

数据库期末项目报告

科目：数据库

教师：郑贵锋

年级	13 级	专业（方向）	移动信息工程
队长学号	13354485	队长姓名	朱琳
队员 1 学号	13354478	队员 1 姓名	周梦辉
队员 2 学号	13354453	队员 2 姓名	张紫琪

一 程序介绍

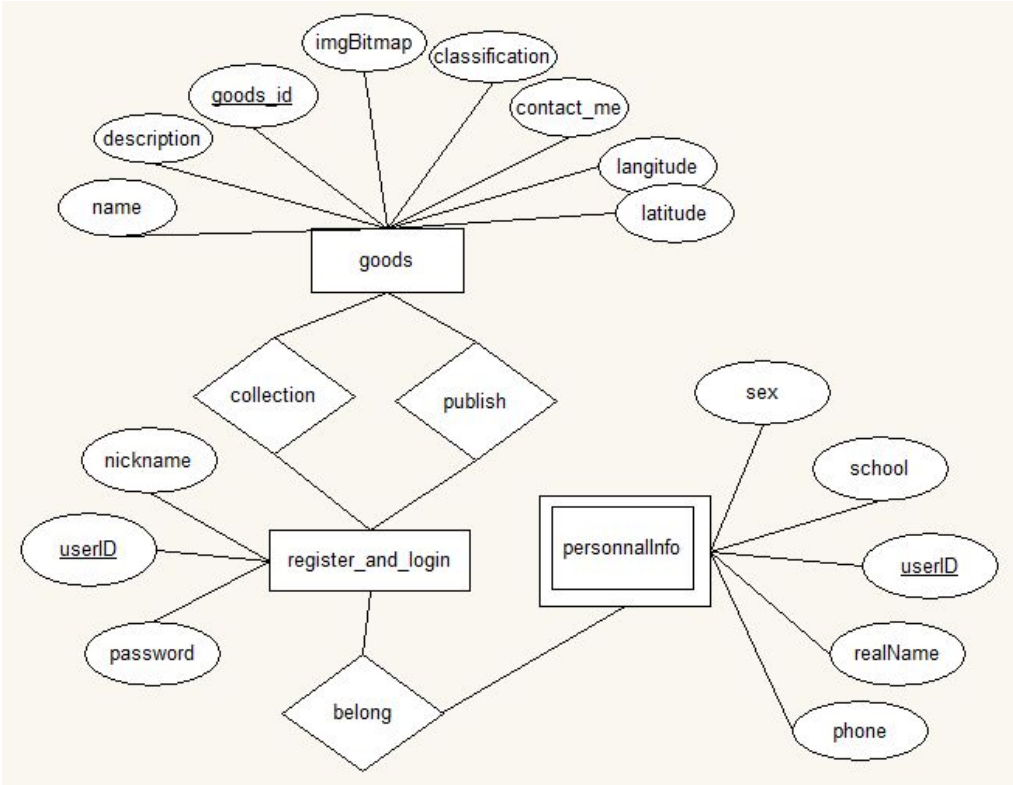
（1）程序简介

这是一款在安卓平台下开发一款基于地理位置的校园闲置物品转让和租赁手机 App。该 App 的主要目标用户为大学生，用户可凭真实身份信息注册和登录，使用昵称在该平台上发布自己的闲置物品信息或征求物品信息，利用定位系统获取周边闲物信息。发布的物品信息需包含物品名称、类别信息，用户可通过文字和图片来对物品进行具体描述，此外还需要填写用户联系方式等等。用户也可通过搜索功能，查找周边的自己需要物品的相关信息。通过该应用，用户可实现校园闲置物品的信息交流和物品交易。

（2）程序功能

- 1.定位功能：**用户登陆后，只要此 App 有定位功能的权限，就会对用户的位置进行定位——用经纬度来评判。因此我们可以根据买家和卖家发布商品的距离来确定两者距离的远近，从而可以自动删选出距离买家比较近的商品。
- 2.发布功能：**用户可以在发布界面发布自己想要出售的商品，包括商品照片，名称，分类，具体描述，联系方式等等。
- 3.查看功能：**用户可以在首页查看自己或者其他发布的二手信息内容，点击内容可以跳转到另一个显示商品具体信息的界面。
- 4.查看删除自己发布的商品的功能：**用户可在我的->我的商品界面查看自己发布过的商品，并且可以进行相应的删改。但是用户没有权限对其他用户发布的内容进行变更。
- 5.收藏功能：**用户在首页界面查看商品时，可以点击每个信息中的收藏按钮将对应商品进行收藏，之后可以在我的->我的收藏界面找到所有收藏的信息。此外，在收藏之后再次点击按钮，即可取消收藏。
- 6.搜索功能：**用户可以在首页的搜索框中添加需要的关键字，即可搜索出具有相关特征的商品。
- 7.摇一摇功能：**定位功能的加强版，用户无需手动搜索，只需摇一摇即可得到附近的人所发布的商品信息。
- 8.登录注册功能：**基础功能，用户凭借账号密码进行注册或者登录。

二 ER 图



三 所涉及到的数据库表

1.register_and_login

表名	register_and_login		
属性	数据类型(精度)	空（非空）	含义
userID	varchar(40)	非空	用户 ID,主键
password	varchar(40)	可空	密码
nick_name	varchar(50)	可空	昵称

【说明】 register_and_login 表格主要保存用户登录注册信息

2. goods

表名	goods		
属性	数据类型(精度)	空（非空）	含义
goods_id	Int(11)	非空	商品 id，主键，从 0 自增
imgURL	varchar(5000)	可空	图片的 bitmap
name	varchar(100)	可空	商品名字
description	varchar(100)	可空	详细描述
classifi	varchar(50)	可空	商品分类
contact_me	varchar(50)	可空	发布者联系方式
userID	varchar(40)	可空	用户 ID(表一的外键)
latitude	double	可空	用户发布商品时纬度
longitude	double	可空	用户发布商品时经度

