核电施工管理App项目设计说明书

目录

一、引言

本文档提供给开发技术人员使用，用于核电施工管理平台项目的设计与开发。

1.1 编写目的

本文档的目的，旨在规范软件开发，推动项目有序正常的进行，使相关人员遵守统一的规范。节省制作相关文档的时间，降低系统实现的风险，加快项目实施进度，做到系统设计的规范性和全面性，以利于系统的设计、实现、测试、维护和版本升级。

1.2 项目范围

本文档用于软件设计阶段的概要设计。

软件概要设计的范围是：核电施工管理平台共六个模块：分别是双日计划管理模块、安全施工管理模块（包括区域检查记录登记录入、上传、检查结果消息推送、整改项查询等功能）、物项管理模块、质量管理模块、通知模块、我的账户模块。

1.3 读者对象

系统设计人员、软件开发人员、客户方的系统设计人员和项目评审人员。

二、设计概述

2.1. 核电施工管理App项目概括

本核电施工管理平台App用于核电工程项目。

2.2. APP功能

1、双日计划
2、安全施工管理
3、质量管理
4、物项管理
5、通知
6、我的账户

三、系统详细需求分析

3.1、双日计划模块：

双日计划模块分别为八个专业编制，具体专业如下：

机械、主系统、管道、电气、仪表、调试、保温、通风。

双日计划中问题由作业组长创建，提交给班长，班长再提交给队部协调工程师；

组长提交问题后，在我的问题等待处理方案；

队部协调工程师判别是物资问题、技术问题、协调问题；

判别后分别提交给物资部工程师、技术部工程师、工程部工程师处理。

3.2、安全施工管理模块

主要针对现场安全问题发现、整改、验证和关闭来进行；

检查结果上传后推送给责任人，进行整改。

相关部门领导与项目领导班子均可查询整改项信息，跟踪进度。

3.3、质量管理模块

核电项目现场质量问题管理模块；

主要针对现场质量问题发现、整改、验证和关闭来进行；

相关部门领导与项目领导班子均对处理过程可查阅。

3.4、物项管理模块

对接ENPower系统中的物项模块，通过ENPower端接口与App关联数据；

主要包括6个模块：

库存日报模块、入库模块、出库模块、退库模块、库存模块、出库记录查询；

可以直接在App里进行物项的入库、出库、退库、盘库及查询。

3.5、通知模块

通知模块主要用于将各项现场重要信息与会议通知通过点对点、点对多的方式进行信息的发送，协助信息接收方快速、有效的在模块内掌握信息内容，提前做好各项准备工作；

通知模块分为三级菜单，一级菜单为通知，耳机为会议通知、通告及图纸作废通知，三个二级菜单下分别分为相应的三级子菜单；

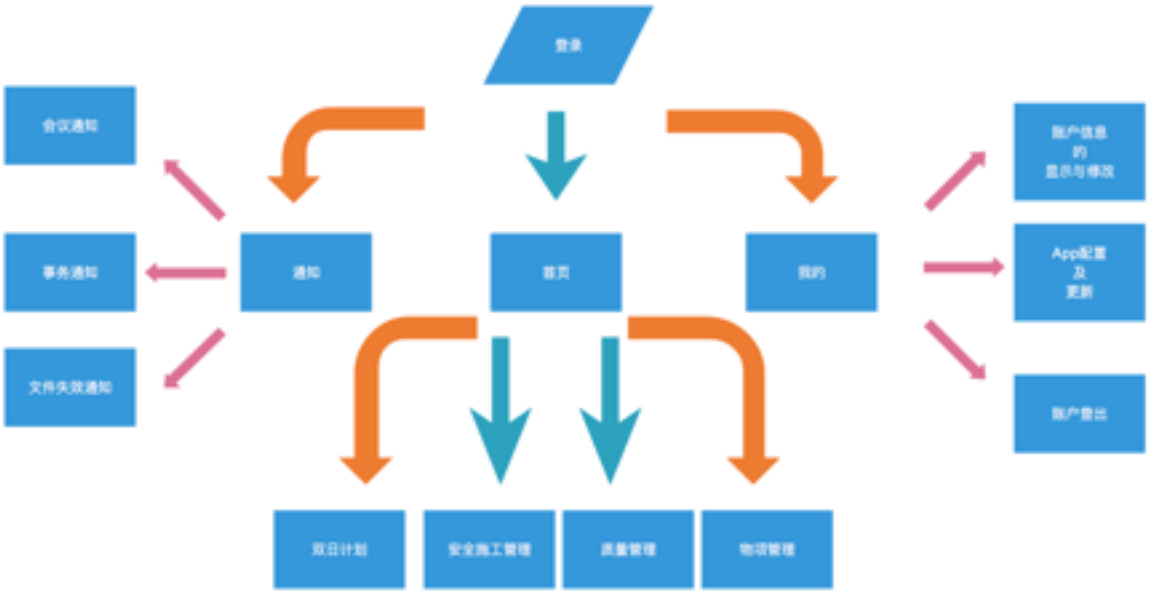
在APP/WEB 端点击三级子菜单可弹出相应的功能窗口，可使用用户进行对应的功能操作。

3.6、我的账户模块

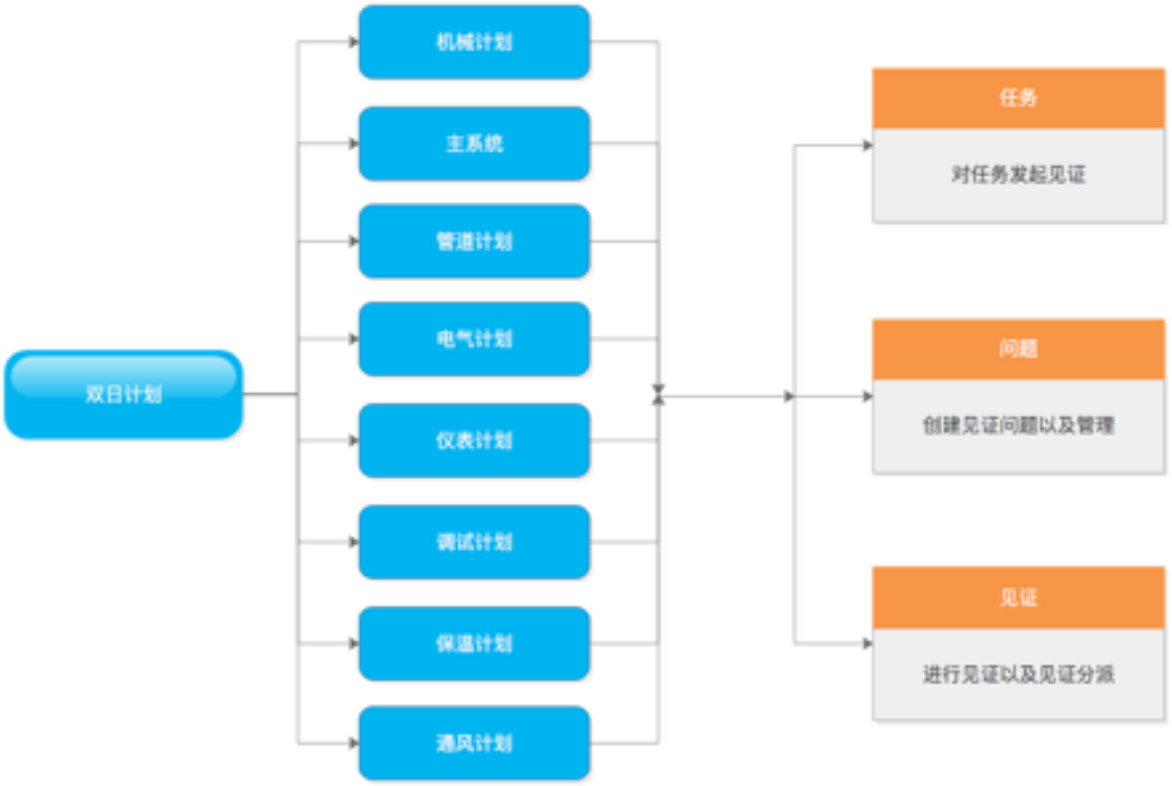
账户信息显示;
账户密码修改;
检测APP是否最新版本，并自动更新;
发送系统日志反馈，便于维护;
退出账户。

四、 总体结构设计

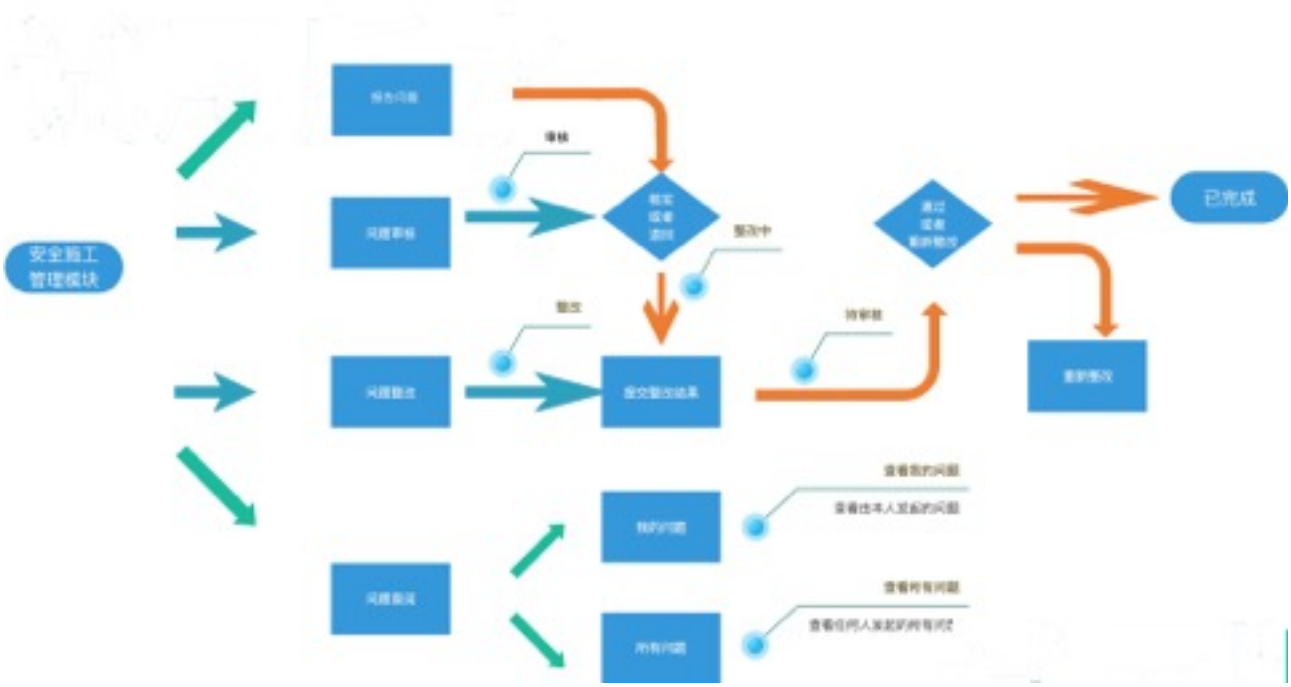
1）总体结构



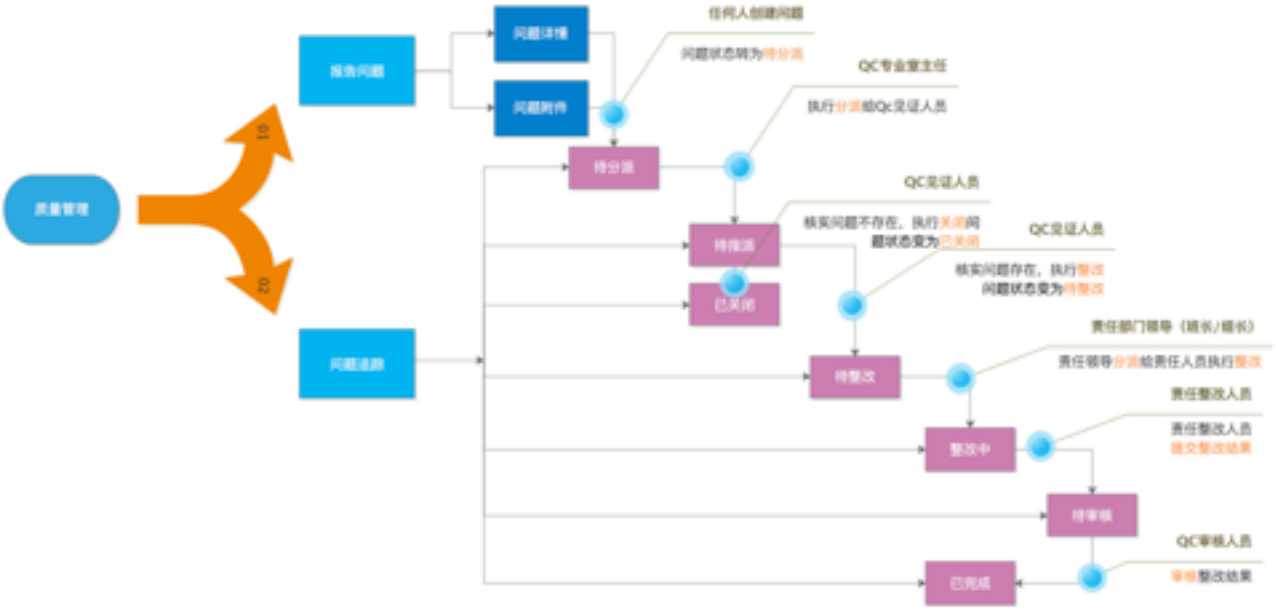
2）双日计划模块


双日计划
机械计划
主系统
管道计划
电气计划
仪表计划
调试计划
保温计划
通风计划
任务
对任务发起见证
问题
创建见证问题以及管理
见证
进行见证以及见证分派

