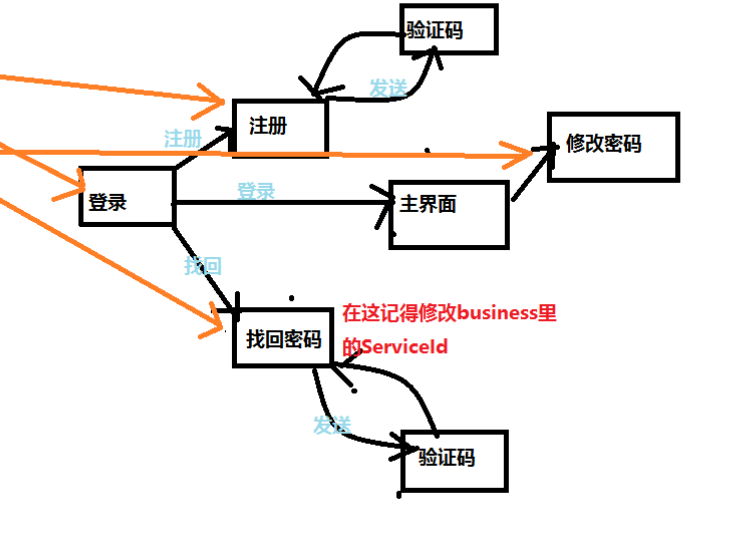
刚开始就构思程序目的，我想写一个能够接单，发布单的一个代练游戏的软件，刚开始想有游戏分类啊什么的，后来在写的时候Order表（保存订单信息）的时候因为疏忽没有加上游戏这一栏，所以这样这个软件先就只能支持一个游戏，因为没有办法对数据进行筛选，分辨不出哪个游戏，因为发现的时候数据部分基本都写的差不多了，只有几个方法还没有写完，所以我也不确定再对表中增加一列会不会对其他方法产生影响，如果有影响任务量太过庞大，保守起见，就决定下个版本解决，因为期末也是没什么时间再推倒重写一遍了，留到我暑假自己慢慢更新版本吧。

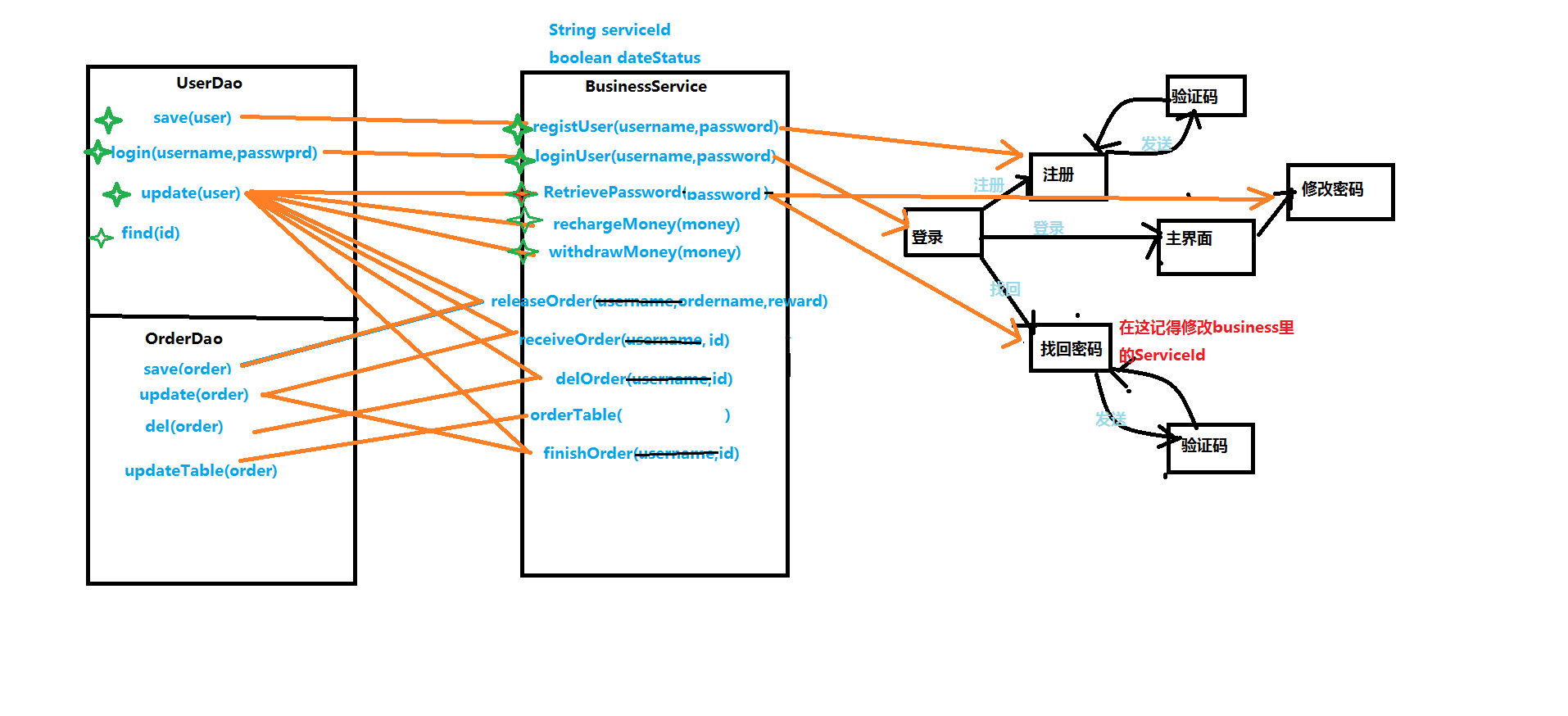
然后既然能接单，发单，那就一定是要有登录界面，那就要有注册啊，修改密码啊等等一系列操作，然后我就简单的画了一个界面之间的关系图



