

项目名称：BBS

姓名：_____高汉平_____ 学号：_____2017210218006_____

一、 项目需求

- 1、登陆识别身份
- 2、浏览所有根问题
- 3、浏览对应问题的回复
- 4、回复
- 5、删除自己的留言
- 6、新建问题

二、 功能分析

BBS 在微信小程序中应用广泛，知乎、微博、论坛等社交娱乐软件中也有类似的功能，通过提出问题，其他人可以在之下的评论区中广泛讨论，发表自己的意见，并有删除自己留言的功能。功能实用性强且应用广泛。

三、 总体设计

用户在输入账号密码后，验证识别身份后登陆主页面，可以新建发布自己的问题，也可以浏览他人的问题以及对应的问题回复，同时可以添加自己的回复或者选择删除自己的回复。

$DB \Leftrightarrow API \Leftrightarrow MP$

用户： *id*、 *username*、 *password*

问题： *id*、 *problem*、 *userid*、 *time*、 *parentid*

四、 实现原理

BBS 设计主要在于用 php 后台实现获取数据并保存到数据库, 需要前端发送

请求到服务器，微信小程序只支持 https 请求，所以调试的时候需要勾选不校验安全域名，然后从数据库获取信息并写入本地缓存，在设计用户登陆时进行识别。

五、 模块划分

通过对 BBS 功能的分析，将其划分为以下几个模块：

- 1、模块一：主界面（包含微信登陆界面以及底部导航）
- 2、模块二：时间版块（用于获取用户提问时间并设置显示格式）
- 3、模块三：根问题版块（用于显示所有根问题以及问题评论）
- 4、模块四：提问版块（用于用户提出新的问题）
- 5、模块五：评论版块（用于用户针对问题编辑留言或者删除自己的留言）
- 6、模块六：登陆版块（用于验证用户账号密码，识别用户身份）

六、 各模块实现过程

a) 模块一：主界面（包含微信登陆界面以及底部导航）

app.json 设置登录界面以及底部导航

```
//app.js
App({
  onLaunch: function() {
    // 展示本地存储能力
    var logs = wx.getStorageSync('logs') || []
    logs.unshift(Date.now())
    wx.setStorageSync('logs', logs)
    // 登录
    wx.login({
      success: res => {
        // 发送 res.code 到后台换取 openId, sessionKey, unionId
      }
    })
    // 获取用户信息
    wx.getSetting({
      success: res => {
        if (res.authSetting['scope.userInfo']) {
          // 已经授权，可以直接调用 getUserInfo 获取头像昵称，不会弹框
        }
      }
    })
  }
})
```

```

wx.getUserInfo({
  success: res => {
    // 可以将 res 发送给后台解码出 unionId
    this.globalData.userInfo = res.userInfo
    // 由于 getUserInfo 是网络请求，可能会在 Page.onLoad 之后才返回
    // 所以此处加入 callback 以防止这种情况
    if (this.userInfoReadyCallback) {
      this.userInfoReadyCallback(res)
    }
  }
})
},
globalData: {
  userInfo: null
}
})

```

b) 模块二：时间版块（用于获取用户提问时间并设置显示格式）

util.js 获取时间

```

Date.prototype.formatStr = function(format) {
  var o = {
    "M+": this.getMonth() + 1, // month
    "D+": this.getDate(), // day
    "h+": this.getHours(), // hour
    "m+": this.getMinutes(), // minute
    "s+": this.getSeconds(), // second
    "q+": Math.floor((this.getMonth() + 3) / 3), // quarter
    "S": this.getMilliseconds()
  };
  if (/ (Y+)/.test(format)) {
    format = format.replace(RegExp.$1, (this.getFullYear() + "").substr(4 - RegExp.$1.length));
  }
  for (var k in o) {
    if (new RegExp("(" + k + ")").test(format)) {
      format = format.replace(RegExp.$1, RegExp.$1.length == 1 ? o[k] : ("00" + o[k]).substr(("" + o[k]).length));
    }
  }
}

