**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： 18-4**

**姓 名： 徐兴乐 学 号： 201801060422**

**2020 年 12 月 30日**

**山 东 科 技 大 学 教 务 处 制**

**实 验 报 告**

**页**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **组 别** |  | **姓 名** | **徐兴乐** | **同组实验者** |  |
| **实验项目**  **名称** | **Android综合型APP设计与实现** | | **实验日期** | **2020/12/30** | |
| **教师评语** |  | | | | |
| **实验成绩：** | | | **指导教师（签名）：**  **年 月 日** | | |
| **一、实验目的：**   根据Android APP的实际需求，运用课程所需知识点，并通过互联网学习扩展知识点，设计、开发满足需求的Android APP。  **二、实验内容：**  **基于Android的书店APP**  团队实现功能描述及系统设计：  1、功能描述：  1）在服务器端接受前端发出的请求，根据请求从数据库中筛选信息进行回传，实现多用户的同时登录。  2）用户登录、注册、找回密码功能，登录，注册使用的用户名、密码、邮箱地址存储在数据库中。  3）登录后有三秒的页面暂留，作为广告位，也可以跳过。  4）书城功能采用二级目录进行书的分类，点击每一本书后会显示存储在数据库书本的详细信息，并有两个功能“立即购买”与“加入购物车”，并相应的加入购物车与订单列表。  5）进入购物车页面，显示用户加入购物车的图书条目，每个条目前有一复选框，根据复选框的情况计算当前选中图书的总价，在购物车内购买选中图书后，删除已购买图书，总价清零。  6）我的界面可以从数据库user表获得用户名并显示，我的订单显示直接购买的记录。  2、系统设计：    个人实现功能：  1）在服务器端接收前端发出的请求，根据请求从数据库中筛选信息进行回传，实现多用户同时登录。  2）进入登录页面，根据底部导航，根据点击事件调用书城，购物车，我的三个碎片。  3）进入书城页面，设置二级目录，每个大类包含六个小类，点击每个小类显示该类型书的条目，点击后任一条目跳转至该书的详情页面。  4）进入购物车页面，显示用户加入购物车的图书条目，每个条目前有一复选框，根据复选框的情况计算当前选中图书的总价。  5）在购物车内购买选中图书后，删除已购买图书，总价清零。  界面展示：      **三、核心代码：**  服务器代码：  Usercontroller：  @RestController public class UserController {  public String myname;  @Autowired  private UserMapper userMapper;  @Autowired  private OrderMapper orderMapper;  @Autowired  private CarMapper carMapper;   @RequestMapping(value ="/login",method = RequestMethod.*POST*)  @ResponseBody  public String Verify(HttpServletRequest request){  String username = request.getParameter("username");  String password = request.getParameter("password");  User u = new User(username,password);  List<User> user = userMapper.SelectUsersByParam(username,password);  myname = username;  if(user.size()==0){  return "0";  }  else {  return "1";  }  }  //register  @RequestMapping(value ="/register",method = RequestMethod.*POST*)  @ResponseBody  public String Register(HttpServletRequest request){  String username = request.getParameter("username");  String password = request.getParameter("password");  String email = request.getParameter("email");  User u = new User(username,password);  List<User> user = userMapper.SelectUsersByParam(username,password);  if(user.size()==0){  userMapper.InsertUser(username,password,email);  return "0";  }  else {  return "1";  }  }  //  @RequestMapping(value ="/find",method = RequestMethod.*POST*)  @ResponseBody  public String Find(HttpServletRequest request){  String email = request.getParameter("email");  System.*out*.println(email);  List<User> user = userMapper.SelectUsersByEmail(email);  System.*out*.println(user.size());  if(user.size()==0){  return "0";  }  else {  return user.get(0).getPassword();  }  }  @RequestMapping(value ="/addorder",method = RequestMethod.*POST*)  @ResponseBody  public String AddMyOrder(HttpServletRequest request){  String myorder = request.getParameter("myorder");  Order order = new Order();  System.*out*.println(myname);  orderMapper.InsertOrder(myname,myorder);  return "1";  }  @RequestMapping(value ="/getorder",method = RequestMethod.*POST*)  @ResponseBody  public String GetMyOrder(HttpServletRequest request){  List<Order> orderList = orderMapper.SelectOrdersByName(myname);  String ans = "";  for(int i = 0;i<orderList.size();i++){  ans += orderList.get(i).getMyorder();  if(i!=orderList.size()-1){  ans+="+";  }  }  return ans+"&"+myname;  }  @RequestMapping(value ="/addcar",method = RequestMethod.*POST*)  @ResponseBody  public String AddCar(HttpServletRequest request){  String mycar = request.getParameter("mycar");  System.*out*.println(mycar);  carMapper.InsertCar(myname,mycar);  return "1";  }  @RequestMapping(value ="/getcar",method = RequestMethod.