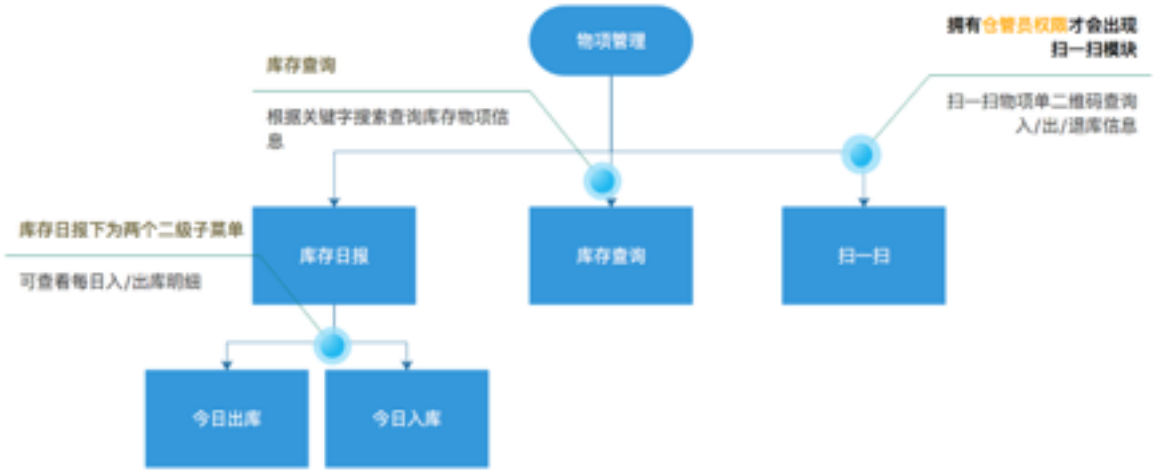
3）安全施工管理模块



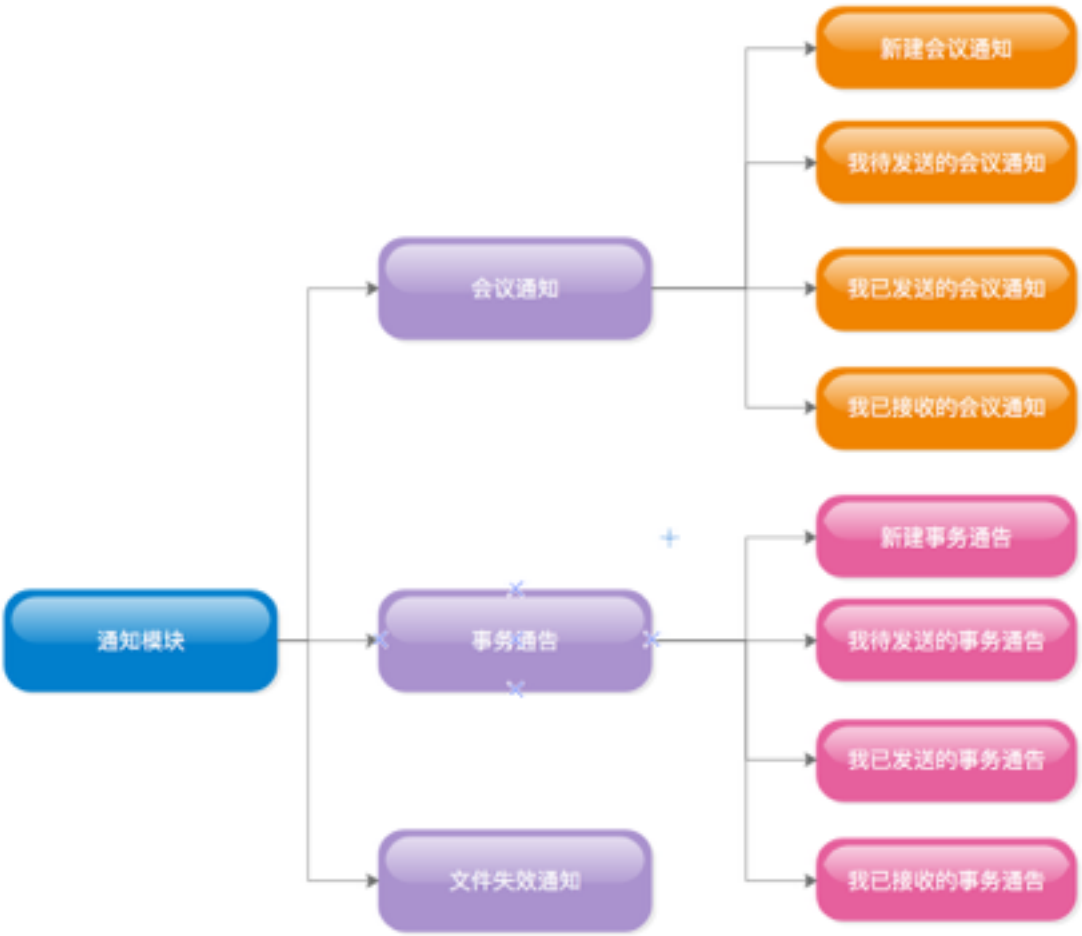
4）质量管理模块


任何人创建问题
问题状态转为待分派
QC专业室主任
执行分派给Qc见证人员
QC见证人员
核实问题不存在，执行关闭问题状态变为已关闭
QC见证人员
核实问题存在，执行整改问题状态变为待整改
责任部门领导（班长/组长）
责任领导分派给责任人员执行整改
责任整改人员
责任整改人员提交整改结果
QC审核人员
审核整改结果

5）物项管理模块



6）通知模块



五、 系统详细设计

5.1、主要页面及页面模块

1）、登录页面

PS：需选择所在项目部，并选择外网环境或内网环境；
外网环境为可连通外网的WIFI以及4G网；
内网环境为局域网内网WIFI。

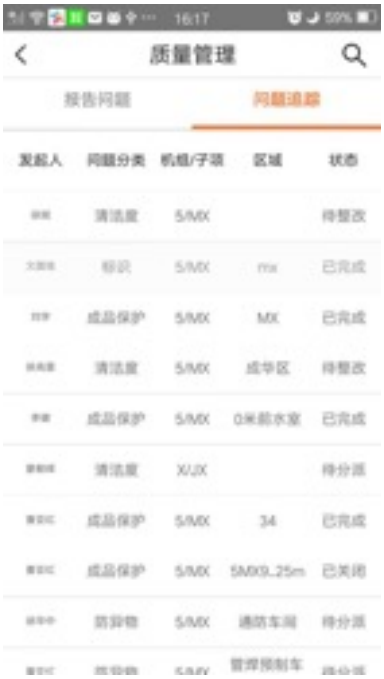
2)、首页主页面

左为实际页面截图； 右为区域标记图
首页主页面分为五个区域:
1 、为滚动轮播图片区域
2 、为双日计划区域
3 、文明施工区域
4 、质量管理区域
5 、物资/物项管理

3）、文明施工管理页面

文明施工模块按功能区分区域，分为报告问题、问题审核、问题整改、问题查阅四类功能区

4）、质量管理页面



根据功能分为报告问题和问题追踪两部分；
问题追踪模块下问题状态分为七种：
待分派、待指派、待整改、整改中、待审核、已完成、已关闭；

5）、物项管理页面

PS：仓管员/物项管理权限下才拥有扫一扫这个功能，可以对物项单上的二维码扫描查看物项信息；

左为仓管员/物项管理权限下 的物项管理页面 ；
右为一般人员的物项管理页面。
6）、通知页面

通知分为会议通知、事务通告、文件失效通知三个子菜单

7）、我的账户页面

我的页面分为六部分区域：

1、账户信息显示：包括人员信息、岗位信息、施工班组信息等等；

2、账户密码修改：修改账户登录密码；

3、退出登录：账户登出并返回到登录页；

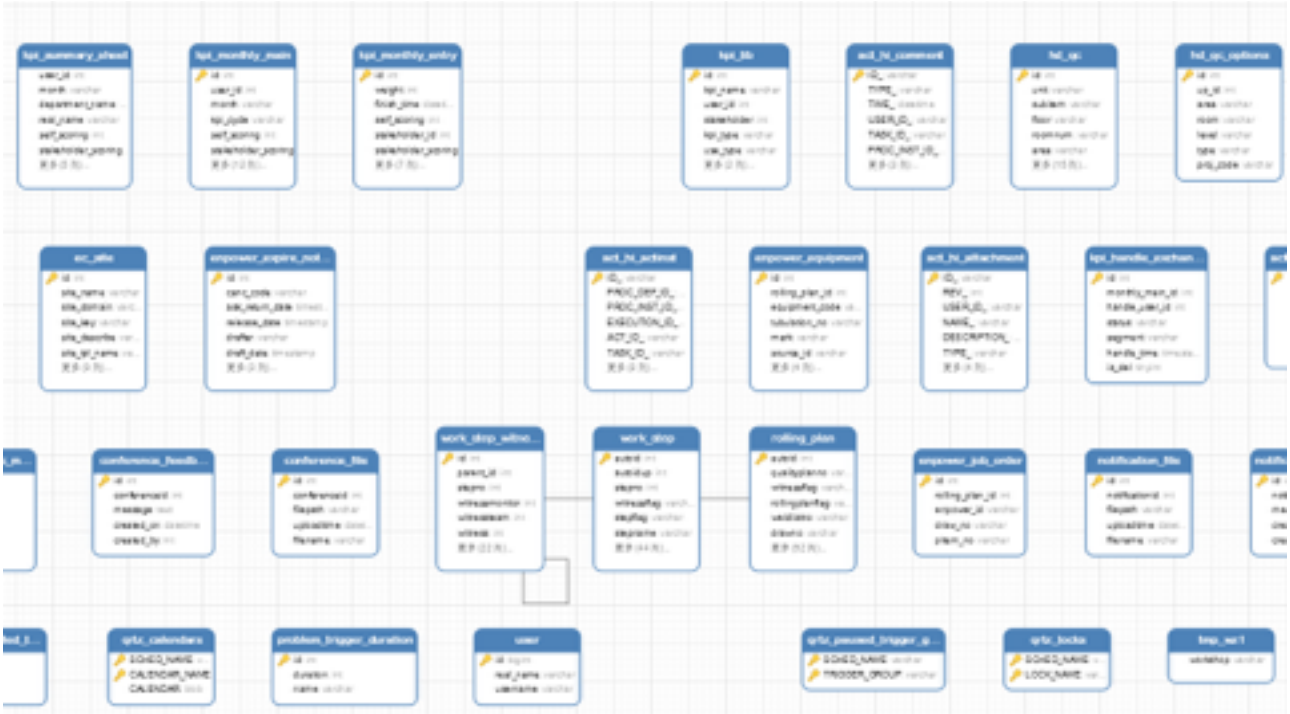
4、发送日志反馈：发送App日志到后台，便于调试；

5、检查新版本：检测当前App是否为最新版本，并选择是否更新到最新版本；

6、版本信息以及项目API信息显示：显示当前App版本以及项目部API信息；

六、 数据库系统设计

6.1、数据库设计



6.2 数据库表单列表

conference	会议表
conference_feedback	会议接收人员表
conference_feedback_mark	会议接收人状态
conference_file	会议文件表
conference_user	会议人员关联表
ec_department	机构部门表
ec_menu	菜单表
ec_privilege	权限表
ec_role	岗位角色表
ec_role_menu	岗位菜单表
ec_site	系统设置表
ec_user	账户人员表
ec_user_role	人员角色表
enpower_equipment	Enpower设置表
enpower_expire_notice	Enpower失效文件通知表
enpower_log	Enpower日志表
enpower_material_info	Enpower材料清单表
enpower_user	Enpower人员同步时间表
hd_qc	质量管理表
hd_qc_file	质量附件表
hd_qc_step	质量管理操作步骤
hse_problem	安全管理表
hse_problem_file	安全附件表
hse_problem_solve	安全附件表
notification	通知表
notification_confirm	通知确认表
notification_feedback	通知接收人员表
notification_feedback_mark	通知接收人状态
notification_file	通知文件表
notification_user	通知人员关联表

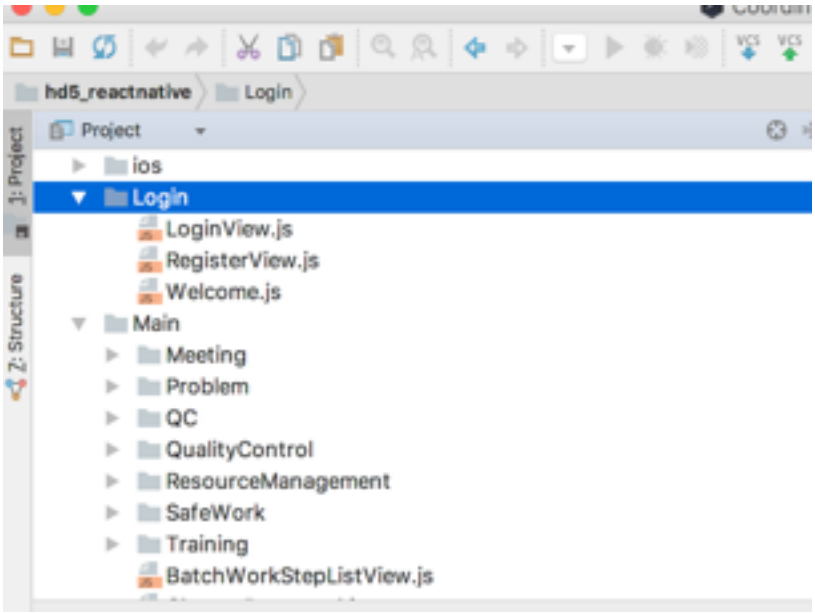
push_log	推送日志
rolling_question	作业问题表
rolling_question_process	作业问题处理流程
rolling_question_solver	作业问题处理状态
user_device	推送设备表
variable_set	设置表
rolling_plan	滚动计划表
work_step	工序表
work_step_witness	工序见证表
work_step_witness_his	工序见证历史表
workstep_witness_file	见证附件表