```

```

return format;
};
module.exports.dateFormat = function(date, format) {
format = format || 'YYYY-MM-DD hh:mm:ss';
return date.formatStr(format);
}

```

c) 模块三：根问题版块（用于显示所有根问题以及问题评论）

(1) *question.js* 添加按钮跳转至 *pages/add/add* 页面和 *pages/comment/comment* 页

面

```

// pages/question/question.js
var util = require('../../utils/util');
// pages/list/list.js
Page({
/**
 * 页面的初始数据
 */
data: {
lists: []
},
/**
 * 生命周期函数--监听页面加载
 */
onLoad: function(options) {
initData(this);
},
/**
 * 生命周期函数--监听页面显示
 */
onShow: function() {
initData(this);
},
/**
 * 编辑事件
 */
edit(e) {
var id = e.currentTarget.dataset.id;
// 跳转 navigateTo
wx.navigateTo({
url: '../add/add?id=' + id,
url2: '../comment/comment?id=' + id
})
}

```

```

    },
    /**
     * 添加事件
     */
    add() {
      wx.navigateTo({
        url: '../add/add',
      })
    },
    add2() {
      wx.navigateTo({
        url: '../comment/comment',
      })
    }
  })
  /**
   * 处理初始化页面列表数据
   */
  function initData(page) {
    var arr = wx.getStorageSync('txt');
    if (arr.length) {
      arr.forEach((item, i) => {
        var t = new Date(Number(item.time));
        item.time = util.dateFormate(t);
      })
      page.setData({
        lists: arr
      })
    }
  }
}

```

(2) *question.wxml* 设置问题页面排版

```

<!--pages/question/question.wxml-->
<view class='page'>
  <scroll-view scroll-y='true' class='lists'>
    <block wx:for="{{lists}}" wx:key="*this.id">
      <view class="list-i" bindtap="edit" data-id="{{item.id}}">
        <view class="content">{{item.content}}</view>
        <view class='time'>时间: {{item.time}}</view>
      </view>
      <view class='time' bindtap='add2'>查看评论</view>
    </block>
  </scroll-view>

```

```
<view class='add' bindtap='add'>
<image src='/images/edit.png'></image>
</view>
</view>
```

(3) *question.wxss* 设置问题页面排版所用格式

```
/* pages/question/question.wxss */
.page {
position: relative;
}
.lists {
position: absolute;
left: 0;
top: 0;
width: 100%;
height: 100%;
}
.list-i {
padding: 18rpx 0;
display: flex;
flex-direction: column;
width: 100%;
height: 200rpx;
background: #e0f6ff;
border-bottom: 3px solid #fff;
}
.content {
padding: 0 28rpx;
height: 118rpx;
line-height: 118rpx;
border-bottom: 1px solid skyblue;
}
.time {
padding: 0 28rpx;
line-height: 88rpx;
color: #88f;
}
.add {
position: absolute;
right: 58rpx;
bottom: 58rpx;
display: flex;
align-items: center;
justify-content: center;
```

```

width: 100rpx;
height: 100rpx;
background: rgb(136, 255, 225);
border: 1px solid #888;
border-radius: 50%;
}
.add image {
width: 48rpx;
height: 48rpx;
}

```

d) 模块四：提问版块（用于用户提出新的问题）

(1) *add.js* 设置添加页面主要函数并添加按钮返回至问题页面

```

// pages/add/add.js
Page({
  /**
   * 页面的初始数据
   */
  data: {
    content: ''
  },
  /**
   * 生命周期函数--监听页面加载
   */
  onLoad: function(e) {
    var id = e.id;
    if (id) {
      getData(id, this);
    } else {
      this.setData({
        id: Date.now()
      })
    }
  },
  /**
   * input change 事件
   */
  change(e) {
    var val = e.detail.value;
    this.setData({
      content: val
    });
  }
});

