# 博青秀代码清单

## 前端代码

### 加载代码

Launch.js

```
import React, { useState, useEffect } from "react";
import { Text, StyleSheet, View, Image, ActivityIndicator, StatusBar } from "react-native";
import { createStackNavigator } from "react-navigation";

const Launch = (prop) => {
  const [time, setTime] = useState(1);

  useEffect(() => {
    let timeChange;
    const clock = () => {
      if (time > 0)
        setTime(time - 1);
      else {
        clearInterval(timeChange);
        prop.navigation.navigate('Login');
      }
    };
    timeChange = setInterval(clock, 1000);
  });

  return (
    <View style={styles.container}>
      <StatusBar hidden={true} />
      <Text style={styles.text}>博青秀</Text>
      <Image
        source={require('../data/img/logo.png')}
        style={styles.picture} />
    </View>
  );
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
  },
  text: {
    color: '#7B9DF6',
    fontSize: 50,
  },
  picture: {
    marginTop: 20,
    marginBottom: 20,
    width: 100,
    height: 90,
  }
});

export default Launch;
```

# 登录代码

Login.js

```
import React, { useState } from "react";
import {
    Text,
    StyleSheet,
    View,
    TextInput,
    TouchableNativeFeedback,
    Image,
    Dimensions,
    StatusBar,
    ImageBackground
} from "react-native";
import UserInformation from "./UserInformation";

const { width, height } = Dimensions.get("screen");
const Login = (prop) => {
    const [account, setAccount] = useState("");
    const [passward, setPassward] = useState("");
    const [secureTextEntry, setSecureTextEntry] = useState(false);

    function changeText(type, text) {
        if (type === "account")
            setAccount(text);
        else if (type === "password") {
            setSecureTextEntry(true);
            setPassward(text);
        }
    }

    /** 对比数据库及登录审核 */



    function login() {
        var userName = account;
        var password = passward;
        // UserInformation.id = id;
        UserInformation.userName = userName;
        UserInformation.password = password;
        var ip = UserInformation.ip;



        /**
         * 现在的登录只有普通用户的登录
         */
        fetch(ip + 'selectUser?name=' + userName + '&password=' + password)
            .then(response => response.json())
            .then(users => {
                console.log(users);
                if (users[0].nUserId != null) {
                    prop.navigation.navigate('Main');
                }
                else {
                    //想要一个密码错误的弹窗
                    console.log("Something is wrong ");
                }
            })





        // prop.navigation.navigate('Main');
    }



    function register() {
        /**审核 */
        prop.navigation.navigate('Register')
    }

    return (
        <View>
            <StatusBar hidden={false} />
            <ImageBackground source={require("../data/img/login.png")} style={{ width, height }}
                <View style={styles.container}>
                    <View style={styles.inputView}>
                        <Image source={require("../data/img/login_1.png")} style={styles.img} />
                        <View style={styles.line}></View>
                        <TextInput
                            value={account}
                            placeholder="请输入登录账号"
                            onChangeText={text => changeText("account", text)}
                            style={styles.input}
                        />
                    </View>
                    <View style={styles.inputView}>
                        <Image source={require("../data/img/login_2.png")} style={styles.img} />
                        <View style={styles.line}></View>
                        <TextInput
                            value={passward}
                            placeholder="请输入登录密码"
                            secureTextEntry={secureTextEntry}
                            onChangeText={text => changeText("password", text)}
                            style={styles.input}
                        />
                    </View>
                    <Text style={styles.forgetPwd}>忘记密码</Text>
                    <TouchableNativeFeedback>
```

```
                        <Text style={styles.btnLogin} onPress={() => login()}>登录</Text>
                    </TouchableNativeFeedback>
                    <Text style={styles.register} onPress={() => register()}>立即注册</Text>
                    <Text style={styles.info}>登录表示你同意该软件
                        <Text style={styles.infoItem}>用户服务协议</Text>和
                        <Text style={styles.infoItem}>隐私政策</Text>
                    </Text>
                </View>
            </ImageBackground>
        </View>
    );
}

const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: "center",
        alignItems: "center",
        marginLeft: 49,
        marginRight: 49,
        // borderWidth: 1,
        // borderColor: "black"
    },
    inputView: {
        borderColor: "#BBB",
        borderWidth: 1,
        borderRadius: 40,
        width: 260,
        height: 45,
        margin: 15,
        flexDirection: "row",
    },
    input: {
        width: 180,
        marginLeft: 10,
        fontSize: 16,
    },
    img: {
        width: 20,
        height: 20,
        marginTop: 11,
        marginLeft: 18,
        marginRight: 13
    },
    line: {
        borderLeftWidth: 1,
        borderLeftColor: "#BBB",
        height: 20,
        marginTop: 11
    },
    forgetPwd: {
        fontSize: 14,
        alignSelf: "flex-end",
        marginRight: 10,
        marginTop: 0
    },
    btnLogin: {
        marginTop: 30,
        borderRadius: 50,
        width: 200,
        height: 40,
        backgroundColor: "#7B9DF6",
        textAlign: "center",
        textAlignVertical: 'center',
        color: "#FFF",
        fontSize: 18
    },
    register: {
        marginTop: 20,
        fontSize: 14,
    },
    info: {
        position: "absolute",
        bottom: 100,
        fontSize: 12
    },
    infoItem: {
        color: "#7B9DF6"
    }
});

export default Login;
```

