应用开发框架设计方案

【2019年3月】

内部资料保密勿泄漏

## 版本记录

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **日期** | **版本号** | **修订说明** | **修订人** | **审核人** | **批准人** |
| 2019.03.12 | V1.0 |  | 唐石元 |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**目录**

[一、 前言 1](#_Toc338346695)

[二、 项目背景及需求分析 2](#_Toc338346696)

[2.1 项目背景 2](#_Toc338346697)

[2.2 项目需求 2](#_Toc338346698)

[2.3 其它需求 2](#_Toc338346699)

[三、 项目设计概要 8](#_Toc338346700)

# 前言概述

本方案的宗旨在于指导项目设计人员，开发人员更好的进行多媒体项目的设计开发及项目的后续维护和改进。本书只讲述主要功能及某些技术关键实现点，详细实现细节本书不一一讲述，如有必要请查阅源码或询问开发人员。

# 项目背景及需求分析

## 项目背景

为规范公司软件开发流程，提升软件开发效率。

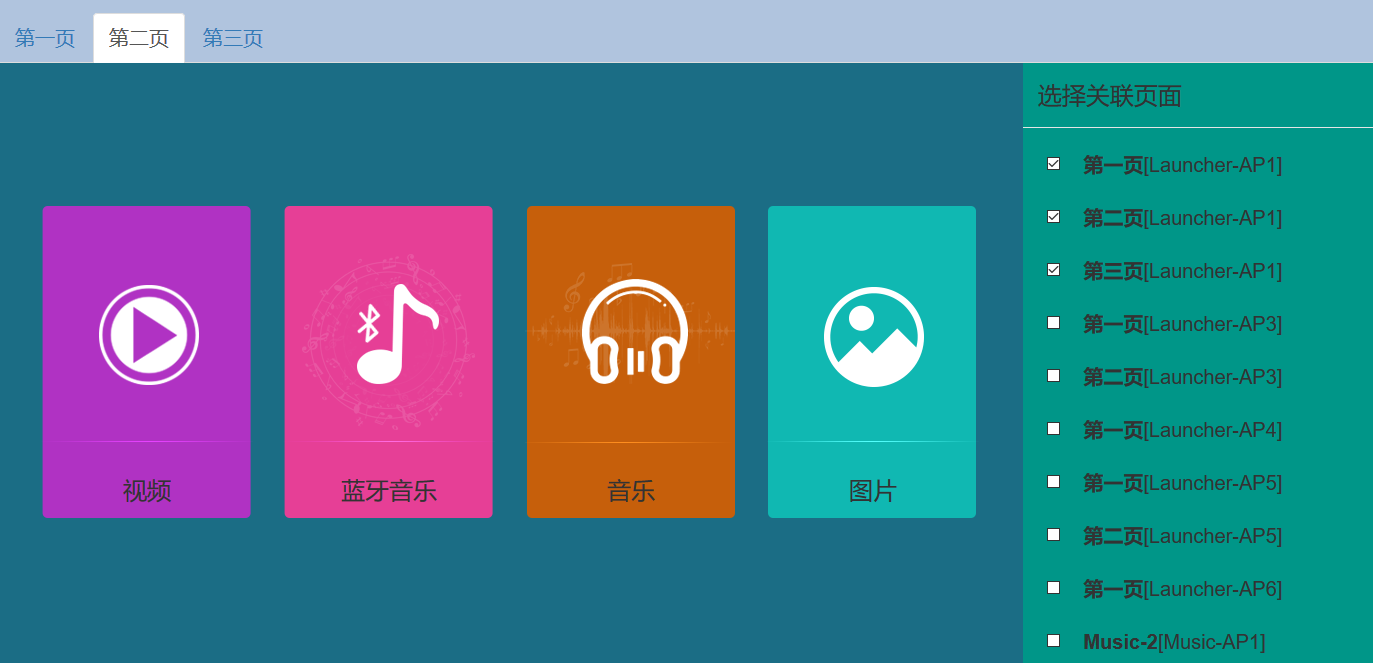
同时也为了解决车载Android平台主应用、蓝牙应用、多媒体应用、收音机等等各种应用需要经常更换UI界面的需求。