【说明】goods 表格主要保存商品信息，以及用户发布商品时的位置信息等

3. personnalInfo

表名	personnalInfo（弱实体）		
属性	数据类型(精度)	空（非空）	含义
userID	varchar(50)	非空	用户 ID(表一的外键),主键，
school	varchar(20)	可空	用户所在学校
sex	varchar(10)	可空	性别
realName	varchar(100)	可空	用户真实姓名
phone	varchar(30)	可空	用户联系方式

【说明】personnalInfo 表格主要用户信息

4.collection

表名	collection		
属性	数据类型(精度)	空（非空）	含义
userID	varchar(40)	非空	用户 ID,表一的外键，主键之一
goods_id	Int(11)	非空	商品 id(表二的外键)，主键之一

【说明】collection 表格主要用户收藏商品的信息

四 数据库实现方式具体介绍

以下是建表语句

1.register_and_login

```
DROP TABLE IF EXISTS register_and_login;
DROP TABLE IF EXISTS collection;
DROP TABLE IF EXISTS goods;
DROP TABLE IF EXISTS personnalInfo;

CREATE TABLE register_and_login (
    userID varchar(40) NOT NULL,
    password varchar(40) DEFAULT NULL,
    nick_name varchar(50) DEFAULT NULL,
    PRIMARY KEY (userID)
);
```

2.商品信息 goods

```
CREATE TABLE goods (
    goods_id int(11) NOT NULL AUTO_INCREMENT,
    imgURL varchar(5000) DEFAULT NULL,
    name varchar(100) DEFAULT NULL,
    description varchar(300) NOT NULL,
    classifi varchar(50) DEFAULT NULL,
    contact_me varchar(50) DEFAULT NULL,
    userID varchar(40) DEFAULT NULL,
    latitude double DEFAULT NULL,
    longitude double DEFAULT NULL,
    PRIMARY KEY (goods_id),
    foreign key(userID) references register_and_login(userID)
) AUTO_INCREMENT=0 ;
```

3.收藏信息 collection

```
CREATE TABLE collection
(
    userID varchar(40),
    goods_id int,
    primary key(userID, goods_id),
    foreign key(goods_id) references goods(goods_id)
);
```

附加:加了一个触发器,是用于在用户删除商品信息之后,相应的收藏信息也会删除。

```
delimiter $
create trigger after_delete
after delete on goods
for each row
begin
    delete from collection where goods_id=old.goods_id;
end $
delimiter ;
```

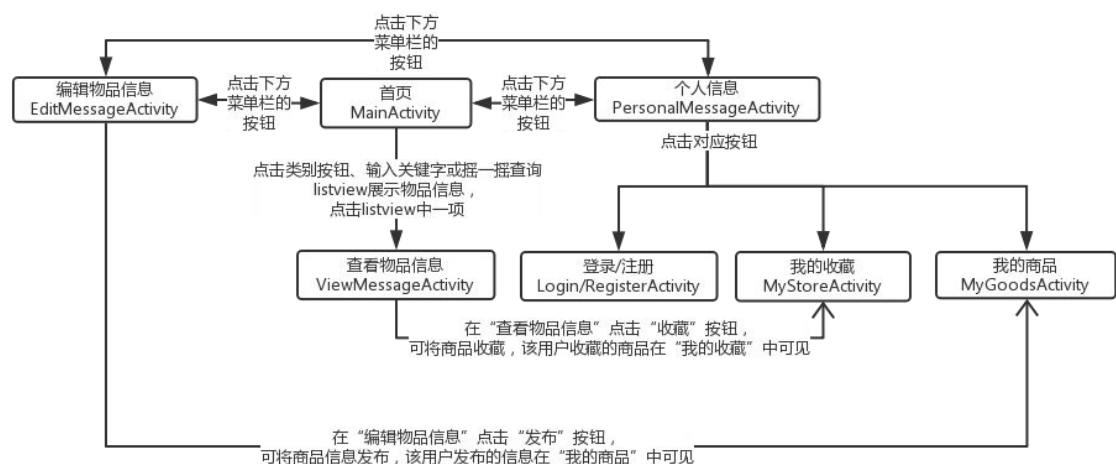
4.用户详细信息 personalInfo

```
CREATE TABLE personalInfo
(
    userID varchar(50),
    school varchar(20),
    sex varchar(10),
    realName varchar(100),
    phone varchar(30),
    foreign key(userID) references register_and_login(userID),
    PRIMARY KEY (userID)
);
```

五 程序实现

(1) 程序项目结构

①程序中各个活动的关系简图



②具体说明

在所有界面下方，都有一条菜单栏，其中包含三个按钮，分别为“主页”、“发布”“我的”。点击对应按钮，可以分别跳转到“主页”、“编辑物品信息”和“个人信息”界面。【界面编程及事件处理】

- 1) **MainActivity:** 首页。用户可在该界面通过以下三种方式查询二手商品的信息，它们分别是：在查询框中输入商品信息、点击一个类别的按钮（首页上有书籍、运动等五个类别的按钮）以及“摇一摇”获得附近的商品信息。使用前两种方式获得的商品信息离用户的距离为2000米及以内，使用“摇一摇”则可获得500米范围内的商品信息。获得的商品信息以listview的方式呈现在首页，它展示了商品的名字、类别以及离用户的距离信息。若想获得某项商品的具体信息，则可以点击listview中的对应一项，即可跳转到“查看物品信息”界面。【ListView，网络，位置服务，加速度传感器】
- 2) **EditMessageActivity:** 编辑物品信息。用户可在该界面输入商品的名称、描述信息，可添加商品的图片，需输入自己的联系方式和选择商品的类别。点击“发布”按钮，即可将商品信息发布。只有登录了的用户才能发布信息，且发布的信息可在“我的商品”