# 注册代码

Register.js

```
import React, { useState } from "react";
import ModalDropdown from 'react-native-modal-dropdown';
import { KeyboardAwareScrollView } from 'react-native-keyboard-aware-scroll-view';
import {
    Text,
    StyleSheet,
    View,
    TextInput,
    TouchableNativeFeedback,
    Image,
    Dimensions,
    ImageBackground
} from "react-native";
import UserInformation from "./UserInformation";

const { width, height } = Dimensions.get("screen");
const Register = (prop) => {
    const [passward, setPassward] = useState("");
    const [confirm, setConfirm] = useState("");
    const [phone, setPhone] = useState("");
    const [captcha, setCaptcha] = useState("");
    const [identity, setIdentity] = useState("");

    function changeText(type, text) {
        if (type === "password")
            setPassward(text);
        else if (type === "confirm")
            setConfirm(text);
        else if (type === "phone")
            setPhone(text);
        else if (type === "captcha")
            setCaptcha(text);
    }

    /** 获取验证码  */
    function gainCaptcha() { }

    /** 注册审核及插入数据库 */
    function register() {
        /**审核 */

        /**
         * 暂时没有添加审核
         */

        var ip = UserInformation.ip;
        // console.log("输出ip地址" + ip);


        if (passward === confirm) {
```

```
        // 暂时还没有加上判断
        if (identity === "管理员") {
            // console.log("进入管理员登录");
            fetch(ip + 'addAdmin?name=' + phone + '&password=' + passward);

            prop.navigation.navigate('Login');
        }
        else if (identity === "普通用户") {
            fetch(ip + 'addUser?name=' + phone + '&password=' + passward);
            prop.navigation.navigate('Login');
        }
        else { // 文化组织
            fetch(ip + 'addOrg?name=' + phone + '&password=' + passward);
            prop.navigation.navigate('Login');
        }
    }
    else {
        console.warn("密码不对,请核实");
    }




}

console.log(identity)
return (
    <ImageBackground source={require("../data/img/login.png")} style={{ width, height }}>
        <View style={styles.container}>
            <KeyboardAwareScrollView style={{ marginTop: 180 }}
                onKeyboardWillShow={(frames) => {
                    console.log('Keyboard event', frames)
                }}>
                <ModalDropdown
                    options={["管理员", "普通用户", "文化组织"]}
                    defaultValue="        请选择身份       ﹀"
                    value={identity}
                    onSelect={(index, value) => setIdentity(value)}
                    animated={false}
                    style={styles.dropdown}
                    textStyle={styles.dropdownText}
                    dropdownStyle={[{ width: 200, marginLeft: 25 }, { height: 38 * 3 }]}
                    dropdownTextStyle={{ fontSize: 16, alignSelf: "center" }}
                />

                <View style={styles.inputView}>
                    <Image source={require("../data/img/login_2.png")} style={styles.img} />
                    <View style={styles.line}></View>
                    <TextInput
                        value={passward}
                        placeholder="请输入登录密码"
                        secureTextEntry={true}
                        onChangeText={text => changeText("password", text)}
                        style={styles.input}
                    />
                </View>
                <View style={styles.inputView}>
                    <Image source={require("../data/img/login_2.png")} style={styles.img} />
                    <View style={styles.line}></View>
                    <TextInput
                        value={confirm}
                        placeholder="请输入确认密码"
                        secureTextEntry={true}
                        onChangeText={text => changeText("confirm", text)}
                        style={styles.input}
                    />
                </View>
                <View style={styles.inputView}>
