

1. 测试有关说明

1.1 测试名称与内容

发布过程

搜索过程

查询过程

收藏过程

1.2 测试用例

【发布过程】：用户登录之后，进入“我的”个人主页，点击“发布”按钮，协调数据库操作，实现发布商品的功能。

```
public class UploadItemActivity extends AppCompatActivity {
    public static final int TAKE_PHOTO = 1;
    public static final int CHOOSE_PHOTO = 2;
    private ImageView picture;
    private Uri imageUri;
    private DatabaseHelper dbHelper;
    private Spinner sp;
    private ContentValues values=new ContentValues();
    final SQLiteDatabase db = dbHelper.getWritableDatabase();
    private void displayImage(String imagePath) {//同时把图片加载到数据库
        if (imagePath != null) {
            Bitmap bitmap = BitmapFactory.decodeFile(imagePath);
            ByteArrayOutputStream baos = new ByteArrayOutputStream();
            bitmap.compress(Bitmap.CompressFormat.PNG, 100, baos);
            picture.setImageBitmap(bitmap);
            values.put("image", baos.toByteArray());
            db.insert("iteminfo", null, values);
        } else {
            Toast.makeText(this, "failed to get image",
Toast.LENGTH_SHORT).show();
        }
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.upload_item);
        picture = (ImageView) findViewById(R.id.item_photo);
        Button finish_upload = (Button) findViewById(R.id.finish);
        Button chooseFromAlbum = (Button)
findViewById(R.id.choose_photo);
        Button takePhoto = (Button) findViewById(R.id.take_photo);
        String[] ctype = new String[] {"生活用品", "学习用品", "电子产品", "体育用品"};
```

```

        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, ctype); //创建一个数组适配器
adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropd
own_item); //设置下拉列表框的下拉选项样式
        Spinner spinner = (Spinner)
super.findViewById(R.id.item_style);
        spinner.setAdapter(adapter);
        sp = (Spinner) findViewById(R.id.item_style);
        final String kind = (String) sp.getSelectedItem();
        finish_upload.setOnClickListener(new View.OnClickListener() {
            @RequiresApi(api = Build.VERSION_CODES.O)
            @Override
            public void onClick(View v) { //上传商品信息插入到数据库
                EditText
title=(EditText)findViewById(R.id.item_name);
                EditText
price=(EditText)findViewById(R.id.item_price);
                EditText
intro=(EditText)findViewById(R.id.item_introduction);
                EditText
contact=(EditText)findViewById(R.id.contact);
                //在登录功能中给 post_userid 赋值
values.put("userId", post_userid);
                values.put("title", title.getText().toString());
                values.put("userId", post_userid);
                values.put("kind", kind);
                values.put("price", price.getText().toString());
                values.put("contact", contact.getText().toString());
                values.put("intro", intro.getText().toString());
                //bmp.compress(Bitmap.CompressFormat.PNG, 100, os);
                // cv.put("pic", os.toByteArray());
                db.insert("iteminfo", null, values);
                Toast.makeText(getApplicationContext(), "发布成功",
Toast.LENGTH_SHORT).show();
                MyItems.actionStart(UploadItemActivity.this);
            }
        });
    });

```

【查询过程】 查询数据库中存储的物品信息，并显示

```

public class MyItems extends AppCompatActivity {
    String TABLENAME = "iteminfo";
    byte[] imagedata;
    Bitmap imagebm; //图片类型必须是 Bitmap
    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

        super.onCreate(savedInstanceState);
        setContentView(R.layout.my_item);
        final DatabaseHelper dbtest = new DatabaseHelper(this);
        final Intent intent = getIntent();
        final SQLiteDatabase db = dbtest.getWritableDatabase();
        ImageView image = (ImageView) findViewById(R.id.item_photo);
        TextView title = (TextView) findViewById(R.id.item_name);
        TextView price = (TextView) findViewById(R.id.item_price);
        TextView intro = (TextView) findViewById(R.id.item_introduce);
        TextView contact = (TextView) findViewById(R.id.contact);
        Button button_return = (Button)
findViewById(R.id.button_return);
        Cursor cursor = db.query(TABLENAME, null, "id=?", new
String[] {intent.getStringExtra("id")}, null, null, null, null); // 根据接
收到的 id 进行数据库查询
        Log.i("商品的 id 是", intent.getStringExtra("id"));
        if (cursor.moveToFirst()) {
            while (!cursor.isAfterLast()) {
                imagedata = cursor.getBlob(6);
                imagebm = BitmapFactory.decodeByteArray(imagedata, 0,
imagedata.length);
                image.setImageBitmap(imagebm); //把图片从数据库加载到
image
                title.setText(cursor.getString(2));
                intro.setText(cursor.getString(4));
                price.setText(cursor.getString(5));
                contact.setText(cursor.getString(7));
                cursor.moveToNext();
            }
            cursor.close();
        }
    }
}