界面中查看到。【网络，手机多媒体（摄像头），数据存取（存储在云端服务器的数据库中）】

- 3) **PersonalMessageActivity:** 个人信息。若用户已登录，界面会出现“您好，xxx（用户昵称）”字样，点击该字样，会进入完善个人信息界面，然后对个人信息进行完善后，可以保存到云端数据库。在完善个人信息界面下方有个退出登录按钮，点击会弹出“是否退出登录”对话框。已登录的用户，可点击“我的商品”和“我的收藏”按钮，查看相关消息。若用户未登录，界面会出现“登录/注册”字样，点击该字样，会跳转到登录界面。
- 4) **EditpersonalActivity:** 完善个人信息。若之前填过个人详细信息则会更新，没有填过，就会插入到云端数据库中。【云端数据库 personnalInfo】
- 5) **ViewMessageActivity:** 查看物品信息。在“主页”、“我的商品”或“我的收藏”界面点击某条商品信息后，会出现跳转到该活动，用于显示物品的具体信息。点击“收藏”按钮，可以将该商品收藏；若已收藏，该按钮字样为“取消收藏”，点击按钮可以取消收藏。只有登录了的用户才能收藏商品信息，且收藏的信息可在“我的收藏”界面中查看到。【网络，数据存取】
- 6) **LoginActivity:** 用户登录。输入账号和密码，点击“登录”按钮登录，登录成功跳转到“个人信息”界面，否则提示“账号或密码错误”信息。若没有账号，点击“立即注册”按钮，跳转到注册界面。【网络】
- 7) **RegisterActivity:** 用户注册。输入账号、密码和昵称，点击“注册”按钮登录，注册成功跳转到“用户登录”界面，若账号已被注册，会有提示信息。【网络，数据存取】
- 8) **MyStoreActivity:** 我的收藏。若用户已登录，以 listview 形式展示用户的收藏商品信息，点击一项，会跳转到“查看物品信息”界面，以查看其具体信息。若用户未登录，会提示“请登录查看”。【ListView，网络，位置服务，数据存取】
- 9) **MyGoodsActivity:** 我的商品。若用户已登录，以 listview 形式展示用户发布的商品信息，点击一项，会跳转到“查看物品信息”界面，以查看其具体信息。长按 listview 中的一项，会弹出“是否删除该商品信息”的对话框。若用户未登录，会提示“请登录查看”。【ListView，网络，位置服务，数据存取】

五 程序关键代码和接口

客户端部分关键代码：

①Goods.java (商品的 class)

```
public class Goods {
    private int id;
    private String goods_name;
    private String goods_class;
    private String description;
    private String pictureId;
    private String contact;
    private double latitude;
    private double longitude;
    private double distance;
    public Goods(int id,String name,String g_class,String des,String picId,String contact,double lati,double longi,double dis){...}
    public int getGoodsId() { return id; }
    public String getGoodsName() { return goods_name; }
    public String getGoodsClass() { return goods_class; }
    public String getDescription() { return description; }
    public String getContact() { return contact; }
    public String getPictureId() { return pictureId; }
    public double getLatitude() { return latitude; }
    public double getLongitude() { return longitude; }
    public double getDistance() { return distance; }
}
```


②主页中的三种搜索方式

关键词搜索【另外两种与之类似】

```
private void SearchGoods(final String info){
    new Thread((Runnable) () -> {
        HttpURLConnection connection = null;
        try{
            URL url=new URL("http://2handbuy.vipsinaapp.com/search");
            connection=(HttpURLConnection) url.openConnection();
            connection.setRequestMethod("POST");
            connection.setConnectTimeout(8000);
            connection.setReadTimeout(8000);
            DataOutputStream out = new DataOutputStream(connection.getOutputStream());
            out.writeBytes("info="+info);

            InputStream in = connection.getInputStream();
            BufferedReader reader=new BufferedReader(new InputStreamReader(in));
            StringBuilder response=new StringBuilder();
            String line;
            while((line=reader.readLine())!=null)
                response.append(line);
            Log.d("res", response.toString());

        }
        try{
            int count=0;
            JSONArray jsonArray=new JSONArray(response.toString());
            for (int i=0;i<javascriptArray.length();i++){
                JSONObject jsonObject = jsonArray.getJSONObject(i);
                int id=jsonObject.getInt("id");
                String name=jsonObject.getString("name");
                String gclass=jsonObject.getString("gclass");
                String des=jsonObject.getString("des");
                String pic=jsonObject.getString("pic");
                String contact=jsonObject.getString("contact");
                double lati=jsonObject.getDouble("lati");
                double longi=jsonObject.getDouble("longi");
                double dis= GetDistance(lati, longi,1a,1e);
                if(dis<=2000){
                    count=count+1;
                    Message message = new Message();
                    Bundle bundle=new Bundle();
                    bundle.putInt("id", id);
                    bundle.putString("name", name);
                    bundle.putString("pic", pic);
                    bundle.putString("gclass", gclass);
                    bundle.putString("des", des);
                    bundle.putString("contact", contact);
                    bundle.putDouble("lati", lati);
                    bundle.putDouble("longi", longi);
                    bundle.putDouble("dis", dis);
                }
            }
        }
    });
}
```

```

        message.what=SHOW_GOODS;
        message.setData(bundle);
        handler.sendMessage(message);
    }

    }

    if(count==0){
        Message message = new Message();
        message.what=NO_GOODS;
        handler.sendMessage(message);
    }else{
        Message message = new Message();
        message.what=HAVE_GOODS;
        handler.sendMessage(message);
    }
} catch (Exception e){
    e.printStackTrace();
}
} catch (Exception e){
    Message message = new Message();
    message.what=NO_INTERNET;
    handler.sendMessage(message);
    e.printStackTrace();
}finally {
    if(connection!=null)
        connection.disconnect();
}
}).start();
}

```

③ListView 的 adapter

```

public class GoodsAdapter extends ArrayAdapter<Goods>{
    private int resourceId;

    public GoodsAdapter(Context context,int textViewResourceId,List<Goods> goods){
        super(context,textViewResourceId,goods);
        resourceId=textViewResourceId;
    }

    private Resources getResources() {
        Resources mResources = null;
        mResources = getResources();
        return mResources;
    }
}