                    <Image source={require("../data/img/register_1.png")} style={styles.img}
                    <View style={styles.line}></View>
                    <TextInput
                        value={phone}
                        placeholder="请输入手机号码"
                        keyboardType="number-pad"
                        onChangeText={text => changeText("phone", text)}
                        style={styles.input}
                    />
                </View>
                <View style={styles.inputView}>
                    <Image source={require("../data/img/register_2.png")} style={styles.img}
                    <View style={styles.line}></View>
                    <TextInput
                        value={captcha}
                        placeholder="请输入验证码"
                        onChangeText={text => changeText("captcha", text)}
                        style={styles.inputCaptcha}
                    />
                    <TouchableNativeFeedback>
                        <Text style={styles.btnCaptcha} onPress={() => gainCaptcha()}>获取验
                    </TouchableNativeFeedback>
                </View>
            </KeyboardAwareScrollView>

            <TouchableNativeFeedback>
                <Text style={styles.btnRegister} onPress={() => register()}>注册</Text>
            </TouchableNativeFeedback>
            <Text style={styles.info}>注册表示你同意该软件
                <Text style={styles.infoItem}>用户服务协议</Text>和
                <Text style={styles.infoItem}>隐私政策</Text>
            </Text>
        </View>
    </ImageBackground>
```

```
    );
}

const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: "center",
        alignItems: "center",
    },
    inputView: {
        borderColor: "#BBB",
        borderWidth: 1,
        borderRadius: 40,
        width: 260,
        height: 45,
        marginHorizontal: 15,
        marginVertical: 10,
        flexDirection: "row",
    },
    input: {
        width: 180,
        marginLeft: 10,
        fontSize: 16,
    },
    inputCaptcha: {
        width: 110,
        marginLeft: 10,
        fontSize: 16,
    },
    img: {
        width: 20,
        height: 20,
        marginTop: 11,
        marginLeft: 18,
        marginRight: 13
    },
    line: {
        borderLeftWidth: 1,
        borderLeftColor: "#BBB",
        height: 20,
        marginTop: 11
    },
    forgetPwd: {
        fontSize: 14,
        alignSelf: "flex-end",
        marginRight: 10,
        marginTop: 0
    },
    btnRegister: {
        position: "absolute",
        bottom: 175,
        borderRadius: 50,
        width: 200,
        height: 40,
        backgroundColor: "#7B9DF6",
        textAlign: "center",
        textAlignVertical: 'center',
        color: "#FFF",
        fontSize: 18
    },
    btnCaptcha: {
        borderRadius: 50,
        width: 82,
        height: 35,
        alignSelf: "center",
        backgroundColor: "#7B9DF6",
        textAlign: "center",
        textAlignVertical: 'center',
        color: "#FFF",
        fontSize: 14
    },
    info: {
        position: "absolute",
        bottom: 100,
        fontSize: 12
    },
    infoItem: {
        color: "#7B9DF6"
    },
    dropdown: {
        borderColor: "#BBB",
        borderWidth: 1,
        borderRadius: 40,
        width: 260,
        height: 45,
        marginHorizontal: 15,
        marginTop: 60,
        marginBottom: 10
    },
    dropdownText: {
        fontSize: 16,
        width: 260,
        height: 45,
        textAlign: "center",
        textAlignVertical: 'center',
        color: "#9E9E9E",
    }
});

export default Register;
```

# 后端代码

## 用户的一些信息存放

UserInformation.js

```
/**
 * 这里存放的是一些全局的信息
 */


// const ip = 'http://192.168.144.249:3000/';//手机热点
const ip = 'http://192.168.2.206:3000/';// 家里wifi

const id = "";
const userName = "";
const password = "";
const userIdentity = "";
export default { ip, id, userName, password, userIdentity };
```