七、 信息编码设计

7.1、代码结构设计

1）App代码结构设计

代码结构按功能模块区分（如下图）：



主目录下分别为8个区域：

1、登录模块 ： 登录页，起始页

2、会议通知模块：三级菜单模式

3、双日滚动计划模块

4、质量管理模块

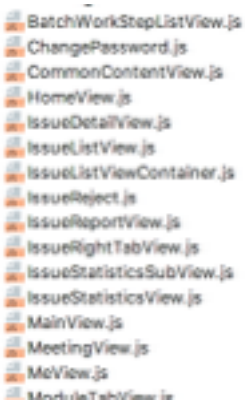
5、物项管理模块

6、安全施工管理模块

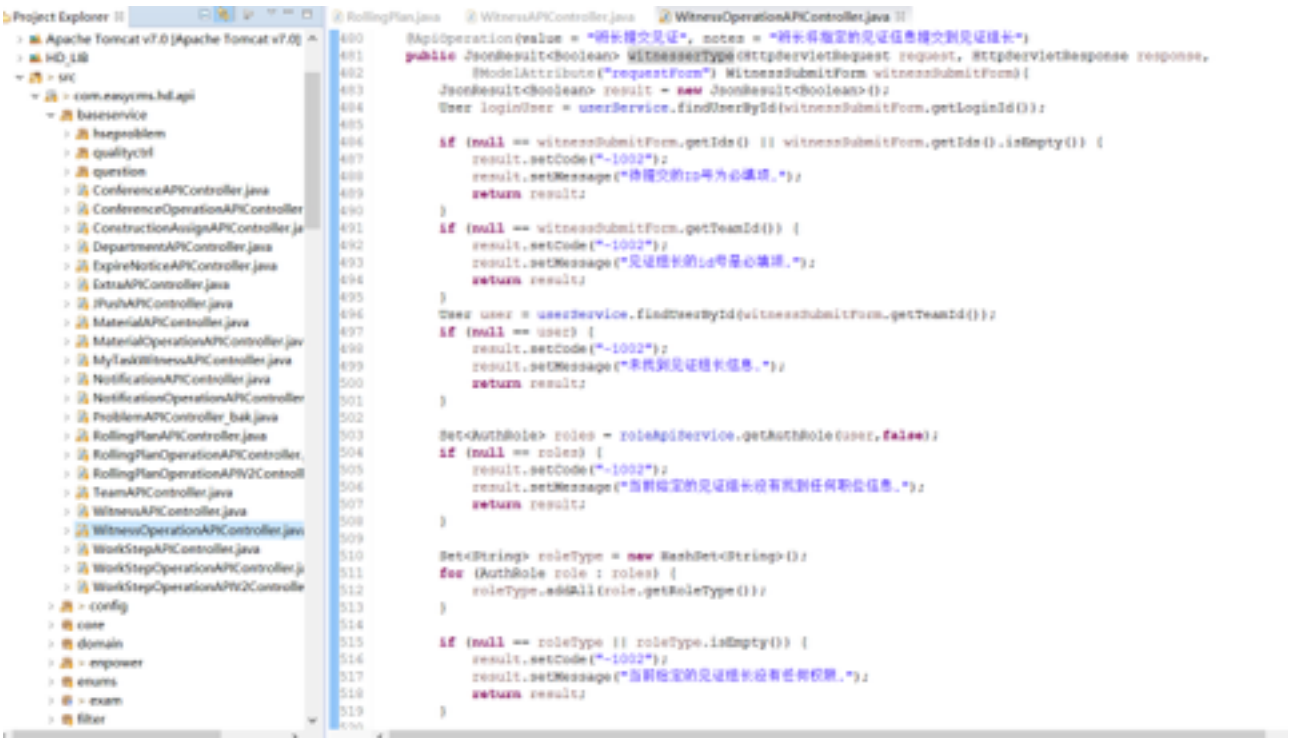
7、公共类文件夹：Http 请求类、可重用自定义页面封装组件、正则格式封装类、全局配置信息等

8、主页面模块：我的账户、主页面

本项目使用react-native 技术开发，使用组件化思想，每个页面以及每个控件都为一个组件。详情如下图：



2）后台代码结构设计



后台代码结构按功能模块区分（如下图）：

主目录下分别为7个模块：

1、登录模块： 登录页

2、设置：系统参数设置、业务参数设置

3、结构设置：人员管理、岗位管理、部门管理

4、施工管理：滚动计划

5、工序见证

6、作业问题管理

7、安全管理

8、质量管理

9、会议通知管理

10、系统管理

7.2　代码编制

1）APP端

主页面代码：TabView.js

```
import React, { Component } from 'react';
import {
    AppRegistry,
    StyleSheet,
    Text,
    View,
    Image,
    BackAndroid
} from 'react-native';


import HomeView from './HomeView';
import MeView from './MeView';
import MeetingView from './MeetingView';
import Navigation from '../common/Navigation';
import TabNavigator from 'react-native-tab-navigator';

export default class TabView extends Component
{
    state =
    {
        selectedTab: 'tab1'
    }

    componentWillMount(){
        var me = this;
        BackAndroid.addEventListener('harwardBackPress', () => {
            const routers = me.props.navigator.getCurrentRoutes();
            if (routers.length > 1) {
                me.props.navigator.pop();
                return true;
            } else {
                    if (routers[0].name == 'MainPage'||routers[0].name == 'LoginView')
{
                      BackAndroid.exitApp();
                      return true;
                    } else {
                      me.props.navigator.pop();
                      return true;
                    }

                  }
                  return false;
      });
    }


      componentWillUnmount() {
                BackAndroid.removeEventListener('hardwareBackPress');
            }

    render()
    {
        return (
            <TabNavigator>
```

```
                <TabNavigator.Item
                    selected={this.state.selectedTab === 'tab1'}
                    title="首页"
                    renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/home_icon.png')} />}
                    renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/home_icon_click.png')} />}
                    badgeText=""
                    selectedTitleStyle={styles.tabBarTintColor}
                    onPress={() => this.setState({ selectedTab: 'tab1' })}>
                    {<HomeView {...this.props}/>}
                </TabNavigator.Item>
                <TabNavigator.Item
                    selected={this.state.selectedTab === 'tab2'}
                    title="通知"
                    renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/meeting_icon.png')} />}
                    renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/meeting_icon_click.png')} />}
                    selectedTitleStyle={styles.tabBarTintColor}
                    onPress={() => this.setState({ selectedTab: 'tab2' })}>
                    {<MeetingView {...this.props}/>}
                </TabNavigator.Item>

                <TabNavigator.Item
                    selected={this.state.selectedTab === 'tab3'}
                    title="我的"
                    renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/me_icon.png')} />}
                    renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/me_icon_click.png')} />}
                    selectedTitleStyle={styles.tabBarTintColor}
                    onPress={() => this.setState({ selectedTab: 'tab3' })}>
                    {<MeView {...this.props}/>}
                </TabNavigator.Item>

            </TabNavigator>

        )
    }
}


const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: 'center',
        alignItems: 'center',
        backgroundColor: '#F5FCFF',
    },
    welcome: {
        fontSize: 20,
        textAlign: 'center',
        margin: 10,
    },
    instructions: {
        textAlign: 'center',
        color: '#333333',
        marginBottom: 5,
```

```
    },
    tabBarTintColor: {

      color: '#0755a6'
    },

});
```

双日计划页面部分业务代码：ModuleTabView.js

```
import React, { Component } from 'react';
import {
    AppRegistry,
    StyleSheet,
    Text,
    View,
    Image,
    BackAndroid,
    DeviceEventEmitter,
} from 'react-native';


import IssueListViewContainer from './IssueListViewContainer';

import WitnessListViewContainer from './WitnessListViewContainer';

import WitnessStatisticsView from './WitnessStatisticsView';
import PlanStatisticsView from './PlanStatisticsView';
import IssueStatisticsView from './IssueStatisticsView';

import WitnessSubViewContainer from './WitnessSubViewContainer';
import PlanStatisticsSubViewContainer from './PlanStatisticsSubViewContainer';
import IssueStatisticsSubView from './IssueStatisticsSubView';

import Navigation from '../common/Navigation';
import TabNavigator from 'react-native-tab-navigator';
import Global from '../common/globals.js'
export default class ModuleTabView extends Component
{
    state =
    {
        selectedTab: 'tab1'
    }

    componentWillMount(){
        var me = this;
        BackAndroid.addEventListener('harwardBackPress', () => {
            const routers = me.props.navigator.getCurrentRoutes();
            if (routers.length > 1) {
                me.props.navigator.pop();
                return true;
            } else {
                    if (routers[0].name == 'MainPage'||routers[0].name == 'LoginView')
{
                      BackAndroid.exitApp();
                      return true;
                    } else {
                      me.props.navigator.pop();
                      return true;
                    }

                  }
                  return false;
      });
    }

    componentDidMount(){
```

```
        subscription = DeviceEventEmitter.addListener('new_issue',(params) =>
this.setState({selectedTab: 'tab2'}))
    }

    componentWillUnmount() {
        BackAndroid.removeEventListener('hardwareBackPress');
        subscription.remove();
    }

    render()
    {
        if (Global.isMonitor(Global.UserInfo)){
            return(
                <TabNavigator>
                    <TabNavigator.Item
                        selected={this.state.selectedTab === 'tab1'}
                        title="任务"
                        renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/task_icon.png')} />}
                        renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/task_icon_click.png')} />}
                        badgeText=""
                        selectedTitleStyle={styles.tabBarTintColor}
                        onPress={() => this.setState({ selectedTab: 'tab1' })}>
                        {<PlanStatisticsSubViewContainer {...this.props}/>}
                    </TabNavigator.Item>
                    <TabNavigator.Item
                        selected={this.state.selectedTab === 'tab2'}
                        title="问题"
                        renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/problem_icon.png')} />}
                        renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/problem_icon_click.png')} />}
                        selectedTitleStyle={styles.tabBarTintColor}
                        onPress={() => this.setState({ selectedTab: 'tab2' })}>
                        {<IssueStatisticsSubView {...this.props}/>}
                    </TabNavigator.Item>

                    <TabNavigator.Item
                        selected={this.state.selectedTab === 'tab3'}
                        title="见证"
                        renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/witness_icon.png')} />}
                        renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/witness_icon_click.png')} />}
                        selectedTitleStyle={styles.tabBarTintColor}
                        onPress={() => this.setState({ selectedTab: 'tab3' })}>
                        {<WitnessSubViewContainer {...this.props}/>}
                    </TabNavigator.Item>

                </TabNavigator>
            )

        } else if (Global.isGroup(Global.UserInfo)){
            return(
                <TabNavigator>
                    <TabNavigator.Item
                        selected={this.state.selectedTab === 'tab1'}
                        title="任务"
```

```
                        renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/task_icon.png')} />}
                        renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/task_icon_click.png')} />}
                        badgeText=""
                        selectedTitleStyle={styles.tabBarTintColor}
                        onPress={() => this.setState({ selectedTab: 'tab1' })}>
                        {<PlanStatisticsSubViewContainer {...this.props}/>}
                    </TabNavigator.Item>
                    <TabNavigator.Item
                        selected={this.state.selectedTab === 'tab2'}
                        title="问题"
                        renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/problem_icon.png')} />}
                        renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/problem_icon_click.png')} />}
                        selectedTitleStyle={styles.tabBarTintColor}
                        onPress={() => this.setState({ selectedTab: 'tab2' })}>
                        {<IssueListViewContainer {...this.props}/>}
                    </TabNavigator.Item>

                    <TabNavigator.Item
                        selected={this.state.selectedTab === 'tab3'}
                        title="见证"
                        renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/witness_icon.png')} />}
                        renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/witness_icon_click.png')} />}
                        selectedTitleStyle={styles.tabBarTintColor}
                        onPress={() => this.setState({ selectedTab: 'tab3' })}>
                        {<WitnessListViewContainer {...this.props}/>}
                    </TabNavigator.Item>

                </TabNavigator>
            )

        }else{

        return (
            <TabNavigator>
                <TabNavigator.Item
                    selected={this.state.selectedTab === 'tab1'}
                    title="任务"
                    renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/task_icon.png')} />}
                    renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/task_icon_click.png')} />}
                    badgeText=""
                    selectedTitleStyle={styles.tabBarTintColor}
                    onPress={() => this.setState({ selectedTab: 'tab1' })}>
                    {<PlanStatisticsView {...this.props}/>}
                </TabNavigator.Item>
                <TabNavigator.Item
                    selected={this.state.selectedTab === 'tab2'}
                    title="问题"
                    renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/problem_icon.png')} />}
                    renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/problem_icon_click.png')} />}
```

```
                    selectedTitleStyle={styles.tabBarTintColor}
                    onPress={() => this.setState({ selectedTab: 'tab2' })}>
                    {<IssueStatisticsView {...this.props}/>}
                </TabNavigator.Item>

                <TabNavigator.Item
                    selected={this.state.selectedTab === 'tab3'}
                    title="见证"
                    renderIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/witness_icon.png')} />}
                    renderSelectedIcon={() => <Image style={{width:24,height:24,}}
source={require('../images/witness_icon_click.png')} />}
                    selectedTitleStyle={styles.tabBarTintColor}
                    onPress={() => this.setState({ selectedTab: 'tab3' })}>
                    {<WitnessStatisticsView {...this.props}/>}
                </TabNavigator.Item>

            </TabNavigator>
        )
            }

    }
}


const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: 'center',
        alignItems: 'center',
        backgroundColor: '#F5FCFF',
    },
    welcome: {
        fontSize: 20,
        textAlign: 'center',
        margin: 10,
    },
    instructions: {
        textAlign: 'center',
        color: '#333333',
        marginBottom: 5,
    },
    tabBarTintColor: {

      color: '#f77935'
    },

});
```

