



南京工業職業技術學院

Nanjing Institute Of Industry Technology

智能终端应用系统开发实训

——Android 日程管理系统设计与开发

(/12/3—/01/3)

项目名称	Android 日程管理系统设计与开发
院 系	计算机与软件学院
班 级	移设 1213
组 别	第五组
学 号	、09、23、48、50、52
学生姓名	顾飞黄锦烽张国浩 张慧敏张燕云朱冰雪
指引教师	张以利

目录

第 1 章 绪论	5
1.1 日程管理来源	5
1.2 日程管理特点	5
1.3 任务分派	5
第 2 章 系统开发环境及技术	6
2.1 系统开发环境	6
2.1.1 硬件环境	6
2.1.2 软件支持	6
2.2 系统开发技术	6
2.2.1 java 简介	6
2.2.2 eclipse 简介	7
2.2.3 SQLite 简介	8
第 3 章 日程管理设计与分析	9
3.1 顾客需求	9
3.2 功能需求	9
3.3 日程管理设计难易点分析	9
3.4 日程管理设计总体框架	10
第 4 章 概要设计	
4.1 系统总体功能设计	11

4.1.1 系统设计原则	11
4.1.2 系统构造功能图	11
4.2 数据库设计	11
4.2.1 分类报表	11
4.2.2 数据库表	12
第 5 章 APP 界面设计	13
5.1 APP 应用图标设计	13
5.2 界面元素	13
第 6 章 详细设计与实现	13
6.1 程序引导加载界面	13
6.2 欢迎界面	15
6.3 注册登录模块	16
6.4 程序主界面模块	18
6.5 滑动菜单模块	19
6.6 添加日程模块	23
6.7 个人中心模块	26
6.8 反馈信息界面	29
6.9 应用分享模块	29
6.10 应用推荐模块	29

6.11 关于咱们界面	30
6.12 设立模块	30
第 7 章 应用测试	33
小结	34
参考文献	38

第 1 章 绪论

1.1 日程管理来源

在没有手机功能齐全时代，咱们日程安排只能通过手写记录，经常会由于没有及时翻阅记录而错过了重要日程，随着手机技术飞速发展，手机应用迅速推广，手机变化了人们生活，工作方式。日程管理就是将每天工作和事务安排在日期中，并做一种有效记录，以便管理日程工作和事务，达到工作备份目。同步也具备对寻常工作进行指引、监督作用。

1.2 日程管理特点

1:绿色免费,程序轻便小巧,只占用几 M 内存!

2:非常灵活日期提示设立

3:全年月日周周期定期循环提示

4:绝不错过未开机状况下提示

5:完善日程查询功能

6:功能完善,简朴易用

1.3 任务分派

表 1.1 任务分派表 可以清晰看到每个人在项目开发过程中所承担任务

顾飞	任务分派 已经做好功能归类 项目设立，编写文档
黄锦烽	设立告知栏常驻 语音辨认
张国浩	欢迎界面 图片资源准备
张慧敏	数据库表设计 设立界面布局
张燕云	图标设计 主界面 引导界面图片设计
朱冰雪	时间日期选取器 记事本功能，编写文档

第2章 系统开发环境及技术

2.1 系统开发环境

2.1.1 硬件环境

处理器 PentiumII 以上；

内存容量 1G 以上；

外存容量 50G 以上；

2.1.2 软件支持

操作系统：windows 7

编程语言：JAVA

数据库： SQLite

开发工具： eclipse

2.2 系统开发技术

2.2.1 java 简介

背景：

Java 是一种应用分布式网络环境中程序设计语言，由 sun 公司开发，Java 语言广为人知，得益于 Internet 迅猛发展。人们开始理解并使用这种神奇语言，普通应用程序、Applet、web 服务器乃至手机嵌入式系统，Java 都足以胜任。Java 对网络提供了强有力支持，这也是这种奇妙语言魅力所在。

特点：

- (1) 面对对象
- (2) 可移植性(universality)
- (3) 安全性(security)
- (4) 多线程(thread)

- (5) 多态
- (6) 解释执行
- (7) 分布性

2.2.2 eclipse 简介

Eclipse 中三个最吸引人地方：

一是它创新性图形 API，即 SWT/JFace，在此之前，我曾经用过 Java AWT/SWING 来开发 Windows 应用程序，AWT/SWING 其界面客观地讲不够美观，并且界面响应速度比较慢；而 SWT/JFace 则大大改进了 Java 在这方面能力。二是它插件机制。三是运用它插件机制开发众多功能强大插件。

