



**课 程 设 计**

课程名称 **物联网软件开发技术**

题目名称\_\_\_**\_ 广工新闻软件 \_**\_\_\_\_\_

学生学院 自动化学院

专业班级\_\_\_\_\_\_物联网2班\_\_\_\_\_\_\_\_

姓名学号\_\_\_吴玉波 3113001629\_\_\_\_

合作组员\_\_\_吴华艺 、曾汝婷 \_\_\_\_

\_\_\_\_王今、郑润媛 \_\_\_\_

指导教师 高明琴

2016 年 6 月 30 日

# 需求分析

如今走在路上，你随时都能看到埋头看手机或者平板的人们。不仅感叹移动互联网时代真的到来了。方便快捷的移动终端，可以让我们很好的利用自己的碎片时间。很多朋友都选择利用自己的这些时间来获取最新的资讯，这一切随着移动互联的到来，变得触手可及。人们看新闻更愿意通过移动设备来浏览，这样更方便，高效。

但是广东工业大学新闻只有PC端的网页版，并没有支持移动端的页面，就更不用说移动APP了，我们要看广东新闻时，就会很不方便。因此，开发一个广工新闻APP是必要的。

实现一个Android平台下的APP，该APP能够连接到广东工业大学新闻栏目，获取新闻内容，并保存到手机的数据库中，给用户查看。具体功能如下：

A． 每隔一段时间（可由用户设定），自动连接广东工业大学新闻栏目，获取最近发布的新闻，保存在手机上；

B． 手机上的新闻应有以下字段：标题、内容、发布单位和发布日期；

C． 已经下载到手机上的新闻不再次获取；

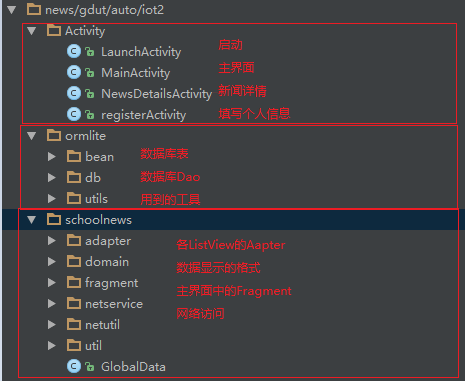
D． 在APP中对已下载的新闻进行查看，可按发布单位查看、按发布日期查看、按关键字查找

E． 要求有简洁、清晰、美观、易于操作的界面。

# 软件设计

APP的系统设计基于MVC思想，可分为三大层：用户界面，控制器，业务模型（数据库）。

基于上面提出的框架，可以得到代码结构如图



# 数据库设计

1. 软件主要考虑到用户，新闻之间的关系，而用户的群体是学生或老师，新闻则是学习的新闻，因此学校，新闻，学生，老师是数据设计中要考虑的因素。考虑到软件的可扩展性，在数据库设计中，添加了一些不必要的字段。为了可以更好地为用户提供服务，我们需要实行访问限制，因此需要一个用户表来对用户的类型进行分类，同时考虑到可以要对软件的功能扩展，可能以后不只是限广东工业大学的师生使用，可能会扩展其他学校的模块，再通过访问限制，这样就可能让各高校的师生都能过这个APP来访问所在学校的新闻。
2. 根据以上的考虑，数据库设计了7个表，分别为：用户、学校、学院、专业、新闻信息、收藏表和发布单位。
3. 数据库E-R图



1. 由E-R图转化为关系模型可得到7个关系模型如下：
   1. User(id, name, sex, professional, school\_id, College\_id, major\_id)
   2. School(id, name, url)
   3. College(id, name, url, school\_id)
   4. Major(id, name, keys)
   5. NewsInform(id, title, type, src\_url, img\_url, content,time, is\_valid, school\_id, department\_id)
   6. Collection(id, user\_id, newsInform\_id, time)
   7. Department（id, name, school\_id）