会议通知部分页面代码：MeetingView.js

```
import React, { Component } from 'react';
import {
    StyleSheet,
    Text,
    View,
    Image,
    TouchableOpacity,
    Platform,
    TouchableNativeFeedback,
    TouchableHighlight,
    Picker,
    AsyncStorage,
    TextInput,
    ScrollView,
    DeviceEventEmitter,
} from 'react-native';

import NavBar from '../common/NavBar'
import Dimensions from 'Dimensions'
import LoginView from '../Login/LoginView'
var Global = require('../common/globals');
var width = Dimensions.get('window').width;
import CommitButton from '../common/CommitButton'
import MeetingListViewContainer from '../Main/Meeting/MeetingListViewContainer';
import NoticeExpiredListView from '../Main/Meeting/NoticeExpiredListView';

import HttpRequest from '../HttpRequest/HttpRequest'
import CreateMeetingView from '../Main/Meeting/CreateMeetingView';
import EditSubjectView from '../Main/Meeting/EditSubjectView';

import CreateNoticeView from '../Main/Meeting/CreateNoticeView';

var meetingModuleData = [
    {
        'index': 0,
        'title': '已接收',
        "type": "receive",
         "tag":'meeting',
        'image': require('../images/received_icon.png')
    },{
        'index': 1,
        'title': '已发送',
        "type": "send",
        "tag":'meeting',
        'image': require('../images/send_icon.png')
    },{
        'index': 2,
        'title': '草稿',
        "type": "draft",
        "tag":'meeting',
        'image': require('../images/draft_icon.png')
    },]

    var noticeModuleData = [
        {
            'index': 0,
            'title': '已接收',
```

```
            "type": "receive",
            "tag":'notice',
            'image': require('../images/received_icon.png')
        },{
            'index': 1,
            'title': '已发送',
            "type": "send",
            "tag":'notice',
            'image': require('../images/send_icon.png')
        },{
            'index': 2,
            'title': '草稿',
            "type": "draft",
            "tag":'notice',
            'image': require('../images/draft_icon.png')
        },]


export default class MeetingView extends Component {
    constructor(props) {
        super(props)
        this.state={
            expired_notice:[]
        }

    }


    back() {
        this.props.navigator.pop()
    }




    createMeeting(tag){
        if (tag == 'meeting') {
            this.props.navigator.push({
                component: CreateMeetingView,
                 props: {
                     type:tag,
                    }
            })
        }else{
            this.props.navigator.push({
                component: CreateNoticeView,
                 props: {
                     type:tag,
                    }
            })

        }

    }

    onGetDataSuccess(response,paramBody){
         Global.log('onGetDataSuccess@@@@')
     var query = this.state.filter;
     if (!query) {
         query = '';
```

```
     }

        var datas = response.responseResult.datas;
        if (datas) {
            this.setState({expired_notice:datas})
        }

    }

    componentDidMount(){
        var paramBody = {
             pagesize:10,
             pagenum:1,
            }



   HttpRequest.get('/enpower/invalidFileList', paramBody,
this.onGetDataSuccess.bind(this),
       (e) => {


          try {
              var errorInfo = JSON.parse(e);
              if (errorInfo != null) {
               Global.log(errorInfo)
              } else {
                  Global.log(e)
              }
          }
          catch(err)
          {
              Global.log(err)
          }

          Global.log('Task error:' + e)
       })
       this.executeStatisticsRequest()

       operationSubscription = DeviceEventEmitter.addListener('operate_meeting',
(param)=>{this.executeStatisticsRequest()})
   }

   componentWillUnmount(){

     operationSubscription.remove();
   }



    onGetStatisticsDataSuccess(response,paramBody){
         Global.log('onGetDataSuccess@@@@')


        var conference = response.responseResult.conference;
        if (conference) {
            this.setState({conference:conference})
        }

        var notification = response.responseResult.notification;
```

```
        if (notification) {
            this.setState({notification:notification})
        }

    }


    executeStatisticsRequest(){

      Global.log('executeStatisticsRequest:')



                 var paramBody = {
                     }

            HttpRequest.get('/statistics/conference', paramBody,
this.onGetStatisticsDataSuccess.bind(this),
                (e) => {

                    //
                    // this.setState({
                    //   dataSource: this.state.dataSource.cloneWithRows([]),
                    //   isLoading: false,
                    // });
                    try {
                        var errorInfo = JSON.parse(e);
                        if (errorInfo != null) {
                         Global.log(errorInfo)
                        } else {
                            Global.log(e)
                        }
                    }
                    catch(err)
                    {
                        Global.log(err)
                    }

                    Global.log('Task error:' + e)
                })
    }



    onModuleItemClick(itemData) {
        this.props.navigator.push({
            component: MeetingListViewContainer,
             props: {
                 data:itemData,
                 tag:itemData.tag,
                }
        })
    }


      enterExpiredNotice(){
          this.props.navigator.push({
              component: NoticeExpiredListView,
               props: {
```

```
                 type:this.props.type,
                }
          })
      }

    onExpriedDetailModuleItemClick(itemData) {
        this.props.navigator.push({
            component: NoticeExpiredListView,
             props: {
                 data:itemData,
                 type:this.props.type,
                }
        })
    }

  renderDot(unread,image){
       if (unread>0) {
           return(<View style={{

           flexDirection: 'row',
           justifyContent: 'space-between',

         }}>
                <Image source={image} style={{ width: 48, height: 48,marginTop:
5,marginLeft:18,marginRight:18}} resizeMode={Image.resizeMode.contain} />
                <Text style={{ left:58,position: 'absolute',fontSize: 12, color:
"#e82628" }}>{unread}</Text>

                </View>)
       }else{
           return(<View style={{
         }}>
                <Image source={image} style={{  width: 48, height: 48,}}
resizeMode={Image.resizeMode.contain} />
                </View>)
       }
  }

  renderItems(moduleData){
      var displayArr = []
      for (var i = 0; i < moduleData.length; i++) {
          var moduleDataItem = moduleData[i]


         var unread = this.getUnreadCount(moduleDataItem)

         displayArr.push(
         <TouchableOpacity key={i} style={styles.cell}
onPress={this.onModuleItemClick.bind(this,moduleDataItem)}>
         <View style={[{width:width/3}, styles.cell]}>

             {this.renderDot(unread,moduleDataItem.image)}

             <Text style={{ fontSize: 12, color: "#282828" }}>{moduleDataItem.title}</
Text>
         </View>
         </TouchableOpacity>)
      }

      return displayArr
```

```
  }


  getUnreadCount(moduleDataItem){
      var unread = 0
      if (moduleDataItem.tag == 'meeting') {
            if (this.state.conference) {
                if (moduleDataItem.type == 'send') {
                    unread = this.state.conference.sendUnread
                }else if(moduleDataItem.type == 'draft'){
                    unread = 0
                }else{
                    unread = this.state.conference.receiveUnread
                }

            }
      }else{
          if (this.state.notification) {
              if (moduleDataItem.type == 'send') {
                  unread = this.state.notification.sendUnread
              }else if(moduleDataItem.type == 'draft'){
                  unread = 0
              }else{
                  unread = this.state.notification.receiveUnread
              }
          }
      }
      if (!unread) {
          return 0
      }
      return unread

  }

  renderExpriedItems(moduleData){
      var displayArr = []
      var len = moduleData.length > 3 ? 3 : moduleData.length



      for (var i = 0; i < len; i++) {
          var moduleDataItem = moduleData[i]
         displayArr.push(
         <TouchableOpacity key={i}
onPress={this.onExpriedDetailModuleItemClick.bind(this,moduleDataItem)}>
         <View style={[{width:width}]}>

             <Text numberOfLines={1} style={{ fontSize: 14, marginTop:10,marginRight:
10,color: "#e82628" }}> {Global.formatDate(moduleDataItem.releaseDate)} 文件失效通知
{moduleDataItem.cancCode}</Text>
         </View>
         </TouchableOpacity>)
      }

      return displayArr
  }
```

```
    render() {
        return (
            <View style={styles.container}>
                <NavBar
                    title="通知"
                     />
                     <ScrollView
                     style={styles.mainStyle}>
                     <View style={styles.itemContainer}>
                     <View style={styles.space}/>
                      <View style={styles.flexContainer}>
                      <Image style={{width:24,height:24,}} source={require('../images/
meetingNoticeIcon.png')} />
                      <Text style={[styles.content,{fontSize:16,color:'#444444',}]}>
                      会议通知
                      </Text>

                      <TouchableOpacity
onPress={this.createMeeting.bind(this,'meeting')} style={{flex:
1,flexDirection:'row',justifyContent:'flex-end', alignItems: 'center',}}>
                      <Text style={[styles.content,{fontSize:14,color:'#1c1c1c'}]}>
                      新建会议
                      </Text>
                      <Image style={{width:24,height:24,}} source={require('../images/
newIcon.png')} />

                      </TouchableOpacity>


                      </View>

                      <View style={styles.divider}/>

                      <View style={styles.statisticsflexContainer}>
                        {this.renderItems(meetingModuleData)}
                      </View>

                      </View>

                       <View style={styles.space}/>


                       <View style={styles.itemContainer}>

                        <View style={styles.flexContainer}>
                        <Image style={{width:24,height:24,}} source={require('../
images/affair_icon.png')} />

                        <Text style={[styles.content,{fontSize:16,color:'#444444',}]}>
                         事务通告
                        </Text>

                        <TouchableOpacity
onPress={this.createMeeting.bind(this,'notice')} style={{flex:
1,flexDirection:'row',justifyContent:'flex-end', alignItems: 'center',}}>
                        <Text style={[styles.content,{fontSize:14,color:'#1c1c1c'}]}>
                        新建通告
                        </Text>
```

```
                        <Image style={{width:24,height:24,}} source={require('../
images/newIcon.png')} />

                        </TouchableOpacity>


                        </View>

                        <View style={styles.divider}/>

                        <View style={styles.statisticsflexContainer}>
                          {this.renderItems(noticeModuleData)}
                        </View>

                        </View>

                         <View style={styles.space}/>

                         <View style={styles.itemContainer}>

                          <TouchableOpacity
onPress={this.enterExpiredNotice.bind(this)} style={styles.flexContainer}>
                          <Image style={{width:24,height:24,}} source={require('../
images/invalidDocumentIcon.png')} />

                          <Text style={[styles.content,{fontSize:16,color:'#444444',}]}
>
                           文件失效通知
                          </Text>

                          <View style={{flex:
1,flexDirection:'row',justifyContent:'flex-end', alignItems: 'center',}}>
                          <Text style={[styles.content,{fontSize:14,color:'#777777'}]}>
                          查看全部
                          </Text>
                          <Image style={{width:24,height:24,}} source={require('../
images/detailsIcon.png')} />

                          </View>


                          </TouchableOpacity>

                          <View style={styles.divider}/>

                          <View style={styles.expired_statisticsflexContainer}>
                          <Text style={[{fontSize:14,color:'#777777'}]}>
                          最新
                          </Text>

                            {this.renderExpriedItems(this.state.expired_notice)}
                          </View>

                          </View>

                        </ScrollView>

            </View>
        )
```

```
    }


}


const styles = StyleSheet.create({

    container: {
        flex: 1,
        justifyContent: 'flex-start',
        alignItems: 'center',
        backgroundColor: '#f2f2f2',
    },
    itemContainer: {

    },
    flexContainer: {
           height: 48,
           width: width,
           backgroundColor: '#ffffff',
           // 容器需要添加direction才能变成让子元素flex
           flexDirection: 'row',
           alignItems: 'center',
           padding:10,
    },
    space: {
    backgroundColor: '#f2f2f2',
    width: width,
    height: 10,
},
    divider: {
    backgroundColor: '#d6d6d6',
    width: width,
    height: 0.5,
    },

    cell: {
        flex: 1,
        height: 84,
        width:width/3,
        justifyContent: "center",
        alignItems: 'center',
         flexDirection: 'column',
    },
    statisticsflexContainer: {
             height: 96,
             backgroundColor: '#ffffff',
             flexDirection: 'row',
             justifyContent: "center",
             alignItems: 'center',
         },
         expired_statisticsflexContainer: {
                  backgroundColor: '#ffffff',
                  padding:10
              },
})
```