*POST*)  @ResponseBody  public String GetCrder(HttpServletRequest request){  List<Car> carList = carMapper.SelectCarsByName(myname);  String ans = "";  for(int i = 0;i<carList.size();i++){  ans += carList.get(i).getCarinfo();  if(i!=carList.size()-1){  ans+="+";  }  }  return ans;  }  @RequestMapping(value ="/deletecar",method = RequestMethod.*POST*)  @ResponseBody  public String DeleteCar(HttpServletRequest request){  String de = request.getParameter("delete");  List<Car> carList = carMapper.SelectCarsByName(myname);  for(int i = 0;i<carList.size();i++){  Car tmp = carList.get(i);  String st[] = tmp.getCarinfo().split(" ");  String str[] = de.split(" ");  for(int j = 0;j<str.length;j++){  if(st[2].equals(str[j])){  carMapper.DeleteCar(myname,tmp.getCarinfo());  }  }  }   return "1";  } }  Bookcontroller：  @RestController public class BookController {  public String imagePath = "";  @Autowired  private BookMapper bookMapper;   @RequestMapping(value ="/search1",method = RequestMethod.*POST*)  @ResponseBody  public String Search(HttpServletRequest request){  String bookk = request.getParameter("bookk");  System.*out*.println(bookk);  List<Book> book = bookMapper.SelectBooksByBookk(bookk);  System.*out*.println(book.size());  String ans = "";  for(int i = 0;i<book.size();i++){  Book tmp = new Book();  tmp = book.get(i);  ans += tmp.getBookname() +" "+tmp.getBookprice()+  " "+tmp.getBookauthor()+ " " +tmp.getBookpress()+  " "+ tmp.getBookkind()+" " + tmp.getImageid();  if(i!=book.size()-1){  ans+="+";  }  }  System.*out*.println(ans);  return ans;  } }  安卓代码：  底部导航：  public class BookFen extends AppCompatActivity implements View.OnClickListener{  private FragmentManager fragmentManager;  public LinearLayout item1,item2,item3;  public ImageView img1,img2,img3;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_book\_fen*);  initView();  fragmentManager = getSupportFragmentManager();  initListener();  }   private void initListener() {  item1.setOnClickListener(this);  item2.setOnClickListener(this);  item3.setOnClickListener(this);  }   private void initView() {  item1 = findViewById(R.id.*item1*);  img1 = findViewById(R.id.*img\_shu*);  item2 = findViewById(R.id.*item2*);  img2 = findViewById(R.id.*img\_car*);  item3 = findViewById(R.id.*item3*);  img3 = findViewById(R.id.*img\_my*);  }  @Override  public void onClick(View view) {  switch (view.getId()){  case R.id.*item1*: {  FragmentTransaction transaction = fragmentManager.beginTransaction();  transaction.replace(R.id.*fag*,new fenlei());  transaction.commit();  setCheck(0);  break;  }  case R.id.*item2*: {  FragmentTransaction transaction = fragmentManager.beginTransaction();  transaction.replace(R.id.*fag*,new shopcart());  transaction.commit();  setCheck(1);  break;  }  case R.id.*item3*: {  FragmentTransaction transaction = fragmentManager.beginTransaction();  transaction.replace(R.id.*fag*,new person());  transaction.commit();  setCheck(2);  break;  }  default:break;  }  }   private void setCheck(int i) {  img1.setColorFilter(Color.*parseColor*("#0f0f0f"));  img2.setColorFilter(Color.*parseColor*("#0f0f0f"));  img3.setColorFilter(Color.*parseColor*("#0f0f0f"));  if(i==0){  img1.setColorFilter(Color.*BLUE*);  }  else if(i==1){  img2.setColorFilter(Color.*BLUE*);  }  else {  img3.setColorFilter(Color.*BLUE*);  }  } }  二级目录实现：  左碎片：  public class LeftFragment extends Fragment{  private String[] s\_lei = {"软件开发 人工智能 操作系统 信息安全 数据库 计算机硬件",  "语言文学 社会学 民俗文化 文化评述 地域文化 新闻传播",  "市场营销 电子商务 金融投资 商业传史 商务沟通 会计计算",  "学生教材 课外读物 作文辅导 练习手册 字典 经典名著",  "战略管理 企业管理 文化管理 城市管理 旅游管理 文化管理",  "人生哲学 心灵与修养 性格与习惯 名人励志 人在职场 人际交往"};  public List<String> leiname = new ArrayList<>();  public View onCreateView(LayoutInflater inflater, ViewGroup container,  Bundle savedInstance){  View view = inflater.inflate(R.layout.