**参赛作品说明书**

**作品名称** ：基于Android的Schedule APP 设计与实现

**完成时间** ：2015年6月1日

目录

1 引言 3

1.1 国内Android课程表APP现状 3

2 客户端Schedule的设计介绍 3

2.1 功能介绍 3

2.1.1课程表显示 4

2.1.2 课程表放大、滚动和课程拖放 5

2.1.3 课程设置 6

2.1.4 课程调整 7

2.1.5快速拨号 7

2.1.6 上课静音和上课提醒 8

2.1.7教学进度设置 9

2.1.8 课程表课程分享 9

3 Android设计架构 11

3.1 UI层 11

3.1.1界面结构 12

3.2应用层 14

3.3数据访问层 16

3.3.1数据封装 16

4 服务器端的设计： 17

# 1 引言

现如今智能手机在不断地改变着人们的生活，影响着师生们的生活习惯。广大师生们都拥有了自己的智能手机，而越来越多的学生和教师使用自己的手机来查看课程表。现在手机应用市场上有不少方便学生们使用的课程表APP(Application)，却极少有针对教师使用习惯设计的课程表APP。

Android系统占据着智能手机的大部分市场，拥有广泛的应用前景，本设计基于Android系统，并参考现有的课程表APP和本校教学实际情况，来实现一款既适合学生使用的，又满足教师使用习惯的Schedule APP。

本文字材料将介绍Schedule APP的设计与实现方法，并列举主要功能的实现。

## 1.1 国内Android课程表APP现状

根据在百度Android APP市场内检索到的课程表APP信息可以了解到，国内Android APP市场已经存在不少的课程表APP，但有不少课程表APP是为某高校定制，需要配合该高校的学生账户使用。虽然也有一部分通用的课程表APP，但是都主要面向学生的使用需求而设计的。

从国内APP市场分析及结果得出，为课程表的另一用户——教师设计的、符合教师使用习惯的课程表APP极少。如下载量达到千万级、位居第一的超级课程表APP，也是针对大学生用户设计的。而这些主流课程表APP，虽然拥有课程表的功能和良好的界面，但主推的都是他们的社交功能。

许多课程表APP附带的功能五花八门，却不是非常实用，比如课堂笔记、记事本、社交等功能，笔记、记事等功能不如印象笔记、WPS等更为专业的商务的APP，而且在课堂上不提倡使用手机，这些功能对于教师和学生是多余的。多余的功能，不仅造成开发成本的浪费，还使APP变得复杂、不易使用。

# 2 Schedule客户端设计

## 2.1 功能介绍

本作品是基于Android的Schedule APP ，采用主流的竖屏界面设计样式，并参考本校的作息时间和上课规律进行开发与实现。

### 2.1.1课程表显示

课程表显示是用到了一个自定义的控件，这个控件名为CourseTable。其本作品的课程表显示机制为：利用Android所提供的绘图方式，继承View控件，重写onDraw方法，最后将课程表“画”出来。

绘制课程表框架的原理和步骤为：

1. 获取控件的宽度。
2. 根据控件宽度来决定课程表文本字体大小。
3. 根据字体大小来决定表格单元格的长度和宽度。
4. 根据单元格的长度、宽度和所在行列，定位单元格的坐标。
5. 根据坐标，使用canvas的drawLine方法绘制表格线条，canvas的drawText来绘制文本。

而绘制课程的实现与课程表原理类似，绘制课程的原理和步骤为：

1. 获取课程表框架的单元格长度和宽度。
2. 根据单元格长度和宽度，课程的上课节次和星期等数据，定位课程坐标。
3. 根据坐标，使用使用canvas的drawRoundRect方法绘制圆角矩形，canvas的drawText来绘制文本。

其中有三个关键的点：

1. Android屏幕的坐标系是左上角为坐标原点，x向右递增，y向下递增。
2. drawText方法可以通过改变其Paint参数的样式来实现文本居中。
3. 由于Android所提供drawText方法为只能绘制单行文本，所以在矩形内绘制文本需要将文本分行，并根据字体和矩形高度、宽度来计算可绘制的行数、列数，若文本过长则进行截取，最后根据行数列数来绘制文本。

### 2.1.2 课程表放大、滚动和课程拖放

由于本作品的课程表为自定义控件，在自定义控件中，往往需要自己捕捉用户的触屏事件来响应用户的操作。我们可以重写View的触屏事件onTouchEvent(MotionEvent event)方法。而参数event中含有用户触碰屏幕的x，y坐标和动作。而View的postInvalidate方法可以回调onDraw方法来刷新控件。