安全文明施工页面代码：SafeWorkHomeView.js

```
import React, { Component } from 'react';
import {
    AppRegistry,
    StyleSheet,
    Text,
    View,
    Image,
    Navigator,
    BackAndroid,
    ListView,
    TouchableOpacity,
    ScrollView,
    // AlertIOS,

} from 'react-native';

import Dimensions from 'Dimensions'
import NavBar from '../../common/NavBar';
import ProblemReview from '../SafeWork/ProblemReview'
import ProblemRectification from '../SafeWork/ProbleRectification'
import ProblemAceess from '../SafeWork/ProblemAccess'
import ProblemReport from '../SafeWork/ProblemReport'
import Global from '../../common/globals'

var width = Dimensions.get('window').width;

var safeModule = [
  {
    'title':"报告问题",
    'image': require('../../images/reportIcon.png'),
    'index': 0,
    "type":"BGWT",
    "right":true,
    "bottom":true,

  },
  {
    'title':"问题审核",
    'image': require('../../images/checkIcon.png'),
    'index': 1,
    "type":"WTSH",
    "right":false,
    "bottom":true,
  },
  {
    'title':"问题整改",
    'image': require('../../images/correctionIcon.png'),
    'index': 2,
    "type":"WTZG",
    "right":true,
    "bottom":false,
  },
  {
    'title':"问题查阅",
    'image': require('../../images/lookIcon.png'),
    'index': 3,
    "type":"WTCY",
```

```
    "right":false,
    "bottom":false,
  },
]

export  default class SafeWorkHomeView extends Component {

  constructor(props) {
      super(props)

      var ds = new ListView.DataSource({ rowHasChanged: (r1, r2) => r1 !== r2 });

      this.state = {
          title: "安全文明任务",
          dataSource: ds,
      }

  }

  componentDidMount() {

      this.setState({
          dataSource: this.state.dataSource.cloneWithRows(safeModule),
      });


  }

  back() {
      this.props.navigator.pop()
  }

  render() {
      return (
          <View style={styles.container}>
              <NavBar
              title={this.state.title}
              leftIcon={require('../../images/back.png')}
              leftPress={this.back.bind(this)}/>
               <ScrollView style={styles.content}>
                      {this.renderHeaderView()}
                     {this.renderToolsView()}
               </ScrollView>

          </View>
      )
  }

  renderHeaderView(){
       return(
         <View style = {{width:width,height:
200,backgroundColor:'#3CDEBA',alignItems:'center',justifyContent:'center'}}>
           <Image source={require('../../images/construction_icon.png')} style={{width:
120,height:120}} resizeMode={Image.resizeMode.contain}></Image>
         </View>
       );
  }

  renderToolsView() {
      return(
```

```
          <ListView
            dataSource={this.state.dataSource}
            renderRow={this.renderRow.bind(this)}
            contentContainerStyle={{
                  justifyContent: 'space-around',
                  flexDirection:'row', //改变ListView的主轴方向
                  flexWrap:'wrap', //换行
                  alignItems:'center', // 必须设置,否则换行不起作用
                  marginLeft:30,
                  marginRight:30,
                  marginTop:50,
                  marginBottom:50,
          }}
            showsVerticalScrollIndicator={false}
            showsHorizontalScrollIndicator={false}
        />
      )
i

  }

  renderRow(item,sectionId,rowId){
      return(
           <View key = {item.index}
                style={[{width:width/2-30,
                        height:width/2-30,
                        borderRightWidth:item.right ? 1 : 0,
                        borderBottomWidth:item.bottom ? 1 : 0,
                        borderColor:'#DDDDDD',
                      }, styles.toolsItem]}>

              {this.renderDot(item)}

           </View>
      )
  }

  _itemClick(item){

// AlertIOS.Global.alert(item.type,item.title);

if (item.type == "WTSH" && !Global.isHSE(Global.UserInfo)) {
   Global.alert("当前用户不是HSE部门成员,无法执行该操作");
   return;
}

var Component;

  switch (item.type) {
    case "BGWT":
    {
     Component =   ProblemReport;
    }
    break;

      break;
    case "WTSH":
      {
```

```
          Component = ProblemReview;
        }
        break;
        case "WTZG":
        {
          Component = ProblemRectification;
        }
            break;
        case "WTCY":
            {
          Component = ProblemAceess;
            }
            break;

  }

  if (Component) {
    this.props.navigator.push({
        component: Component,

    })
  }


  }

  renderDot(item){

    return(
      <TouchableOpacity style={{alignSelf:'center'}}
        onPress={this._itemClick.bind(this,item)}>
          <Image source={item.image} style={styles.circle_outter}
resizeMode={Image.resizeMode.contain}></Image>
           <Text style={styles.item}> {item.title} </Text>
      </TouchableOpacity>
    );

  }

}

const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: 'flex-start',
        alignItems: 'center',
        backgroundColor: '#fff',
    },
    content: {
      flex: 1,
      backgroundColor: '#fff',
    },
    toolsItem: {
        backgroundColor: "#fff",
        justifyContent: "center",
        alignItems: "center",

    },
    item: {
      color: "#FBAC2B",
```

```
      alignSelf:'center',
      fontSize:16,
    },
    circle_outter:{
      marginBottom: 6,
      alignSelf:'center',
      width: 88,
      height: 88,
    },

  });
```

质量管理部分页面代码：ProblemView.js

```
import React, { Component } from 'react';
import {
    StyleSheet,
    Text,
    View,
    Image,
    TouchableOpacity,
    Platform,
    ScrollView,
    RefreshControl,
    ActivityIndicator,
    TouchableNativeFeedback,
    TouchableHighlight,
    InteractionManager,
} from 'react-native';
import HttpRequest from '../../HttpRequest/HttpRequest'
import Dimensions from 'Dimensions';
import NavBar from '../../common/NavBar'
import LoadMoreFooter from '../../common/LoadMoreFooter.js'
import CircleLabelHeadView from '../../common/CircleLabelHeadView';
import px2dp from '../../common/util'
import dateformat from 'dateformat'
import Global from '../../common/globals.js';
import ConstMapValue from '../../common/ConstMapValue.js';

const isIOS = Platform.OS == "ios"
var width = Dimensions.get('window').width;
import QualityCheckList from './QualityCheckList'
import ProblemReport from './ProblemReport'
import   ScrollableTabView  from 'react-native-scrollable-tab-view';

var LOADING = {};


export default class ProblemView extends Component {
    constructor(props) {
        super(props)


        this.state = {

            title: "质量管理",
            keyword:'',
```

```
    }


}


    back() {
        this.props.navigator.pop()
    }



componentDidMount() {


}




rendTabs(){
    return( <ScrollableTabView locked={true}
           tabBarUnderlineStyle={{backgroundColor: '#f77935'}}
           tabBarBackgroundColor='#FFFFFF'
           tabBarActiveTextColor='#f77935'
           tabBarInactiveTextColor='#777777'>

     {this.renderFeedbackView('报告问题',0)}
     {this._renderQuetionView('问题追踪',1)}
</ScrollableTabView>

    )


}

render() {
    return (
        <View style={styles.container}>
            <NavBar
            title={this.state.title}
            leftIcon={require('../../images/back.png')}
            searchMode={true}
            onSearchChanged={(text) => this.onSearchChanged(text)}
            onSearchClose = {this.onSearchClose.bind(this)}
            leftPress={this.back.bind(this)} />
            {this.rendTabs()}
        </View>
    )
}

onSearchChanged(text){
  if (this.refs.myQuestionlist) {
        this.refs.myQuestionlist.onSearchChange(text);
  }
}

onSearchClose(){
```

```
  if (this.refs.myQuestionlist) {
        this.refs.myQuestionlist.onSearchChange('');
  }
 }

 renderFeedbackView(label,index) {
     return (
         <ProblemReport
         tabLabel={label}
          style={{flex:1}}
          type={this.props.type}
          navigator={this.props.navigator}
          />

     )
 }

 _renderQuetionView(label){

return( <View  tabLabel={label} style={styles.container}>


  {this.renderContent()}
  {this.renderListView()}

    </View>)


 }

 renderListView() {

     var userId = '';

     return (<QualityCheckList
        style={{alignSelf:'stretch',flex:1}}
        detailType={"1003"}
        navigator={this.props.navigator}
        ref="myQuestionlist"
        />)
 }

 renderContent(){

    return(
      <View>
      <View style={styles.statisticsflexContainer}>

      <View style={styles.cell}>
       <Text style={{color:'#1c1c1c',fontSize:12,marginBottom:2,}}>
          发起人
        </Text>
      </View>

      <View style={styles.cell}>
      <Text style={{color:'#1c1c1c',fontSize:12,marginBottom:2,}}>
        问题分类
      </Text>
      </View>
```

```
          <View style={styles.cell}>
          <Text style={{color:'#1c1c1c',fontSize:12,marginBottom:2,}}>
            机组/子项
          </Text>
          </View>

          <View style={styles.cell}>
           <Text style={{color:'#1c1c1c',fontSize:12,marginBottom:2,}}>
                     区域
             </Text>
           </View>

           <View style={styles.cell}>
           <Text style={{color:'#1c1c1c',fontSize:12,marginBottom:2,}}>
                      状态
           </Text>
            </View>

          </View>

          <View style={{backgroundColor:'d6d6d6',height:0.5,width:width}}>
          </View>

          </View>

        )

    }



}


const styles = StyleSheet.create({
    container: {
        flex: 1,
        justifyContent: 'flex-start',
        alignItems: 'center',
        backgroundColor: '#f2f2f2',
    },
    topView: {
        height: 150,
        width: width,
    },

    divider: {
    backgroundColor: '#d6d6d6',
    width: width,
    height: 0.5,
    },
    label:{
            color: '#ffffff',
            fontSize:16,
    },
    content:{
            color: '#1c1c1c',
```

```
            fontSize:16,
    },
    scrollSpinner: {
       marginVertical: 20,
     },
    flexContainer: {
           height: 54,
           width: width,
           backgroundColor: '#ffffff',
           // 容器需要添加direction才能变成让子元素flex
           flexDirection: 'row',
           alignItems: 'center',
           padding:10,
    },
    itemContainer: {
            flex:1,
            marginTop:0.5,
    },
     statisticsflexContainer: {
              height: 57.5,
              backgroundColor: '#ffffff',
              flexDirection: 'row',
          },

    cell: {
        flex: 1,
        height: 57.5,
        width:width/3,
        justifyContent: "center",
        alignItems: 'center',
         flexDirection: 'column',
    },

    cellLine: {
        width: 2,
        height: 14,
        backgroundColor: '#cccccc',
    },

});
```

物项管理部分页面代码：ResourceManageView.js

```
import React, { Component } from 'react';
import {View, TouchableOpacity, Image, Text, ScrollView} from 'react-native';

import Dimensions from 'Dimensions';
import NavBar from '../../common/NavBar';
import CardView from 'react-native-cardview'
import ScanQRCodeView from './ScanQRCodeView';
import DailyReportView from './DailyReportView';
import InfoInputView from './InfoInputView';
import Global from '../../common/globals';

var width = Dimensions.get('window').width;

export default class ResourceManageView extends Component{


   renderScanView(){
          if(!Global.isKeeper(Global.UserInfo)){

               return;
          }

          return(<TouchableOpacity onPress = {() => this.goToScanQRCode()}>
                        <CardView
                               style = {{marginTop: 10, marginLeft: 10, marginRight: 10,
backgroundColor: '#ffffff', height: width*0.28, flexDirection: 'row', alignItems:
'center'}}
                               cardElevation = {4}
                               cornerRadius = {6}>
                               <Image
                                      style = {{width: 80, height: 80, marginLeft: 10}}
                                      source = {require('../../images/scan_icon.png')} />
                               <View>
                                      <Text style={{color: '#444444', fontSize: 18,
fontWeight: 'bold'}}>扫一扫</Text>
                                      <Text style={{color: '#777777', fontSize: 14,
width: width*0.67}}>查看物项信息，入库、出库、退库处理</Text>
                               </View>
                        </CardView>
                 </TouchableOpacity>);

   }

       render(){
              return(
                     <View style={{flex:1,flexDirection: 'column', backgroundColor:
'#f2f2f2'}}>
                            <NavBar
                           title={'物项管理'}
                           leftIcon={require('../../images/back.png')}
                           leftPress={this.back.bind(this)} />
                            <Image
                                   style={{width: width, height: width*0.53}}
                                   source={require('../../images/images.png')} />
                            <TouchableOpacity onPress = {() => this.goToDailyReport()}>
                                   <CardView
```

```
                                        style = {{marginTop: 10, marginLeft: 10,
marginRight: 10, backgroundColor: '#ffffff', height: width*0.28, flexDirection: 'row',
alignItems: 'center'}}
                                        cardElevation = {4}
                                        cornerRadius = {6}>
                                        <Image
                                              style = {{width: 80, height: 80,
marginLeft: 10}}
                                              source = {require('../../images/
daily_icon.png')} />
                                        <View>
                                              <Text style={{color: '#444444', fontSize:
18, fontWeight: 'bold'}}>库存日报</Text>
                                              <Text style={{color: '#777777', fontSize:
14}}>可查看每日入/出库明细</Text>
                                        </View>
                                  </CardView>
                            </TouchableOpacity>
                            {this.renderScanView()}
                            <TouchableOpacity onPress = {() => this.goToStoreQuery()}>
                                  <CardView
                                        style = {{marginTop: 10, marginLeft: 10,
marginRight: 10, backgroundColor: '#ffffff', height: width*0.28, flexDirection: 'row',
alignItems: 'center'}}
                                        cardElevation = {4}
                                        cornerRadius = {6}>
                                        <Image
                                              style = {{width: 80, height: 80,
marginLeft: 10}}
                                              source = {require('../../images/
inquire_icon.png')} />
                                        <View>
                                              <Text style={{color: '#444444', fontSize:
18, fontWeight: 'bold'}}>库存查询</Text>
                                              <Text style={{color: '#777777', fontSize:
14, width: width*0.67}}>搜索查询物项信息</Text>
                                        </View>
                                  </CardView>
                            </TouchableOpacity>
                      </View>
                );
        }

        back(){
                this.props.navigator.pop();
        }

        goToScanQRCode(){
                this.props.navigator.push({
                      component: ScanQRCodeView
                })
        }

        goToDailyReport(){
                this.props.navigator.push({
                      component: DailyReportView
                })
        }
```

```
	goToStoreQuery(){
		this.props.navigator.push({
			component: InfoInputView,
			props: {
				title: '物项查询',
				type: 2
			}
		});
	}
}
```

