****

安卓课程设计报告

——仿12306

成员： 李佳敏 张燕华

班级： 计算机163

指导老师： 邓飞

日期： 2019.1.4

1. **设计背景****及开发环境**

**1、设计背景**

随着交通行业以及旅游行业的发展，越来越多的人需要利用各种交通工具前往不同的地方。因此，如何便利的订票购票，是每个人共同的需求。在智能手机普及的今天，一个火车订票APP应该提供及时、广泛的信息服务，加快信息的更新效率，及时查询火车票信息，防止引起出行不便。在手机上直接购票可以减少排队买票的时间和精力，也避免了网上购票的不及时性，满足了广大群众随时随地购票的愿望。在现代21世纪智能手机普及的今天十分有必要。我们的项目主要针对的是火车票的查询与购买

由上述背景可知，手机移动购票已经成为大家的日常所需。设计一个快捷的方便的订票app很有必要。我们的app就是在此需求下诞生的。我们的目标是更快更方便的查询火车票信息，方便人们的购票和出行，由此，我们设计了这个仿12306的购票app。

**2、开发环境**

Eclipse、Android studio、mysql

1. **需求：功能清单**

**1、实现登录、注册界面；**

实现注册按钮动作、实现登录按钮动作；

1.1登录：

1.2:注册



**2、首页（查询界面）**，实现该界面；（结合下面两图）

实现起始地按钮、终点按钮的动作；

实现出发日期选择动作

实现查询按钮



**3、实现查询结果界面；**



**4、实现日期选择界面;**



**5、确认订单界面；**(将查询结果里面listview的item选择点击结果返回出来)

实现确认订单按钮