课程表滚动的实现原理及步骤如下：

1. 在绘制课程表的时候，给所有包含x,y坐标参数的绘制方法加上一个分别加上一个moveX,moveY的整型变量（int）。
2. 捕捉用户第一次触碰屏幕的x,y坐标并保存为lastX,lastY。
3. 捕捉用户的滑屏动作，并实时记录每次滑动屏幕，将最新的坐标与之前的坐标相减，然后累加到moveX, moveY变量中，并更新上次坐标lastX,lastY。
4. 使用View的postInvalidate方法重新绘制控件，刷新显示。

课程表放大的操作的实现原理与滚动基本相同，不同的是，课程表放大捕捉只捕捉用户触碰屏幕两点的情况，并且计算方式也不同。由于课程表所有的绘制坐标都可以追溯到绘制的字体大小，所以可以在字体大小上加一个放大的变量。放大的原理和步骤如下：

1. 给绘图的字体大小加上一个放大浮点(float)变量scaleSize。
2. 捕捉用户第一次触碰的两点坐标，并根据这两点计算两点之间的长度，记录到一个浮点变量scaleLastLenght中。
3. 捕捉用户在触碰的时候，滑动屏幕的两点坐标，并根据这两点计算两点之间的长度，记录到一个浮点变量scaleCurrentLenght中。根据公式5.1，来计算应该放大的比例scaleSize。

scaleSize+=(scaleCurrentLenght - scaleLastLenght) / 控件宽度（5.1）

1. 使用View的postInvalidate方法重新绘制控件，刷新显示。

课程拖放的实现与课程表滚动实现原理基本相同，其关键是在用户触碰的时候优先检索该触碰点是否在课程上，若不是，继续执行，若是，执行拖动操作。其他非关键的实现本节不作讨论。

### 2.1.3 课程设置

课程设置的添加、删除和修改均由存储策略负责实现。而在界面设计上使用到自定义的控件原理基本与主界面课程表相同。如图所示，用到了选择上课周次的自定义控件，在没有设置上课周次的情况下其默认为6行6列，其行列受上课周次影响。

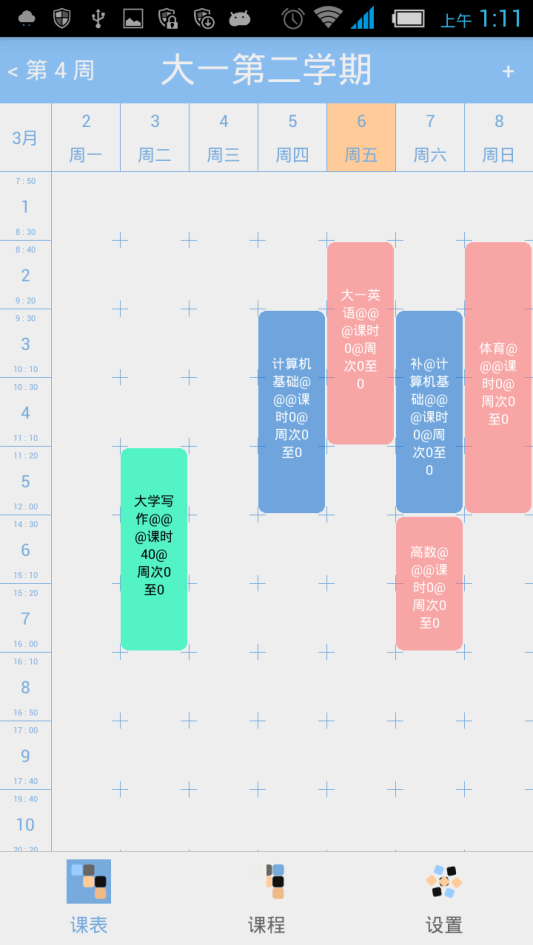
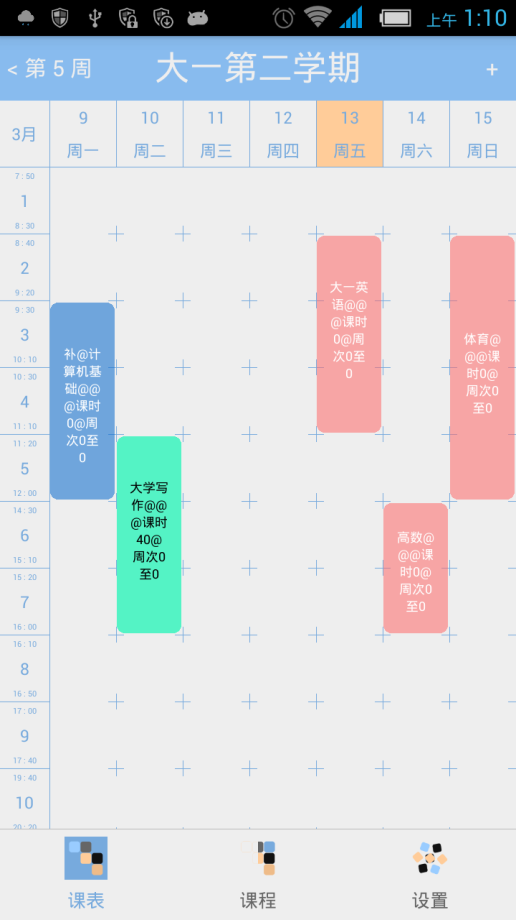