2）Web后端

2.1 双日滚动计划页面：list_rollingPlan.html

```
[@EC.breadcrumbs/]

<div id="page-content" class="clearfix">
	<div class="page-header position-relative">
		<h1>双日滚动计划信息管理</h1>

	</div>
	<!--/page-header-->

	<div class="row-fluid">
		<div class="span12">
			<div class="widget-box">
				<div
					class="widget-header widget-header-blue widget-header-flat
wi1dget-header-large">
					<h4 class="lighter">滚动计划信息列表</h4>
					<div class="widget-toolbar">
						<div class="btn-group">
							<input type="text" id="d4323"
onFocus="WdatePicker({startDate:'%y-%M-01',alwaysUseStartDate:true,maxDate:'#F{$dp.$D(\'d4324\',
{d:-1});}'})" value="" class="Wdate" style="font-weight: bold;"/> 到
							<input type="text" id="d4324"
onFocus="WdatePicker({startDate:'%y-%M-01',alwaysUseStartDate:true,minDate:'#F{$dp.$D(\'d4323\',{d:
1});}'})" value="" class="Wdate" style="font-weight: bold;"/> 
							<button class="btn btn-small btn-info"
								onclick="dataTable.refresh();">
								<i class="icon-refresh"></i>刷新
							</button>
						</div>
					</div>
				</div>
				<div class="widget-body">
					<div class="widget-main">
						<table id="rollingPlanTable"
							class="table table-striped table-bordered table-
hover">
							<thead>
```

```
                                                    <tr>
                                                        <th class="center"><label> <input
type="checkbox"

onclick="checkAll(this);" /> <span class="lbl"></span></label></th>
                                                        <th>序号</th>
                                                        <th>作业条目编号</th>
                                                        <th>机组号</th>
                                                        <th>子项</th>
                                                        <th>图纸号</th>
                                                        <th>系统号</th>
                                                        <th>房间号</th>
                                                        <th>工程量类别</th>
                                                        <th>工程量名称</th>
                                                        <th>工程量编号</th>
                                                        <th>焊口号</th>
                                                        <th>工程量</th>
                                                        <th>规格</th>
                                                        <th>材质</th>
                                                        <th>核级</th>
                                                        <th>单位</th>
                                                        <th>点值</th>
                                                        <th>质量计划号</th>
                                                        <th>计划施工日期</th>
                                                        <th>施工班组</th>
                                                        <th>状态</th>
                                                        <th>备注</th>
                                                        <th>操作</th>
                                                    </tr>
                                                </thead>

                                            </table>
                                        </div>
                                    </div>
                                </div>
                            </div>
                            <!-- PAGE CONTENT ENDS HERE -->
                        </div>
                    </div>

<script type="text/javascript">
                    $(function() {
                            dataTable.initTable({
                                    "tableID" : "#rollingPlanTable",
                                    "lengthChange": true,
                                    "processing" : true,
                                    "serverSide" : true,
                                    "autoWidth": true,
                                    //对每列设置排序规则
                                    //列的长度必须对应，如果表格有7列，必须写7个，否则会报错
                                    "columns" : [ {
                                            "data" : "id",
                                            "sortable" : false
                                    }, {
                                            "data" : "no",
```

```
  "sortable": false
}, {
        "data" : "weldlistno",
  "sortable": false
}, {
        "data" : "unitno",
  "sortable": false
}, {
        "data" : "subItem",
  "sortable": false
}, {
        "data" : "drawno",
  "sortable": false
}, {
        "data" : "systemno",
  "sortable": false
}, {
        "data" : "roomno",
  "sortable": false
}, {
        "data" : "projectType",
  "sortable": false
}, {
        "data" : "projectname",
  "sortable": false
}, {
        "data" : "projectno",
  "sortable": false
}, {
        "data" : "weldno",
  "sortable": false
}, {
        "data" : "projectcost",
  "sortable": false
}, {
        "data" : "specification",
  "sortable": false
}, {
        "data" : "materialtype",
  "sortable": false
}, {
        "data" : "corelevel",
  "sortable": false
}, {
        "data" : "unit",
  "sortable": false
}, {
        "data" : "spot",
  "sortable": false
}, {
        "data" : "qualityplanno",
  "sortable": false
}, {
        "data" : "plandate",
  "sortable": false
}, {
        "data" : "consteam",
  "sortable": false
}, {
        "data" : "status",
```

```
                    "sortable": false
                }, {
                        "data" : "remark",
                    "sortable": false
                }, {
                        "data" : "links",
                        "sortable" : false
                } ],
                "url" : "${path}${current_uri}"
        });

        var searchInput=$(".dataTables_filter :input :first");
        searchInput.unbind();

        searchInput.bind('blur keyword', function() {
                isSearch(function(){
                        var searchInput=$(".dataTables_filter :input :first");
                        $('#rollingPlanTable').DataTable().search(searchInput.val()).draw();
                });
        });

        searchInput.keypress(function(e) {
                if (e.which == 13) {
                        var searchInput=$(".dataTables_filter :input :first");
                        $('#rollingPlanTable').DataTable().search(searchInput.val()).draw();
                }
        });
    })
</script>
```

2.2 通知模块页面：list_notification.html

```
[@EC.breadcrumbs/]

<div id="page-content" class="clearfix">
  <div class="page-header position-relative">
    <h1>通知管理</h1>

  </div><!--/page-header-->

  <div class="row-fluid">
    <div class="span12">
      <div class="widget-box">
          <h4 class="lighter">通知信息列表</h4>
                                <div class="widget-header widget-header-blue widget-header-flat
wi1dget-header-large">
                                    <div style="float: left;">
                                        <ul class="nav nav-tabs" id="myTab" >
                                            <li class="active"><a data-toggle="tab"
href="#dataMain" onclick="dataTable.reload('type=SEND', true);" >
                                                已发送
                                                </a></li>
                                            <li><a data-toggle="tab" href="#dataMain"
onclick="dataTable.reload('type=RECEIVE', true);" >
                                                已接收
                                                </a></li>
                                            <li><a data-toggle="tab" href="#dataMain"
onclick="dataTable.reload('type=DRAFT', true);">
                                                草稿
```

```
                                                                                    </a></li>
                                                                </ul>
                                                            </div>
                                                            <div class="widget-toolbar">
                                <div class="btn-group">
                                [@ec_privilege_check uri="${current_uri}/add"]
                                    <button class="btn btn-small btn-success" onclick="loadFile('${path}$
{AUTH_MENU.privilege.uri!}');" >
                                        <i class="icon-plus"></i>${AUTH_MENU.name}
                                    </button>
                                [/@ec_privilege_check]
                                [@ec_privilege_check uri="${current_uri}/delete"]
                                    <button class="btn btn-small btn-danger"
onclick="batchConfirm(dataTable.tableID,'${path}${AUTH_MENU.privilege.uri!}')">
                                        <i class="icon-remove"></i>${AUTH_MENU.name}
                                    </button>
                                [/@ec_privilege_check]
                                    <button class="btn btn-small btn-info" onclick="dataTable.refresh();">
                                        <i class="icon-refresh"></i>刷新
                                    </button>
                                </div>
                            </div>
                                        </div>
                                        <div class="widget-body">
                                          <div id="dataMain" class="tab-pane in active">
                                                <div class="widget-main">
                        <table id="notificationTable" class="table table-striped table-bordered table-hover">
                            <thead>
                                <tr>
                                    <th class="center"><label>
                                        <input type="checkbox" onclick="checkAll(this);"/>
                                        <span class="lbl"></span></label></th>
                                    <th>主题</th>
                                    <th>部门</th>
                                    <th>开始时间</th>
                                    <th>结束时间</th>
                                    <th>操作</th>
                                </tr>
                            </thead>

                        </table>
                    </div>
                                          </div>
                                        </div>
                </div>
            </div>
            <!-- PAGE CONTENT ENDS HERE -->
        </div>
    </div>
    <script type="text/javascript">
            $(function() {
                    dataTable.initTable({
                            "serverSide": true,
                            "tableID":"#notificationTable",
                            "searching": false,
                            //对每列设置排序规则
                            //列的长度必须对应，如果表格有7列，必须写7个，否则会报错
                            "columns": [
                                    {
```

```
                                        "data":"id",
                                        "sortable": false
                                },
                                {
                                        "data":"subject",
                                        "sortable": false
                                },
                                {
                                        "data":"department",
                                        "sortable": false
                                },
                                {
                                        "data":"startTime",
                                        "sortable": false
                                },
                                {
                                        "data":"endTime",
                                        "sortable": false
                                },
                                {
                                        "data":"links",
                                        "sortable": false
                                }
                ],
                "url": "${path}${current_uri}"
                    });
        })
</script>
```

2.3 质量管理页面：list_quality_problem.html

```
[@EC.breadcrumbs/]

<div id="page-content" class="clearfix">
   <div class="page-header position-relative">
      <h1>质量管理</h1>
   </div>

   <div class="row-fluid">
      <div class="span12">
         <div class="widget-box">
            <div class="widget-header widget-header-blue widget-header-flat wi1dget-header-large">
            <h4 class="lighter">质量问题列表</h4>
               <div class="widget-toolbar">
                  <div class="btn-group">
                        [@ec_privilege_check uri="/hd/quality/report"]
                                                <button class="btn btn-small btn-success"
onclick="report();">
                                                        统计报表
                                                </button>
                                                [/@ec_privilege_check]
                                        </div>
                                        <div class="btn-group">
```

```
                                                    <button class="btn btn-small btn-info dropdown-
toggle"
                                                            data-toggle="dropdown">
                                                            状态<span class="caret"></span>
                                                    </button>
                                                    <ul class="dropdown-menu" style="min-width:
65px;" role="menu">
                                                            <li><a
href="javascript:dataTable.reload('status=', true);">所有</a></li>
                                                            <li><a
href="javascript:dataTable.reload('status=PreQCLeaderAssign', true);">待分派</a></li>
                                                            <li><a
href="javascript:dataTable.reload('status=PreQCAssign', true);">待指派</a></li>
                                                            <li><a
href="javascript:dataTable.reload('status=PreUpRenovete', true);">待整改</a></li>
                                                            <li><a
href="javascript:dataTable.reload('status=PreRenovete', true);">整改中</a></li>
                                                            <li><a
href="javascript:dataTable.reload('status=PreQCverify', true);">待审核</a></li>
                                                            <li><a
href="javascript:dataTable.reload('status=Closed', true);">已关闭</a></li>
                                                            <li><a
href="javascript:dataTable.reload('status=Finished', true);">已完成</a></li>
                                                    </ul>
                                            </div>
                                            <div class="btn-group">
                    <button class="btn btn-small btn-info" onclick="dataTable.refresh();">
                      <i class="icon-refresh"></i>刷新
                    </button>
                  </div>
                </div>
              </div>
              <div class="widget-body">
                <div class="widget-main">
                  <table id="qualityProblemTable" class="table table-striped table-bordered table-hover">
                    <thead>
                      <tr>
                        <th class="center"><label>
                          <input type="checkbox" onclick="checkAll(this);"/>
                          <span class="lbl"></span></label>
                        </th>
                        <th>序号</th>
                        <th>机组</th>
                        <th>子项</th>
                        <th>楼层</th>
                        <th>房间号</th>
                        <th>系统</th>
                        <th>责任部门</th>
                        <th>责任班组</th>
                        <th>问题描述</th>
                        <th>创建时间</th>
                        <th>是否质量问题单</th>
                        <th>当前状态</th>
                        <th>操作</th>
                      </tr>
                    </thead>
```

```
                </table>
              </div>
            </div>
          </div>
        </div>
        <!-- PAGE CONTENT ENDS HERE -->
    </div>
</div>

<script type="text/javascript">
        $(function() {
                dataTable.initTable({
                        "serverSide": true,
                        "tableID":"#qualityProblemTable",
                        'bSort':false,
                        "lengthChange": true,
                        "processing" : true,
                        "serverSide" : true,
                        "autoWidth": true,
                        "bFilter" : false,
                        //对每列设置排序规则
                        //列的长度必须对应，如果表格有7列，必须写7个，否则会报错
                        "columns": [
                                {
                                        "data":"id",
                                        "sortable": false
                                },
                                {
                                                "data":"no"
                                }
                                ,
                                {
                                        "data":"unit"
                                },
                                {
                                        "data":"subitem"
                                },
                                {
                                        "data":"floor"
                                },
                                {
                                        "data":"roomnum"
                                },
                                {
                                        "data":"system"
                                },
                                {
                                        "data":"responsibleDept"
                                },
                                {
                                        "data":"responsibleTeam"
                                },
                                {
                                   "data":"problemDescription"
                                 },
                                 {
                                                "data":"createDate"
                                        },
                                {
```

```
                                                    "data":"qualityFlag"
                                                },
                                        {
                                            "data":"status"
                                        },
                                {
                                        "data":"links",
                                        "sortable": false
                                }
                ],
                "url": "${path}${current_uri}"
                    });
        })

                //导出指标库Excel
        function report(){
                var url = "${path}/hd/quality/report";
                $.ajax({
                    type: "GET",
                    url: url,
                    success: function(response, status, request) {
                        var disp = request.getResponseHeader('Content-Disposition');
                        if (disp && disp.search('attachment') != -1) {  //判断是否为文件
                            var form = $('<form method="POST" action="' + url + '">');
                            form.append($('<input type="hidden" name="" value="">'));
                            $('body').append(form);
                            form.submit(); //自动提交
                        }
                    }
                });
        }

        //返回记住的上一次的页面;
        var timesRunPage = 0;
        var intervalPageBack = setInterval(dataTable.initPageBack,500);
</script>
```

