**信达网站前端开发**

**项目设计文档**

**项目组长：刘高峰**

**设计小组：昕左岸小组**

**编制小组：昕左岸小组**

**编制人员：刘高峰 尹浩民 孟庆凯 徐金源 杨丽芬**

**编制日期：2017年9月12日**

目录

[1.引言 3](#_Toc495665381)

[1.1 规范说明 3](#_Toc495665382)

[1.2 适用范围 3](#_Toc495665383)

[1.3 基本准则 3](#_Toc495665384)

[2.开发技术 3](#_Toc495665385)

[2.1前端框架：Vue 3](#_Toc495665386)

[2.2基础运行环境：Node.js 4](#_Toc495665387)

[2.3打包工具：Webpack 4](#_Toc495665388)

[2.4语法编译：Babel 5](#_Toc495665389)

[2.5 CSS预处理：Less 5](#_Toc495665390)

[2.6 HTTP操作：Axios 5](#_Toc495665391)

[2.7 UI框架：Element 6](#_Toc495665392)

[3.项目功能树形图 7](#_Toc495665393)

[4.文件组织结构图 8](#_Toc495665394)

[5. 系统架构图 9](#_Toc495665395)

[6. 功能流程图 10](#_Toc495665396)

[6.1功能总流程图 10](#_Toc495665397)

[6.2登录流程 10](#_Toc495665398)

[6.3注册流程 12](#_Toc495665399)

[6.4购物流程 13](#_Toc495665400)

[7. 系统约束 14](#_Toc495665401)

[7.1 通用规则 14](#_Toc495665402)

[7.2 信息验证规则 14](#_Toc495665403)

[7.3 文件规范 15](#_Toc495665404)

[7.4 html书写规范 15](#_Toc495665405)

[7.5 css书写规范 17](#_Toc495665406)

[7.6 JavaScript书写规范 20](#_Toc495665407)

[7.7 图片规范 21](#_Toc495665408)

[7.8 注释规范 22](#_Toc495665409)

[7.9 开发及测试工具约定 22](#_Toc495665410)

**1.引言**

1.1 规范说明

为提高团队协作效率，便于后台人员添加功能及前端后期优化维护，输出高质量的文档，搭建一个更好的前端架构，特制订此文档。本规范文档一经确认，前端开发人员必须按本文档规范进行前台页面开发。本文档如有不对或者不合适的地方请及时提出，经讨论决定后方可更改。

1.2 适用范围

此为前端开发团队遵循和约定的代码书写规范。

1.3 基本准则

符合web标准，语义化html， 采用模块化开发，结构表现行为分离，兼容性优良。页面性能方面，代码要求简洁明了有序，尽可能的减小服务器负载，保证最快的解析速度。

**2.开发技术**

2.1前端框架：Vue

Vue.js是一套构建用户界面的渐进式框架，采用自底向上增量开发的设计。由于本项目开发的是一个比较复杂的单页应用(SPA)，交互处理多，页面内容需要根据用户的操作动态变化，业务模型较复杂，利用Vue的话，因为Vue有声明式响应式的数据绑定与组件化的开发，且使用了Virtual DOM技术，会自动响应数据的变化情况，而且可以提供更多的易维护性，让基于网页的前端应用程序开发起来方便很多。

项目快速构建：Vue-cli，它是一个Vue的脚手架工具，可以自动生成好项目目录，配置好Webpack，以及各种依赖包的工具。

路由控制：Vue-router，它是Vue的一个前端路由器，利用它可以方便地管理请求入口和页面映射关系，它可以实现对页面局部进行无刷新的替换，让用户感觉就像切换到了网页一样。

状态管理模式：Vuex，是一个专为 Vue.js 应用程序开发的状态管理模式。它采用集中式存储管理应用的所有组件的状态，并以相应的规则保证状态以一种可预测的方式发生变化。本项目是一个SPA，因此为了更好地在组件外部管理状态，使用Vuex 将会成为自然而然的选择。