选择周次控件

### 2.1.4 课程调整

课程调整的实现原理比较简单，在课程数据里面添加一个调整对象列表，用这个列表来记录调整的日期、节次和备注信息。在课程绘制的时候根据这些记录来决定是否显示课程，或者是否额外显示调整后的课程。这个功能由后面提到的Course具体观察者对象来完成。

如效果图1，周四的计算机基础课为原本的课程安排，周六的为调整补课，多一个补字。而效果图2中，计算机基础课原本的课程安排停课，周一补课。

课程调整效果1 课程调整效果2

### 2.1.5快速拨号

Android系统默认提供了两种拨号方式：第一种是通过使用Intent设置电话号码来调用系统的拨号，而第二种可以在TextView的布局样式中设置android:autoLink="phone"的拨号属性。第二种方式比较简洁，但是会改变TextView的样式，其样式如图所示。

本设计使用了第二种方式来实现拨打号码。



TextView拨号样式

### 2.1.6 上课静音和上课提醒

上课静音功能和上课听功能实现原理相同，其中涉及Android的3个重要组件，这些组件为：

1. AlarmManager。这是Android时钟管理器，它可以为APP提供系统的时钟服务，让APP在某一个未来的时间执行任务。AlarmManager的set方法可以设置一次任务，setRepeating方法可以设置重复任务。
2. BroadcastReceiver。这是AlarmManager执行任务的载体，通过继承AlarmManager重写其onReceive，可以执行一些如启动APP、播放音乐的任务。
3. PendingIntent。PendingIntent可以指定由哪个BroadcastReceiver执行AlarmManager的定时任务。

本设计中使用继承自BroadcastReceiver 的两个子对象，CourseAlarmReceiver和DailySchuduleReceiver来分别负责上课提醒和上课静音的功能实现。

### 2.1.7教学进度设置

本设计中的教学进度设置与课程调整类似，也使用了一个CourseSchedule的列表来存放教学进度相关的内容。其代码如下：

public class CourseSchedule implements Serializable{

public int startPeriod=0;

public int endPeriod=0;

public String content="";

public String ID="";

}

其中startPeriod为该教学进度的开始课时，endPeriod为教学进度的结束课时，content为保存教学进度的文本，ID为教学进度的标识ID。在进入课程详情界面的时候，将计算已经上过的课时，当课时的值落在某项CourseSchedule的startPeriod和endPeriod之间的情况下，将显示该项的content文本内容。其对象关系结构如图所示。

