]. 电脑编程技巧与维护, 2013(6):12-13.
6. 陈莉君. Linux 操作系统内核分析[M]. 北京：人民邮电出版社, 2000：12-19.
7. 许瑾. 基于 Android 平台音乐播放器的设计与实现[D]. 北京：北京邮电大学硕士学位论文,2011：42-55.
8. 胡俊波. 基于 Android 的智能家居系统 APP 的研究与设计[D]. 桂林：桂林电子科技大学硕士学位论文, 2015：38-39.
9. 徐大明. 基于机器学习的跨平台水产精准养殖管理系统设计[D]. 太原：太原理工大学硕士学位论文, 2016:29-32.