**课程编号：0006991**



**BEIJING UNIVERSITY OF TECHNOLOGY**

**Android应用程序设计**

|  |  |
| --- | --- |
| **题目名称：** | Android 课程表应用设计 |
| **学 院：** | 信息学部 |
| **报 告 人：** | 15071005 王涵 |
|  | 15071018 刘云 |
| **报告日期：** | 2017年 5 月 31日 |

|  |  |  |
| --- | --- | --- |
| **学期:** 2016–2017学年第2学期 | | **成绩：** |
| **评语：** | | |
| **教师签字：** | **评阅日期：** | |

目录

[一、 总体设计 2](#_Toc483475303)

[二、 详细设计，关键代码提供流程图。 2](#_Toc483475304)

[三、 运行界面 5](#_Toc483475305)

[四、 收获体会 6](#_Toc483475306)

### 总体设计

本应用是一个课程表应用。支持从北京工业大学教务管理系统或本地文件导入个人课程表。主界面显示本周课表和今日课表。支持课程的增加、编辑、删除操作。学期内周数可自动匹配。

使用Java语言编写，使用了Android Support Library提供的大量组件搭建界面。使用Jsoup处理课表，将网页转换成数据库的记录，使用RecyclerView显示课程列表，使用RxJava、OkHttp、Retrofit进行网络请求，使用OrmLite进行数据库操作。使用ColorPicker给课程编辑界面加入了选择颜色的功能。

### 详细设计

1. MainActivity

主界面由Toolbar、TabLayout、ViewPager组成。TodayCoursesFragment和WeekCoursesFragment显示在ViewPager中。MainActivity中实现了一个简单的FragmentPagerAdapter用于管理两个Fragment。Toolbar上的Spinner用于显示和选择周数。

List<String> ***weeks*** = RefStreams.*iterate*(1, ***i*** -> ***i*** + 1)  
 .limit(20)  
 .map(***i*** -> String.*format*(getResources().getString(R.string.***week\_num***), ***i***))  
 .collect(Collectors.*toList*());  
ArrayAdapter<String> ***adapter*** = **new** ArrayAdapter<>(**this**, R.layout.***spinner\_item***, ***weeks***);  
***adapter***.setDropDownViewResource(android.R.layout.***simple\_spinner\_dropdown\_item***);  
**spinner**.setAdapter(***adapter***);  
**spinner**.setSelection(**preferences**.getInt(Constant.***CURRENT\_WEEK***, 0));  
**spinner**.setOnItemSelectedListener(**new** AdapterView.OnItemSelectedListener() {  
 @Override  
 **public void** onItemSelected(AdapterView<?> ***parent***, View ***view***, **int *position***, **long *id***) {  
 **preferences**.edit().putInt(Constant.***CURRENT\_WEEK***, ***position***).apply();  
 }  
});

TodayCoursesFragment

显示今日课表。使用Java 8 Streams进行筛选操作。筛选出星期是今日星期，且在起始周和终止周范围内的课程，并加入到RecyclerView的Adapter中，显示出来。

**int *today*** = Calendar.*getInstance*().get(Calendar.***DAY\_OF\_WEEK***);  
**adapter**.**items**.clear();  
**manager** = **new** CourseManager(getContext());  
List<Course> ***list*** = **manager**.getAllCourse();  
**if** (***list*** != **null**) {  
 StreamSupport.*stream*(***list***)  
 .flatMap(***course*** -> StreamSupport.*stream*(***course***.getSpacetimes()))  
 .filter(***spacetime*** -> ***spacetime***.getWeekday() == ***today***)

.filter(***spacetime*** -> ***spacetime***.getStartWeek() <= ***currentWeek*** && ***spacetime***.getEndWeek() >= ***currentWeek***)

.forEach(***spacetime*** -> **adapter**.**items**.add(***spacetime***.getCourse()));  
}  
**adapter**.notifyDataSetChanged();

WeekCoursesFragment

显示本周课表。筛选操作同上。对屏幕尺寸进行15等分，将表头和内容的空间按一定比例进行分配（表头占1份，每个课程占2份，一周7天，一共15份），使课程表刚好占满屏幕。