```



```

@Override
public View getView(int position, View convertView, ViewGroup parent) {
    Goods goods = getItem(position);
    View view= LayoutInflater.from(this.getContext()).inflate(resourceId, null);
    ImageView goodsImage=(ImageView) view.findViewById(R.id.goods_image);
    TextView goodsName=(TextView)view.findViewById(R.id.goods_name);
    TextView distance=(TextView)view.findViewById(R.id.dis);

    //byte[] bytes = Base64.decode(goods.getPictureId(), Base64.DEFAULT);
    //Bitmap bitmap = BitmapFactory.decodeByteArray(bytes, 0, bytes.length);
    //goodsImage.setImageBitmap(bitmap);

    if(goods.getGoodsClass().equals("book"))
        goodsImage.setImageResource(R.mipmap.book);
    else if(goods.getGoodsClass().equals("sport"))
        goodsImage.setImageResource(R.mipmap.sport);
    else if(goods.getGoodsClass().equals("furniture"))
        goodsImage.setImageResource(R.mipmap.furniture);
    else if(goods.getGoodsClass().equals("clothes"))
        goodsImage.setImageResource(R.mipmap.clothes);
    else
        goodsImage.setImageResource(R.mipmap.food);

    if(goods.getGoodsName().length()<=10)
        goodsName.setText(goods.getGoodsName());
    else
        goodsName.setText(goods.getGoodsName().substring(0,10)+"...");
    distance.setText(""+goods.getDistance()+"m");
    return view;
}
}

```

④位置服务，获取经纬度信息之后，根据两地经纬度计算两地距离的方法

```

public static double GetDistance(double lat1, double lng1, double lat2, double lng2)
{
    double radLat1 = rad(lat1);
    double radLat2 = rad(lat2);
    double a = radLat1 - radLat2;
    double b = rad(lng1) - rad(lng2);

    double s = 2 * Math.asin(Math.sqrt(Math.pow(Math.sin(a / 2), 2) +
        Math.cos(radLat1) * Math.cos(radLat2) * Math.pow(Math.sin(b / 2), 2)));
    s = s * EARTH_RADIUS;
    //s = Math.round(s * 1000000) / 1000000;
    s=Math.round(s * 1000);
    return s;
}

```

⑤调用手机摄像头和相册

```
add_picture_1.setOnClickListener((v) -> {  
    File outputImage = new File(Environment.getExternalStorageDirectory(), "tempImage.jpg");  
    try{  
        if(outputImage.exists()){  
            outputImage.delete();  
        }  
        outputImage.createNewFile();  
    }catch (IOException e) {  
        e.printStackTrace();  
    }  
    imageUri = Uri.fromFile(outputImage);  
    Intent intent = new Intent("android.media.action.IMAGE_CAPTURE");  
    intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);  
    startActivityForResult(intent, TAKE_PHOTO);  
});
```

```
add_picture_2.setOnClickListener((v) -> {  
    File outputImage = new File(Environment.getExternalStorageDirectory(), "output_image.jpg");  
    try{  
        if(outputImage.exists()){  
            outputImage.delete();  
        }  
        outputImage.createNewFile();  
    }catch (IOException e) {  
        e.printStackTrace();  
    }  
    imageUri = Uri.fromFile(outputImage);  
    Log.d("pic", imageUri.toString());  
    Intent intent = new Intent("android.intent.action.GET_CONTENT");  
    intent.setType("image/*");  
    intent.putExtra("crop", true);  
    intent.putExtra("scale", true);  
    intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);  
    startActivityForResult(intent, CROP_PHOTO);  
});
```

```
@Override  
protected void onActivityResult(int requestCode, int resultCode, Intent data) {  
    switch(requestCode){  
        case TAKE_PHOTO:  
            if(resultCode==RESULT_OK){  
                Intent intent = new Intent("com.android.camera.action.CROP");  
                intent.setDataAndType(imageUri, "image/*");  
                intent.putExtra("scale", true);  
                intent.putExtra(MediaStore.EXTRA_OUTPUT, imageUri);  
                startActivityForResult(intent, CROP_PHOTO);  
            }  
            break;  
        case CROP_PHOTO:  
            if(resultCode == RESULT_OK){  
                try{  
                    Bitmap bitmap = BitmapFactory.decodeStream(getContentResolver().openInputStream(imageUri));  
                    imageString=ReadImgToBinary2.imgToBase64(bitmap);  
                    Log.d("xxx", imageString);  
                    byte[] bytes = Base64.decode(imageString, Base64.DEFAULT);  
                    Bitmap bitmap1 = BitmapFactory.decodeByteArray(bytes, 0, bytes.length);  
                    pic.setImageBitmap(bitmap1);  
                    //imageString=URLEncoder.encode(imageString, "utf-8");  
  
                    imageString=imageString.replace("+", "!");  
                    imageString=imageString.replace("/", "*");  
                }catch(FileNotFoundException e){  
                    e.printStackTrace();  
                }  
            }  
        }  
    }
```

```

    }
    break:
default:
    break:
}
}

```

(二) 服务器端关于数据库调用的代码(使用 python 语言)

①用户登录：注册成功向客户端发送该账户对应的昵称；否则返回 “no”

```

class login:
    def POST(self):
        para = web.input()
        username = para['username']
        password = para['password']
        con = Connect()
        query='select * from register_and_login where userID="'+str(username)+'"'
        ok = Exec(con,query)
        if len(ok)==0:
            return 'no'
        else:
            if ok[0][1]==password:
                return ok[0][2]
            else:
                return 'no'

```

②用户注册：注册成功，返回”ok”；否则返回 “no”

```

class regist:
    def POST(self):
        para = web.input()
        username = para['username']
        password = para['password']
        nickname = para['nickname']
        u=str(username)
        p=str(password)
        n=nickname
        query='insert into register_and_login values ("'+u+"','"+p+"','"+n+"') '
        #TODO:...
        con = Connect()
        try:
            ok = Exec(con,query)
        except:
            return 'no'

        con.commit()
        return 'ok'