**6、实现确认支付界面**，和下面继续支付和取消订单按钮

实现返回所订票的信息：

（若取消订单则返回首页订票界面）



**8、实现订单成功提交界面**及各按钮



**9、实现选择查询车站列表界面**



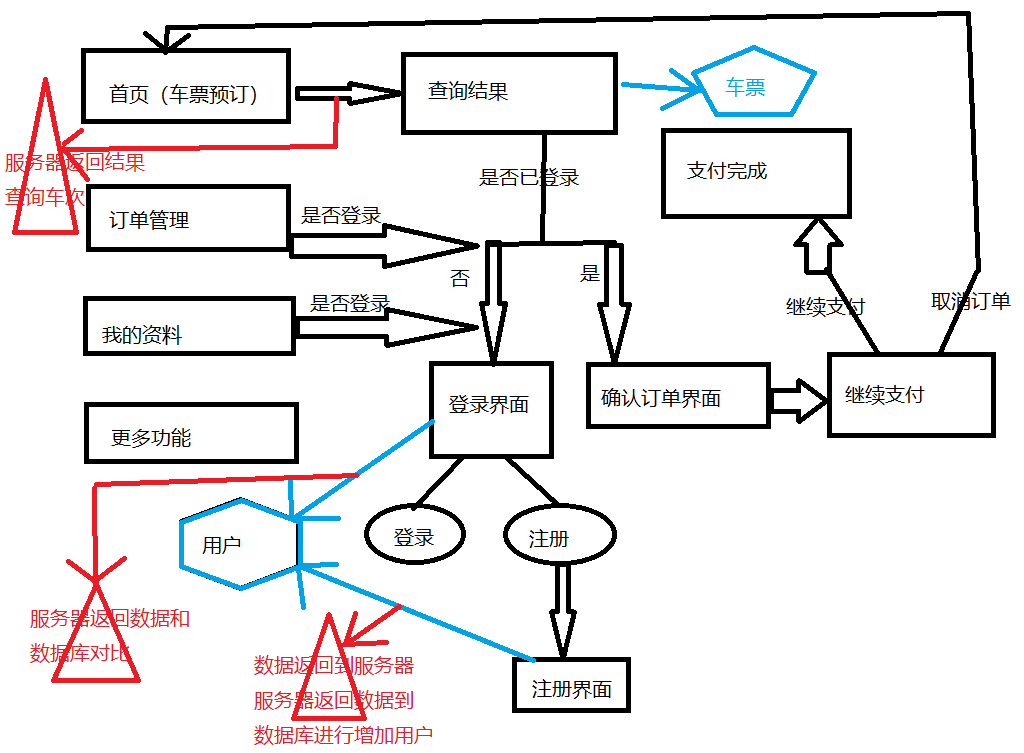
1. **完成情况小结**

|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 功能 | 功能明细 | 是否实现 |
| 1 | 输入查询条件 | 1. 点击起点、终点 2. 点击日期，切换日期 | 实现1) 2) |
| 2 | 查询 | 1. 点击“查询”按钮，进入查询结果页面 | 实现1) |
| 3 | 功能按钮 | 1. 分为“首页”（车票预订）、”订单管理“、“我的资料”、“更多功能”四块 | 实现1） |
|  |  |  |  |

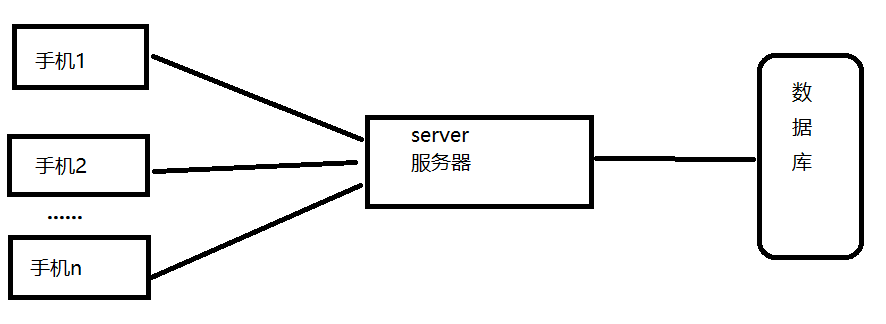
|  |  |  |  |
| --- | --- | --- | --- |
| 编号 | 功能 | 功能明细 | 是否实现 |
| 1 | 查询条件切换 | 1. 交换发到站 2. 切换日期 3. 前一天、后一天   A)切换过快，会有提示  4）网络错误提示 | 实现2）4） |
| 2 | 排序 | 按发时、历时、到时、票价等排序 | 实现发时和到时(未实现，按数据库信息顺序直接显示在列表上) |
| 4 | 选择车票 | 订票，进入订单模块   1. 未登录，要求登录 2. 确认订单，显示车票信息在确认订单界面 | 实现1）、2） |