**private void** showCourses() {  
 **int *currentWeek*** = **preferences**.getInt(Constant.***CURRENT\_WEEK***, -1);  
 List<Course> ***list*** = **manager**.getAllCourse();  
 **if** (***list*** != **null**) {  
 StreamSupport.*stream*(***list***)  
 .flatMap(***course*** -> StreamSupport.*stream*(***course***.getSpacetimes()))  
 .filter(***spacetime*** -> ***spacetime***.getStartWeek() <= ***currentWeek*** && ***spacetime***.getEndWeek() >= ***currentWeek***)  
 .forEach(**this**::addCourseToView);  
 }  
}

**private void** addCourseToView(Spacetime ***spacetime***) {  
 Course ***course*** = ***spacetime***.getCourse();  
 LinearLayout ***v*** = (LinearLayout) LayoutInflater.*from*(getContext()).inflate(R.layout.***item\_course\_view***, **coursesLayout**, **false**);  
 TextView textView = ButterKnife.*findById*(***v***, R.id.***course\_name***);  
 textView.setText(***course***.getName() + **"\n"** + ***spacetime***.getLocation());  
 ***v***.setBackgroundColor(***course***.getColor());  
 **int *weekday*** = ***spacetime***.getWeekday() == 1 ? 7 : ***spacetime***.getWeekday() - 1;  
 ***v***.setX((***weekday*** - 1) \* **baseWidth**);  
 ***v***.setY((***spacetime***.getStartTime() - 1) \* **baseHeight**);  
 **coursesLayout**.addView(***v***);  
 ***v***.getLayoutParams().**height** = (***spacetime***.getEndTime() - ***spacetime***.getStartTime() + 1) \* **baseHeight**;  
 ***v***.getLayoutParams().**width** = **baseWidth**;  
 }

1. CourseDetailActivity

查看和编辑课程详情。同一课程的不同时间的添加和删除。在创建Activity时在Intent中传入flag以标志是否为新课程。新课程时执行创建操作，非新课程时执行修改操作。

**if** (**flag**) {  
 **course** = **new** Course();  
 **manager**.insertCourse(**course**);  
} **else** {  
 **course** = **manager**.getCourse(**courseId**);  
}  
loadSpacetime();  
**toolbar**.setOnMenuItemClickListener(***menuItem*** -> {  
 **switch** (***menuItem***.getItemId()) {  
 **case** R.id.***action\_confirm***:  
 **course**.setName(**nameView**.getText().toString());  
 **course**.setTeacher(**teacherView**.getText().toString());  
 **manager**.updateCourse(**course**);  
 **for** (Spacetime ***st*** : **adapter**.**items**) **manager**.updateSpacetime(***st***);  
 finish();  
 **break**;  
 **case** R.id.***action\_delete\_forever***:  
 showDeleteConfirm();  
 **break**;  
 **case** R.id.***action\_add\_spacetime***:  
 Spacetime ***spacetime*** = **new** Spacetime();  
 ***spacetime***.setCourse(**course**);  
 **manager**.insertSpacetime(***spacetime***);  
 **adapter**.**items**.add(***spacetime***);  
 **adapter**.notifyDataSetChanged();  
 **break**;  
 }  
 **return true**;  
});

1. CourseManageActivity

课程的导入、添加、清空。使用RecyclerView显示所有课程。从网络导入，自动登录教务系统，下载本学期课表。使用RxJava和Retrofit进行网络请求，简化了代码。当教务系统无法访问时，可从文件导入，读取用户保存好的HTML文件，导入到数据库。

private void loadCourses() {

adapter.items.clear();

manager = new CourseManager(this);

List<Course> list = manager.getAllCourse();

if (list != null)

adapter.items.addAll(manager.getAllCourse());

adapter.notifyDataSetChanged();

}

**final** ZfService ***zfService*** = ZfRetrofit.*getZfService*();  
 **disposable** = ***zfService***.login(***loginMap***)  
 .flatMap(***s*** -> {  
 **return *zfService***.getSchedule(***scheduleMap***);  
 })  
 .subscribeOn(Schedulers.*io*())  
 .observeOn(AndroidSchedulers.*mainThread*())  
 .subscribe(***text*** -> {  
 ***manager***.clearCourse();  
 CourseParser.*parse*(***text***, ***manager***);  
 loadCourses();  
 },  
 ***throwable*** -> {  
 dismissProgressDialog();  
 },  
 () -> dismissProgressDialog(),  
 ***dis*** -> **progressDialog** = ProgressDialog.*show*(CourseManageActivity.**this**, **"Schedule"**, **"Now loading..."**, **true**, **true**, ***dialogInterface*** -> ***dis***.dispose()));  
}

