**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：胡浩然 学 号：201801060105**

**年 月 日**

**山 东 科 技 大 学 教 务 处 制**

**实 验 报 告**

**页**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **组 别** |  | **姓 名** | **胡浩然** | **同组实验者** |  |
| **实验项目**  **名称** | **BookStore大作业** | | **实验日期** | **1.1** | |
| **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。   1. **个人实现** 2. 实现litepal框架，通过litepal进行数据库的。      1. 实现购物车的实体类，其中记录书籍id和用户id。      1. 实现购物车界面布局   通过recylcerview展示购物车中的商品，下方支付按钮可以实现显示当前商品的总价格。  *<?*xml version="1.0" encoding="utf-8"*?>* <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"  xmlns:tools="http://schemas.android.com/tools"  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  tools:context=".FragmentCart">   <LinearLayout  android:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:orientation="vertical">  <TextView  android:layout\_width="match\_parent"  android:layout\_height="45dp"  android:gravity="center"  android:textSize="30dp"  android:text="购物车"/>  <View  android:layout\_width="match\_parent"  android:layout\_height="1dp"  android:background="#808080"/>  <androidx.recyclerview.widget.RecyclerView  android:id="@+id/recy\_gou"  android:layout\_width="match\_parent"  android:layout\_height="600dp"/>  <Button  android:id="@+id/zhifu"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:textColor="@android:color/white"  android:textStyle="bold"  android:background="@drawable/cart\_icon"  android:text="支付" />  </LinearLayout> </FrameLayout>   1. 通过litepal获取当前用户的购物车。      1. 创建子线程，利用购物车通过访问服务器端获取其购物车中所有的书籍信息      1. 根据获取的书籍信息，初始化recycleView，将购物车的信息显示出来        1. 应的通过book信息，动态设置recycleView的每个item，并通过谷歌Glide框架从服务器加载对应商品图片。      1. 点击对应商品进入商品的详情页面，在购物车适配器中实现      1. 点击每个Item的差号，将该商品从数据库中删除，并通过notifyDataSetChanged函数动态刷新recycleView。      1. 在适配其中分析RedioButton的点击状态，动态设置下方支付按钮的总价格。     （11）点击支付功能，对于选择商品进行删除，并通过网络，对用户余额、书籍销量、订单等通过网络进行对应修改，并将对应商品从android数据库中删除，并刷新recycleView。  private void SCDD() {  new Thread(new Runnable() {  @Override  public void run() {  list = new LinkedList<>();  for (int i = 0; i < bookList.size(); i++) {  if (isSlected.get(i) == false)  continue;  list.add(i);  Bill bill = new Bill();  bill.setBid(bookList.get(i).getBid());  bill.setUid(HttpUtil.*MyUser*.getUid());  DataSupport.*deleteAll*(Gou.class, "bid = ?", String.*valueOf*(bookList.get(i).getBid()));  Date date = new Date();  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  String nowTime = sdf.format(date);  bill.setTime(nowTime);  HttpUtil.*insertBill*(bill);  HttpUtil.*addxiao*(bookList.get(i).getBid());  HttpUtil.*jianqian*(HttpUtil.*MyUser*.getUid(),bookList.get(i).getBvalue());  }  updataUI1();  }  }).start(); }  购物车页面实现总代码：  package com.example.bookstore;  import android.app.AlertDialog; import android.content.DialogInterface; import android.content.Intent; import android.os.Bundle;  import androidx.annotation.NonNull; import androidx.fragment.app.Fragment; import androidx.recyclerview.widget.LinearLayoutManager; import androidx.recyclerview.widget.RecyclerView;  import android.view.LayoutInflater; import android.view.View; import android.view.ViewGroup; import android.widget.Button; import android.widget.ImageView; import android.widget.RadioButton; import android.widget.TextView; import android.widget.Toast;  import com.bumptech.glide.Glide;  import org.litepal.crud.DataSupport;  import java.io.IOException; import java.math.BigDecimal; import java.math.RoundingMode; import java.text.SimpleDateFormat; import java.util.Date; import java.util.LinkedList; import java.util.List;  public class FragmentCart extends Fragment {   private static final String *ARG\_PARAM1* = "param1";  private static final String *ARG\_PARAM2* = "param2";  private String mParam1;  private String mParam2;   List<Boolean> isSlected;  List<Book> bookList;  List<Gou> gouList;  List<Integer> list;  RecyclerView recyclerView;  LinearLayoutManager layoutManager;  AdapterGou adapterGou;  View view;  double Money;  Button zhifu;   public FragmentCart() {  }   public static FragmentCart newInstance(String param1, String param2) {  FragmentCart fragment = new FragmentCart();  Bundle args = new Bundle();  args.putString(*ARG\_PARAM1*, param1);  args.putString(*ARG\_PARAM2*, param2);  fragment.setArguments(args);  return fragment;  }   @Override  public void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  if (getArguments() != null) {  mParam1 = getArguments().getString(*ARG\_PARAM1*);  mParam2 = getArguments().getString(*ARG\_PARAM2*);  }  }   @Override  public View onCreateView(LayoutInflater inflater, ViewGroup container,  Bundle savedInstanceState) {  isSlected = new LinkedList<>();  bookList = new LinkedList<>();  gouList = new LinkedList<>();  view = inflater.inflate(R.layout.*fragment\_cart*, container, false);  recyclerView = view.findViewById(R.id.*recy\_gou*);  layoutManager = new LinearLayoutManager(view.getContext());  recyclerView.setLayoutManager(layoutManager);  zhifu = view.findViewById(R.id.