**实 验 报 告**

**课程名称：** **移动应用开发**

**学 院： 计算机科学与工程学院**

**专 业：计算机科学与技术** **班 级： 18级1班**

**姓 名：樊睿 学 号：201801060102**

**年 月 日**

**山 东 科 技 大 学 教 务 处 制**

**实 验 报 告**

**页**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **组 别** |  | **姓 名** | **樊睿** | **同组实验者** |  |
| **实验项目**  **名称** | **数据存储** | | **实验日期** | **12.18** | |
| **教师评语** |  | | | | |
| **实验成绩：** | | | **指导教师（签名）：**  **年 月 日** | | |
| 1. **app功能**   （1）项目连接云端服务器，实现将数据存储云端。  （2）支持多用户。用户可注册，注册后可登录。    （3）用户登录后呈现图书的列表，可以根据书籍姓、作者、价格区间等信息进行筛选。    （4）点击某本图书后可以显示图书的详情（详情内容自定义）。在详情页面可以将该图书加入购物车。    （4）提供结算功能，根据购物车中的图书及数量，显示总额。点击后进行支付。    （5） 提供书籍评论功能，显示每位用户对于不同书籍进行评论，并通过与服务器的交互查看其它用户的评论。    (6) 分类页面可以根据书籍类别显示各类书籍。    （7）我的页面可以查看用户信息，订单，修改签名等功能。     1. **系统设计**   **数据库设计**  **User表**（表示用户信息）  uid 用户编号  uname 用户姓名  upassword 用户密码  umoney 用户余额  utype 用户类型  qian 用户签名；  **Book表**（书籍信息）  bid 书籍编号  bname 书名  bwriter 作者  bdetail 详细信息  bvalue 价格  xiao 销量  btype书籍类型；  **Bill表**（订单表）  oid 订单编号  uid 用户编号  bid 书籍编号  time 订单生成时间  **Remark表**（评论信息）  rid 评论编号  uid 用户编号  bid 书籍编号  comment 评论内容  **系统架构**  （1） 服务器端和android端进行两个项目的编写。  （2） 服务器端搭建基于SSM框架的web项目，在服务器通过mysql进行数据的存储，通过Mybatis在web项目里对数据库进行操作，并实现与android端进行的数据交互，将数据传输给android，将android传输过来的数据进行存储。  （3） android完成前端页面的实现，并通过OkHttp框架进行与服务器的数据交互。  **小组分工**  韩晨晨：进行数据库设计。服务器端代码书写，完成在android端与服务器端进行数据交互的接口，根据android传来信息对服务器端数据库进行修改或返回给android端数据。云服务器的搭建。  胡浩然：购物车界面的实现。在通过Litepal架构进行对android原生数据库的操作，将购物车的信息存入数据库和读出，实现购物车页面的各种功能的实现，对用户余额、书籍销量、订单等通过网络进行对应修改。  袁康裕：分类界面的实现，从服务器端读取书籍信息，通过不同类型的选择，在listview中展示该类别的书籍。商品详情界面的实现，可以查看对应书籍信息，实现加入购物车，购买等功能。评论功能的实现。  刘建豪：我的界面的实现，完成根据登录的用户展示改用户表信息，用户个性签名的修改。订单功能的实现，对于购买的书籍生成响应订单，并在我的页面中可以展示。  樊睿：商品浏览界面的实现，展示所有的书籍，可以通过模糊搜索对应的书名，根基商品价格、销量进行排序，根据作者、书籍类别、价值区间进行筛选。  **商品浏览界面：**  **package** com.example.bookstore;  **import** androidx.appcompat.app.AppCompatActivity;  **import** android.app.AlertDialog; **import** android.content.Context; **import** android.content.DialogInterface; **import** android.content.Intent; **import** android.graphics.drawable.BitmapDrawable; **import** android.graphics.drawable.Drawable; **import** android.os.Bundle; **import** android.view.LayoutInflater; **import** android.view.View; **import** android.widget.Button; **import** android.widget.TextView; **import** android.widget.Toast;  **import** com.bumptech.glide.Glide; **import** com.example.bookstore.scoll.AutoScrollViewPager; **import** com.example.bookstore.scoll.BannerIndicator; **import** com.example.bookstore.scoll.ViewPagerAdapter; **import** com.google.gson.Gson; **import** com.rengwuxian.materialedittext.MaterialEditText;  **import** org.w3c.dom.Text;  **import** java.io.IOException; **import** java.sql.Timestamp; **import** java.text.SimpleDateFormat; **import** java.util.ArrayList; **import** java.util.Date; **import** java.util.List;  **import** okhttp3.FormBody; **import** okhttp3.OkHttpClient; **import** okhttp3.Request; **import** okhttp3.RequestBody; **import** okhttp3.Response;  **public class** BookDetail **extends** AppCompatActivity {  **private** AutoScrollViewPager **autoScrollViewPager**;  **private** List<Drawable> **drawableList** = **new** ArrayList<>();  **private** BannerIndicator **bannerIndicator**;  ViewPagerAdapter **pagerAdapter**;  TextView **bookname**;  TextView **bookvalue**;  TextView **bookdetail**;  TextView **bookwriter**;  TextView **booktype**;  TextView **bookxiao**;  Context **mContext**;  Button **ping**;  Button **kan**;  Button **jie**;  Button **gou**;  Book **book**;   @Override  **protected void** onCreate(Bundle savedInstanceState) {  **super**.onCreate(savedInstanceState);  setContentView(R.layout.***activity\_book\_detail***);  Intent intent = getIntent();  **mContext** = **this**;   **int** bookid = intent.getIntExtra(**"BookId"**, 1);   **bookname** = findViewById(R.id.***book\_name***);  **bookvalue** = findViewById(R.id.***book\_value***);  **booktype** = findViewById(R.id.***book\_type***);  **bookdetail** = findViewById(R.id.***tit\_desc\_xiangqing***);  **bookxiao** = findViewById(R.id.***book\_xiao***);  **bookwriter** = findViewById(R.id.***book\_wirter***);  **autoScrollViewPager** = findViewById(R.id.***book\_image***);  **bannerIndicator** = findViewById(R.id.***indicator***);  **ping** = findViewById(R.id.***pinglun***);  **kan** = findViewById(R.id.***kanping***);  **jie** = findViewById(R.id.***jiesuan***);  **gou** = findViewById(R.id.