忽略里面的所有文字和箭头，后面会讲

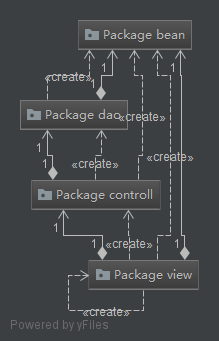
然后界面框架出来之后就开始构思数据部分了，因为某些操作要对数据库进行频繁切相同的访问，比如查询等操作，所以就使用了MVC结构，Model在程序中被命名为UserDao（负责对User表进行操作），和OrderDao（负责对Order表进行操作）他们都在dao包中，然后就是Controller，在程序中的BusinessSeriver类，这里面对数据进行了简单的操作然后去调用dao包中的方法，BusinessSeriver在contaoll包中，然后就是view包，里面保存了界面的所有类

然后大概画了一个图



其中黑色的线是界面之间的从属关系，蓝色是方法，在后面的实际执行中，增加了一些方法和界面，没有添加进去，红色是之前打算在BusinessService里存下当前服务的User ID

，用来提醒自己需要添加，但是后来改变策略，把整个User 传进去，就不存在这个问题了。



然后是实际操作，先从数据部分开始写，因为从view写的话有很多不可控的事情有可能发生比如窗口之间数据的传递，都会很乱，而且不利于以后的维护，所以先写UserDao（这里因为登录界面最先用到的就是User表），Userdao里面设计了几个方法

public void test() //测试数据

public boolean save(User user) //保存User 到数据库中

public User find(Srting id) //根据id查找用户

Public User findname(String username) //根据用户查找用户

Public User login(String username,

String password) //根据传来的帐号密码查询用户

Public boolean del(String id) //根据id删除用户

Public boolean update(User user) //修改用户信息

Public boolean release(Connection conn,

PreparedStatement stmt,

Resultset rs) //释放资源的方法

然后设计OrderDao

public boolean save(Order order) //保存数据

Public boolean update(Order order) //更新数据

Public boolean del(Order order) //删除数据

Public ArrayList getList( ) //查询数据（返回所有order的list 表）

Public Order find(String orderId) //根据id查找订单

Public boolean release(Connection conn,

PreparedStatement stmt,

Resultset rs) //释放资源的方法

和数据库打交道的两个类都做完了，因为User和Order 所包含的数据很多，传来传去很麻烦，就给他们封装起来，直接传这个对象就可以了，所以就有了bean.Order 和 bean.User

两个类,里面分别有各自数据的get和set方法，就不一一罗列了。

然后就是BusinessService 这个类目的是对View和Dao之间的数据交换做简单的处理

Public User registUser(String username,

String password) //注册用户

Public boolean retrievePasswordBuUsername(

String uaername,String passwoed) //根据username修改密码

Public void retrievePassword(User user,

String newPassword) //根据id找到用户然后修改密码

Public boolean rechargeMoney(User user,

Double money) //充值

Public boolean withdowMoney(User user,

Double money) //提现

Public Order releaseOrder(User user,

String ordername,

Double reward) //发布订单

Public boolean updateOrderState(

User user,

String orderId,

String state) //更改订单状态

Public boolean derOrder(User user,

String orderId) //删除订单

Public boolean receiveOrder(

User user,

String orderId) //接受订单

Public boolean finfshOrder(User user,

String orderId) //完成订单

Public ArrayList<Order> getStateList(

String state) //拿到所有状态为state的list订单

Public ArrayList<Order> getRelease\_idList(

String release\_id) //拿到所有release\_id的list订单

Public ArrayList<Order> getReceive\_idList(

String receive\_id) //拿到所有receive\_id的list订单

然后View部分主要都是一些界面，还有监视器的部分，因为主界面有5个界面要显示，所以View里面也有一个Controll，用来进行简单的判断，然后执行相应的操作

下面是view部分的类图