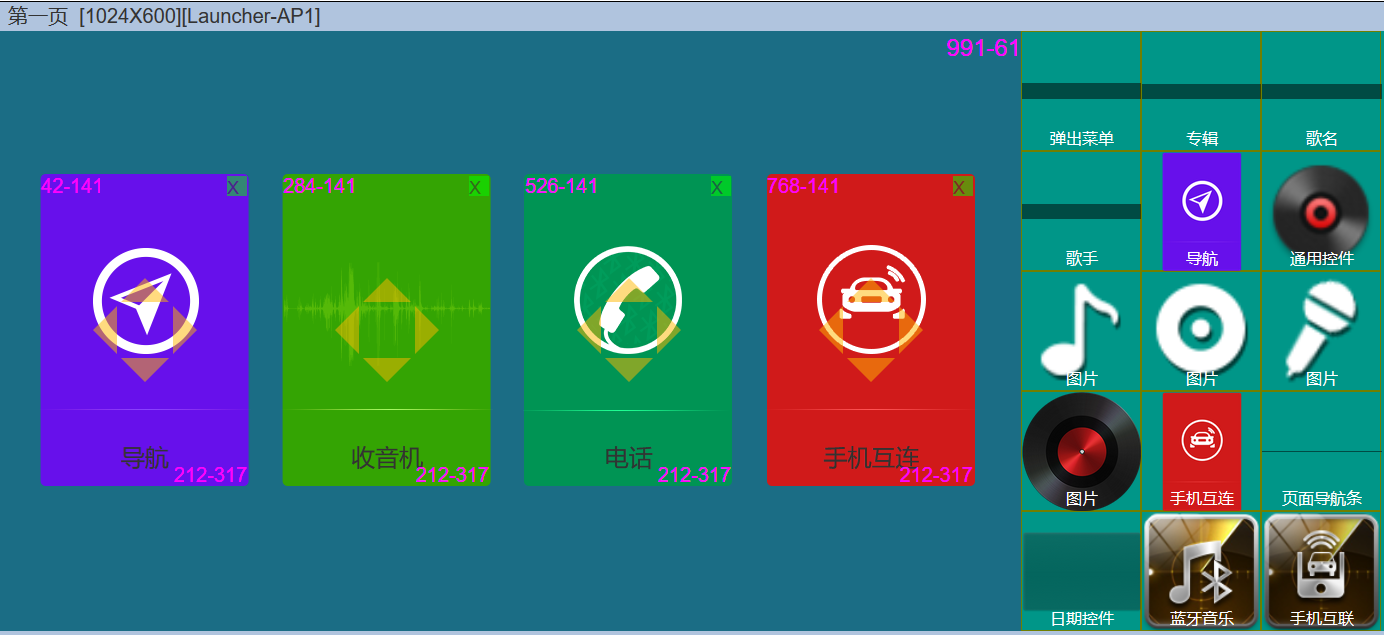
公司软件开发以此项目为框架展开，在将UI和功能分离后，可以将主要精力投入到逻辑功能模块的开发上。在遵循统一框架的基础上，能够提升公司软件开发质量和开发效率，避免重复开发和不规范开发。