## 数据库连接

routes.js

```
const express = require('express');
const bodyParser = require('body-parser');
const mysql = require('mysql');


const connection = mysql.createPool({
    /**
     * 有时候可以连接上，有时候连接不上，很烦！
     */


    /*
       host: '101.34.44.163', // Your connection adress (localhost).
       user: 'mysqld',     // Your database's username.
       password: 'kqaz1234',        // Your database's password.
       database: 'boqingxiu'   // Your database's name.

   */

    host: 'localhost', // Your connection adress (localhost).
    user: 'root',     // Your database's username.
    password: '123456',        // Your database's password.
    database: 'boqingxiu'   // Your database's name.



});

const app = express();


app.use(bodyParser.json());



app.get('/selectUser', function (req, res) {
    // Connecting to the database.
    // console.log(req.query.name);
    connection.getConnection(function (err, connection) {
        // const sql = "select * from classroom where cname ='" + cname + "'";
        // var id = req.query.id;
        var name = req.query.name;
        var password = req.query.password;

        // var cname = req.query.cname;

        // const sql = `select * from normalUser where nUserId = ${id} and nUserPwd = '${passwor
        const sql = `select * from normalUser where nUserName = '${name}' and nUserPwd = '${pass
        // Executing the MySQL query (select all data from the 'users' table).
        connection.query(sql, function (error, results, fields) {
            // If some error occurs, we throw an error.
            if (error) throw error;

            // Getting the 'response' from the database and sending it to our route. This is wer
            res.send(results)
        });
    });
});


/**
 * 添加管理员
 */

app.get('/addAdmin', function (req, res) {

    console.log("开始插入");


    var sql = 'insert into admin set  adminName=? , adminPwd=?'; // 这边的"?"是SQL的模板语法

    var params = [req.query.name, req.query.password]  // 这边的数组参数与上边的"?"一一映射

    console.log(req.query.name, req.query.password);
    connection.query(sql, params, function (err, result) {
        try {
            res.send('增加管理员成功');
        } catch (err) {
            console.log('新增数据失败');
        }
    });

    console.log("执行sql语句结束");
})

/**
 * 添加普通用户
 */

app.get('/addUser', function (req, res) {

    console.log("开始插入");
    var sql = 'insert into  normalUser set  nUserName=? , nUserPwd=?,userPhone=?, totalTime =?
    var params = [req.query.name, req.query.password, req.query.name, 0.0]  // 这边的数组参数与上
    console.log(req.query.name, req.query.password);
    connection.query(sql, params, function (err, result) {
        try {
            res.send('增加普通用户成功');
        } catch (err) {
            console.log('新增数据失败');
        }
    });
})


/**
 * 添加文化组织
 */

app.get('/addOrg', function (req, res) {
    var sql = 'insert into organization set orgName=? , orgPwd=?,orgPhone=?'; // 这边的"?"是SQL的
    var params = [req.query.name, req.query.password, req.query.name]  // 这边的数组参数与上边的"?
    connection.query(sql, params, function (err, result) {
        try {
            res.send('增加组织成功');
        } catch (err) {
            console.log('新增数据失败');
        }
    });
})



/**
 * 查询招募模块 根据 appointId
 */


app.get('/selectAppointment', function (req, res) {
    var sql = 'select * from appointment where appointId = ?'; // 这边的"?"是SQL的模板语法
    var params = [req.query.id]  // 这边的数组参数与上边的"?"——映射
    connection.query(sql, params, function (err, result) {
        try {
            res.send(result);
        } catch (err) {
            console.log('查询发生错误');
        }
    });
})



/**
 * 查询公告模块，查询出来所有的模块，应该根据系统时间来进行变化
 */
app.get('/selectRecruitboard', function (req, res) {
    var sql = 'select * from recruitboard'; // 这边的"?"是SQL的模板语法
    // var params = [req.query.id]  // 这边的数组参数与上边的"?"——映射
    connection.query(sql, function (err, result) {
        try {
            res.send(result);
        } catch (err) {
            console.log('查询发生错误');
        }
    });
})



/**
 * 查询出来初始时间段  根据id
 */
app.get('/selectTime', function (req, res) {
    var sql = 'select * from time where timeId = ?'; // 这边的"?"是SQL的模板语法
    var params = [req.query.id]  // 这边的数组参数与上边的"?"一一映射
    connection.query(sql, params, function (err, result) {
        try {
            res.send(result);
        } catch (err) {
            console.log('查询发生错误');
        }
    });
})



/**
 * 查询出来组织 根据 id
 */
app.get('/selectOrg', function (req, res) {
    var sql = 'select * from time where orgId = ?'; // 这边的"?"是SQL的模板语法
    var params = [req.query.id]  // 这边的数组参数与上边的"?"一一映射
    connection.query(sql, params, function (err, result) {
        try {
            res.send(result);
        } catch (err) {
            console.log('查询发生错误');
        }
    });
})






app.listen(3000, () => {
    console.log('Go to http://localhost:3000 so you can see the data.');
});
```