2.2基础运行环境：Node.js

Node.js 是一个基于 Chrome V8 引擎的 JavaScript 运行环境，利用Node.js 的包管理器 npm可以方便管理依赖所需的包。是打包构建工具的基础运行环境。

2.3打包工具：Webpack

本项目作为SPA，拥有着复杂的JavaScript代码和依赖包，为了简化复杂度使用了模块化与预处理器等方法，而利用Webpack可以分析项目结构，找到JavaScript模块以及其它一些浏览器不能直接运行的拓展语言，并将其打包为合适的格式以供浏览器使用。且由于HTTP是应用层协议，其下层TCP是运输层协议，TCP传输过程消耗时间可能比下载资源本身还要长，所以利用Webpack将一些小文件资源全部打包成一个文件，只要一次TCP传输过程，就可把多个资源下载下来，形成规模效应，可以让网页展现更快，用户体验更好。另外Webpack的热加载技术，可以在修改代码保存后，浏览器页面自动刷新变化，不需要去手动刷新。

2.4语法编译：Babel

本项目为了开发的方便使用了ES6的一些新特性，然而有的浏览器还不支持这些新特性，所以要用Babel这个转换器来将ES6转换成浏览器能支持的老版本JavaScript语言，以便让页面正常运行。

2.5 CSS预处理：Less

Less 是一门 CSS 预处理语言，它扩展了 CSS 语言。

CSS 需要书写大量看似没有逻辑的代码，不方便维护及扩展，不利于复用。而LESS在 CSS 的语法基础之上，引入了变量，Mixin，运算以及函数等功能，大大简化了 CSS 的编写，使 CSS 更易维护和扩展。可以用更少的代码做更多的事情。

2.6 HTTP操作：Axios

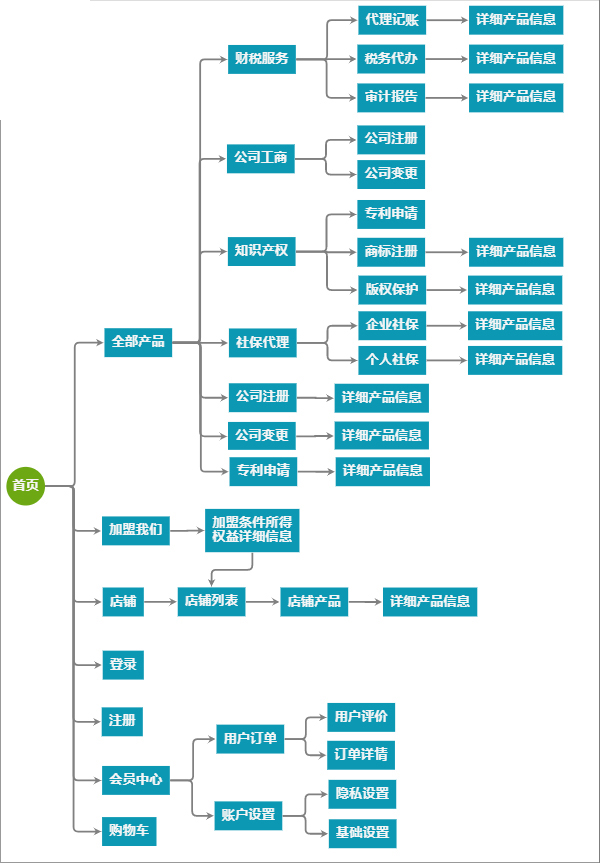
Axios 是一个基于Promise 用于浏览器和 nodejs 的 HTTP 客户端。单页应用程序中的数据需要通过ajax的方式获取，也需要通过ajax的方式提交到后端。vue本身没有封装ajax操作库，要通过Axios来进行ajax操作。Axios 使用了 Promise，而 jQuery自3.0 版本才支持 Promise,如果只想使用jQuery的ajax的话，引入整个 jQuery 是不小的负担，所以使用Axios是一个不错的选择。而且Axios是vue官方推荐的ajax库。