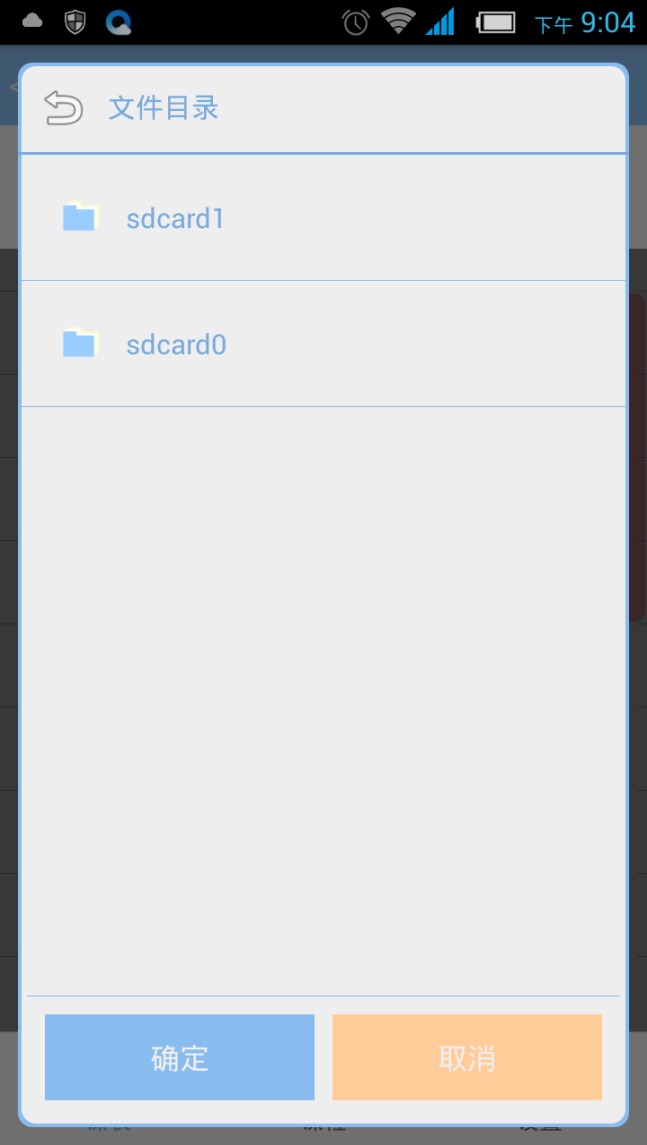
### 2.1.8 课程表课程分享

本作品的课程表分享功能采用导入、导出文件的形式实现，并交给存储策略对象负责。由于Android没有像Windows那样提供访问文件的对话框，也没有提供文件浏览控件，并且本作品的分享功能采用文件存储方式，所以，必须自定义一个文件浏览器。

在本设计中， 使用PopupWindow、ListView作为界面的主要控件，使用File对象来获取文件目录，来实现一个简易的文件浏览对话框。

PopupWindow为一个弹出界面，它是一个轻量级的窗口，可以使用xml的样式布局。而ListView为一个列表空控件，需要配合BaseAdapter（基本适配器）使用，BaseAdapter可以使用xml的样式布局，也就是说任意界面都可以成为ListView的列表项。File对象包含了一个文件的信息，可以获取其文件的子路径。

而文件浏览器的任务就是返回用户的当前路径或选择文件的路径。文件管理器FileBox的实现效果如图所示。



FileBox文件浏览器效果

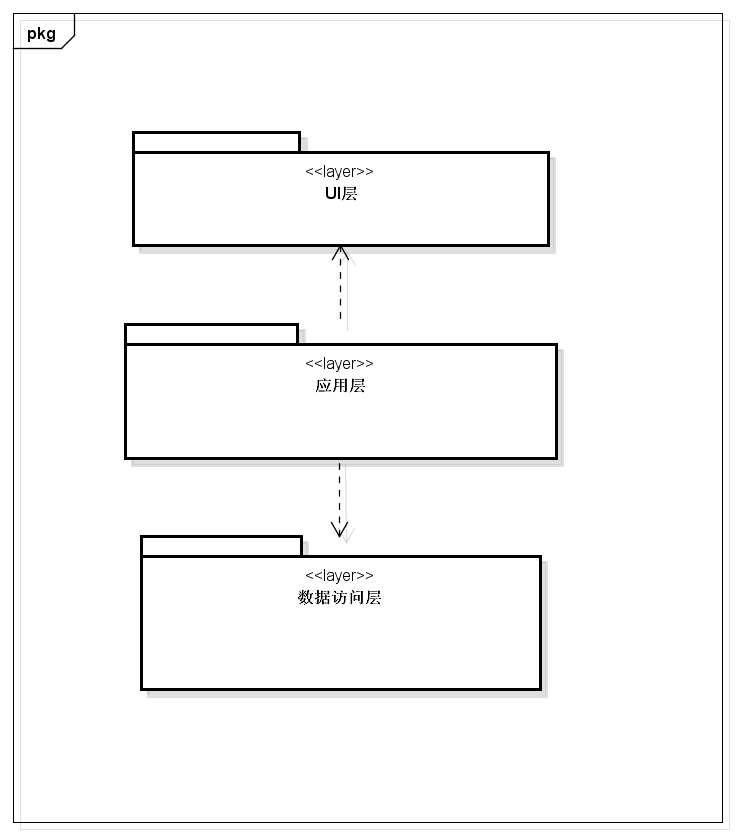
本作品设计的Schedule APP对课程表能够一目了然，看起来很直观。加上可以自由拖动、拉伸、快速调整课程的功能，简化了课程调整带来的繁琐操作，此外课程表支持放大功能，方便手机屏幕小和实力不好的用户使用，其他的课程表APP无此功能。