```

③依据关键词获取商品信息，打包成 json 格式发回给客户端

【摇一摇依据位置、点击类别依据类别于此类似】

```

class search:
    def POST(self):
        para = web.input()
        gclass = para['info']
        con = Connect()
        query='select * from goods where description like "%'+gclass+'%" or name like "%'+gclass+'%"'
        res = Exec(con,query)
        data=dict()
        s=""
        for i in range (len(res)):
            data['id']=res[i][0]
            data['name']=res[i][2]
            data['pic']=res[i][1]
            data['gclass']=res[i][4]
            data['des']=res[i][3]
            data['contact']=res[i][5]
            data['lati']=res[i][7]
            data['longi']=res[i][8]
            if i!=len(res)-1:
                s=s+json.dumps(data)+',';
            else:
                s=s+json.dumps(data)
        s=s+']'
        return s

```

④查看某用户发布的商品信息，打包成 json 格式发回给客户端

```
class getMyGoods:

    def POST(self):
        para = web.input()
        userId = para['userId']
        con = Connect()
        query='select * from goods where userID="'+str(userId)+'"'
        res = Exec(con,query)
        data=dict()
        s="["
        for i in range (len(res)):
            data['id']=res[i][0]
            data['name']=res[i][2]
            data['pic']=res[i][1]
            data['gclass']=res[i][4]
            data['des']=res[i][3]
            data['contact']=res[i][5]
            data['lati']=res[i][7]
            data['longi']=res[i][8]
            if i!=len(res)-1:
                s=s+json.dumps(data)+',';
            else:
                s=s+json.dumps(data)
        s=s+']'
        return s
```

⑤用户删除某条商品信息

```
class delete:

    def POST(self):
        para = web.input()
        goods_id = para['id']
        query='delete from goods where goods_id="'+str(goods_id)+'"'
        con = Connect()
        ok = Exec(con,query)
        con.commit()
        return 'ok'
```

⑥用户收藏某条商品信息

```
class store:

    def GET(self):
        return 'ok'

    def POST(self):
        para = web.input()
        userid = para['userId']
        goods_id = para['id']
        query='insert into collection values ("'+str(userid)+'","'+str(goods_id)+'")'
        #TODO:...
        con = Connect()
        try:
            ok = Exec(con,query)
        except:
            return 'no'

        con.commit()
        return 'ok'
```

⑦ 用户完善个人信息

```
def POST(self):
    para = web.input()
    realName = para['realName']
    sex = para['sex']
    sclass = para['sclass']
    phone= para['phone']
    userId = para['userId']
    select='select * from personalInfo where userID="'+str(userId)+'"'
    con = Connect()
    sel=Exec(con,select):
    if len(sel)==0:
        query='insert into personalInfo(userID,realName,school,phone,sex) values ("'+str(userId)+' ',''+str(realName)+' ',''+str(sclass)+' ',''+str(phone)+' ',''+str(sex)+' ')"
    else:
        query='update personalInfo set realName="'+str(realName)+' ',school="'+str(sclass)+' ',phone="'+str(phone)+' ',sex="'+str(sex)+' ' where userID="'+str(userId)+' ' '

    try:
        ok = Exec(con, query)
    except:
        return 'no'

    con.commit()
    return 'ok'
```