*activity\_left\_fragment*,container,false);  ListView listView = view.findViewById(R.id.*da\_lei*);  init();  LeiAdapter adapter = new LeiAdapter(getActivity(),R.layout.*activity\_lei\_adapter*,leiname);  listView.setAdapter(adapter);  listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  @Override  public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  RightFragment rightFragment = (RightFragment)getFragmentManager().findFragmentById(R.id.*right\_lei*);  rightFragment.refresh(s\_lei[position],position);  }  });  return view;  }   private void init() {  leiname.add("计算机/网络");  leiname.add("人文社科");  leiname.add("经济管理");  leiname.add("教育");  leiname.add("管理");  leiname.add("成功励志");  leiname.add("生活");  leiname.add("中小学教辅");  leiname.add("工具书");  leiname.add("小说");  leiname.add("名人传记");  leiname.add("艺术");  leiname.add("古籍");  leiname.add("法律");  leiname.add("心理学");  leiname.add("童书");  }  }  右碎片：  public class RightFragment extends Fragment {  private View view;  public int id;  public String idx;  public String ans;  Button s\_lei1,s\_lei2,s\_lei3,s\_lei4,s\_lei5,s\_lei6;  public View onCreateView(LayoutInflater inflater, ViewGroup container,  Bundle savedInstance) {  view = inflater.inflate(R.layout.*activity\_right\_fragment*, container, false);  s\_lei1 = view.findViewById(R.id.*s\_lei1*);  s\_lei2 = view.findViewById(R.id.*s\_lei2*);  s\_lei3 = view.findViewById(R.id.*s\_lei3*);  s\_lei4 = view.findViewById(R.id.*s\_lei4*);  s\_lei5 = view.findViewById(R.id.*s\_lei5*);  s\_lei6 = view.findViewById(R.id.*s\_lei6*);  s\_lei1.setOnClickListener(new View.OnClickListener(){  public void onClick(View view){  idx = id + "1";  submitWithOkHttp();  }  });  s\_lei2.setOnClickListener(new View.OnClickListener(){  public void onClick(View view){  idx = id + "2";  submitWithOkHttp();  }  });  s\_lei3.setOnClickListener(new View.OnClickListener(){  public void onClick(View view){  idx = id + "3";  submitWithOkHttp();  }  });  s\_lei4.setOnClickListener(new View.OnClickListener(){  public void onClick(View view){  idx = id + "4";  submitWithOkHttp();  }  });  s\_lei5.setOnClickListener(new View.OnClickListener(){  public void onClick(View view){  idx = id + "5";  submitWithOkHttp();  }  });  s\_lei6.setOnClickListener(new View.OnClickListener(){  public void onClick(View view){  idx = id + "6";  submitWithOkHttp();  }  });   return view;  }  private void submitWithOkHttp() {  ans = "";  new Thread(new Runnable() {  @Override  public void run() {  try{  OkHttpClient client = new OkHttpClient();  RequestBody requestBody = new FormBody.Builder()  .add("bookk",idx)  .build();  Request request = new Request.Builder()  .url("http://10.17.112.177:8080/search1")  .post(requestBody)  .build();  Response response = client.newCall(request).execute();  String responseData = response.body().string();  System.*out*.println(responseData);  Intent intent= new Intent(getActivity(),BookShow.class);  intent.putExtra("ans",responseData);  startActivity(intent);  ans = responseData;  } catch (Exception e) {  e.printStackTrace();  }  }  }).start();  }  public void refresh(String s\_lei,int x){  id = x;  String str[] = s\_lei.split(" ");  s\_lei1 = view.findViewById(R.id.*s\_lei1*);  s\_lei2 = view.findViewById(R.id.*s\_lei2*);  s\_lei3 = view.findViewById(R.id.*s\_lei3*);  s\_lei4 = view.findViewById(R.id.*s\_lei4*);  s\_lei5 = view.findViewById(R.id.*s\_lei5*);  s\_lei6 = view.findViewById(R.id.