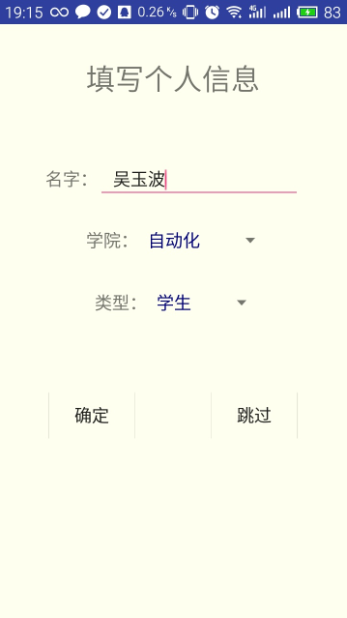
# 软件实现

## 界面设计

* 界面跳转设计流程



* 各界面显示：
  + 分别为启动界面、信息填写界面、新闻列表界面、收藏夹界面、个人信息界面和新闻详情界面。







## ORMLite

1. 在项目中我们要使用数据库，为了提高开发效率，当然少不了数据库ORM框架了，这里使用ORMLite对数库进行操作。
2. 去ORMLite官网下载jar包下载并添加第三方库
   1. ormlite-android-4.48.jar 和 ormlite-core-4.48.jar
3. 配置数据库的表类

OrmLite 要通过类来映射数据中的每一个表，能关键字@DatabaseTable(tableName = "表名")进行映射，而表中的每一个字段通过关键字@DatabaseField(columnName = "字段名")进行映射。

* 1. 用户

@DatabaseTable(tableName = "tb\_user")  
public class User   
{  
 @DatabaseField(generatedId = true)  
 private int id;  
 @DatabaseField(columnName = "name")  
 private String name;  
 @DatabaseField(columnName = "sex")  
 private String sex;  
 @DatabaseField(columnName = "professional")  
 private String professional;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "school\_id", foreignAutoRefresh = true)  
 private School school;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "college\_id", foreignAutoRefresh = true)  
 private College college;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "major\_id", foreignAutoRefresh = true)  
 private Major major;

* 1. 学校

@DatabaseTable(tableName = "School")  
public class School {  
 @DatabaseField(generatedId = true)  
 private int id;  
 @DatabaseField(columnName = "name")  
 private String name;  
 @DatabaseField(columnName = "url")  
 private String url;

* 1. 学院

@DatabaseTable(tableName = "College")  
public class College {  
 @DatabaseField(generatedId = true)  
 private int id;  
 @DatabaseField(columnName = "name")  
 private String name;  
 @DatabaseField(columnName = "url")  
 private String url;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "school\_id", foreignAutoRefresh = true)  
 private School school;

* 1. 专业

@DatabaseTable(tableName = "Major")  
public class Major {  
 @DatabaseField(generatedId = true)  
 private int id;  
 @DatabaseField(columnName = "name")  
 private String name;  
 @DatabaseField(columnName = "keys")  
 private String keys;

* 1. 新闻信息

@DatabaseTable(tableName = "NewsInform")  
public class NewsInform {  
 @DatabaseField(generatedId = true)  
 private int id;  
 @DatabaseField(columnName = "title")  
 private String title;  
 @DatabaseField(columnName = "type")  
 private String type;  
 @DatabaseField(columnName = "src\_url")  
 private String src\_url;  
 @DatabaseField(columnName = "img\_url")  
 private String img\_url;  
 @DatabaseField(columnName = "content")  
 private String content;  
 @DatabaseField(columnName = "is\_valid")  
 private Boolean is\_valid;  
 @DatabaseField(columnName = "time")  
 private String time;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "school\_id", foreignAutoRefresh = true)  
 private School school;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "department\_id", foreignAutoRefresh = true)  
 private Department department;

* 1. 收藏

@DatabaseTable(tableName = "Collection")  
public class Collection {  
 @DatabaseField(generatedId = true)  
 private int id;  
 @DatabaseField(columnName = "name")  
 private String name;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "user\_id", foreignAutoRefresh = true)  
 private User user;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "newsInform\_id", foreignAutoRefresh = true)  
 private NewsInform newsInform;  
 @DatabaseField(columnName = "time")  
 private String time;

* 1. 发布单位

@DatabaseTable(tableName = "Department")  
public class Department {  
 @DatabaseField(generatedId = true)  
 private int id;  
 @DatabaseField(columnName = "name")  
 private String name;  
 @DatabaseField(canBeNull = true, foreign = true, columnName = "school\_id", foreignAutoRefresh = true)  
 private School school;

1. DatabaseHelper
   1. 原生的数据库操作，需要继承SQLiteOpenHelper。
   2. 为了保证app中只存在一个SQLite Connection，整个DatabaseHelper使用单例只对外公布出一个对象，
   3. 继承OrmLiteSqliteOpenHelper，其实就是间接继承了SQLiteOpenHelper