## 项目实现需求

### PC端通过网页拖动图片方式配置UI界面。

如下列Web页面显Launcher配置截图，可随意配置图片文字等内容，拖动位置单位精确到像素以满足任意设计需要。





### Android端根据JSON数据解析生成UI界面。

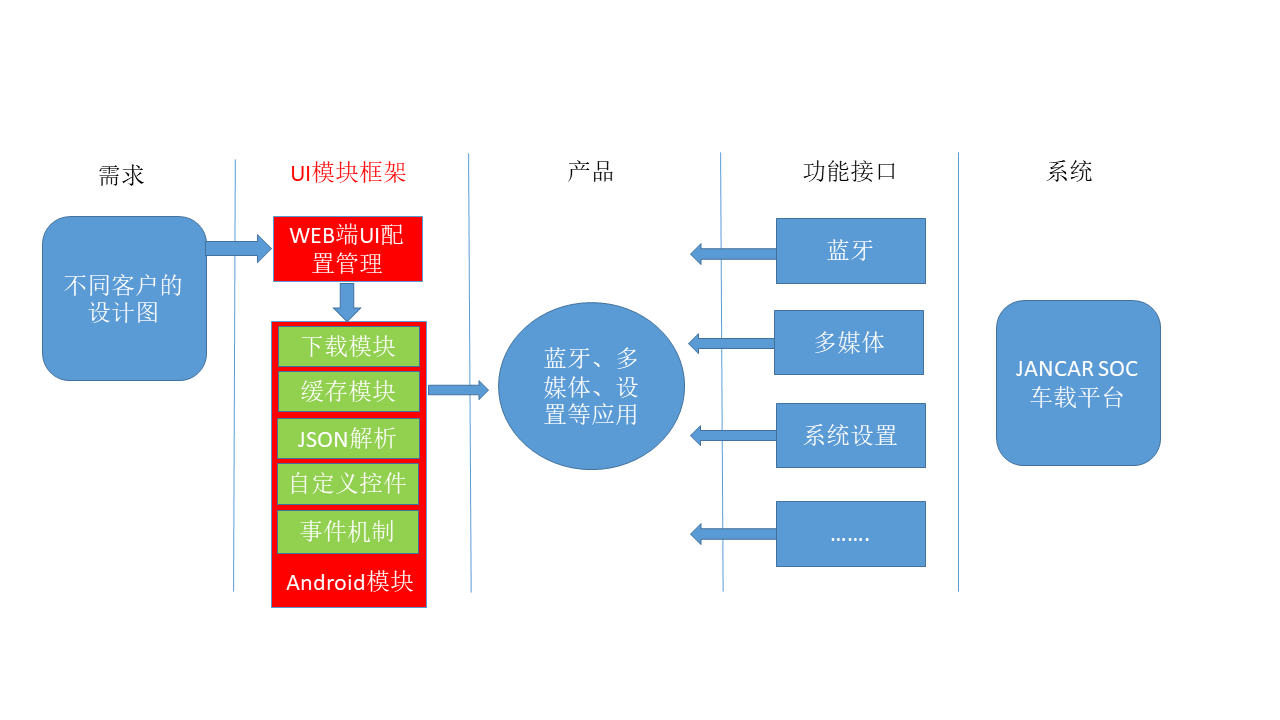
Android端通过访问Web服务端提供的接口，获取配置Json文本以及所关联包含的图片、文字信息内容等资源，最终解析生成应用UI界面。



# 项目设计概要

## 基本说明

### 基本关系简图



### 后台源码目录说明

/application

.../admin //**后台管理入口**

…/api //**网络访问接口**

…/auth //**RABC权限管理模块**

…/flyui /**/UI配置管理模块**

…/home //**公司网站管理后台**

…/index //**公司网站**

…/command.php

…/common.php

…/config.php

**…/database.php //数据库配置文件**

…/route.php

…/tags.php

/config //一些配置文件，可自行定义

/downfile //给客服下载的文件放在这里，自定义，可修改

/extend //自定义目录，放了一些可重用的公共函数

/public //网站公共资源,js css img等

/runtime //thinkphp运行需要目录，注意需要写权限

/thinkphp

/uploads //图片上传保存目录

/vendor //thinkphp扩展模块

### Android端源码目录说明

## 后台接口

### 后台接口安全策略

用户访问接口模块为API，目前接口访问安全策略为GET无限制，POST、PUT、DELETE需要登陆后才可操作。具体代码如下：今后需要视需求做进一步扩展。

**if**(Request::*instance*()->isGet()){  
}**else if**(Request::*instance*()->isPost()){  
 **if** (!session(**'userid'**)) {  
 $this->error(**"没有访问权限!"**);  
 }  
}**else if**(Request::*instance*()->isPut()){  
 **if** (!session(**'userid'**)) {  
 $this->error(**"没有访问权限!"**);  
 }  
}**else if**(Request::*instance*()->isDelete()){  
 **if** (!session(**'userid'**)) {  
 $this->error(**"没有访问权限!"**);  
 }  
}**else**{  
 **if** (!session(**'userid'**)) {  
 $this->error(**"没有访问权限!"**);  
 }  
}