```

```

    },
    /**
     * cancel 事件
     */
    cancel() {
      wx.navigateBack();
    },
    sure() {
      var reg = /^s*$/g;
      if (!this.data.content || reg.test(this.data.content)) {
        console.log('不能为空');
        return;
      }
      this.setData({
        time: Date.now()
      });
      setValue(this);
      wx.navigateBack();
    }
  })
  /**
   * 根据跳转的 url 中的 id 获取编辑信息回填
   */
  function getData(id, page) {
    var arr = wx.getStorageSync('txt');
    if (arr.length) {
      arr.forEach((item) => {
        if (item.id == id) {
          page.setData({
            id: item.id,
            content: item.content
          })
        }
      })
    }
  }
  /**
   * 设置填写的信息，分编辑、新增
   */
  function setValue(page) {
    var arr = wx.getStorageSync('txt');
    var data = [],
    flag = true;
    if (arr.length) {

```



```

arr.forEach(item => {
  if (item.id == page.data.id) {
    item.time = Date.now();
    item.content = page.data.content;
    flag = false;
  }
  data.push(item);
})
}
if (flag) {
  data.push(page.data);
}
wx.setStorageSync('txt', data);
}

```

(2) *add.wxml* 设置添加问题页面排版

```

<!--pages/add/add.wxml-->
<view class="page">
  <view class="page-t">
    <textarea class="content" placeholder="编辑问题: " bindinput="change"
    value="{{content}}" maxlength="-1"></textarea>
  </view>
  <view class="btns">
    <view class="sure" bindtap="sure">确定</view>
    <view class="cancel" bindtap="cancel">取消</view>
  </view>
</view>

```

(3) *add.wxss* 设置主页页面排版所用格式

```

/* pages/add/add.wxss */
.page {
  display: flex;
  flex-direction: column;
}
.page-t {
  width: 100%;
  height: 90%;
}
.content {
  margin: 10rpx;
  height: 100%;
  width: auto;
}
.btns {

```

```

flex: 1;
display: flex;
align-items: center;
justify-content: space-around;
background-color: skyblue;
}

```

e) 模块五：评论版块（用于用户针对问题编辑留言或者删除自己的留言）

(1) *comment.js* 设置评论页面主要函数并添加按钮返回至问题页面

```

// pages/comment/comment.js
Page({
  data: {
    inputVal: "",
    msgData: [],
    msgData1: ""
  },
  changeInputVal(ev) {
    this.setData({
      inputVal: ev.detail.value
    });
  },
  addMsg() {
    var list = this.data.msgData; //获取所有留言
    list.push({ //向list中添加当前添加的留言
      msg: this.data.inputVal
    });
    this.setData({
      msgData: list,
      inputVal: ""});
  },
  deleMsg(ev) {
    var list = this.data.msgData;
    var n = ev.target.dataset.index; list.splice(n, 1);
    this.setData({
      msgData: list
    });
  },
  deleMsg1(ev) {
    var list = this.data.msgData1;
    var n = ev.target.dataset.index;
    list.splice(n, 1);
    this.setData({
      msgData1: list
    });
  }
});

```

```

});
},
/**
 * 生命周期函数--监听页面加载
 */
onLoad: function(options) {
  /**把 storage 中的内容取出并赋值给 msgData1 */
  var that = this;
  wx.getStorage({
    key: 'msgData1',
    success: function(res) {
      that.setData({
        msgData1: res.data
      });
    },
  })
},
})

```

(2) *comment.wxml* 设置评论页面排版

```

<!--pages/comment/comment.wxml-->
<view class="msg-box">
  <view class="send-box">
    <input value="{{inputVal}}" bindinput='changeInputVal' class="input"
      type="text" placeholder='输入评论: ' placeholder-class='place-input' />
    <button size='mini' type="primary" bindtap='addMsg'>发表</button>
  </view>
  <text class="msg-info" wx:if="{{msgData.length==0}}">暂无评论</text>
  <view class="list-view">
    <view class="item" wx:for="{{msgData}}" wx:key="{{index}}">
      <text class='text1'>{{item.msg}}</text>
      <icon data-index="{{index}}" class="close-btn" bindtap='deleMsg'
        type="cancel"></icon>
    </view>
  </view>
</view>