然后需要实现两个方法：

1、onCreate(SQLiteDatabase database,ConnectionSource connectionSource)

创建表，我们直接使用ormlite提供的TableUtils.createTable(connectionSource, User.class);进行创建~

2、onUpgrade(SQLiteDatabase database, ConnectionSource connectionSource, int oldVersion, int newVersion)

更新表，使用ormlite提供的

TableUtils.dropTable(connectionSource, User.class, true);进行删除操作~

1. 编写DAO类
   1. 对每个Bean创建一个XXXDao来处理当前Bean的数据库操作，当然真正去和数据库打交道的对象，通过上面代码中的getDao（T t）进行获取
   2. getDao为一个泛型方法，会根据传入Class对象进行创建Dao，并且使用一个Map来保持所有的Dao对象，只有第一次调用时才会去调用底层的getDao()。
   3. 条件查询QueryBuilder的使用
      1. 简单的where等于

XXXDaoOpe.queryBuilder().where().eq("user\_id", userId).query();直接返回XXX的列表

* 1. 其他操作updateBuilder、deleteBuilder

使用queryBuilder是希望执行完成查询直接返回List<Bean>集合；

对于updateBuilder和deleteBuilder并不关注返回值，直接使用

* 1. 新闻信息的Dao（NewsInformDao）实现

关键操作

* + 1. 获取数据中的新闻列表（包括排序，筛选等功能）。

public List<NewsInform> getNewsInformList(NewsInformOptions opt){  
  
 if(!opt.isFilter){//select all  
 return getListByNoFilter(opt);  
 }  
 //select with a filter  
 if (opt.filterindex == NewsInformOptions.*DEPARTMENT*) { //filter is department  
 return getListByFilterDepartment(opt);  
 }  
 if (opt.filterindex == NewsInformOptions.*ID* ) {  
 return getListByFilterIdOrTime(opt);  
 }  
 //filter is ID or TIME  
 return getListByFilterNormal(opt);  
}

* + 1. 查找全部新闻

private List<NewsInform> getListByNoFilter(NewsInformOptions opt){  
 List<NewsInform> newsInformList = null;  
 try {  
 newsInformList = newsInformDaoOpe.queryBuilder().orderBy(NewsInformOptions.*getColumnName*(opt.orderindex),  
 opt.ordertype).where().eq("school\_id", opt.school).and().like("title",opt.seachString).query();  
 }catch (SQLException e){  
  
 }  
 return newsInformList;  
}

* + 1. 按发布单位筛选新闻

private List<NewsInform> getListByFilterDepartment(NewsInformOptions opt){  
 List<NewsInform> newsInformList = null;  
 try {  
 newsInformList = newsInformDaoOpe.queryBuilder().orderBy(NewsInformOptions.*getColumnName*(opt.orderindex),  
 opt.ordertype).where().eq(NewsInformOptions.*getColumnName*(opt.filterindex),  
 opt.department).and().eq("school\_id",  
 opt.school).and().like("title",opt.seachString).query();  
 }catch (SQLException e){  
  
 }  
 return newsInformList;  
}

* + 1. 按时间等其他条件筛选新闻

private List<NewsInform> getListByFilterNormal(NewsInformOptions opt){  
 List<NewsInform> newsInformList = null;  
 try {  
 newsInformList = newsInformDaoOpe.queryBuilder().orderBy(NewsInformOptions.*getColumnName*(opt.orderindex),  
 opt.ordertype).where().eq(NewsInformOptions.*getColumnName*(opt.filterindex),  
 opt.getColumnStringValue(opt.filterindex)).and().eq("school\_id",  
 opt.school).and().like("title",opt.seachString).query();  
 }catch (SQLException e){  
  
 }  
 return newsInformList;  
}

* + 1. 其中NewsInformOptions类用于设置新闻查找的操作（包括排序，筛选等）。

1. 编写操作接口类
   1. 为了屏蔽Dao的细节，只对外提供DataBaseOpt类对数据库进行操作。
   2. DataBaseOpt类包含了所有Bean的Dao实例，可以通过调用Dao的方法对数据库进行操作。

## 新闻抓取

* 1. 添加第三方库Jsoup：<http://jsoup.org/>
  2. 通过Url获得到对应的Document对象

public static Document getDocumentByUrl(String url){  
 try {  
 Document document = Jsoup.*connect*(url).userAgent(*UESRAGENT\_PHONE*).get();  
 return document;  
 } catch (IOException e) {  
 e.printStackTrace();  
 }  
 return null;  
}