Eclipse 平台体系构造：

Eclipse 平台是一种具备一组强大服务框架，这些服务支持插件，例如 Java 开发环境插件（JDT）、插件开发环境（PDE）和其它第三方插件（Newtool）。它由几种重要某些构成：平台运营库、工作区、工作台、团队支持和协助。

Eclipse 平台体系构造：

平台运营库是内核，它在启动时检查已安装了哪些插件，并创立关于它们注册表信息。为减少启动时间和资源使用，它在实际需要任何插件时才加载该插件。除了内核外，其她每样东西都是作为插件来实现。

工作区是负责管理顾客资源插件。这涉及顾客创立项目、那些项目中文献，以及文献变更和其她资源。工作区还负责告知其她插件关于资源变更信息，例如文献创立、删除或更改。

工作台为 Eclipse 提供顾客界面。它是使用原则窗口工具包（SWT）和一种更高档 API（JFace）来构建；SWT 是 Java Swing/AWT GUI API 非原则代替者，JFace 则建立在 SWT 基本上，提供顾客界面组件。SWT 已被证明是 Eclipse 最具争议某些。SWT 比 Swing 或 SWT 更紧密地映射究竟层操作系统本机图形功能，这不但使得 SWT 更迅速，并且使得 Java 程序具备更像本机应用程序外观和感觉。使用这个新 GUI

API 也许会限制 Eclipse 工作台可移植性,但是针对大多数流行操作系统 SWT 移植版本已经可用。Eclipse 对 SWT 使用只会影响 Eclipse 自身可移植性——使用 Eclipse 构建任何 Java 应用程序都不会受到影响,除非它们使用 SWT 而不是使用 Swing/AWT。

团队支持组件负责提供版本控制和配备管理支持。它依照需要添加视图,以容许顾客与所使用任何版本控制系统(如果有话)交互。大多数插件都不需要与团队支持组件交互,除非它们提供版本控制服务。

2.2.3 Sqlite 简介

SQLite,是一款轻型数据库,是遵守 ACID 关联式数据库管理系统,它设计目的是嵌入式,并且当前已经在诸多嵌入式产品中使用了它,它占用资源非常低,在嵌入式设备中,也许只需要几百 K 内存就够了。它可以支持 Windows/Linux/Unix 等等主流操作系统,同步可以跟诸多程序语言相结合,例如 Tcl、C#、PHP、Java 等,尚有 ODBC 接口,同样比起 Mysql、PostgreSQL 这两款开源世界知名数据库管理系统来讲,它解决速度比她们都快。SQLite 第一种 Alpha 版本诞生于 5 月。至今已有 13 个年头,SQLite 也迎来了一种版本 SQLite 已经发布。

SQLite 是遵守 ACID 关系型数据库管理系统,它包括在一种相对小 C 库中。它是 D.RichardHipp 建立公有领域项目。

不像常用客户-服务器范例,SQLite 引擎不是个程序与之通信独立进程,而是连接到程序中成为它一种重要某些。因此重要通信合同是在编程语言内直接 API 调用。这在消耗总量、延迟时间和整体简朴性上有积极作用。整个数据库(定义、表、索引和数据自身)都在宿主主机上存储在一种单一文献中。它简朴设计是通过在开始一种事务时候锁定整个数据文献而完毕。

第 3 章 日程管理设计与分析

3.1 顾客需求

(1) 功能需求: 本系统完毕对自己日程进行记录, 每天日程安排以及查看修改工作。

(2) 权限需求: 顾客管理员拥有最高权限, 可以对系统中所有数据进行查看、添加、修改和删除等操作; 普通顾客只可以对自己进程进行查看、添加、修改和删除等操作。

3.2 功能需求

(1) 顾客登录、顾客注册: 顾客通过登录方式进行口令校验安全进入在线日程管理系统: 顾客注册是顾客为进入该系统时将个人信息提交到数据库中。

(2) 日程管理: 其中使用 ListView 控件, 对某一天或某段时间日程进程查看, 和对某个日程地点、开始时间、重要性及联系电话进行修改, 更新等操作。

3.3 日程管理设计难易点分析

数据库操作 (增删改查)