```

(3) *comment.wxss* 设置评论页面排版所用格式

```

/* pages/comment/comment.wxss */
.msg-box {
  padding: 20px;
}
.send-box {
  display: flex;

```

```

}
.input {
border: 1px solid #ccc;
padding: 5px;
border-radius: 5px;
}
.msg-info {
display: block;
margin: 10px 0 0 0;
color: #390;
}
.place-input {
color: salmon;
}
.list-view {
margin: 20px 0 0 0;
}
.item {
overflow: hidden;
border-bottom: 1px dashed #ccc;
height: 30px;
line-height: 30px;
}
.text1 {
float: left;
}
.close-btn {
float: right;
margin: 5px 5px 0 0;
}

```

f) 模块六：登陆版块（用于验证用户账号密码，识别用户身份）

(1) *login.js* 设置连接数据与数据库接口及函数

```

// pages/login/login.js
Page({
/**
 * 页面的初始数据
 */
data: {
browserroot: '',
browsereplybyid: '',
userid: '',

```

```

parentid: '',
deleteproblembyid: ''
},
bindusername: function(e) {
console.log(e.detail.value)
this.setData({
username: e.detail.value
})
},
bindpassword: function(e) {
console.log(e.detail.value)
this.setData({
password: e.detail.value
})
},
clicktoreg: function() {
wx.request({
url: 'http://172.22.130.33/index.php/Ajaxapi/Ajaxapi/browserroot',
success: function(res) {
console.log(res.data.data)
}
})
},
/**
 * 生命周期函数--监听页面加载
 */
onLoad: function(options) {
wx.getStorage({
key: 'counts',
success: function(res) {
console.log(res)
},
})
},
/**
 * 生命周期函数--监听页面初次渲染完成
 */
onReady: function() {
},
/**
 * 生命周期函数--监听页面显示
 */
onShow: function() {
},

```

```

/**
 * 生命周期函数--监听页面隐藏
 */
onHide: function() {
},
/**
 * 生命周期函数--监听页面卸载
 */
onUnload: function() {
},
/**
 * 页面相关事件处理函数--监听用户下拉动作
 */
onPullDownRefresh: function() {
},
/**
 * 页面上拉触底事件的处理函数
 */
onReachBottom: function() {
},
/**
 * 用户点击右上角分享
 */
onShareAppMessage: function() {
}
})

```

(2) *login.wxml* 设置登陆页面排版

```

<!--pages/login/login.wxml-->
<view class="wrapper">
<view>
<view class="image">
<image style="border-radius:30%;"></image>
</view>
<view class='top'>
<open-data type="userAvatarUrl"></open-data>
</view>
<view class="text">
<open-data type="userNickName"></open-data>
</view>
</view>
</view>
<view class="content">账号: </view>

```

```

<view>
<input style="border:1px solid black" class="weui-input" maxlength="20"
bindinput="bindusername" />
</view>
<view class="content">密码: </view>
<view>
<input password="true" style="border:1px solid black" class="weui-input"
maxlength="20" bindinput="bindpassword" />
</view>
<image style="width:1px; height:1px;" bindload="imageLoad"
src='/images/empty.png'></image>
<button type="primary" bindtap="clicktoreg"> 登陆 </button>

```

(3) *login.wxss* 设置登陆页面排版所用格式

```

/* pages/login/login.wxss */
.wrapper {
background: #18c259;
padding: 25px 0;
}
.image {
width: 100px;
height: 30px;
}
.text {
text-align: center;
color: #fff;
}
.content {
display: flex;
margin-top: 10px;
margin-bottom: 10px;
margin: 20rpx;
}
.top {
width: 250rpx;
height: 250rpx;
margin: 0 auto;
}

```

七、 运行结果



凛冬将至

BBS

1+1=?

确定

取消



凛冬将至

账号:

密码:

登陆

1+1=?

时间: 2019-10-23 23:14:06

查看评论



等于2

发表

暂无评论

输入评论:

发表

等于2