1. **主体框架**

**（1）、大的架构**

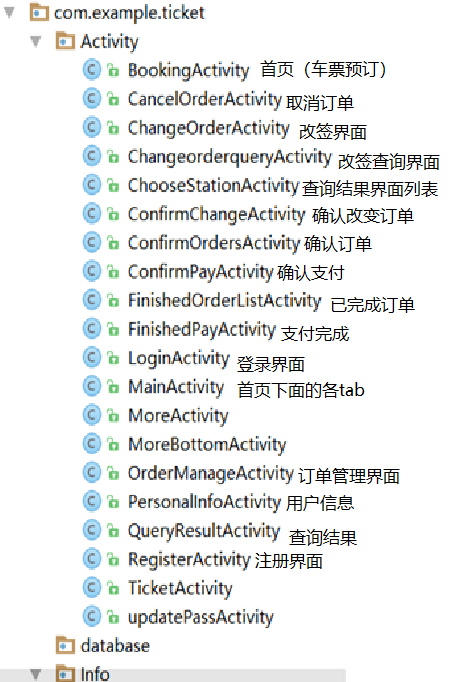


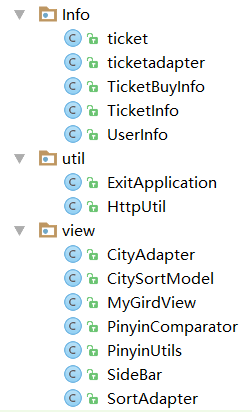
**（2）、数据流向**

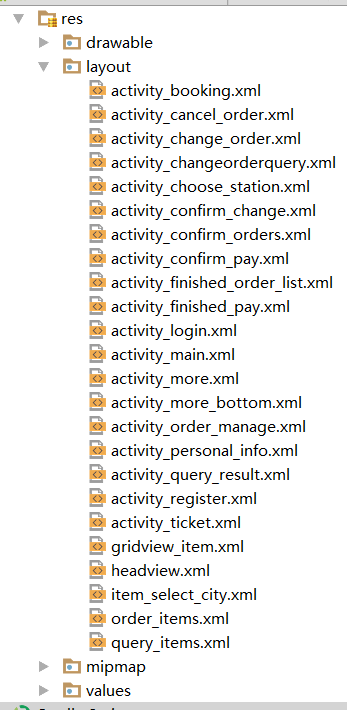


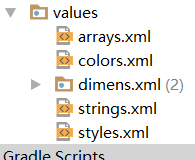
**（3）、文件清单**

**Android端：**

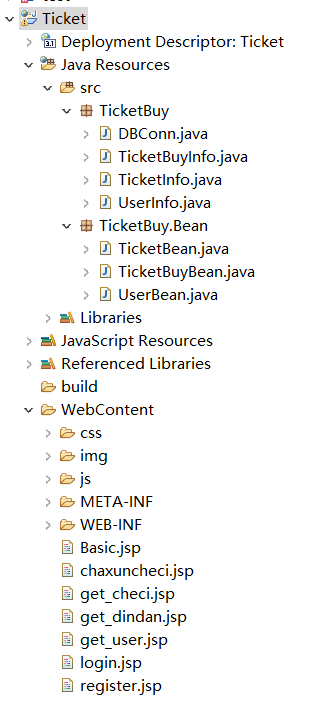




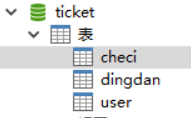




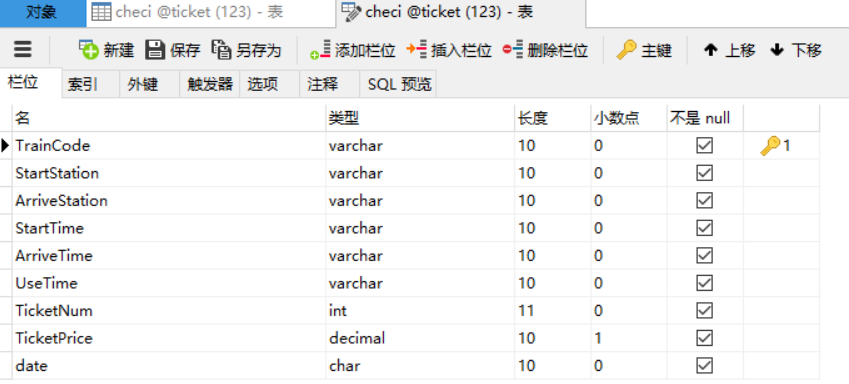
**Eclipse端：**



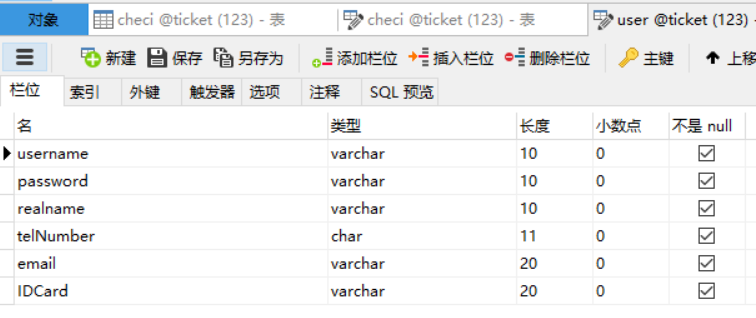
**数据库端：**



**车次表（checi）：**



**用户表（user）：**



1. **使用说明**

首先可以在首页先进行查询车票信息（选择相应的车站和时间进行查询），

然后点击查询按钮可以看见所选择的条件下的各车次信息，可以点击各车次进行购票（若已经登录则直接跳转到确认订单界面，若未登录的情况下，则会跳转到登录界面，要求进行登录。）

在跳转到确认订单界面的时候，会显示你所选择的车次的具体车票信息，并且是否确认支付；

在点击确认支付后将跳转到确认支付界面，然后会将车票信息显示在此界面上，并且点击继续支付按钮可以进行继续支付操作

接下来会跳转到支付完成界面，由此，购买车票完成。

此外，一开始在订单管理和我的资料界面显示的是各界面，若在未登录情况下点击这两个界面，显示的将是登录界面，你可以选择登录（若还没有账号的情况下，可以选择注册，注册之后，你的注册信息将通过客户端上传到服务器上，再在服务器上传返回到数据库中，存储你的注册信息，并在下一次登录的时候可以调用。）