### 后台接口简要说明

**请求地址：/api/app?type=launcher&themeName=Launcher-AP1&version=V1.0.0&acode**

**参数说明：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **参数** | **中文名称** | **参数类型** | **Y/可选** | **参数说明** |
| type | 应用类型 | 字符 | Y | 应用类型 |
| themeName | 模板名称 | 字符 | Y | 模板名称 |
| version | 版本号 | 字符 | Y | 应用版本号 |
| acode | 区域 | 字符 | N | 地理区域标识 |
| userId | 用户标识 | 字符 | N | 用户ID |
| token | 授权码 | 字符 | N | 授权码 |

如下接口地址：

/api/cell

/api/page

/api/celltype

/api/theme

对应cell,page,celltype,theme数据表，遵循Restful

如下接口地址：

/api/pagecell

/api/cellpagecell

/api/themetopcell

结合中间表，返回页面和cell的关联位置关系。

## 网页端设计

### 图片上传功能

图片上传功能将采用图片MD5值做为唯一区分标志，并根据MD5值生成图片URL，避免重复上传，Android客服端根据URL设计图片缓存，提升内存分配的合理性。

Web端将图片上传功能封装为单一控件模块，JS源码为flyimage.js，对应API接口/api/imagefile，完成图片上传功能。

如下图：





点击灰色区域会自动跳转到文件选择界面，选择图片即完成图片上传，同时会返回图片地址和图片尺寸等数据 。点击X按钮设计为点击事件，会返回回调方法，具体删除操作需在回调中实现。

代码调用示例如下：

**$**(**'.FlyImageInput'**).flyinput({  
 **ajaxurl**: **"{:url('/api/imagefile')}"**,  
 **autoup**: **true**,  
 **showView**: **true**,  
 **postData**: **false**,  
 **imageUrl**: **""**,  
 **imageWidth**: **""**,  
 **imageHeight**: **""**,  
 **width**: **"100%"**,  
 **height**: **"300px"**}).bind(**"success"**, **function** (e, data) {  
 **$**(**'#imageurl'**).val(data.saveName);  
}).bind(**"del"**, **function** (e, data) {  
 **$**(**'#imageurl'**).val(**""**);  
});

参数说明：

**ajaxurl**: **undefined**,//实现图片上传的接口，具体实现看源码  
**autoup**: **true**,//是否自动上传图片，默认自动上传  
**showView**: **true**,//是否显示预览，不显示为input控件  
**postData**: **false**,//是否生成上传url,图片宽高等input控件  
**imageUrl**: **""**,//默认显示图片  
**imageWidth**: **""**,//默认图片宽度  
**imageHeight**: **""**,//默认图片高度  
**width**: **"100%"**,//控件宽度  
**height**: **"240px"**//控件高度

回调：

bind(**"success"**, **function** (e, data))//上传成功回调

data.savaName//上传成功后返回的图片地址

data.width//上传图片的宽度

data.height//上传图片的高度

bind(**"del"**, **function** (e, data))//删除按钮回调

data.num组件cell标识

data.subnum组件cell.images 标识

两个参数用于指定哪个控件执行了删除操作  
如表达不足请结合源码理解。

### UI拖动布局界面

Web端将拖动布局功能封装为单一控件模块，JS源码位于flyscreen.js

## Android端设计

### 功能模块划分

### 顶部页面设计

每个模块都设计实现了一个顶部页面，顶部页面顾名思义位于应用展示最顶部。

### 事件机制

类似于EventBus

### 图片管理模块

功能包括，图片下载，图片内存缓存，磁盘缓存。