(3) 程序运行截图

1. 打开应用，出现一个延迟 3 秒的开始页面，然后进入“初始”页面。



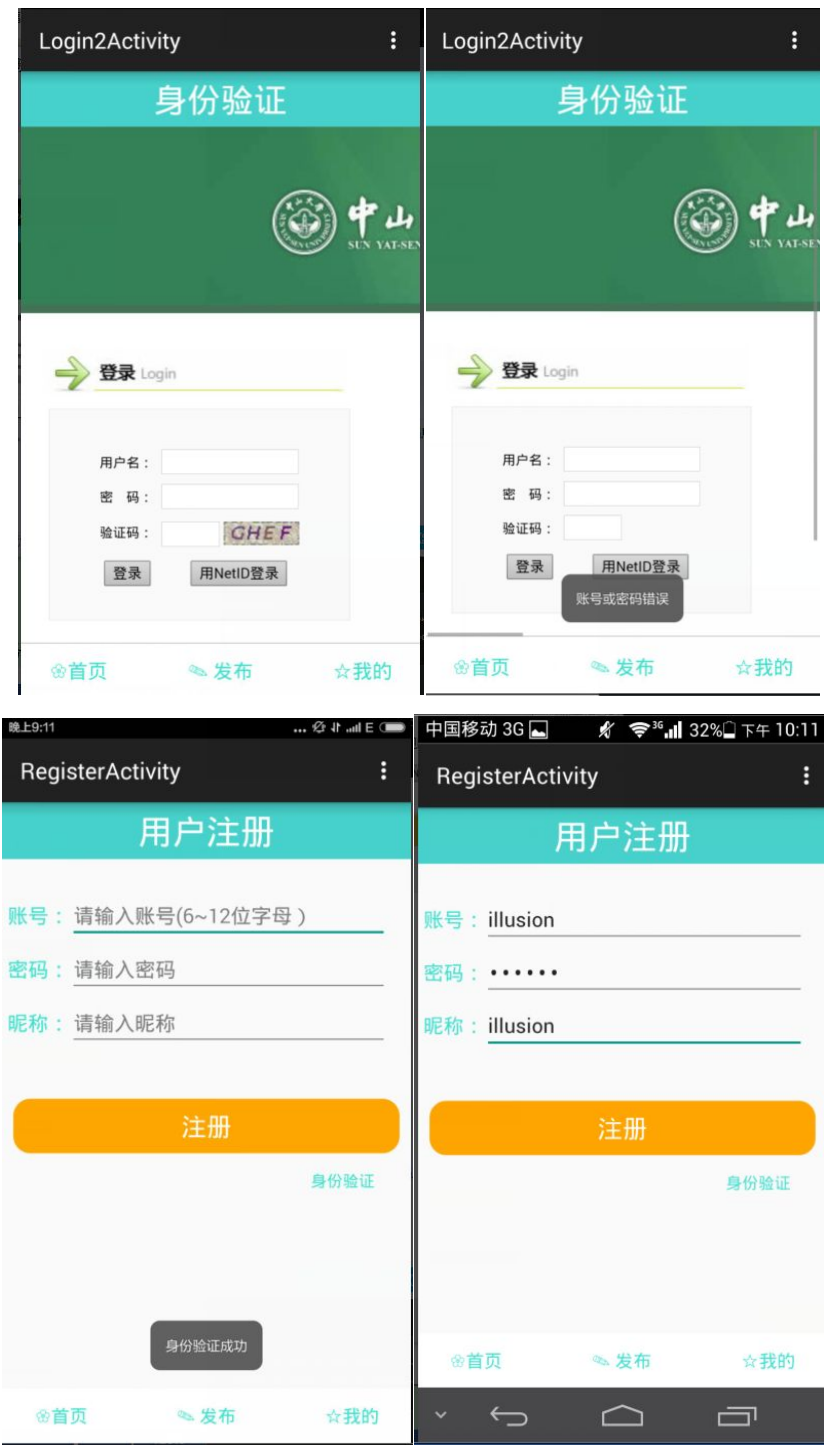
2. 为了更好的使用该款 App，需要进行注册页面进行注册。

首先进入个人信息页面，点击登陆/注册按钮，会进入用户注册页面，系统会提示没有进行身份验证，无法成功注册，点击右下角的身份验证，进入身份验证的页面。



3. 本款应用主要针对大学生，所以注册需先进入学校教务系统进行大学生验证。

该款 App 目前只面向中山大学学生，点击身份验证按钮后，会进入中山大学的教务系统，输入用户名和密码，如果成功登陆教务系统，则身份验证成功，可以进入下一步，注册账号，并要求输入相关信息。



4. 注册后，可以进入个人信息页面，登陆自己刚才注册的账号。



这时候再点击您好，xxx 那个按钮，会出现完善个人信息界面



更新成功后在云端数据库可以看到被更新的信息

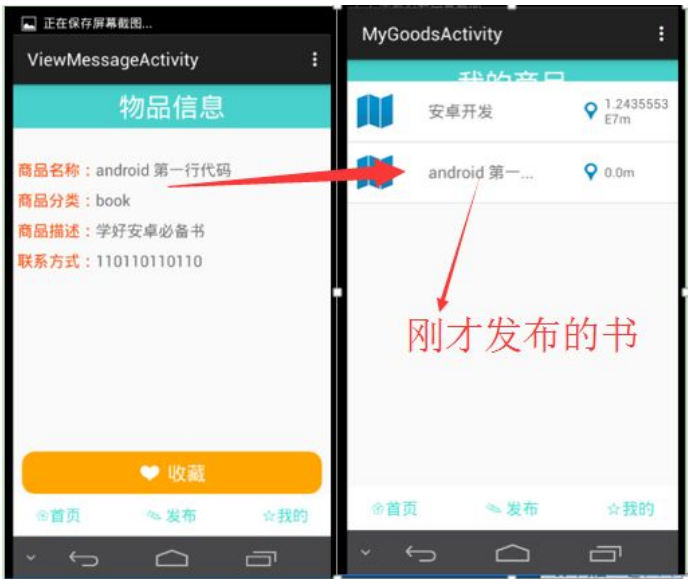
			userId	school	sex	realName	phone
<input type="checkbox"/>				zhuhai	男	xx	1372623
<input type="checkbox"/>			19283	zhuhai	男	xx	1372623
<input type="checkbox"/>			niexiaoning	zhuhai	女	hhh	13726231932
<input type="checkbox"/>			zhulin	north	女	tt	rrr