1. **小组工作情况**
2. **每个人的工作**
3. 李佳敏：（主要工作）

进行eclipse服务器端订单查询块及订票块的代码编写

进行android studio的订单查询、查询结果、确认支付功能、部分确认订单界面代码编写

进行mysql数据库表中数据的填充

进行课设文档的编写

（2）张燕华：（主要工作）

进行eclipse服务器端java代码编写及注册、登录jsp编写；

进行android studio客户端的注册、登录功能、部分确认订单界面编写

进行mysql数据库及表的创建

进行doxygen的编写

1. **评估百分比贡献**

每人各50%

1. **使用技术**

1、使用json、gson、okhttp进行订单查询

Android端代码：(部分)

**private** String **date**;  
**private** ListView **lv**;  
**public** ArrayList<ticket> **listticket**;  
ticket **ticketBean**;

**lv** = (ListView)findViewById(R.id.***query\_listview***);  
init();

*//获取json数据部分***private void** init(){  
 **new** Thread(**new** Runnable() {  
 @Override  
 **public void** run() {  
 OkHttpClient okHttpClient = **new** OkHttpClient();  
 Intent intent = getIntent();  
 String startStation = intent.getStringExtra(**"from"**).toString().trim();  
 String arriveStation = intent.getStringExtra(**"to"**).toString().trim();  
 String datel = intent.getStringExtra(**"datel"**).toString().trim();  
 RequestBody requestBody = **new** FormBody.Builder()  
 .add(**"StartStation"**,startStation)  
 .add(**"ArriveStation"**,arriveStation)  
 .add(**"date"**,datel)  
 .build();  
 *//服务器返回的地址* Request request = **new** Request.Builder()  
 .url(**"http://10.0.2.2:8080/Ticket/chaxuncheci.jsp"**)  
 .post(requestBody)  
 .build();  
 **try** {  
 Response response = okHttpClient.newCall(request).execute();  
 *//获取到数据* **date** = response.body().string();  
 *//把数据传入解析json数据方法* Message message = **new** Message();  
 message.**what** = 1;  
 **han**.sendMessage(message);  
 Gsonjx(**date**);  
 }**catch** (IOException e){  
 e.printStackTrace();  
 }  
 }  
 }).start();  
}  
  
**public** Handler **han** = **new** Handler(){  
 @Override  
 **public void** handleMessage(Message msg) {  
 **switch** (msg.**what**){  
 **case** 1:  
 *//Toast.makeText(QueryResultActivity.this,""+date,Toast.LENGTH\_LONG).show();* **break**;  
 **case** 2:  
 **lv**.setAdapter(**new** ticketadapter(QueryResultActivity.**this**,**listticket**));*//调用listview适配器* **break**;  
 }  
 }  
};  
  
**private void** Gsonjx(String date){  
 JsonParser parser = **new** JsonParser();  
 JsonArray jsonArray = parser.parse(date).getAsJsonArray();  
 Gson gson = **new** Gson();  
 **listticket** = **new** ArrayList<>();  
 **for** (JsonElement ticket:jsonArray){  
 **ticketBean** = gson.fromJson(ticket,ticket.**class**);  
 **listticket**.add(**ticketBean**);  
 }  
 Message message = **new** Message();  
 message.**what** = 2;  
 **han**.sendMessage(message);  
}

Eclipse端代码（部分）：

// 读取数据

request.setCharacterEncoding("UTF-8");

String StartStation = request.getParameter("StartStation");

String ArriveStation = request.getParameter("ArriveStation");

String date = request.getParameter("date");

TicketBean bean = **new** TicketBean();

//Connection conn = DBConn.getConnection();

List<TicketInfo> data = **new** ArrayList<TicketInfo>();

data = bean.getResult(StartStation, ArriveStation, date);

System.out.println("json");

System.out.println("]");

//封装成JSON

out.println("[");

Iterator< TicketInfo > iter = data.iterator();