*s\_lei6*);  s\_lei1.setText(str[0]);  s\_lei2.setText(str[1]);  s\_lei3.setText(str[2]);  s\_lei4.setText(str[3]);  s\_lei5.setText(str[4]);  s\_lei6.setText(str[5]);  }  }  图书显示：  public class BookShow extends AppCompatActivity{  public List<Book> bookList = new ArrayList<>();  public String ans;  public String title = "";  public ListView listView;  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_book\_show*);  Intent intent = getIntent();  ans = intent.getStringExtra("ans");  submitWithOkHttp();   }  private void submitWithOkHttp() {   new Thread(new Runnable() {  @Override  public void run() {  try{  String str[] = ans.split("\\+");  for(int i = 0;i<str.length;i++){  String now[] = str[i].split(" ");  String url1 ="http://10.17.112.177:8080/"+now[5]+".jpg";  OkHttpClient client1 = new OkHttpClient();  Request request1 = new Request.Builder()  .url(url1)  .build();  Response response1 = null;  try {  response1 = client1.newCall(request1).execute();  } catch (IOException e) {  e.printStackTrace();  }  InputStream inputStream1 = response1.body().byteStream();//得到图片的流  Bitmap bitmap1 = BitmapFactory.*decodeStream*(inputStream1);  Book tmp = new Book(now[0],now[1],now[2],now[3],now[4],now[5],bitmap1);  title = now[4];  bookList.add(tmp);  }  toastTest();   } catch (Exception e) {  e.printStackTrace();  }  }  }).start();  }  private void toastTest() {  runOnUiThread(new Runnable() {  @Override  public void run() {  TextView lei\_title = findViewById(R.id.*lei\_title*);  lei\_title.setText(title);  listView = findViewById(R.id.*show\_list*);  BookAdapter bookAdapter = new BookAdapter(BookShow.this,R.layout.*activity\_book\_adapter*,bookList);  listView.setAdapter(bookAdapter);  listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {  @Override  public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  Book nex = bookList.get(position);  String detail = nex.getBookname()+" "+nex.getBookauthor()+" "+nex.getBookpress()  +" "+nex.getBookkind()+" "+nex.getBookprice()+" "+bitmapToString(nex.getImg()) +" "+nex.getImageid();  Intent intent = new Intent(BookShow.this,BookDetail.class);  intent.putExtra("detail",detail);  startActivity(intent);  }  });  }  });  }  public String bitmapToString(Bitmap bitmap){  //将Bitmap转换成字符串  String string=null;  ByteArrayOutputStream bStream=new ByteArrayOutputStream();  bitmap.compress(Bitmap.CompressFormat.*PNG*,100,bStream);  byte[]bytes=bStream.toByteArray();  string= Base64.*encodeToString*(bytes,Base64.*DEFAULT*);  return string;  }  }  购物车实现：  public class CarDetail extends AppCompatActivity {  public List<Car> carList = new ArrayList<>();  public ListView listView;  public TextView car\_all;  public String ans = "";  CarAdapter carAdapter;  public List<String> de = new ArrayList<>();  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.*activity\_car\_detail*);  submitWithOkHttp();  Button pay = findViewById(R.id.*pay*);  pay.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  for(int i = 0;i<de.size();i++){  ans+=de.get(i);  if(i!=de.size()-1){  ans+=" ";  }  for(int j = 0;j<carList.size();j++){  if(de.get(i).equals(carList.get(j).getImageid())){  carList.remove(j);  }  }  }  killitWithOkHttp();  car\_all.setText("0");  Toast.*makeText*(getApplicationContext(), "购买成功", Toast.