不同于其他课程表APP，以学生为主要用户进行功能开发，本作品将课程表的另一方用户——教师的需求考虑了进来，添加了课程的调整和教学进度设置功能。

而其他课程表APP的网络社交功能，和一些不适宜在课堂上使用的功能均不在本作品设计的考虑范围内。

# 3 Schedule架构设计

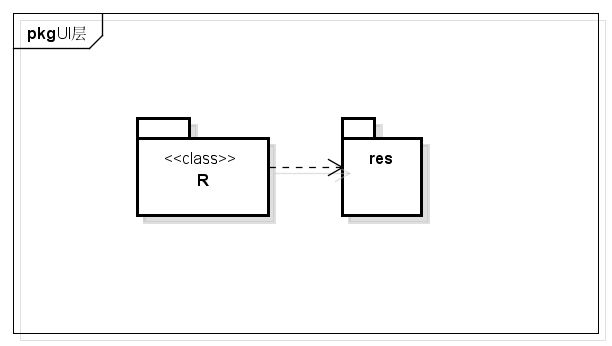
我们遵循数据与界面分离的理念来设计架构，按照Android提供的开发方式，本作品的逻辑架构采用了三层架构，分为UI层、应用层、和数据访问层。



逻辑架构图

## 3.1 UI层

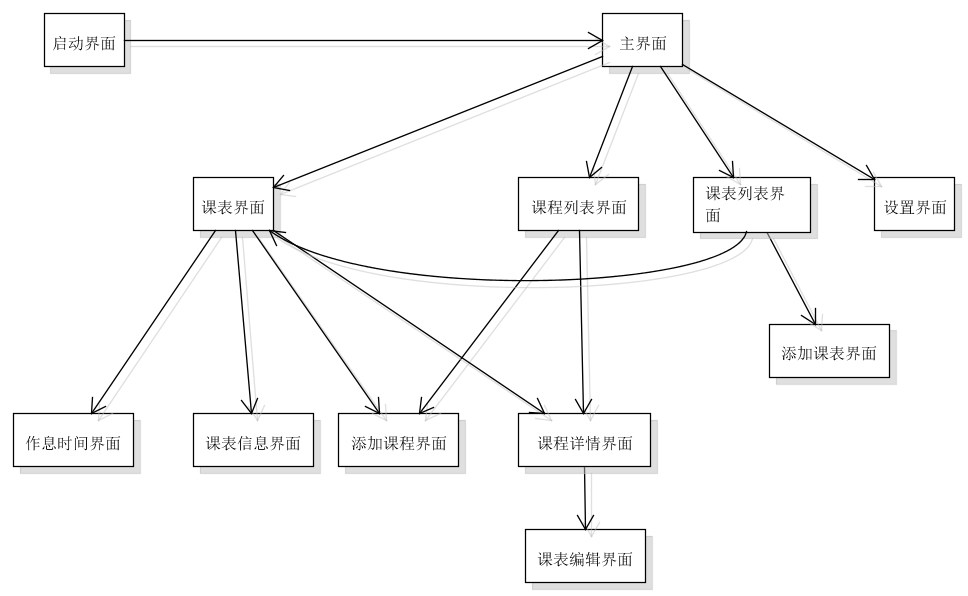
UI层中的res为Android的资源包，包含了APP的界面、样式和动画等文件，供应用层调用。R文件为自动生成的资源标识对象，R里面包含了所有在res文件夹中所有的文件、图片和控件的标识，这些标识供应用层来找到UI控件。



UI层包图

### 3.1.1界面结构

本作品的界面采用采用Activity与Fragment相结合的方式。启动界面为一个单独的Activity，主界面和其他界面均为Activity嵌套Fragment，其中主界面的Activity只负责管理3个主界面的Fragment，其他Fragment均由一另个Activity来管理。这三个Activity分别为LauncherActivity（启动活动） 、MainActivity（主界面活动）和ContainerActivity。



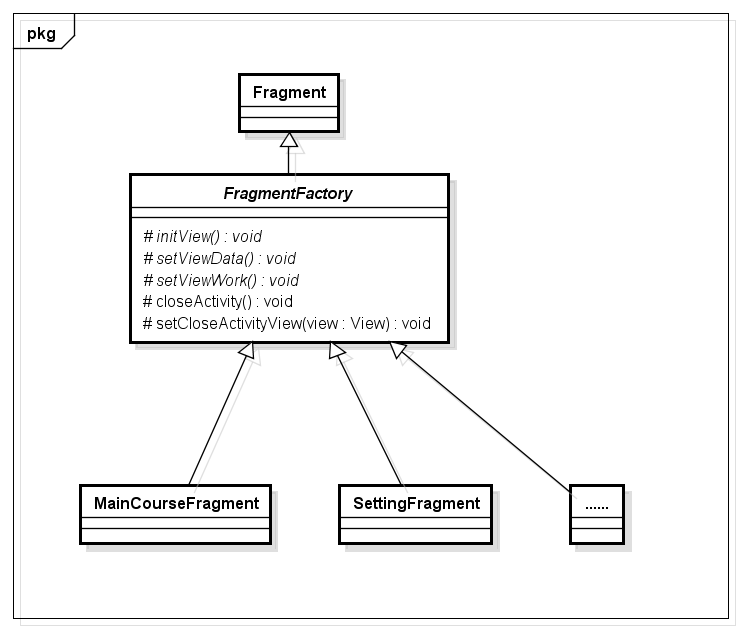
除了启动的界面由Activity负责，其余均使用Activity嵌套Fragment的形式，其对应的对象如下：