* 1. 获取广工新闻网页并解析
     1. 广东新闻的Url：<http://gdutnews.gdut.edu.cn/>
     2. 通过GetNewServive类抓取最近几页的新闻，解析后存到数据库中。
     3. 其中保存新闻的标题，摘要，发布时期，发布单位，新闻Url等信息到数据库中。
     4. 抓取过程中发现数据中已存在的新闻就会丢弃。
     5. 关键代码如下（按页读取网络中的新闻，并存到数据库）：

public Boolean getNewsByPage( int page){  
 String[] strtemp;  
 try{  
 Document document = *getDocumentByUrl*(*newsUrl*+*HomeJsp*+"&PAGENUM="+page+"&urltype=tree.TreeTempUrl&wbtreeid=1013");  
 if(document == null) return false;  
 Elements ul = document.select(".listtextul");  
 Document divcontions = Jsoup.*parse*(ul.toString());  
 Elements element = divcontions.getElementsByTag("li");  
 for(Element links : element)  
 {  
 String title = links.getElementsByTag("a").text();  
 String link = links.select("a").attr("href").replace("/", "").trim();  
 String url = *school*.geturl()+*ContextJsp*+link;  
 String date = links.getElementsByTag("span").text();  
 if(title == null || link==null || url==null || date==null){  
 System.*out*.println("---------网络故障----------------");  
 continue;  
 }  
 if(title.isEmpty() || link.isEmpty() || url.isEmpty() || date.isEmpty()){  
 System.*out*.println("----------------网络故障------");  
 continue;  
 }  
 strtemp = date.split("\\[");  
 if(strtemp.length == 2){  
 date = strtemp[1].split("\\]")[0];  
 }  
 if(!dbOpt.*isNewExistByUrl*(url)){  
 Document cdocument = *getDocumentByUrl*(url);  
 String detail = cdocument.select("#ctl00\_ContentPlaceHolder1\_tbxIntro").text();  
 String depatment = cdocument.select("#ctl00\_ContentPlaceHolder1\_tbxCopyFrom").text();  
 strtemp = depatment.split("：");  
 if(strtemp.length == 2){  
 depatment = strtemp[1];  
 }  
 Department dp = dbOpt.*getDepartmentByNameOrCreate*(depatment, *school*);  
 dbOpt.*addNewsInform*(title,"normal", url, "..", detail, true, date, *school*, dp);  
 }  
 }  
 }catch(Exception e){  
 FileUtil.*addFile*(e.getStackTrace().toString());  
 e.printStackTrace();  
 return false;  
 }  
 return true;  
 }

## 新闻列表显示

1. ListView显示



1. Adapter提供数据
   1. NewsListViewAdapter继承BaseAdapter，为ListView提供数据。
   2. 关键代码getView()

@Override  
public View getView(int position, View convertView, ViewGroup parent) {  
 News news = newss.get(position);  
 if(convertView == null){  
 convertView = View.*inflate*(context,R.layout.*listview\_fragment\_news*, null);  
 }  
 ((TextView)convertView.findViewById(R.id.*fragment\_news\_listview\_title*)).setText(news.getTitle());  
 ((TextView)convertView.findViewById(R.id.*fragment\_news\_listview\_content*)).setText(news.getSource());  
 ((TextView)convertView.findViewById(R.id.*fragment\_news\_listview\_time*)).setText(news.getDate());  
 ((TextView)convertView.findViewById(R.id.*fragment\_news\_listview\_department*)).setText(news.getDepartment());  
  
 return convertView;  
}

1. 异步操作从数据库中获取新闻列表
   1. AsyncTask

*/\*\*  
 \* 异步获取新闻列表集合  
 \** ***@author*** *linpeng123l  
 \*  
 \*/*public class MyAsyncTaskGetNews extends AsyncTask<Integer, String, List<News>>{  
 @Override  
 protected List<News> doInBackground(Integer... pages) {  
 List<News> newss = getNews(pages[0]);  
 return newss;  
 }  
 @Override  
 protected void onPostExecute(List<News> newss) {  
 if(newss.size()>0){  
 adapter.addNews(newss);  
 adapter.notifyDataSetChanged();  
 page++;  
 }  
 isLoading = false;  
 }  
}