界面整体设计: 找不到适合框架, 自己又不会写

调试程序时候经常性遇到异常

3.4 日程管理总体框架

日程管理 APP 开发流程包括 应用定位、项目规划、UI 设计数据库设计, 功能类实现以及工具类附加

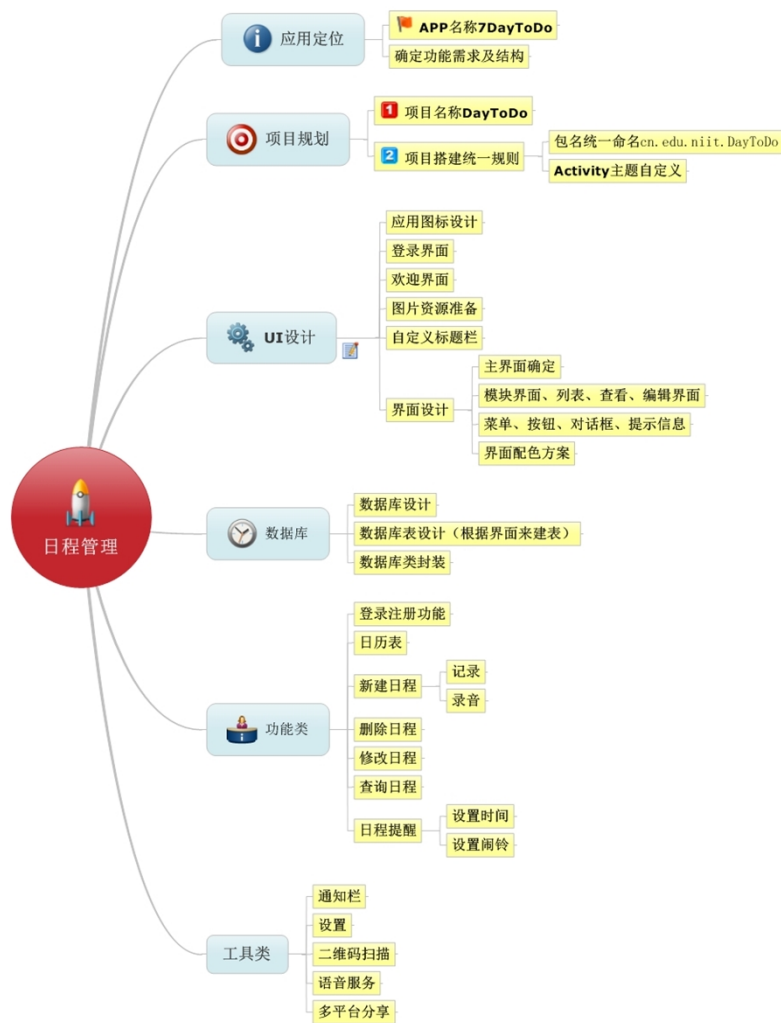


图3.1 总体框架图

第4章 概要设计

4.1 系统总体功能设计

4.1.1 系统设计原则

日程管理 APP 重要分为两大块功能：

日程记录功能：日程添加 闹钟提示

个人中心：可以进行注册 登录 并且可以对自己信息进行修改

系统设立：设立主题 告知栏启动 清除缓存等功能

应用分享：依照第三方提供 share 分享服务 可以将 APP 分享到别平台

4.1.2 系统机构功能图

日程管理概要设计包括四大模块：日程管理，个人中心，系统设立，应用分
享

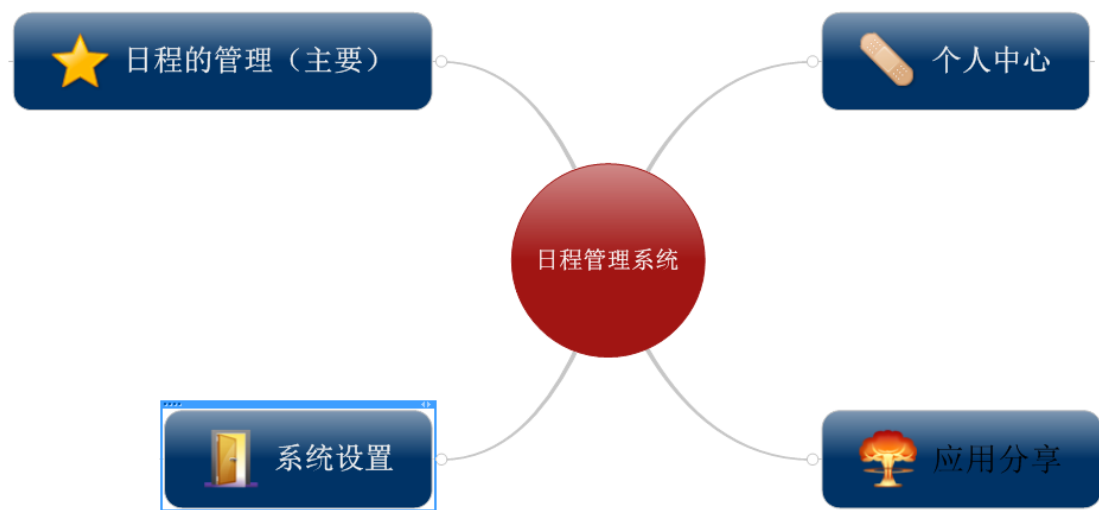


图 4.1 系统功能图

4.2 数据库设计

4.2.1 分类报表

本次实训采用了一张数据库两张表（分别是日程表和顾客表）

4.2.2 数据库表

数据库采用了两张数据库表 分别是日程管理表（Remind表）和顾客信息表（Usertab表）

Remind表

字段	数据类型	备注
title	varchar	标题

remindDate	varchar	设立闹钟日期
remindTime	varchar	设立闹钟时间
serviceflag	integer	闹钟服务开关
_id	integer	日程 id
remindCounts	integer	闹钟计数
note	varchar	正文

图表 4.2 Remind表数据表内容

Usertab表

字段	数据类型	备注
id	int	顾客 id
username	varchar	顾客名
password	varchar	密码
nickname	varchar	昵称
regist_time	varchar	注册时间

图表 4.3 Usertab表数据表内容

第 5 章 APP 界面设计

5.1 APP应用图标设计

图标设计含义：一周有7天，咱们要筹划着做好每一天事情，因此APP名字为7DayToDo

绘制工具：Ps

绘制办法：钢笔工具切片图层，填充色块而成



图5.1 应用图标

5.2 界面元素

图标设计伪扁平

蓝色+灰色 朴素大方

设计比较简洁，易于查看

第 6 章 详细设计与实现

6.1 程序引导加载页面

每次进入应用时会显示一种程序加载页面



图6.1 程序引导加载图

重要代码（设立线程休眠后进行跳转）

```
final Intent intent = new Intent(this, GuideActivity.class);
```

// 如果之前启动过这个Activity，并还没有被destroy话，而是无论与否存在，
都重新启动新activity

```
intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
```

// 创立一种新线程来显示欢迎动画，指定期间后结束，跳转至指定界面

```
new Thread(new Runnable() {
```

```
    @Override
```

```
    public void run() {
```

```
        try {
```

```
            Thread.sleep(3000); // 用线程暂停3秒来模仿做了一种耗时3秒
```

检测操作

```
            SharedPreferences preferences =
```

```
            getSharedPreferences("isFirstUse", MODE_WORLD_READABLE);
```

```
            isFirstUse = preferences.getBoolean("isFirstUse", true);