```

【搜索过程】用户使用软件，打开主页，主页下方会随机显示推送的商品，上方有一个搜索框，在搜索框内输入商品的大致信息就会跳转页面展示商品列表

```

public class HomeFragment extends Fragment
    implements SearchView.OnQueryTextListener,
    AdapterView.OnItemClickListener {

    private List<Commodity> commodityList = new ArrayList<>();
    private ListView comResult;

    private SearchView searchview;

```

```

        private ListView searchResult;
        //数据库
        private String[] search_data = {"游戏机", "二手课本", "辅导习题",
"ershou", "sse", "ssa", "ssd"};
        //
        private List<String> result = new ArrayList<String>();
        private ArrayAdapter<String> search_adapter;

    public HomeFragment() {
        // Required empty public constructor
    }

    @Override
    public void onStart() {
        super.onStart();
        //搜索列表初始化
        searchview = (SearchView)
getView().findViewById( R.id.search_view );
        searchview.setSubmitButtonEnabled( true );
        searchview.setIconifiedByDefault( false );
        searchview.setOnQueryTextListener( this );
        searchResult = (ListView)
getView().findViewById( R.id.search_result );
        search_adapter = new ArrayAdapter<String>( getContext(),
android.R.layout.simple_list_item_1, result );
        searchResult.setAdapter( search_adapter );
        searchResult.setOnItemClickListener( this );
        //猜你想要 初始化
        initCommodity();
        Commodity_adapter commodityAdapter = new
Commodity_adapter( getContext(), R.layout.commodity_list,
commodityList );
        comResult = (ListView)
getView().findViewById( R.id.comResult );
        comResult.setAdapter( commodityAdapter );
        comResult.setOnItemClickListener( new
AdapterView.OnItemClickListener() {
            @Override
            public void onItemClick(AdapterView<?> parent, View view,
int position, long id) {
                Intent intent = new Intent( getActivity(),
ResultActivity.class );
                startActivity( intent );
            }
        }
    }

```

```

        } );
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
                                Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        return inflater.inflate( R.layout.fragment_home, container,
false );
    }

    //单击搜索按钮后
    @Override
    public boolean onQueryTextSubmit(String query) {
        result.clear();
        for (String s : search_data) {
            if (s.toLowerCase().indexOf( query.toLowerCase() ) >= 0)
                result.add( s );
        }
        search_adapter.notifyDataSetChanged();
        return false;
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        return false;
    }

    //搜索结果可点击，实现跳转
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
position, long id) {
        if (id >= 0) {
            String s = result.get( (int) id );
            //得到搜索的结果
            Toast t;
            t = (Toast) Toast.makeText( getContext(), "您点击了" + s,
Toast.LENGTH_SHORT );
            t.show();
            Intent intent = new Intent( getActivity(),
SearchActivity.class );
            startActivity( intent );
        }
    }

```

```

    }
}
//后期修改数据库
private void initCommodity() {
    for (int i = 0; i < 5; i++) {
        Commodity a = new Commodity( "shufa", R.drawable.a1, "50r" );
        commodityList.add( a );
        Commodity b = new Commodity( "shux", R.drawable.a2, "300r" );
        commodityList.add( b );
        Commodity c = new Commodity( "xsg", R.drawable.a3, "90r" );
        commodityList.add( c );
    }
}
}

```

【评论展示】将用户评论通过适配器用 recyclerview 将其放入 cardview 中自动展示出来

```

public class InfoAdapter extends
RecyclerView.Adapter<InfoAdapter.ViewHolder> {
    private Context context;
    private List<Info> mInfoList;
    private Dialog dia;
    static class ViewHolder extends RecyclerView.ViewHolder {
        CardView cardView;
        ImageView infoImage;
        ViewHolder(View itemView) {
            super(itemView);
            cardView = (CardView) itemView;
            infoImage = itemView.findViewById(R.id.info_image);
        }
    }
    public InfoAdapter(List<Info> infoList) {
        mInfoList = infoList;
    }
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType)
    {
        if (context == null) {
            context = parent.getContext();
        }
        View view = LayoutInflater.from(context).inflate(R.layout.info,
parent, false);
        final ViewHolder holder = new ViewHolder(view);
        holder.cardView.setOnClickListener(new View.OnClickListener()
        {

```

```

        @Override
        public void onClick(View v) {
            int position = holder.getAdapterPosition();
            Info info = mInfoList.get(position);
            Intent intent = new Intent(context,
ImageActivity.class);
            intent.putExtra(ImageActivity.Image_ID,
info.getImageId());
            context.startActivity(intent);
        }
    });
    return holder;
}
@Override
public void onBindViewHolder(ViewHolder holder, int position) {
    Info info = mInfoList.get(position);

Glide.with(context).load(info.getImageId()).into(holder.infoImage);
}
@Override
public int getItemCount() {
    return mInfoList.size();
}
}

```