2.4 安全文明施工接口：HseProblemAPIController.java

```
package com.easycms.hd.api.baseservice.hseproblem;

import java.util.Calendar;
import java.util.List;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
```

```
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.bind.annotation.ResponseBody;
import org.springframework.web.multipart.commons.CommonsMultipartFile;

import com.easycms.common.exception.ExceptionCode;
import com.easycms.common.util.CommonUtility;
import com.easycms.core.util.Page;
import com.easycms.hd.api.enums.hseproblem.HseProbelmTypeEnum;
import com.easycms.hd.api.enums.hseproblem.HseProblemSolveStatusEnum;
import com.easycms.hd.api.enums.hseproblem.HseProblemStatusEnum;
import com.easycms.hd.api.request.base.BaseRequestForm;
import com.easycms.hd.api.request.hseproblem.HseProblemAssignRequestForm;
import com.easycms.hd.api.request.hseproblem.HseProblemListRequestFormByStatus;
import com.easycms.hd.api.request.hseproblem.HseProblemRequestForm;
import com.easycms.hd.api.response.JsonResult;
import com.easycms.hd.api.response.hseproblem.HseProblemCreatUiDataResult;
import com.easycms.hd.api.response.hseproblem.HseProblemCreatUiDataResult2;
import com.easycms.hd.api.response.hseproblem.HseProblemPageDataResult;
import com.easycms.hd.api.response.hseproblem.HseProblemResult;
import com.easycms.hd.api.service.HseProblemApiService;
import com.easycms.hd.hseproblem.domain.HseProblem;
import com.easycms.hd.hseproblem.domain.HseProblemSolve;
import com.easycms.hd.hseproblem.domain.HseProblemSolveStep;
import com.easycms.hd.hseproblem.domain.HseProblemStep;
import com.easycms.hd.hseproblem.service.HseProblemFileService;
import com.easycms.hd.hseproblem.service.HseProblemService;
import com.easycms.hd.hseproblem.service.HseProblemSolveService;
import com.easycms.management.user.domain.User;
import com.easycms.management.user.service.UserService;
import com.wordnik.swagger.annotations.Api;
import com.wordnik.swagger.annotations.ApiOperation;
import com.wordnik.swagger.annotations.ApiParam;

import lombok.extern.slf4j.Slf4j;

@Controller
@RequestMapping("/hdxt/api/hse")
@Api(value="HseProblemAPIController",description="文明施工问题模块相关的api")
@Slf4j
@javax.transaction.Transactional
public class HseProblemAPIController {
	@Autowired
	private HseProblemService hseProblemService;
	@Autowired
	private HseProblemApiService hseProblemApiService;
	@Autowired
	private HseProblemFileService hseProblemFileService;
	@Autowired
	private UserService userService;
	@Autowired
	private HseProblemSolveService hseProblemSolveService;

	@ResponseBody
	@RequestMapping(value = "createUI", method = RequestMethod.GET)
	@ApiOperation(value = "创建安全文明问题界面", notes = "用于获取机组、厂房、责任部门、责任班
组")
```

```
	public JsonResult<HseProblemCreatUiDataResult> createUI (HttpServletRequest request,
HttpServletResponse response,
				@ModelAttribute("requestForm") BaseRequestForm baseRequestForm,
				@ApiParam(required = false, name = "safe", value = "是否安全模块")
@RequestParam(value="safe",required=false) String safe,
				@ApiParam(required = false, name = "responsibleDeptId", value = "责任部门ID")
@RequestParam(value="responsibleDeptId",required=false) Integer responsibleDeptId){
		JsonResult<HseProblemCreatUiDataResult> result = new
JsonResult<HseProblemCreatUiDataResult>();
		HseProblemCreatUiDataResult hseProblemCreatUiResult = null;

		try{

			hseProblemCreatUiResult =
hseProblemApiService.getHseProblemCreatUiResult(responsibleDeptId,safe);
		}catch(Exception e){
			e.printStackTrace();
			result.setMessage("操作异常:"+e.getMessage());
			result.setResponseResult(hseProblemCreatUiResult);
			return result;
		}

		result.setResponseResult(hseProblemCreatUiResult);
		return result;

	}

	@ResponseBody
	@RequestMapping(value = "v2/createUI", method = RequestMethod.GET)
	@ApiOperation(value = "创建安全文明问题界面", notes = "用于获取机组、厂房、责任部门、责任班
组")
	public JsonResult<HseProblemCreatUiDataResult2> createUI_v2 (HttpServletRequest request,
HttpServletResponse response,
				@ModelAttribute("requestForm") BaseRequestForm baseRequestForm,
				@ApiParam(required = false, name = "safe", value = "是否安全模块")
@RequestParam(value="safe",required=false) String safe,
				@ApiParam(required = false, name = "responsibleDeptId", value = "责任部门ID")
@RequestParam(value="responsibleDeptId",required=false) Integer responsibleDeptId){
		JsonResult<HseProblemCreatUiDataResult2> result = new
JsonResult<HseProblemCreatUiDataResult2>();
		HseProblemCreatUiDataResult2 hseProblemCreatUiResult = null;

		try{

			hseProblemCreatUiResult =
hseProblemApiService.getHseProblemCreatUiResult_v2(responsibleDeptId,safe);
		}catch(Exception e){
			e.printStackTrace();
			result.setMessage("操作异常:"+e.getMessage());
			result.setResponseResult(hseProblemCreatUiResult);
			return result;
		}

		result.setResponseResult(hseProblemCreatUiResult);
		return result;

	}

	@ResponseBody
```

```
	@RequestMapping(value = "create", method = RequestMethod.POST)
	@ApiOperation(value = "创建安全文明问题", notes = "用于创建安全文明问题")
	public JsonResult<Boolean> create(HttpServletRequest request, HttpServletResponse response,
			@ModelAttribute("requestForm") HseProblemRequestForm
hseProblemRequestForm,@RequestParam(value="file",required=false) CommonsMultipartFile[] files){
		JsonResult<Boolean> result = new JsonResult<Boolean>();
		Boolean status = false ;
		Integer loginId = hseProblemRequestForm.getLoginId();
		if (null == loginId || loginId.intValue()==0) {
			result.setCode("-1001");
			result.setMessage("Error/s occurred, loginId is not available");
			return result;
		}
		if(!CommonUtility.isNonEmpty(hseProblemRequestForm.getUnit())){
			result.setCode("-1001");
			result.setMessage("Error/s occurred, Unit is not available");
			return result;
		}
		if(!CommonUtility.isNonEmpty(hseProblemRequestForm.getWrokshop())){
			result.setCode("-1001");
			result.setMessage("Error/s occurred, Wrokshop is not available");
			return result;
		}
//		if(!CommonUtility.isNonEmpty(hseProblemRequestForm.getEleration())){//标高 20180422  a
 p p 选填
//			result.setCode("-1001");
//			result.setMessage("Error/s occurred, Eleration is not available");
//			return result;
//		}
//		if(!CommonUtility.isNonEmpty(hseProblemRequestForm.getRoomno())){//房间号 20180422
 a p p 选填
//			result.setCode("-1001");
//			result.setMessage("Error/s occurred, Roomno is not available");
//			return result;
//		}
		try{

			status = hseProblemApiService.createHseProblem(hseProblemRequestForm, files);

		}catch(Exception e){

			e.printStackTrace();
			result.setMessage("操作异常:"+e.getMessage());
			result.setResponseResult(false);
			return result;
		}

		if(!status){
			result.setCode("-1001");
			result.setMessage("create problem error!");
			return result;
		}
		result.setResponseResult(status);
		return result;
	}

	@ResponseBody
	@RequestMapping(value = "problemList", method = RequestMethod.GET)
```

```
	@ApiOperation(value = "查询安全文明问题分页列表", notes = "用于查询各状态安全文明问题列表,
问题进度状态：need_handle:待处理;renovating:整改中;need_check:待审查;finish:已完成;none:不需要处理")
	public JsonResult<HseProblemPageDataResult> problemList(HttpServletRequest request,
HttpServletResponse response,
			@ModelAttribute("requestForm") HseProblemListRequestFormByStatus
hseProblemRequestForm){
		JsonResult<HseProblemPageDataResult> result = new
JsonResult<HseProblemPageDataResult>();
		HseProblemPageDataResult hseProblemPageDataResult = new
HseProblemPageDataResult();
		Page<HseProblem> page = new Page<HseProblem>();
		String problemStatus = null;
		String solveStatus = null;
		String problemType = null;
		//问题类型判断
		if(hseProblemRequestForm.getProbelmType()!=null){
			problemType = hseProblemRequestForm.getProbelmType().toString();
		}
		//问题状态判断
		if(hseProblemRequestForm.getProblemStatus()!=null){
			problemStatus = hseProblemRequestForm.getProblemStatus().toString();
		}
		//判断处理情况是否为空
		if(hseProblemRequestForm.getProblemSolveStatus()!=null){
			solveStatus = hseProblemRequestForm.getProblemSolveStatus().toString();
		}

		if (null != hseProblemRequestForm.getPagenum() && null !=
hseProblemRequestForm.getPagesize()) {
			page.setPageNum(hseProblemRequestForm.getPagenum());
			page.setPagesize(hseProblemRequestForm.getPagesize());
		}
		try{
			if(hseProblemRequestForm.getLoginId()!=null){
				User user =
userService.findUserById(hseProblemRequestForm.getLoginId());
				if(user == null){
					result.setCode("-1001");
					result.setMessage("用户不存在！");
					return result;
				}

				if(HseProbelmTypeEnum.mine.toString().equals(problemType)){
					hseProblemPageDataResult =
hseProblemApiService.getHseProblemResultByUser(page, problemStatus, user);
				}else{
					hseProblemPageDataResult =
hseProblemApiService.getHseProblemPageDataResult(page, problemStatus,solveStatus,user);
				}

			}
		}catch(Exception e){
			e.printStackTrace();
			result.setCode("-1001");
			result.setMessage("操作异常："+e.getMessage());
		}
		result.setResponseResult(hseProblemPageDataResult);
		return result;
	}
```

```
	@ResponseBody
	@RequestMapping(value = "problem/{id}", method = RequestMethod.GET)
	@ApiOperation(value = "查询安全文明问题", notes = "用于查询单个安全文明问题")
	public JsonResult<HseProblemResult> problemListOne(HttpServletRequest request,
HttpServletResponse response,
			@ModelAttribute("requestForm") BaseRequestForm
hpRequestForm,@PathVariable("id") Integer id) throws Exception {
		JsonResult<HseProblemResult> result = new JsonResult<HseProblemResult>();
		HseProblemResult hpResult = null;
		HseProblem hp = hseProblemService.findById(id);
		if(hp != null){
			hpResult = hseProblemApiService.transferForList(hp, null);
			result.setResponseResult(hpResult);
		}else{
			result.setCode("-1001");
			result.setMessage("查询的安全问题不存在！");
		}

		return result;
	}


	@ResponseBody
	@RequestMapping(value = "assign", method = RequestMethod.POST)
	@ApiOperation(value = "HSE执行分派", notes = "用于HSE分派整改任务,responsibleTeamId责任班组
id，problemId问题id")
	public JsonResult<Boolean> assign (HttpServletRequest request, HttpServletResponse response,
			@ModelAttribute("requestForm") HseProblemAssignRequestForm
hseProblemAssignRequestForm){
		JsonResult<Boolean> result = new JsonResult<Boolean>();
		boolean status = false;
		Integer loginId = null;
		if(hseProblemAssignRequestForm.getLoginId() != null){
			loginId = hseProblemAssignRequestForm.getLoginId();
		}
		User loginUser = userService.findUserById(loginId);
		if(loginUser == null){
			result.setCode("-1000");
			result.setMessage("当前用户不存在！");
			return result;
		}
		try{
			//登录用户是否是HSE部门成员
			if(!userService.isDepartmentOf(loginUser, "HSEDepartment")){
				result.setCode("-1000");
				result.setMessage("当前用户不是HSE部门成员，无法执行该操作！");
				return result;
			}
		}catch(Exception e){
			log.error("something error");
			e.printStackTrace();
			result.setMessage("操作异常:"+e.getMessage());
			result.setResponseResult(false);
			return result;
		}
		if(hseProblemAssignRequestForm.getProblemId() == null){
			result.setCode("-1000");
			result.setMessage("必要参数problemID为空！");
```

```
                return result;
            }

            status = hseProblemApiService.hseAssignTasks(hseProblemAssignRequestForm);
            if(!status){
                result.setCode("-1001");
                result.setMessage("执行分派出错！");
            }
            result.setResponseResult(status);
            return result;

        }

        @ResponseBody
        @RequestMapping(value = "unAssign", method = RequestMethod.POST)
        @ApiOperation(value = "HSE执行不需要处理问题", notes = "用于HSE将问题设为不需要处理状态，
problemId问题id")
        public JsonResult<Boolean> unAssign(HttpServletRequest request, HttpServletResponse
response,
                @ApiParam(required = true, name = "problemId", value = "文明施工问题ID")
@RequestParam(value="problemId",required=true) Integer problemId,@ModelAttribute("requestForm")
BaseRequestForm baseRequestForm){
            JsonResult<Boolean> result = new JsonResult<Boolean>();
            boolean status = false;
            try{
                Integer loginId = null;
                if(baseRequestForm.getLoginId() != null){
                    loginId = baseRequestForm.getLoginId();
                }
                User loginUser = userService.findUserById(loginId);
                if(loginUser == null){
                    result.setCode("-1000");
                    result.setMessage("当前用户不存在！");
                    return result;
                }
                //登录用户是否是HSE部门成员
                if(!userService.isDepartmentOf(loginUser, "HSEDepartment")){
                    result.setCode("-1000");
                    result.setMessage("当前用户不是HSE部门成员，无法执行该操作！");
                    return result;
                }

                //将问题状态改为不需要处理状态
                HseProblem hseProblem = hseProblemService.findById(problemId);
                boolean isNeedHandle = false;
                if(hseProblem != null){
                    isNeedHandle =
hseProblem.getProblemStatus().equals(HseProblemStatusEnum.Need_Handle.toString());

                }else{
                    result.setCode("-1001");
                    result.setMessage("当前问题不存在！");
                    return result;
                }
                if(!isNeedHandle){
                    result.setCode("-1001");
                    result.setMessage("当前状态不允许关闭问题！");
                    return result;
```

```
                        }
                        hseProblem.setProblemStatus(HseProblemStatusEnum.None.toString());
                        hseProblem.setProblemStep(HseProblemStep.isFinished.toString());//问题关闭
                        hseProblem.setFinishDate(Calendar.getInstance().getTime());//问题关闭时间
                        hseProblemService.update(hseProblem);

                        //将HSE处理情况更新为已处理
                        List<HseProblemSolve> hseProblemSolves
=hseProblemSolveService.findByProblenIdAndSolveStep(problemId,
HseProblemSolveStep.hseConfirm.toString());
                        if(hseProblemSolves != null && !hseProblemSolves.isEmpty()){
                                HseProblemSolve hseProblemSolve = hseProblemSolves.get(0);
                                hseProblemSolve.setSolveUser(baseRequestForm.getLoginId());
                                hseProblemSolve.setSolveDate(Calendar.getInstance().getTime());

hseProblemSolve.setSolveStatus(HseProblemSolveStatusEnum.Done.toString());
                                hseProblemSolveService.update(hseProblemSolve);
                        }
                        status=true;
                }catch(Exception e){
                        log.error("something error");
                        e.getMessage();
                }
                if(!status){
                        result.setCode("-1001");
                        result.setMessage("操作失败！");
                }
                result.setResponseResult(status);
                return result;
        }

        @ResponseBody
        @RequestMapping(value = "submitRenovateResult", method = RequestMethod.POST)
        @ApiOperation(value = "提交整改结果", notes = "用于整改责任部门提交整改结果，problemId问题
id")
        public JsonResult<Boolean> submitRenovateResult(HttpServletRequest request,
HttpServletResponse response,
                        @ApiParam(required = true, name = "problemId", value = "文明施工问题ID")
@RequestParam(value="problemId",required=true) Integer problemId,
                        @ApiParam(required = true, name = "description", value = "整改描述")
@RequestParam(value="description",required=true) String description,
                        @RequestParam(value="file") CommonsMultipartFile[] files){
                JsonResult<Boolean> result = new JsonResult<Boolean>();
                boolean status = false;

                status = hseProblemApiService.processRenovateResult(problemId, description, files);
                if(!status){
                        result.setCode("-1001");
                        result.setMessage("提交结果失败！");
                }
                result.setResponseResult(status);
                return result;
        }

        @ResponseBody
        @RequestMapping(value = "checkRenovateResult", method = RequestMethod.POST)
        @ApiOperation(value = "HSE审查整改结果", notes = "用于HSE审查整改结果，problemId问题id,
checkResult：通过或者重新整改")
```

```
	public JsonResult<Boolean> checkRenovateResult(HttpServletRequest request,
HttpServletResponse response,
				@ApiParam(required = true, name = "problemId", value = "文明施工问题ID")
@RequestParam(value="problemId",required=true) Integer problemId,
				@ApiParam(required = true, name = "checkResult", value = "审查结果，1-通过，2-
不通过") @RequestParam(value="checkResult",required=true) Integer checkResult,
				@ModelAttribute("requestForm") BaseRequestForm baseRequestForm){
			JsonResult<Boolean> result = new JsonResult<Boolean>();
			Integer loginId = null;
			if(baseRequestForm.getLoginId() != null){
				loginId = baseRequestForm.getLoginId();
			}
			User loginUser = userService.findUserById(loginId);
			if(loginUser == null){
				result.setCode("-1000");
				result.setMessage("当前用户不存在！");
				return result;
			}
			//登录用户是否是HSE部门成员
			if(!userService.isDepartmentOf(loginUser, "HSEDepartment")){
				result.setCode("-1000");
				result.setMessage("当前用户不是HSE部门成员，无法执行该操作！");
				return result;
			}

			boolean status = false;

	try{
	    status = hseProblemApiService.checkRenovateResult(baseRequestForm,problemId,checkResult);

	}catch(Exception e){
	   e.printStackTrace();
				result.setCode(ExceptionCode.E4);
	   result.setMessage("操作异常:"+e.getMessage());
	   result.setResponseResult(status);
	   return result;
	}

			if(!status){
				result.setCode("-1001");
				result.setMessage("操作失败！");
			}

			result.setResponseResult(status);
			return result;
		}


}
```