5. 可以进入发布页面发布商品信息。

点击页面下方的发布按钮，进入发布页面，输入合适信息，并选取商品种类，之后点击发布按钮（黄色长条），进行发布。

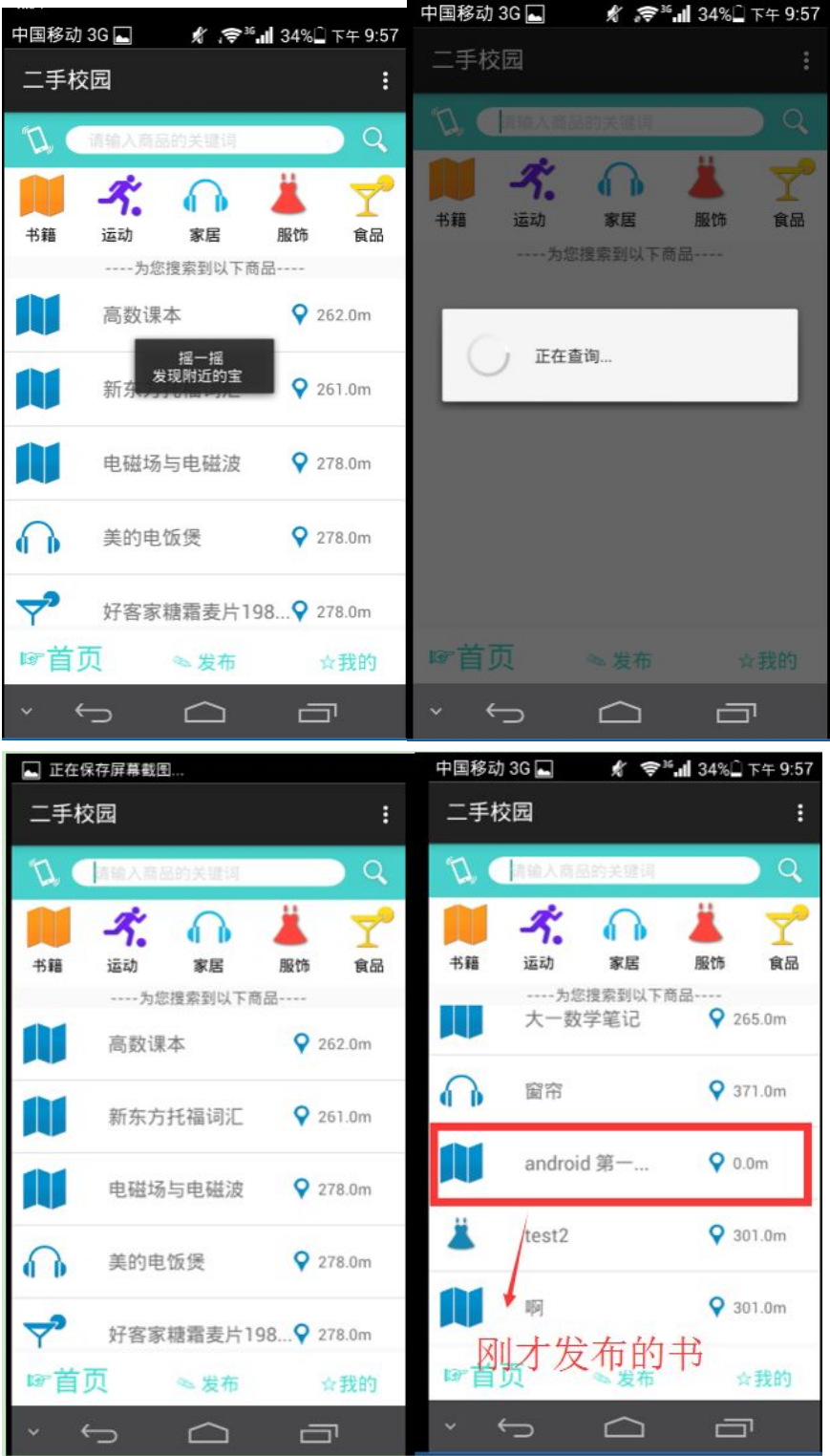


6. 发布后的商品信息会出现在“我的商品”中。

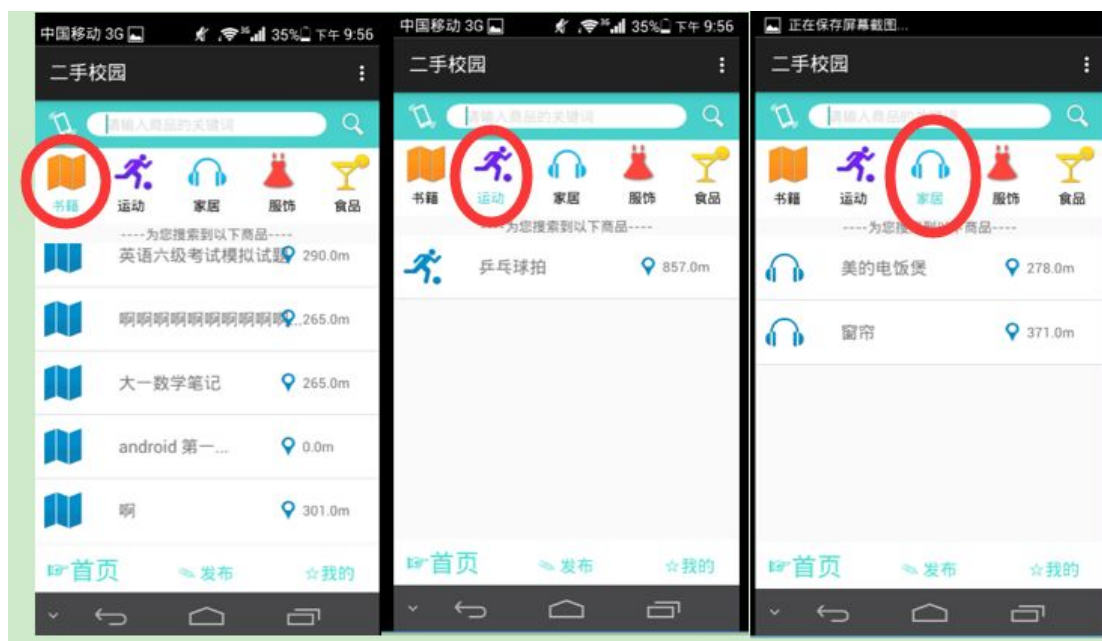


7. 如果需要购买商品，可以使用搜索功能，因为该款 App 面向大学生，并且致力于减少诈骗风险和运输途中的各种问题，搜索出的物品都是距离比较小的周边物品。

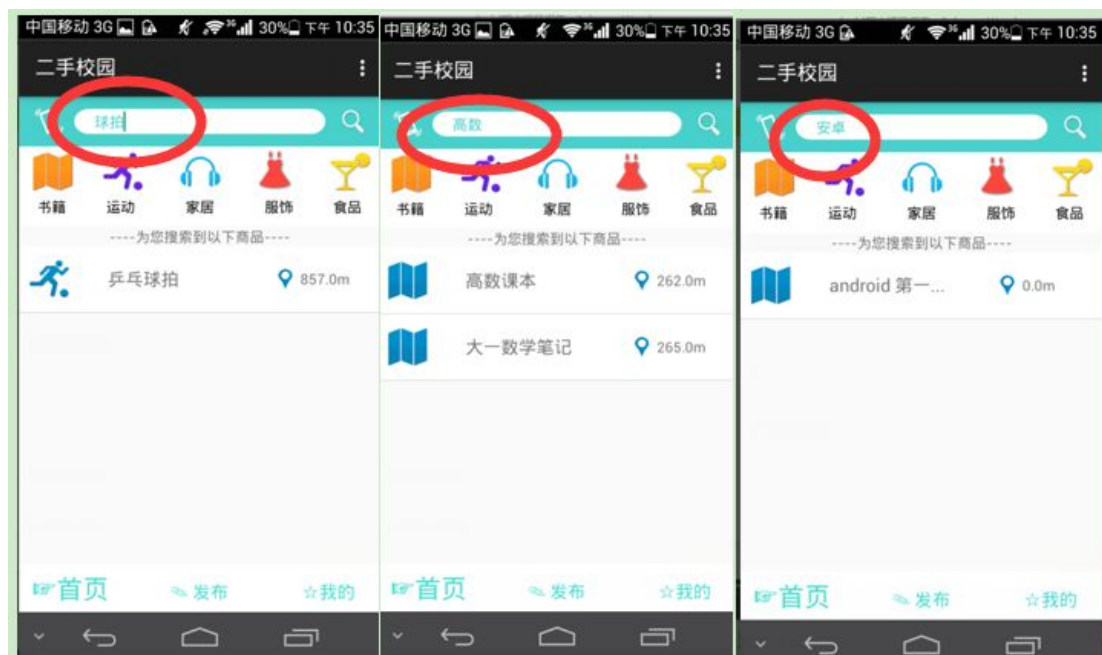
8. 使用摇一摇功能，无差别搜索周边商品。



10 直接点击提供的分类物品。



11 关键字搜索功能。



12 如果找到中意的物品，可以点击收藏，保存到个人账户的“我的收藏”中。



六 各组员项目感想

【朱琳】

本次关于数据库的 project 是跟我们的安卓课程的 project 连在一起的,所以也是密不可分的。开始我们的数据库设计比较复杂,后来经过层层的选择和精简,我们数据库的表缩减为 4 个,其中收藏功能,自己发布的商品,其他人发布的商品,这些东西都使用 goods 来存储,只是在收藏功能中记录了 goods_id 和 userId,这样的话就省去了原来 ER 图中的一些 belongTo 的属性,从而有利于优化存储空间。我发现精简数据库有很多好处,节省存储空间,简单明了,节省调用语句等等。另外关于 app 实现部分,数据库方面的优化也使得 app 的界面更加流畅。

【周梦辉】

Project 终于做完了,从最初的创意构思到实现基本功能再到最后功能的测试和改进,我们小组耗费了许多心思。这个过程,虽然辛苦,但也有很多收获,对自己能力的提升很有作用。因为开始做的时候,还有很多知识在安卓课上暂时没讲到的,比如说 http 请求、位置服务还有手机多媒体等等,这都需要我自己去看书一点点的学习和模仿。尤其是网络通信方面,几乎贯穿了我们整个 APP,但在最开始起步的时候特别难。

关于客户端与服务器的通信,当时不仅没学 http 请求,而且服务器那边一开始也不太会写,而且需要使用 python 语言,而我压根没接触过这种语言,因此无奈之下,又开始查阅资料,自己学习 Python,不过好在 python 语言并不复杂,又加上有 c++ 的语言基础,学习起来虽然伤神,但也能搞定,话虽如此,在这个过程中,仍然是反反复复遇到了很多困难。