**while**( iter.hasNext() ){

TicketInfo info = iter.next();

out.println("{");

out.println("\"TrainCode\":\"" + info.getTrainCode() + "\",");

out.println("\"StartStation\":\"" + info.getStartStation() + "\",");

out.println("\"ArriveStation\":\"" + info.getArriveStation() + "\",");

out.println("\"StartTime\":\"" + info.getStartTime() + "\",");

out.println("\"ArriveTime\":\"" + info.getArriveTime() + "\",");

out.println("\"UseTime\":\"" + info.getUseTime() + "\",");

out.println("\"TicketNum\":\"" + info.getTicketNum() + "\",");

out.println("\"TicketPrice\":\"" + info.getTicketPrice() + "\",");

out.println("\"date\":\"" + info.getDate()+"\"");

out.println("}");

**if**( iter.hasNext() ){

out.print(",");

}

}

out.println("]");

%>

2、使用okhttp进行用户注册代码：（部分）

安卓端：

**registerBtn**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 *//获取输入信息  
 //验证用户名，真实姓名，手机号码，邮箱，身份证是否已经被注册  
 //将信息添加到数据库  
 //跳转到下一个页面显示* String res\_name = **name**.getText().toString().trim();  
 String res\_pass1 = **pass1**.getText().toString().trim();  
 String res\_pass2 = **pass2**.getText().toString().trim();  
 String res\_realname = **realname**.getText().toString().trim();  
 String res\_phone = **phone**.getText().toString().trim();  
 String res\_email = **email**.getText().toString().trim();  
 String res\_idcard = **idCard**.getText().toString().trim();  
 **if**(res\_name.equals(**""**)){  
 Looper.*prepare*();  
 Toast.*makeText*(RegisterActivity.**this**,**"用户名不能为空"**,Toast.***LENGTH\_SHORT***).show();  
 Looper.*loop*();  
 }**else if**(!res\_pass1.equals(res\_pass2)){  
 Looper.*prepare*();  
 Toast.*makeText*(RegisterActivity.**this**,**"密码不一致"**,Toast.***LENGTH\_SHORT***).show();  
 Looper.*loop*();  
 }**else**{  
 RequestBody requestBody = **new** FormBody.Builder()  
 .add(**"username"**,res\_name)  
 .add(**"password"**,res\_pass1)  
 .add(**"realname"**,res\_realname)  
 .add(**"telNumber"**,res\_phone)  
 .add(**"email"**,res\_email)  
 .add(**"IDCard"**,res\_idcard).build();  
 HttpUtil.*getByOkHttp*(**"http://10.0.2.2:8080/Ticket/register.jsp"**, requestBody, **new** Callback() {  
 @Override  
 **public void** onFailure(Call call, IOException e) {  
 Show(**"Fail"**);  
  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
 String data = response.body().string().trim();  
 **if**(data.equals(**"1"**)){  
 Looper.*prepare*();  
 Toast.*makeText*(RegisterActivity.**this**,**"注册成功"**,Toast.***LENGTH\_SHORT***).show();  
 Intent intent = getIntent();  
 intent.setClass(RegisterActivity.**this**,PersonalInfoActivity.**class**);  
 startActivity(intent);  
 Looper.*loop*();  
 }  
 }  
 });  
 }

//查询结果由起始站、终点站、和时间决定

**public** ArrayList<TicketInfo>getResult(String StartStation,String ArriveStation,String date){

ArrayList<TicketInfo> data=**new** ArrayList<TicketInfo>();

Connection conn=DBConn.*getConnection*();

**try** {

String sql = "select TrainCode,StartStation, ArriveStation, "

+ "StartTime, ArriveTime, "

+ " UseTime, TicketNum,TicketPrice,date"

+ " from checi"

+ " where StartStation like ?"

+ " and ArriveStation like ?"