* 1. 获取新闻列表的关键代码

private List<News> getNews(int showpage){  
 int i, skip, take;  
 List<News> newss = new ArrayList<News>();  
 List<NewsInform> informList;  
 skip = showpage\*5;  
 take = 5;  
 i = 0;  
 informList = dbOpt.*getNewsInformList*(opt);  
 if(informList == null || informList.isEmpty()) {  
 newss.add(new News("没有新闻，请连接网络重启", "",  
 "", "....","今天", "..."));  
 return newss;  
 }  
 for(NewsInform newsInform : informList){  
 if(i++ < skip && showpage!= -1) {  
 continue;  
 }  
 String date = newsInform.gettime();  
 String department = newsInform.getdepartment().getName();  
 String detail = newsInform.getcontent();  
 if(detail.length() > GlobalData.*newsContextShow*){  
 detail = detail.substring(0,GlobalData.*newsContextShow*-6);  
 detail += "......";  
 }  
 newss.add(new News(newsInform.gettitle(), newsInform.getsrc\_url(),  
 newsInform.getimg\_url(), detail,date, department));  
 if(i >= take+skip && showpage != -1){  
 break;  
 }  
 }  
 setFooterView(informList.size());  
 return newss;  
}

## 新闻详情

1. 获取网页
   1. 由于通过Url获取的网页是PC端的网页，需要转换成适应手机显示。
      1. 关键代码如下：

private void getNewDetail(){  
 int tryCount = 0;  
 Document document = null;  
 while(document == null){  
 tryCount++;  
 document = *getDocumentByUrl*(url);  
 if(++tryCount >= 3){  
 break;  
 }  
 }  
 if(document == null){  
 noNetWork();  
 return;  
 }  
 Elements title = document.select(".title");  
 Elements time = document.select("#ctl00\_ContentPlaceHolder1\_tbxUpdateTime");  
 Elements from = document.select("#ctl00\_ContentPlaceHolder1\_tbxCopyFrom");  
 Elements autho = document.select("#ctl00\_ContentPlaceHolder1\_tbxEditor");  
 Elements content = document.select("#contentText");  
 Elements jpgs = content.select("img[src$=.jpg]"); // 所有 jpg 的图片  
 String imgUrl = null;  
 for (Element links : jpgs) {  
 imgUrl = GdutUrl + links.attr("src").toString();  
 links.attr("src", imgUrl);  
 links.attr("width", "100%");  
 }  
 Elements pngs = content.select("img[src$=.png]"); // 所有 jpg 的图片  
 for (Element links : pngs) {  
 imgUrl = GdutUrl + links.attr("src").toString();  
 links.attr("src", imgUrl);  
 links.attr("width", "100%");  
 }  
 String fin =  
 title.toString() + //标题  
 "时间：" + time.toString() + "<br>"//时间  
 + from.toString() + "<br>"//来源  
 + autho.toString() + "<br>"//作者  
 + content.toString();//正文  
 showNewDetail(fin, 1);  
}

* 1. 没有网络的情况下只显示数据库中的新闻概要。
     1. 关键代码如下：

private void noNetWork(){  
 String text = "<p>=\_=咦，你的网络不给力，请检查你的网络设置。</p>"  
 +"<h1>"+newsInform.gettitle()+"</h1>"  
 + "<p>来源："+newsInform.getdepartment().getName()+"</p>"  
 + "<p>时间："+newsInform.gettime()+"</p>"  
 +"<p>"+newsInform.getcontent()+"</p>";  
 showNewDetail(text, 0);  
}

1. WebView显示网页
   1. 通过Handler来更新显示，关键代码如下：

class myHandler extends Handler {  
 @Override  
 public void handleMessage(Message msg) {  
 if(msg.what == 0){  
 toastShow("你的网络不给力，请检查你的网络设置");  
 }  
  
 detailNews.loadData(msg.obj.toString(),"text/html; charset=UTF-8;",null);  
 System.*out*.println(msg.obj.toString());  
 isloading.setVisibility(View.*INVISIBLE*);  
 super.handleMessage(msg);  
 }  
}

## 新闻查找、排序与过滤

* 1. Search 控件
     1. 用于关键字输入。

<SearchView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/searchView"  
 android:layout\_weight="0.99"  
 android:imeOptions="actionSearch" />