*LENGTH\_SHORT*).show();  carAdapter.notifyDataSetChanged();  }  });  }  private void killitWithOkHttp(){  new Thread(new Runnable() {  @Override  public void run() {  try{  OkHttpClient client = new OkHttpClient();  RequestBody requestBody = new FormBody.Builder()  .add("delete",ans)  .build();  Request request = new Request.Builder()  .url("http://10.17.112.177:8080/deletecar")  .post(requestBody)  .build();  Response response = client.newCall(request).execute();  String responseData = response.body().string();  } catch (Exception e) {  e.printStackTrace();  }  }  }).start();  }  private void submitWithOkHttp() {  new Thread(new Runnable() {  @Override  public void run() {  try{  OkHttpClient client = new OkHttpClient();  RequestBody requestBody = new FormBody.Builder()  .add("g\_car","give me car")  .build();  Request request = new Request.Builder()  .url("http://10.17.112.177:8080/getcar")  .post(requestBody)  .build();  Response response = client.newCall(request).execute();  String responseData = response.body().string();  String str[] = responseData.split("\\+");   for(int i = 0;i<str.length;i++){  String nex[] = str[i].split(" ");  System.*out*.println(str[i]);  Car tmp = new Car();  tmp.setCar\_name(nex[0]);  tmp.setCar\_price(nex[1]);  tmp.setImageid(nex[2]);  String url1 ="http://10.17.112.177:8080/"+nex[2]+".jpg";  request = new Request.Builder()  .url(url1)  .build();  response = null;  try {  response = client.newCall(request).execute();  } catch (IOException e) {  e.printStackTrace();  }  InputStream inputStream1 = response.body().byteStream();//得到图片的流  Bitmap bitmap = BitmapFactory.*decodeStream*(inputStream1);  System.*out*.println(bitmap);  tmp.setCar\_image(bitmap);  carList.add(tmp);  }  toastTest();  } catch (Exception e) {  e.printStackTrace();  }  }  }).start();  }   private void toastTest() {  runOnUiThread(new Runnable() {  @Override  public void run() {  car\_all = findViewById(R.id.*car\_zong*);  listView = findViewById(R.id.*car\_list*);  car\_all.setText("0");  carAdapter = new CarAdapter(CarDetail.this,R.layout.*activity\_car\_adapter*,carList);  listView.setAdapter(carAdapter);  }  });  }   public class CarAdapter extends ArrayAdapter<Car> {  private int resourceId;  public CarAdapter(@NonNull Context context, int resourceId, @NonNull List<Car> objects) {  super(context, resourceId, objects);  this.resourceId = resourceId;  }  public View getView(final int position, View convertView, ViewGroup parent){  final Car car = getItem(position);  View view;  final ViewHolder viewHolder;  if(convertView == null){  view = LayoutInflater.*from*(getContext()).inflate(resourceId,parent,false);  viewHolder = new ViewHolder();  viewHolder.carimage = view.findViewById(R.id.*car\_image*);  viewHolder.carname = (TextView) view.findViewById(R.id.*car\_name*);  viewHolder.carprice = (TextView) view.findViewById(R.id.*car\_price*);  viewHolder.carcheck = view.findViewById(R.id.*car\_select*);  view.setTag(viewHolder);  }  else{  view = convertView;  viewHolder = (ViewHolder) view.getTag();  }   viewHolder.carcheck.setOnClickListener(new View.OnClickListener() {  @Override  public void onClick(View v) {  if(viewHolder.carcheck.isChecked()){  Car cc = getItem(position);  de.add(cc.getImageid());  String str[] = car.getCar\_price().split("人");  double x = Double.*parseDouble*(str[0]);  double y = Double.*parseDouble*(car\_all.getText().toString());  y = x+y;  String k = String.*format*("%.2f", y);  car\_all.setText(k);  }  else {  Car cc = getItem(position);  for(int i = de.size()-1;i>=0;i--){  if(cc.getImageid().equals(de.get(i))){  de.remove(i);  }  }  String str[] = car.getCar\_price().split("人");  double x = Double.*parseDouble*(str[0]);  double y = Double.*parseDouble*(car\_all.getText().toString());  y = y - x;  String k = String.*format*("%.2f", y);  car\_all.setText(k);  }  }  });  viewHolder.carimage.setImageBitmap(car.getCar\_image());  viewHolder.carname.setText(car.getCar\_name());  viewHolder.carprice.setText("¥"+car.getCar\_price());   return view;  }   class ViewHolder{  CheckBox carcheck;  ImageView carimage;  TextView carname;  TextView carprice;  }  }  } | | | | | |