1. 启动界面。对应LauncherActivity。仅由一个LauncherActivity来负责。
2. 主界面。对应一个MainActivity。主界面嵌套了三个Fragment界面，其分别是MainCourseFragment（课程表）、CourseAndTableDataListFragment（课程列表和课程表列表）和SettingFragment（设置）。
3. 课程表界面。在主界面的课程表为MainActivity嵌套MainCourseFragment的结构，在课程表列表进入的课程表则为ContainerActivity嵌套MainCourseFragment的结构。
4. 作息时间界面。为ContainerActivity嵌套CourseTableDayliySchedule-

ViewFragment的结构。

1. 课程表信息界面。为ContainerActivity嵌套CourseTableDataFragment的结构。
2. 添加课程表界面。为ContainerActivity嵌套AddCourseTableDataFragment的结构。
3. 课程详情界面。在当用户为学生用户，其结构为ContainerActivity嵌套StudentCourseDetailFragment的结构，当用户为教师用户的时候，其结构为ContainerActivity嵌套TeacherCourseDetailFragment的结构。
4. 添加课程界面。在当用户为学生用户，其结构为ContainerActivity嵌套StudentAddCourseFragment的结构，当用户为教师用户的时候，其结构为ContainerActivity嵌套TeacherAddCourseFragment的结构。
5. 编辑课程界面。在当用户为学生用户，其结构为ContainerActivity嵌套StudentEditCourseFragment的结构，当用户为教师用户的时候，其结构为ContainerActivity嵌套TeacherEditCourseFragment的结构。

**本作品的Fragment结构模型：**



## 3.2应用层

在应用层中：

1) Golbar。存放全局数据的对象，其存放了静态共有的属性，其他对象均可以访问其属性。

2) publicTools。公用的工具包，其包含了一些公用的对象和静态方法，供其他对象调用。

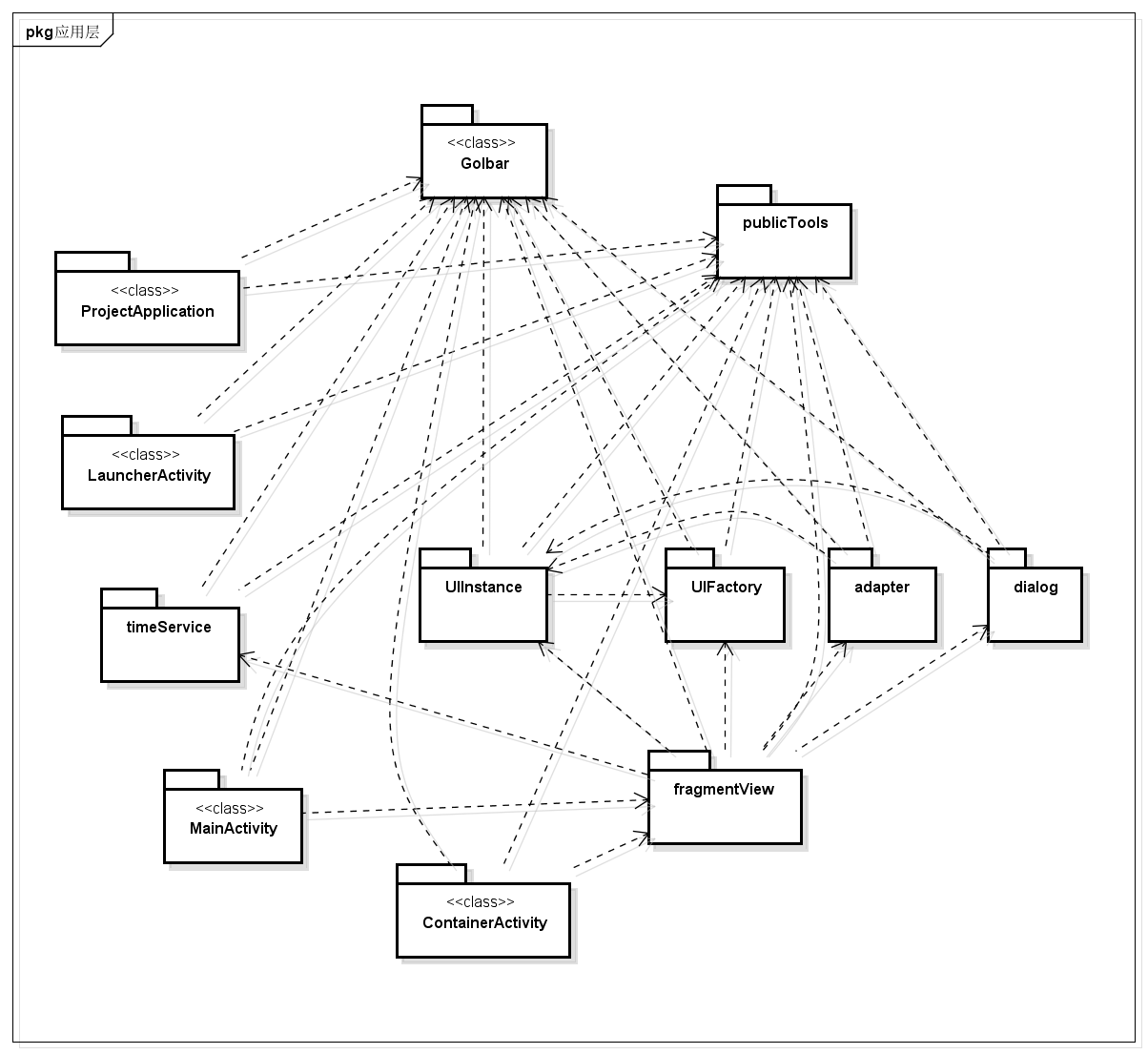
3) ProjectApplication。继承自Application的对象，负责调用接口来执行APP启动需要运行的任务。