+ " and date like ?";

/\*Statement st = conn.createStatement();

ResultSet rs = st.executeQuery(sql);\*/

java.sql.PreparedStatement st =

conn.prepareStatement( sql );

st.setString(1, StartStation);

st.setString(2, ArriveStation);

st.setString(3, date);

ResultSet rs = st.executeQuery();

System.***out***.println( "查询结果为：" );

**while**( rs.next() ){

String TrainCode = rs.getString("TrainCode");

String StartTime = rs.getString("StartTime");

String ArriveTime = rs.getString("ArriveTime");

String UseTime = rs.getString("UseTime");

**int** TicketNum = rs.getInt("TicketNum");

**double** TicketPrice = rs.getDouble("TicketPrice");

TicketInfo info = **new** TicketInfo();

info.setTrainCode(TrainCode);

info.setStartStation(StartStation);

info.setArriveStation(ArriveStation);

info.setStartTime(StartTime);

info.setArriveTime(ArriveTime);

info.setUseTime(UseTime);

info.setTicketNum(TicketNum);

info.setTicketPrice(TicketPrice);

info.setDate(date);

data.add( info );

String str = String.*format*(

"(%s,%s, %s, %s,%s,%s,%d,%.1f,%s)\n",

TrainCode,StartStation, ArriveStation, StartTime, ArriveTime,UseTime,TicketNum,TicketPrice,date);

System.***out***.printf( str );

}

} **catch** (SQLException e) {

e.printStackTrace();

System.***out***.println("数据库查询失败\n"+e.getMessage());

// **TODO**: handle exception

}**finally** {

**if**( conn != **null** ){

**try**{

conn.close();

}

**catch**( SQLException e ){

System.***out***.printf( "关闭连接失败\n" + e.getMessage() );

}// try

}// if

}//finally

**return** data;

}

eclipse端：（部分）

<%

request.setCharacterEncoding("UTF-8");

String username = request.getParameter("username");

String password = request.getParameter("password");

String realname = request.getParameter("realname");

String telNumber = request.getParameter("telNumber");

String email = request.getParameter("email");

String IDCard = request.getParameter("IDCard");

//执行

UserBean bean = **new** UserBean();

UserInfo info = **new** UserInfo(username,password,realname,telNumber,email,IDCard);

**boolean** result = bean.add(info);

**if**(result){

out.println("1");//注册成功

}

**else**{

out.println("0");//注册失败

}

%>

**public** **boolean** add( UserInfo info ){

// 获取连接

Connection conn = DBConn.*getConnection*();

**int** count = -1;

**try**{

String sql = " insert user "

+ " (username,password,realname,telNumber,email,IDCard) "

+ " values(?, ?, ?, ?, ?, ?) ";

PreparedStatement st = conn.prepareStatement( sql );

**int** i = 1;

// 用户名

st.setString( i, info.getUsername() );

++i;

// 密码

st.setString(i, info.getPassword() );

++i;

// 真实名

st.setString( i, info.getRealname() );

++i;

// 电话号码

st.setString(i, info.getTelNumber());

++i;

//电子邮箱

st.setString(i, info.getEmail());

++i;

//身份证号码

st.setString(i, info.getIDCard());

++i;

System.***out***.printf("sql = %s\n", st.toString());

count = st.executeUpdate( );// 执行语句

System.***out***.printf( "新增%d条记录", count );

}

**catch**( SQLException e ){

System.***out***.printf( "新增失败:" + e.getMessage() );

**return** **false**;

}

**finally**{

**if**( conn != **null** ){

**try**{

conn.close();

}

**catch**( SQLException e ){

System.***out***.printf( "关闭连接失败\n" + e.getMessage() );

}// try

}// if

}// finally

**return** **true**;

}

1. 使用okhttp进行登录

Android端代码：（部分）

**logins**.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View v) {  
 Intent intent = getIntent();  
 **final** String username = **etName**.getText().toString().trim();  
 String password = **etPass**.getText().toString().trim();  
  
 **if**(username.equals(**""**)){  
 Looper.*prepare*();  
 Toast.*makeText*(LoginActivity.**this**,**"用户名不能为空"**,Toast.***LENGTH\_SHORT*** ).show();  
 Looper.*loop*();  
 }**else if**(password.equals(**""**)){  
 Looper.*prepare*();  
 Toast.*makeText*(LoginActivity.**this**,**"密码不能为空"**,Toast.***LENGTH\_SHORT*** ).show();  
 Looper.*loop*();  
 }**else**{  
 RequestBody requestBody = **new** FormBody.Builder()  
 .add(**"username"**,username)  
 .add(**"password"**,password)  
 .build();  
 HttpUtil.*getByOkHttp*(**"http://10.0.2.2:8080/Ticket/login.jsp"**, requestBody, **new** Callback() {  
 @Override  
 **public void** onFailure(Call call, IOException e) {  
 Show(**"Fail"**);  
  