*zhifu*);  Money=0;  zhifu.setText("¥" + "0.00");   gouList = DataSupport.*select*("bid")  .where("uid = ?", String.*valueOf*(HttpUtil.*MyUser*.getUid()))  .find(Gou.class);   getBooks();   zhifu.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  showNormalDialog();  }  });  return view;  }  private void showNormalDialog() {  AlertDialog.Builder normalDialog =  new AlertDialog.Builder(view.getContext());  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) {  Toast.*makeText*(view.getContext(),"支付成功",Toast.*LENGTH\_SHORT*).show();  SCDD();  }  });  normalDialog.setNegativeButton("关闭",  new DialogInterface.OnClickListener() {  @Override  public void onClick(DialogInterface dialog, int which) {   }  });  normalDialog.show();  }  private void SCDD() {  new Thread(new Runnable() {  @Override  public void run() {  list = new LinkedList<>();  for (int i = 0; i < bookList.size(); i++) {  if (isSlected.get(i) == false)  continue;   list.add(i);  Bill bill = new Bill();  bill.setBid(bookList.get(i).getBid());  bill.setUid(HttpUtil.*MyUser*.getUid());   DataSupport.*deleteAll*(Gou.class, "bid = ?", String.*valueOf*(bookList.get(i).getBid()));   Date date = new Date();  SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");  String nowTime = sdf.format(date);  bill.setTime(nowTime);   HttpUtil.*insertBill*(bill);  HttpUtil.*addxiao*(bookList.get(i).getBid());  HttpUtil.*jianqian*(HttpUtil.*MyUser*.getUid(),bookList.get(i).getBvalue());  }  updataUI1();  }  }).start();   }   private void updataUI1() {  getActivity().runOnUiThread(new Runnable() {  @Override  public void run() {  for(int i=0;i<list.size();i++)  {  bookList.remove(list.get(i)-i);  isSlected.remove(list.get(i)-i);  }  list.clear();  adapterGou.notifyDataSetChanged();  Money = 0;  zhifu.setText("¥" + "0.00");  }  });  }   private void getBooks() {  new Thread(new Runnable() {  @Override  public void run() {  try {  for (int i = 0; i < gouList.size(); i++) {  bookList.add(HttpUtil.*getOnlyBook*(gouList.get(i).getBid()));  }  updateUI();  } catch (IOException e) {  e.printStackTrace();  }  }  }).start();  }   void updateUI() {  getActivity().runOnUiThread(new Runnable() {  @Override  public void run() {  for (int i = 0; i < bookList.size(); i++) {  isSlected.add(false);  }  adapterGou = new AdapterGou(bookList);  recyclerView.setAdapter(adapterGou);  }  });  }   class AdapterGou extends RecyclerView.Adapter<AdapterGou.ViewHolder> {  private List<Book> bookList;   public class ViewHolder extends RecyclerView.ViewHolder {  ImageView Image;  TextView Name;  TextView Value;  TextView Writer;  ImageView Cha;  View productView;  RadioButton gouRaio;   public ViewHolder(View view) {  super(view);  Image = view.findViewById(R.id.*gou\_image*);  Name = view.findViewById(R.id.*gou\_bookname*);  Value = view.findViewById(R.id.*gou\_value*);  Writer = view.findViewById(R.id.*gou\_writer*);  gouRaio = view.findViewById(R.id.*gou\_radio*);  Cha = view.findViewById(R.id.*cha*);  productView = view;  }  }   public AdapterGou(List<Book> books) {  bookList = books;  }   @NonNull  @Override  public AdapterGou.ViewHolder onCreateViewHolder(@NonNull final ViewGroup parent, int viewType) {  View view = LayoutInflater.*from*(parent.getContext()).inflate(R.layout.*gou\_item*, parent, false);  final AdapterGou.ViewHolder holder = new AdapterGou.ViewHolder(view);  holder.productView.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  int position = holder.getAdapterPosition();  Intent intent = new Intent(parent.getContext(), BookDetail.class);  intent.putExtra("BookId", bookList.get(position).getBid());  parent.getContext().startActivity(intent);  }  });  return holder;  }   @Override  public void onBindViewHolder(@NonNull AdapterGou.ViewHolder holder, int position) {  Book book = bookList.get(position);  holder.Name.setText(book.getBname());  holder.Writer.setText(book.getBwriter());  Glide.*with*(holder.productView).load(HttpUtil.*IP*+ "/static/image/" + String.*valueOf*(book.getBid()) + "-1.jpg").into(holder.Image);  holder.Value.setText("¥" + String.*valueOf*(book.getBvalue()));  holder.gouRaio.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if (holder.gouRaio.isSelected()) {  holder.gouRaio.setSelected(false);  holder.gouRaio.setChecked(false);  Money -= bookList.get(position).getBvalue();  isSlected.set(position, false);  double vv = new BigDecimal(Money).setScale(2, RoundingMode.*DOWN*).doubleValue();  zhifu.setText("¥" + vv);  } else {  holder.gouRaio.setSelected(true);  Money += bookList.get(position).getBvalue();  isSlected.set(position, true);  double vv = new BigDecimal(Money).setScale(2, RoundingMode.*DOWN*).doubleValue();  zhifu.setText("¥" + vv);  }  }  });  holder.Cha.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  DataSupport.*deleteAll*(Gou.class, "bid = ?", String.*valueOf*(bookList.get(position).getBid()));  bookList.remove(position);  isSlected.remove(position);  notifyDataSetChanged();  }  });  }  @Override  public int getItemCount() {  return bookList.size();  }  } } | | | | | |