4) LauncherActivity。负责显示APP启动界面和执行启动界面相关任务。

5) MainActivity。负责调用fragmentView里的主界面来显示APP主界面和执行主界面相关任务。

6) ContainerActivity。负责调用fragmentView里的界面来显示APP界面。

7) fragmentView。fragmentView为Fragment界面对象的包。其被ContainerActivity调用，来显示APP的部分界面和界面控制事务。



应用层包图

8）adapter。adapter为Android控件需要使用到的适配器包。被fragmentView的界面调用。

9）UIFactory。UIFactory为自定义抽象UI控件包。里面存放了自定义10）控件的抽象类，为UIInstance包里面的部分对象的父类。

11）UIInstance。UIInstance为自定义UI控件对象包。被adapter、UIFactory、adapter这三个包内的与界面相关的对象调用。

12）dialog。dialog为自定义对话框对象的包，用于界面相关的事务，被fragmentView调用。

13）timeService。timeService作为提供时间任务处理服务的对象包，为APP提供系统时间任务处理接口，如上课静音和上课提醒。

## 3.3数据访问层

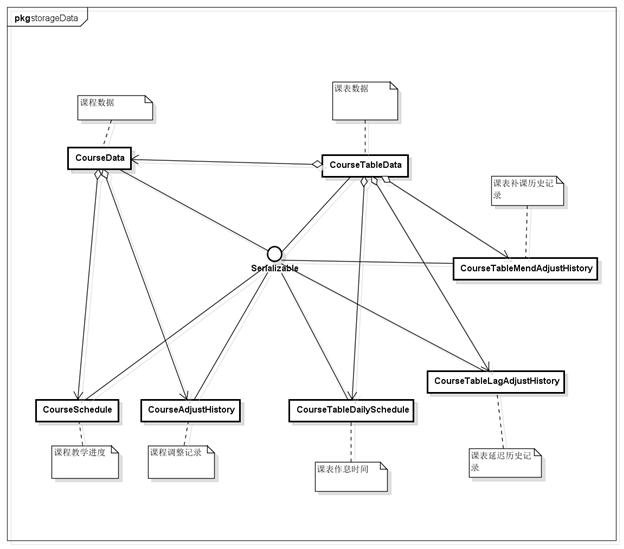
在数据存储方面，由于在现实生活中，一个课程的表的课程一般不会超过20门，而按照小学至大学16年每年2个学期的计算，一共有32个课程表。即最多有620门课程。

假设一门课的数据大小为10kb，课程表大小为200kb，则数据大小总和为12800kb，其占用空间刚好同两首普通mp3文件格式歌曲所占用空间的大小。可见在数据规模上， APP的需要的存储空间并不高。