```

```
            if (isFirstUse) {
```

```

        startActivity(new Intent(Welcome.this,
GuideActivity.class));

    } else {

        startActivity(new Intent(Welcome.this,
MainActivity.class));

    }

    // 获取应用上下文，生命周期是整个应用，应用结束才会结束
//
    getApplicationContext().startActivity(intent); // 跳转
    finish(); // 结束本欢迎画面Activity

    Editor editor = preferences.edit();

    editor.putBoolean("isFirstUse", false);

    editor.commit();

} catch (InterruptedException e) {

    e.printStackTrace();

}

}).start();

```

6.2 第一次进入时欢迎界面

第一次使用应用所看到欢迎漫画



妈妈，我做功课，忘记休息了



这是赏你的



你打我干嘛，多少次了



妈妈你记得用**7DayToDo**
这样就不会忘记打我了

注册

登录

跳过

图6.2 欢迎界面图

重要代码（创立一种引导图片切换适配器来管理几种 view）

```
private void initView() {  
  
    LayoutInflater mLi = LayoutInflater.from(this);  
  
    view1 = mLi.inflate(R.layout.gudie_view01, null);  
  
    view2 = mLi.inflate(R.layout.gudie_view02, null);  
}
```

```

        view3 = mLi.inflate(R.layout.guide_view03, null);

        view4 = mLi.inflate(R.layout.guide_view04, null);

        viewPager = (ViewPager) findViewById(R.id.viewpager);

        views = new ArrayList<View>();

        vpAdapter = new ViewPagerAdapter(views);

        regist = (Button) view4.findViewById(R.id.regist);

        login = (Button) view4.findViewById(R.id.login);

        startBt = (Button) view4.findViewById(R.id.startBtn);

    }

    private void initDate() {

        viewPager.setOnPageChangeListener(this);

        viewPager.setAdapter(vpAdapter);

        views.add(view1);

        views.add(view2);

        views.add(view3);

        views.add(view4);
    }

```

6.3 注册登录模块

注册登录所展示 UI 界面以及数据库写入实现办法

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。

如要下载或阅读全文，请访问：

<https://d.book118.com/068030021103006117>