3）代码接口编制规则

滚动计划岗位配置参数：

witness_team_qc1	QC1见证组长
witness_member_qc1	QC1见证组员
witness_team_qc2	QC2见证组长
witness_member_qc2	QC2见证组员

supervisor	滚动计划作业问题技术领导(技术室主任)
solver	滚动计划作业问题技术人员
coordinator	滚动计划的协调工程师
solver3	滚动计划作业问题技术部经理
solver4	滚动计划作业问题副总经理（技术）
solver5	项目部总经理

质量管理：

QCManager	质量管理QC专业室主任
qc_member	质量管理人员

物项管理：

keeper	仓管员权限

安全施工问题状态：

problemStatus	字段
‘Need_Handle'	待处理
‘Renovating'	整改中
‘Need_Check'	待审核
‘None'	不需处理
‘Finish’	已完成

用户身份判断：

roleType	字段
‘monitor'	班长
‘team’	组长
‘captain’	队长
'member'	成员

八、 系统配置

8.1、配置原则

能良好地为移动App管理平台系统服务，便于开发。

8.2、硬件配置

后台服务器：

1）16G内存

2）2T 硬盘

8.3、软件配置

1）WebStorm 开发工具

2）SourceTree 版本控制系统工具

3）Eclipse 后台开发工具

4）Navicat Premium 数据库管理工具

5）MySQL 关系型数据库管理系统

九、 关键技术

9.1 关键技术的提出

1）React-Native

2）Bugly

9.2 关键技术的实现方案

1）React-Native

1.React Native (简称RN)是Facebook于2015年4月开源的跨平台移动应用开发框架，是Facebook早先开源的JS框架 React 在原生移动应用平台的衍生产物，目前支持iOS和安卓两大平台。RN使用Javascript语言，类似于HTML的JSX，以及CSS来开发移动应用，因此熟悉Web前端开发的技术人员只需很少的学习就可以进入移动应用开发领域。

React Native使你能够在Javascript和React的基础上获得完全一致的开发体验，构建世界一流的原生APP。

2）Bugly

腾讯 Bugly，是腾讯公司为移动开发者开放的服务之一，面向移动开发者提供专业的 Crash 监控、崩溃分析等质量跟踪服务。Bugly 能帮助移动互联网开发者更及时地发现掌控异常，更全面的了解定位异常，更高效的修复解决异常。

针对移动应用，腾讯 Bugly 提供了专业的 Crash、Android ANR (application not response)、iOS 卡顿监控和解决方案。移动开发者 (Android / iOS) 可以通过监控，快速发现用户在使用过程中出现的 Crash (崩溃)、Android ANR 和 iOS 卡顿，并根据上报的信息快速定位和解决问题。

1.异常反馈上报

腾讯Bugly，为移动开发者提供专业的异常上报和运营统计，帮助开发者快速发现并解决异常，同时掌握产品运营动态，及时跟进用户反馈。

2.应用升级

提供完整的应用升级解决方案，通过热更新 & 弹窗提醒方式及时修复线上 Bug，提升新版本升级率。

十、 实现后的功能流程

详情图示功能流程，请参考应用操作图文手册；

10.1 账户登录功能流程

步骤1：登录页输入用户名和密码 —> 选择所在项目部 —> 选择网络环境—> 登录

10.2 双日计划功能流程

1）分派任务

步骤 1:班长登录 在任务----待分派任务----选择日期， 组长，---确认分派，将任务分派给组长。

2）见证

说明:组长对任务发起见证，由班长分派给 qc1 见证组长 /qc2 见证组长，qc1 组长分派给 qc1 组员，qc2 组长分派 给 qc2 组员;再由 QC1 组员见证，qc2 组员见证自己的， 如果 qc2 没有分派给 czecqc/czecqa/paec 就可以

代替见 证 czecqc/czecqa/paec 当所有见证都合格后，由组长确认见证，任务变为已完成如果有一个见证不合格，由组长重新发起见证，后面流程不变。

批量见证:组长登录，任务---未完成任务---选择相应任 务，点击批量见证----出现不同任务相同的工序，选择工 序---确认

一般见证:

步骤 2:组长登录 在任务----未完成任务----勾选任务 点---发起见证---选择工序--确认进行见证分派。选择时间， 输 入 地 点 ---提交

如果实际见证人还没有针对某工序见证时，此时还没有任何见证记录，作业组长可以针对此工序已经发起的见证进行取消。

步骤 3:班长登录 在见证---qc1 组长待 交，qc2 组长待 交选择对应见证组长 以及见证工序---- 提交

步骤 4:qc1 组长登录 在见证分派---选 择 qc1 组员和工序--- 提交

步骤 5:qc2 组长登录 在见证分派---选 择 qc2 组员和工序--- 提交

步骤 6:qc1 组员登录，见证分派--待见 证--选择见证进入见证详情--选择见证时 间，输入见证地点，选择见证结果-- 提交

步骤 7:qc2 组员登录，见证分派-- czecqc/czecqa/paec 待分派的见证--分别 选择见证人员-- 提交

步骤 8:qc2 组员登录，我的见证--待 交 见证--点击见证进入见证详情--点击回填- -选择时间，输入地点，选择结果-- 提交。 如果步骤 7 都分给 czecqc/czecqa/paec 步 骤 8，只需见证 qc2 其他三个没有回填显示

步骤 9:czecqc/czecqa/paec 登录，待见 证--点击见证进入见证详情--选择时间， 输入地点，选择结果-- 提交。

步骤 10: 组长登录，任务--已完成的任 务--点击任务进入任务详情--输入焊工号- - 交。

见证合格:必须所有工序都合格，
见证不合格:只有有一个工序不合格，就 需要组长重新发起见证。 见证不合格步骤: 跳过步骤 1，步骤 3，步骤 4，步骤 5，后面从步骤 6 开始走 完

重新走步骤 2，步骤 2:中选择工序页面-- 选框前有文字:再次见证。

3）问题

步骤 1:组长登录 任务--未完成任务--点 击任务进入任务详情--问题创建--选择问 题类型，输入问题描述-- 提交。 任务由施 工中/未施工变为--停滞中。

步骤 2:班长登录 问题--问题列表--对应 组长点击进入--待指派的问题--选择一个 问题进入问题详情--选择协调人员-- 提交。

步骤 3:协调人员登录 问题列表--相应组 长点击进入--待指派的问题--点击问题详 情--选择问题解决人-- 提交。

步骤 4:技术人员登录 待处理的问题-- 点击问题进入问题详情--输入问题反馈-- 提交。

步骤 5:组长登录 问题--待确认的问题-- 选择一个问题进入问题详情--接受反馈。

步骤 6:组长登录 问题--待确认的问题-- 选择一个问题进入问题详情--不接受反 馈。

步骤 7:技术领导登录 问题列表--对应技 术人员--管道问题--未解决--点击一个问 题进入问题详情--直接处理

步骤 8:重新走步骤 5

步骤 9:技术领导登录 问题列表--对应技 术人员--管道问题--未解决--点击一个问 题进入问题详情--改派--选择其他技术人 员确定--提 交

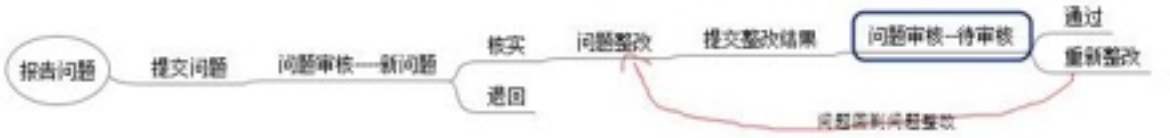
步骤 10:技术人员登录 待处理的问题-- 点击问题进入问题详情--输入问题反馈-- 交。

步骤 11:重新走步骤 5

组长接受问题后，该任务由停滞中变为施工中。

10.3 安全文明施工功能流程

流程图：



1）报告问题：

步骤 1:点击进入--问题填写页面--输入和选择必填项--提交

2）问题审核：

步骤 2:点击进入--待处理--新问题--进入问题详情--选择班组，整改期限--退回，该问题在不 需处理可查看

步骤 3:点击进入--待处理--新问题--进入问题详情--选择班组，整改期限--核实

3）问题整改：

步骤 4:点击进入—待处理—点击问题详情--输入整改描述、整改照片--提交整改结果。

4）问题审核：

步骤 5:进入问题审核--待处理--待审核--进入问题详情--通过。

步骤 6:进入问题审核--待处理--待审核--进入问题详情--重新整改。后面的步骤从步骤 4 开 始

5）问题查阅：

步骤 7:点击进入--我的问题/所有问题 ----查看问题详情

10.4 质量管理功能流程：

登录分为 **5** 个角色，分角色登录。

领导登录:问题汇总和报告问题，报告问题包括了上报问题和自己相关问题流程的追踪;

责任单位登录:问题待处理和报告问题，报告问题包括了上报问题和自己相关问题流程的追踪;

QC 专业室主任登录:问题待处理和报告问题，报告问题包括了上报问题和自己相关问题流程的追踪 ;

QC1 登录:问题待处理和报告问题，报告问题包括了上报问题和自己相关问题流程的追踪 ;

普通员工登录:报告问题包括了上报问题和自己相关问题流程的追踪

10.5 物项管理流程

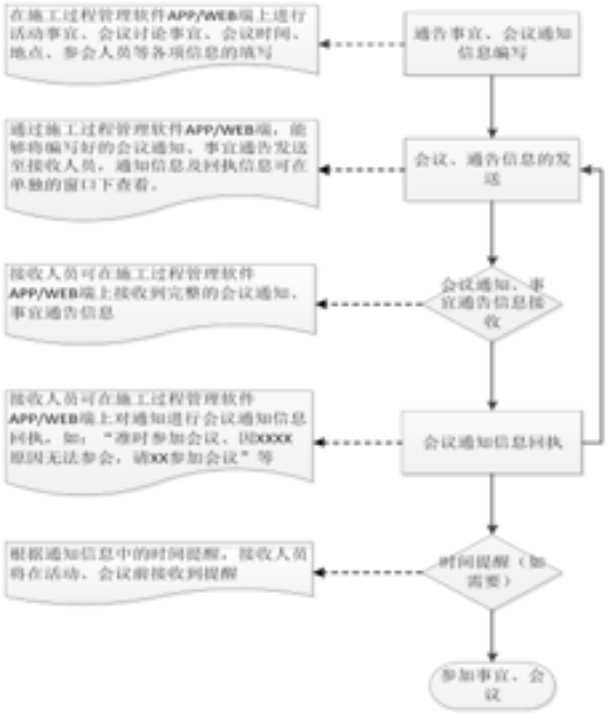
1）库存日报: 在库存日报查看今日入库/出库的详情数据

2）扫一扫: 在扫一扫，扫描物项入库单/出库单/退库单二维码，查询物项信息，入库、出库、退库处理信息

3）库存查询: 输入查询的关键字消息 ，搜索查询物项信息

10.6 会议通知流程

会议通知业务流程图:



二级模块——文件失效通知

ENPower里文件回收管理\未销毁\文件失效通知单 中的数据定时流转到此模块， 依靠组织机构的设置，实现部门管理人员能够查看本部门已借阅失效的图纸，个人能 够查看个人已借阅失效的图纸，且有提醒功能。

1） 三级模块——会议通知编制

三级模块——会议通知编制模块主要用于实现用户对会议通知的编制与发送

2） 三级模块——我待发的会议通知

点击软件内相应菜单下通知模块树状结构中的三级模块——我待发的会议通知， 可弹出已保存的会议通知编制草稿，具体操作界面样式请参考附件 1:三级模块—— 会议通知编制操作界面样式。

3）三级模块——我发送的会议通知

三级模块——我发送的会议通知模块主要用于实现用户对已发送会议通知的查看。

4）三级模块——我接收的会议通知

三级模块——我接收的会议通知模块主要用于实现用户对已接收的会议通知进行查看，同时可进行会议通知的的回执，并在会前接收到提醒。

5） 三级模块——通告编制 三级模块——通告编制模块主要用于实现用户对通告的编制与发送

6） 三级模块——我待发的通告

点击软件内相应菜单下通知模块树状结构中的三级模块——我待发的通告，可弹出已保存的会议通知编制草稿。

7）三级模块——我发送的通告 三级模块——我发送的通告模块主要用于实现用户对已发送通告的查看

8）三级模块——我接收的通告 三级模块——我接收的通告模块主要用于实现用户对已接收的通告进行查看，并在接收到提醒。