***add\_shopcars***);   getBookMessage(bookid); //评价按钮  **ping**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  showCustomizeDialog();//调用下面的自定义函数  }  }); //查看评论  **kan**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Intent intent1 = **new** Intent(**mContext**, KanPing.**class**);  //跳转到看评论的Activity  intent1.putExtra(**"bid"**, **book**.getBid());  startActivity(intent1);  }  }); //加入购物车  **gou**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  Gou gou = **new** Gou(bookid, HttpUtil.*MyUser*.getUid());  //通过自定义的接口获取相关信息  gou.save();  Toast.makeText(v.getContext(), **"加入成功"**, Toast.LENGTH\_SHORT).show();  }  }); //支付按钮  **jie**.setOnClickListener(**new** View.OnClickListener() {  @Override  **public void** onClick(View v) {  showNormalDialog();  }  });  } //通过http接口和id获得相应的详情信息  **private void** getBookMessage(**int** bookid) {  **new** Thread(**new** Runnable() {  @Override  **public void** run() {  **try** {  **book** = HttpUtil.*getABook*(bookid);  showResponse();  } **catch** (Exception e) {  System.out.println(**"error"**);  e.printStackTrace();  }  }  }).start();  } //详情显示并将信息存入list中  **private void** showResponse() {  runOnUiThread(**new** Runnable() {  @Override  **public void** run() {  **bookname**.setText(**"书名:"** + **book**.getBname());  **bookdetail**.setText(**book**.getBdetail());  **booktype**.setText(**"类别:"** + **book**.getBtype());  **bookvalue**.setText(**"¥"** + String.valueOf(**book**.getBvalue()));  **bookwriter**.setText(**"作者:"** + **book**.getBwriter());  **bookxiao**.setText(**"销量:"** + String.valueOf(**book**.getXiao()));   **drawableList**.add(**new** BitmapDrawable(**book**.getPic1()));  **drawableList**.add(**new** BitmapDrawable(**book**.getPic2()));  **drawableList**.add(**new** BitmapDrawable(**book**.getPic3()));  **pagerAdapter** = **new** ViewPagerAdapter(**drawableList**, **mContext**);  **autoScrollViewPager**.setAdapter(**pagerAdapter**);  **bannerIndicator**.setUpWidthViewPager(**autoScrollViewPager**);  **autoScrollViewPager**.startAutoPlay();  }  });  } //弹出一个发布评论窗口  **private void** showCustomizeDialog() {  AlertDialog.Builder customizeDialog =  **new** AlertDialog.Builder(**mContext**);  **final** View dialogView = LayoutInflater.from(**mContext**)  .inflate(R.layout.***dia\_ping***, **null**);  customizeDialog.setTitle(**"评论"**);  customizeDialog.setView(dialogView);  customizeDialog.setPositiveButton(**"发布"**,  **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  MaterialEditText pp = dialogView.findViewById(R.id.***d\_ping***);  postPing(pp.getText().toString());  }  });  customizeDialog.show();  }   **private void** postPing(String ppp) {  **new** Thread(**new** Runnable() {  @Override  **public void** run() {  **try** {   Remark remark = **new** Remark(1, HttpUtil.*MyUser*.getUid(), **book**.getBid(), ppp);  Gson gson1 = **new** Gson();  String jsons = gson1.toJson(remark);  OkHttpClient client = **new** OkHttpClient();  RequestBody requestBody = **new** FormBody.Builder()  .add(**"remark"**, String.valueOf(jsons)).build();  Request request = **new** Request.Builder()  .post(requestBody)  .url(HttpUtil.*IP* + **"/postremark"**)  .build();  Response response = client.newCall(request).execute();  } **catch** (Exception e) {  System.out.println(**"error"**);  e.printStackTrace();  }  }  }).start();  } //支付提示  **private void** showNormalDialog() {  AlertDialog.Builder normalDialog =  **new** AlertDialog.Builder(**mContext**);  normalDialog.setTitle(**"是否支付"**);  normalDialog.setMessage(**"你确定要支付吗？"**);  normalDialog.setPositiveButton(**"确定"**,  **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {  Bill order = **new** Bill();  //修改订单信息  order.setBid(**book**.getBid());  order.setUid(HttpUtil.*MyUser*.getUid());  Date date = **new** Date();  //获取订单产生时间  SimpleDateFormat sdf = **new** SimpleDateFormat( **"yyyy-MM-dd HH:mm:ss"** );  String nowTime = sdf.format(date);  order.setTime(nowTime);  insertOrder(order);//插入订单列表  }  });  normalDialog.setNegativeButton(**"关闭"**,  **new** DialogInterface.OnClickListener() {  @Override  **public void** onClick(DialogInterface dialog, **int** which) {   }  });  normalDialog.show();  } //插入订单函数并将数据存入数据库  **private void** insertOrder(Bill order) {  **new** Thread(**new** Runnable() {  @Override  **public void** run() {  HttpUtil.*insertBill*(order);  HttpUtil.*addxiao*(**book**.getBid());  HttpUtil.*jianqian*(HttpUtil.*MyUser*.getUid(),**book**.getBvalue());  }  }).start();  } } | | | | | |