@Override  
public boolean onQueryTextChange(String newText) {  
 opt.seachString = "%"+newText+"%";  
  
 if (newText.isEmpty()) {  
 NewsUpdate();  
 }  
  
 return false;  
}

* + 1. 关键字查找，通过数据库存的模糊查找，
       1. queryBuilder().like(“title”, “%关键词%”).

newsInformList = newsInformDaoOpe.queryBuilder().like("title",opt.seachString).query();

* 1. Spinner控件
     1. 提供筛选选项和排序选项。

<Spinner  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/spinner\_order" />

* + 1. 设置筛选选项，从数据中查找所有选项（时间列表/发布单位列表）对spinner进行初始化。

private void sourceFilter(){  
 List<Department> departmentList = dbOpt.*getDepartmentAllBySchool*(GlobalData.*currentSchool*);  
 if(departmentList == null || departmentList.isEmpty()){  
 return;  
 }  
 List<News> newsList = getNews(-1);  
  
 for(Department department:departmentList){  
 if(department.getName().isEmpty()) continue;  
 apStrs.add(department.getName());  
 }  
 String date = "";  
 for(News news:newsList){  
 if( date.equals(news.getDate()) || news.getDate().isEmpty()) continue;  
 date = news.getDate();  
 daStrs.add(date);  
 }  
  
 apAdapter.setNotifyOnChange(true);  
 daAdapter.setNotifyOnChange(true);  
}

* + 1. 通过NewsInformOptions设置选项。

if(str == "发布单位"){  
 opt.isFilter = false;  
}  
else {  
 opt.isFilter = true;  
 opt.filterindex = NewsInformOptions.*DEPARTMENT*;  
 opt.department = dbOpt.*getDepartmentByNameOrCreate*(str, GlobalData.*currentSchool*);  
}

* + 1. queryBuilder().where().eq(“字段名”, “过滤字符串”).

newsInformList = newsInformDaoOpe.queryBuilder().where().eq(NewsInformOptions.*getColumnName*(opt.filterindex),opt.getColumnIntValue(opt.filterindex)).query();

1. 设置排序选项，通过NewsInformOptions设置选项。

if(str == "默认排序"){  
 opt.orderindex = NewsInformOptions.*TIME*;  
 opt.ordertype = false;  
}  
else if(str == "日期降序"){  
 opt.orderindex = NewsInformOptions.*TIME*;  
 opt.ordertype = false;  
}  
else if(str == "日期升序"){  
 opt.orderindex = NewsInformOptions.*TIME*;  
 opt.ordertype = true;  
}  
else if(str == "发布单位"){  
 opt.orderindex = NewsInformOptions.*DEPARTMENT*;  
 opt.ordertype = true;  
}

1. queryBuilder().orderBy(“字段名”,ture/false)(升序/降序).

newsInformList = newsInformDaoOpe.queryBuilder().orderBy(NewsInformOptions.*getColumnName*(opt.orderindex),opt.ordertype)..query();

## 收藏

1. 新闻收藏
   1. 浏览新闻详情前，选检查数据收藏表是否已经对此新闻收藏了。

private boolean checkColletiion(){  
 List<Collection> collectionList = dbOpt.*getColletionByNewsInfom*(dbOpt.*getNewsInformBySrcUrl*(detailUrl));  
 if(collectionList == null || collectionList.isEmpty()){  
 return false;  
 }  
 return true;  
}

* 1. 如果已经收藏，则显示已收藏图标D:\wuyubo_life\Android\schoolNews-6-27\SchoolNews-6-27\app\src\main\res\drawable\icon_collection_press.png，否则显示未收藏图标D:\wuyubo_life\Android\schoolNews-6-27\SchoolNews-6-27\app\src\main\res\drawable\icon_collection_normal.png。
  2. 如果未收藏，可以通过点击D:\wuyubo_life\Android\schoolNews-6-27\SchoolNews-6-27\app\src\main\res\drawable\icon_collection_normal.png收藏当前新闻，。并设置显示已收藏图标D:\wuyubo_life\Android\schoolNews-6-27\SchoolNews-6-27\app\src\main\res\drawable\icon_collection_press.png。
  3. 如果已经收藏可以通过点击D:\wuyubo_life\Android\schoolNews-6-27\SchoolNews-6-27\app\src\main\res\drawable\icon_collection_press.png取消收藏。并设置显示未收藏图标D:\wuyubo_life\Android\schoolNews-6-27\SchoolNews-6-27\app\src\main\res\drawable\icon_collection_normal.png。
  4. 收藏后的新闻可在收藏夹中查看