后来上了关于利用安卓自带传感器的实验课,我又对这款 App 添加了摇一摇的搜索功能,我根据实验文档和查阅了相关的资料,通过安卓手机自带的加速度传感器最终实现这个功能,这样能够让用户更方便快捷的搜索物品,不再像市场上的应用一样,只提供关键字搜索和类别搜索,这个摇一摇的搜索功能,虽然功能比较单一,但我仍然认为这款安卓二手交易 App

一个亮点。我计划在这学期结束后，对所有的物品添加一个属性，“热度”，也即是受欢迎程度，综合搜索量，交易量等等属性，摇一摇功能就是将所有的物品，按照“热度”递减的标准，依次显示出来，这样能够让用户查找物品更方便。

再者关于商品分类的搜索功能，这个功能实现起来比较简单，在用户发布商品信息的时候，只需要就自动在后台将物品归类到对应的种类，在其他用户点击分类标签的时候，就将对应的物品全部显示出来。

最后是关于关键字搜索，我的做法是先从商品的名称中查找关键字，如果找到对应的关键字，就将对应商品添加到要显示的物品中。再查找剩下的商品的说明文字，如果能够找到对应的关键字，同样扔到要显示的物品中。此外，因为许多商品的信息中都有中文字符，在客户端和服务期之间的传递，我遇到了很多问题，也耗费了很多心思，因为服务器是放在云平台上，所以关于后台的代码修改耗费了很多时间，而且在客户端和服务端间传递中文，还需要先编码才行，此外我还了解 JSON 格式的解析等等，好在最后都一点一点的解决了。

【张紫琪】

这学期数据库学习了挺多内容的，从 ER 图关系模型的建立，新建数据库，新建表格，关系代数关系演算到 SQL 模型，学会了对基本数据库的操作（新建、增删改查等）。

从布置数据库 project 开始，我们选择与移动应用开发课程联系起来一起做。选择主题的时候花费了很多时间，最后选定做二手交易平台。我们的应用的数据库主要有

- ①保存用户登录注册信息表；
- ②保存商品信息的表格，主要保存商品的图片，描述，类别（书籍、运动商品、家具、衣服等），发布商品的用户的的信息（如姓名，联系方式）；
- ③保存用户个人信息的表格
- ④用户和商品之间的联系；
- ⑤当删除商品时，用户收藏与之相关的商品也一并删除；

我们用到的数据库 sql 语句包含了上课讲的所有内容。然后是构建服务端，我们购买了阿里云服务器，在云端用 python 写了登录注册以及数据库的操作。搭建好云服务器之后，就是编写 application 了，我们用到了 android 课上讲的 listView, Intent, Bundle, 网络服务，距离定位等等。其中的界面设计也花费了不少心思，写代码是一个艰难的过程，期间出现了各种 bug，最后我们还是挺过来了！编写应用的过程中，让我知道了还有很多我们可以实现的功能，也学会了不少东西，比如怎么保存图片到云服务器然后再把它显示在手机客户端，我们用的是 url 保存链接。最后感谢队友的努力付出，让我们的 app 有了更多的功能和更美的界面。