1. SettingsActivity

设置教务管理系统的用户名和密码。另外，在这里可以启用和禁用启动画面，切换应用主题，查看当前版本。使用PreferenceFragment实现。

*bindPreferenceSummaryToValue*(**userNamePreference**);  
*bindPreferenceSummaryToValue*(**psdNamePreference**);  
**splashScreenPreference**.setOnPreferenceChangeListener((***preference***, ***boo***) -> {  
 PackageHelper.changeMain(getContext(), (**boolean**) ***boo***);  
 **return true**;  
});  
**themePreference**.setOnPreferenceClickListener(***preference*** -> {  
 getContext().startActivity(**new** Intent(getContext(), ThemeActivity.**class**));  
 **return true**;  
});  
**sourcePreference**.setOnPreferenceClickListener( ***preference*** -> {  
 startActivity(**new** Intent(Intent.***ACTION\_VIEW***, Uri.*parse*(getResources().getString(R.string.***source\_code\_url***))));  
 **return true**;  
});

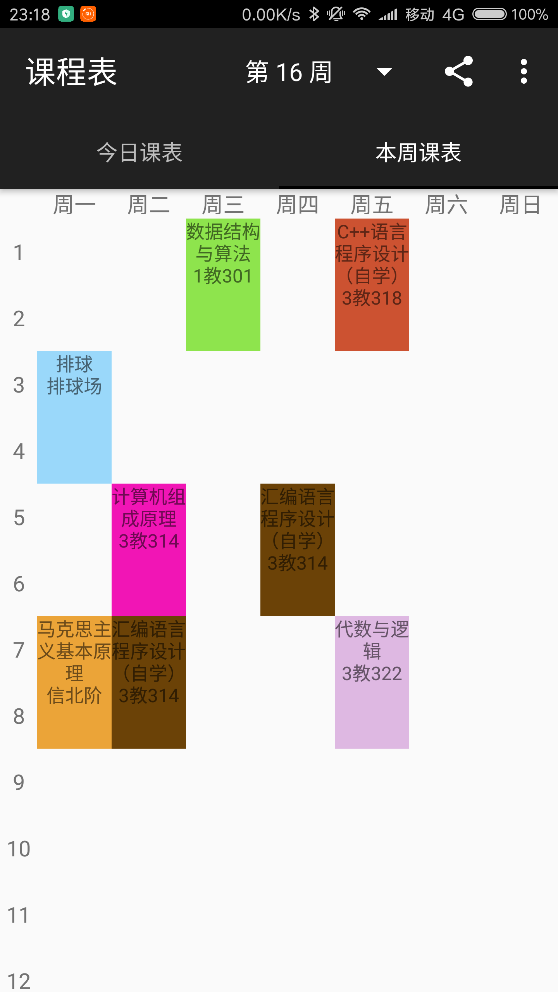
1. CourseParser

解析教务管理网页，获得课程数据，添加到数据库中。流程图如下。



### 运行界面

1.今日课表/本周课表

2.新建课程/编辑课程

3.管理课程/设置

### 收获体会

通过Android应用程序设计这门课，加深了对Android编程的理解，熟练了Android编程，学习了大量新的组件与API，拓展了Android编程的视野，让以后的编程能够更加顺利。这门课上课讲的知识点并不深入，只是Android的编程基础，大部分知识需要我们自行学习。通过这门课，很大程度的锻炼了我们的自学能力，这是作为一个程序员最重要的能力之一。

通过详细设计这个课程表App，让我们学会了做一个成型App需要的详细步骤。通过使用一些Java 8的新特性，如lambda表达式，提高了代码的可读性。通过使用一些第三方库，改善了源代码的结构，提高了编程的效率。动态地向界面中添加元素，以显示出本周课表，是这个设计最大的难点。我们研究了大量网络上的源代码，最终搞清了原理，解决了问题。在15周的时候，教务系统关闭了“学生个人课表”的访问，我们只好临时加入了从文件导入的功能。此外，通过解决了一个个难题，修好各种各样奇怪的闪退bug，让我们学会了如何写一段更好的、更清晰的代码，让程序更加稳定的运行，极大的提高了我们的编程水平。通过对GitHub的运用，让我们熟悉了一些Git的知识，为今后的团队协作提供了很大帮助。并且，在最后程序做出来的时候，给予了我们极大的信心与喜悦。