 }  
  
 @Override  
 **public void** onResponse(Call call, Response response) **throws** IOException {  
 String data = response.body().string().trim();  
 *//登录成功跳转到下一个页面* **if**(data.equals(**"1"**)){  
 Looper.*prepare*();  
 Toast.*makeText*(LoginActivity.**this**,**"登录成功"**,Toast.***LENGTH\_SHORT*** ).show();  
 SharedPreferences.Editor editor = getSharedPreferences(**"checkLogin"**,  
 ***MODE\_PRIVATE***).edit();  
 editor.putBoolean(**"login"**,**true**);  
 editor.commit();  
 Intent intent = getIntent();  
 String tabId = intent.getStringExtra(**"tabId"**);  
 **if** (tabId.equals(**"tab1"**)){  
 intent.setClass(LoginActivity.**this**,ConfirmChangeActivity.**class**);  
 startActivity(intent);  
 }**else if** (tabId.equals(**"tab2"**)){  
 intent.setClass(LoginActivity.**this**,OrderManageActivity.**class**);  
 startActivity(intent);  
 }**else if** (tabId.equals(**"tab3"**)) {  
 intent.setClass(LoginActivity.**this**, TicketActivity.**class**);  
 startActivity(intent);  
 }**else if** (tabId.equals(**"tab4"**)) {  
 intent.setClass(LoginActivity.**this**, MoreBottomActivity.**class**);  
 startActivity(intent);  
 }  
 Looper.*loop*();  
 }**else**{  
 *//Toast.makeText(LoginActivity.this,"该用户不存在，请先注册",Toast.LENGTH\_SHORT).show();* }  
 }  
 });  
 }  
 *//获取用户输入的内容  
 //判断非空  
 //调用okhttp3的登录验证方法，如果为真则存在用户，否则该用户不存在，提示注册* }  
 });  
}

Eclipse端代码：（部分）

<%

request.setCharacterEncoding("UTF-8");

//接受参数Android端

String username = request.getParameter("username");

String password = request.getParameter("password");

//执行

UserBean bean = **new** UserBean();

UserInfo info = **new** UserInfo(username,password,"","","","");

String \_username = bean.select(username,password);//数据库端

**if**(username.equals(\_username)){

out.println("1");//登录成功返回1

}

**else**{

out.println("0");//登录失败

}

%>

**public** String select(String \_name,String \_password) {

String username = **null**;

//获取连接

Connection conn = DBConn.*getConnection*();

**try** {

String sql = "select username from user where username = ? and password = ?";

PreparedStatement st = conn.prepareStatement(sql);

**int** i = 1;

st.setString(i, \_name);

i++;

st.setString(i, \_password);

ResultSet rs = st.executeQuery();//执行查询语句

**while**(rs.next()){

username = rs.getString("username");

}

} **catch** (SQLException e) {

System.***out***.println("查找失败"+e.getMessage());

username = **null**;

**return** username;

// **TODO**: handle exception

}**finally** {

**if**(conn!=**null**){

**try** {

conn.close();

} **catch** (SQLException e) {

System.***out***.print("关闭连接失败\n"+e.getMessage());

// **TODO**: handle exception

}

}

}

**return** username;

}

1. **小结**

通过本次课程设计，我们学习到了很多之前没有掌握的知识，比如说进行okhttp联网操作、连接mysql数据库，（之前一直用的都是sqlite数据库）、还有包括怎么通过gson解析json格式数据。还更加深刻的认识到了服务器端和安卓客户端的连接以及传输数据等操作，像是以前我们对于eclipse只学会了连接sql server数据库，但是通过这次实验学会了如何连接上mysql数据库。

同时对于安卓端的各控件的运用也更加熟悉了，对于安卓的各activity的带参数、不带参数的各种跳转页有了更加深刻的认识。

但是本次课设也有很多的遗憾，由于时间的限制，我们很多预想的功能也还没有来得及实现它们，比如说该用户的本人信息、改签等功能都没来得及去实现。当然这次实验也让我深刻分意识到了我们本身能力的不足，还需要更加深入的去学习android开发，通过更加深入的学习来完善我们现阶段没有完成的一些功能。