经过对数据规模的分析和几种数据存储方式的比较，最终本作品使用了JSON文本格式文件的存储方案。

### 3.3.1数据封装

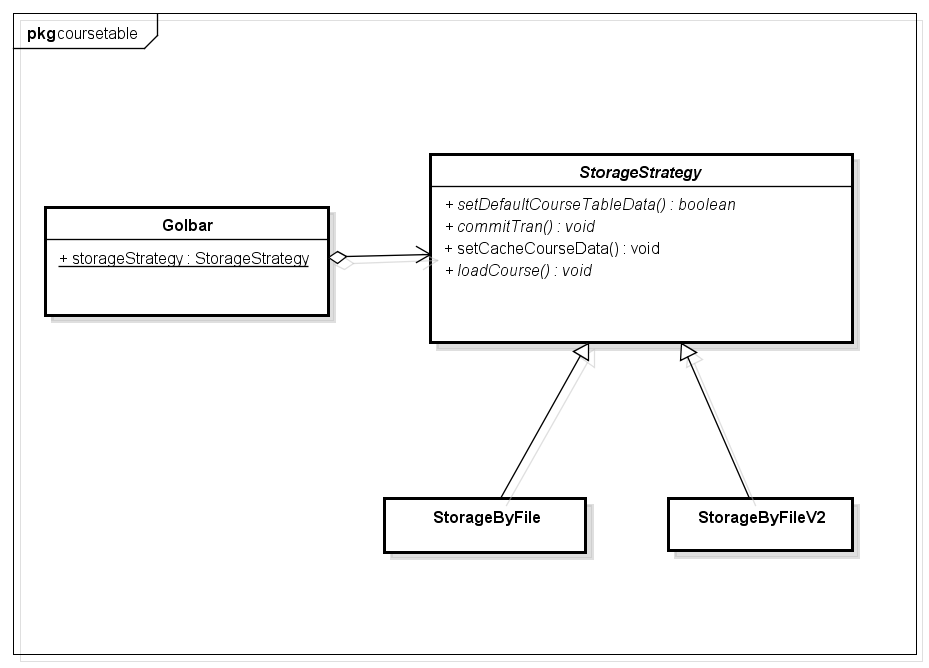
本作品将需要存储的课程数据和课程表数据均独立封装成了对象。如图：



Schedule数据类图

图中的对象均实现了Java的Serializable接口。CourseTableData含有CourseAdjustHistory、CourseTableDailySchedule、CourseTableLagAdjustHistory、CourseTableMendAdjustHistory对象的List集合。而CourseData中包含了CourseSchedule和CourseAdjustHistory两个对象的List集合。

为方便APP存储方式的扩展和变更，本作品在数据存储上使用了策略模式，并采用数据库事务的概念来定义和使用接口。



# 4 服务器端的设计：

## 4.1服务器端响应

本作品服务器端采用了Apache Tomcat服务器，通过编写servlet服务器端程序，进行交互式的浏览和修改数据。主要过程为：

客户端发送请求至服务器端；

服务器将请求信息发送至Servlet；

Servlet生成响应内容并将其传给服务器。响应内容动态生成，通常取决于客户端的请求；

服务器将响应返回给客户端。

Servlet看起来像是通常的Java程序。Servlet导入特定的属于Java Servlet [API](http://baike.soso.com/ShowTitle.e?sp=SAPI)的包。因为是对象字节码，可动态地从网络加载，可以说Servlet对Server就如同Applet对Client一样，但是，由于Servlet运行于Server中，它们并不需要一个图形用户界面。从这个角度讲，

# 5测试

5.1功能测试

此处列举本设计主要功能的测试用例，并给出结果。



### 5.1.1 查看课程表

用户查看课程表功能的测试用例及测试结果如表所示。

查看课程表测试用例



其中，按周次查看课程表的测试用例如表所示。

按周次查看课程表测试用例



### 5.1.2 课程表放大

用户课程表放大功能的测试用例及测试结果如表所示。

表课程表放大测试用例



### 5.1.3 课程设置

课程设置包含了课程的添加、添加到课程表、查看、删除、从课程表上删除和编辑这6种不同的操作，这些操作与所在UI位置有关，所以在此处设计的测试用例把UI作为一个条件。该测试用例如表所示。

课程表课程设置用例



其中课程相关的部分界面UI的显示又与用户有关，其测试用例如表所示。

课程表课程设置用例



### 5.1.4 上课静音

上课静音功能的测试用例及测试结果如表所示。

上课静音测试用例



### 5.1.5 上课提醒

上课提醒功能，分为每日提醒和每周提醒，它们的测试用例及测试结果如表所示。

每日提醒测试用例



### 5.1.6 课程表分享

课程表分享功能的测试用例及测试结果如表所示。

课程表分享功能测试用例



6总结

6.1 总结

经过参加这次比赛，我们学到了很多的东西，也得到了很多的锻炼。在不断修改Bug的过程中，不断的学习新知识。在智能手机普及兴盛的时代背景下，作为主流的Android智能手机，其应用前景非常美好。

本作品设计基于Android的Schedule APP，从市面上缺少符合教师使用习惯的课程表APP出发，来设计一款不仅符合学生用户使用习惯的，也符合教师用户使用习惯的课程表APP。