1. 数据库插入与删除
   1. 添加收藏

public static Boolean addCollection(String name, User user, NewsInform newsInform, String time){  
 Collection collection = new Collection();  
 *collectionDao* = CollectionDao.*getInstence*(*mcontext*);  
 collection.setId(1);  
 collection.setUser(user);  
 collection.setNewsInform(newsInform);  
 collection.settime(time);  
 *collectionDao*.add(collection);  
 return true;  
}

* 1. 取消收藏

DeleteBuilder<Collection, Integer> deleteBuilder = collectionDaoOpe.deleteBuilder();  
deleteBuilder.where().eq("user\_id", user).and().eq("newsInform\_id", newsInform);  
deleteBuilder.delete();

1. 收藏夹显示，查找收藏中的所有新闻，并通过listView显示出来。



## 个人信息

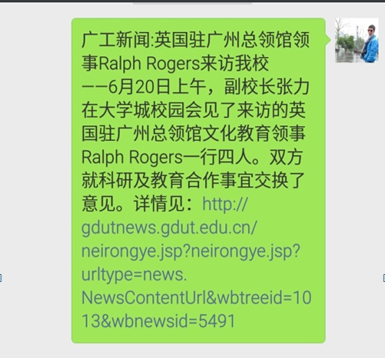
1. 个人信息填写
   1. 为了设置访问限制，需要填写个人的昵称、学院、及专业。
2. 个人信息显示
   1. 查看自己的设置的个人信息。

## 分享

1. 新闻分享
   1. 在新闻详情中点击D:\wuyubo_life\Android\schoolNews-6-27\SchoolNews-6-27\app\src\main\res\drawable\icon_share_normal.png图标即可以分享。

@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
 switch (item.getItemId()) {  
 case R.id.*menu\_share*:  
 Intent intent = new Intent(Intent.*ACTION\_SEND*);  
 intent.setType("text/plain");  
 intent.putExtra(Intent.*EXTRA\_TEXT*, GlobalData.*APP\_NAME*+":"+newsInform.gettitle()  
 +"\r\n——"+newsInform.getcontent()+"详情见："+detailUrl);  
 startActivity(Intent.*createChooser*(intent, getTitle()));  
 break;

* 1. 以字符串方式，通过短信、微信、QQ等向好友分享当前新闻的标题、概要、Url等信息。



# 软件测试

1. 网页抓取出错
   1. 软件启动时开始抓取网页时崩溃
   2. 原因：主线程中不能访问网络
   3. 解决：把网络访问放在service中，另开一个线程进行网页抓取。
2. 图标设计问题
   1. 原来设置的logo太丑。
   2. 重新设计过新的logo
   3. 🡪🡪
3. 界面设计问题
   1. 由杂乱的界面转向小清新风格。

🡪 

1. 没网络网页显示问题
   1. 由于通过Url获取新详情后再显示，但是在网络差的时候，软件会崩溃。
   2. 原因：获取网页失败后操作了doc的空引用。

解决：获取网页失败时只显示数据库中存在的标题、发布时间、发布单位和新闻概要。

# 总结

这不是我的第一个APP，但是第一个合作完成并能完善功能的APP，对于团队合作也并不轻松，可能对大家的技能都不太了解，也没有实现代码统一管理，所以合作起来，沟通是一大难题，分工不能很明确，这也是每次课程设计必须要面对的问题。

对于这次课程设计，自从题目出来之后就开始设计了。可能是对安卓开发有一定的兴趣，在这学期的课设计中，这次是最认真、花最多时间的一次。

这是一次系统的APP开发过程，可以感受得到，安卓开发对设备的要求还是挺高的，对于我这台配置太低的电脑，调试周期真的太长了，每次编译都要等半天，真是需要不少时间。同时，对于java语言，个人觉得比C语言难搞不少，查找BUG的难度大很多，因此安卓开发是需要耐心和时间的。

另外，虽然我个人的发展方向不打算走安卓开发，但对于物联网应用的开发，安卓还是必不可少的。在这次课程设计中，好在能够认真地对待，感觉还是学到了不少东西，对安卓开发有了全新的认识，也有满满的自豪感。感谢老师给我这次机会，现在学到的东西，在以后做开发的过程中肯定能用上的。

# 参考文献

《疯狂Android讲议》第3版 李刚 著

CSDN论坛