2.7 UI框架：Element-UI/Mint-UI

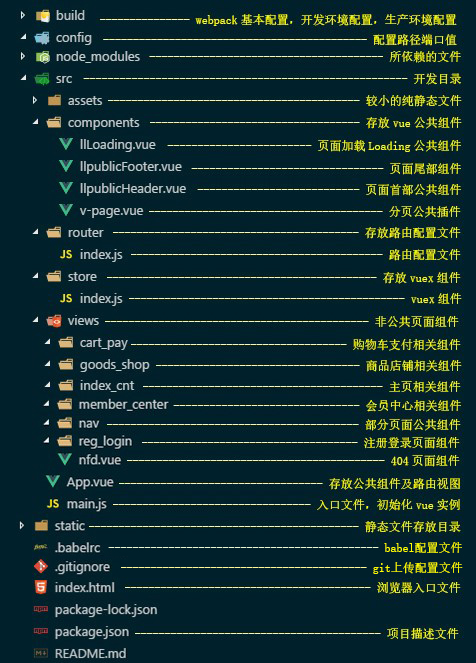
Element是一套基于Vue 2.0的组件库，很好地支持Vue2.0，并且提供了配套设计资源，可以帮助网站快速成型，提高项目的效率。

Mint-UI是一套基于 Vue.js 的移动端组件库，通过它，可以快速构建出风格统一的页面，提升开发效率。真正意义上的按需加载组件。考虑到移动端的性能门槛，Mint UI 采用 CSS3 处理各种动效，避免浏览器进行不必要的重绘和重排。依托 Vue.js 高效的组件化方案，Mint UI 做到了轻量化。

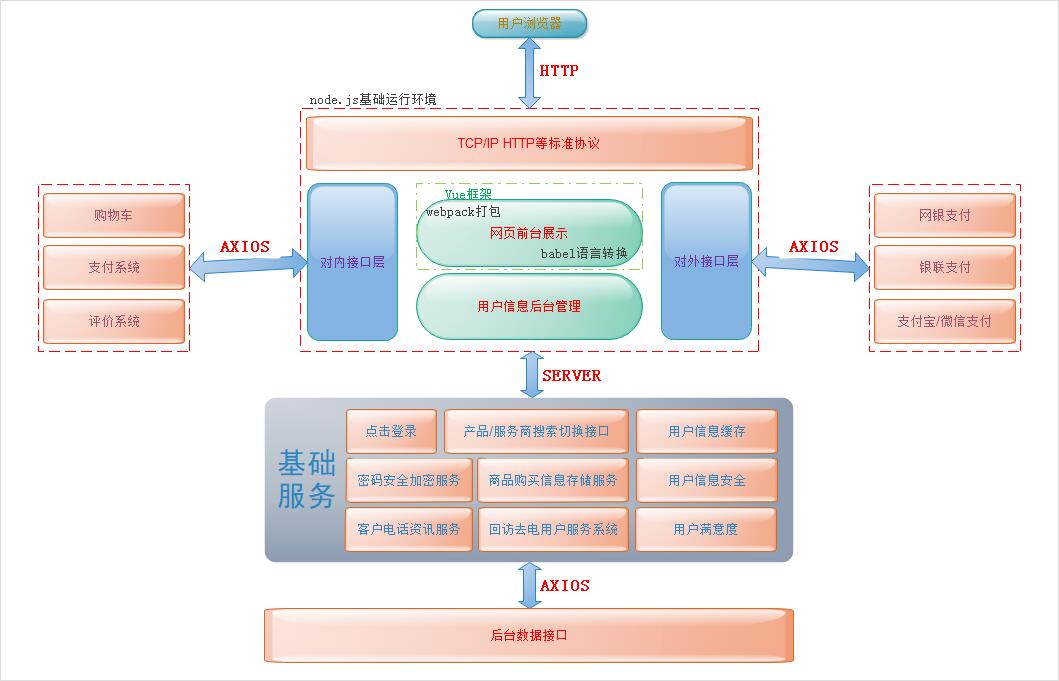
**3.项目功能树形图**



**4.文件组织结构图**

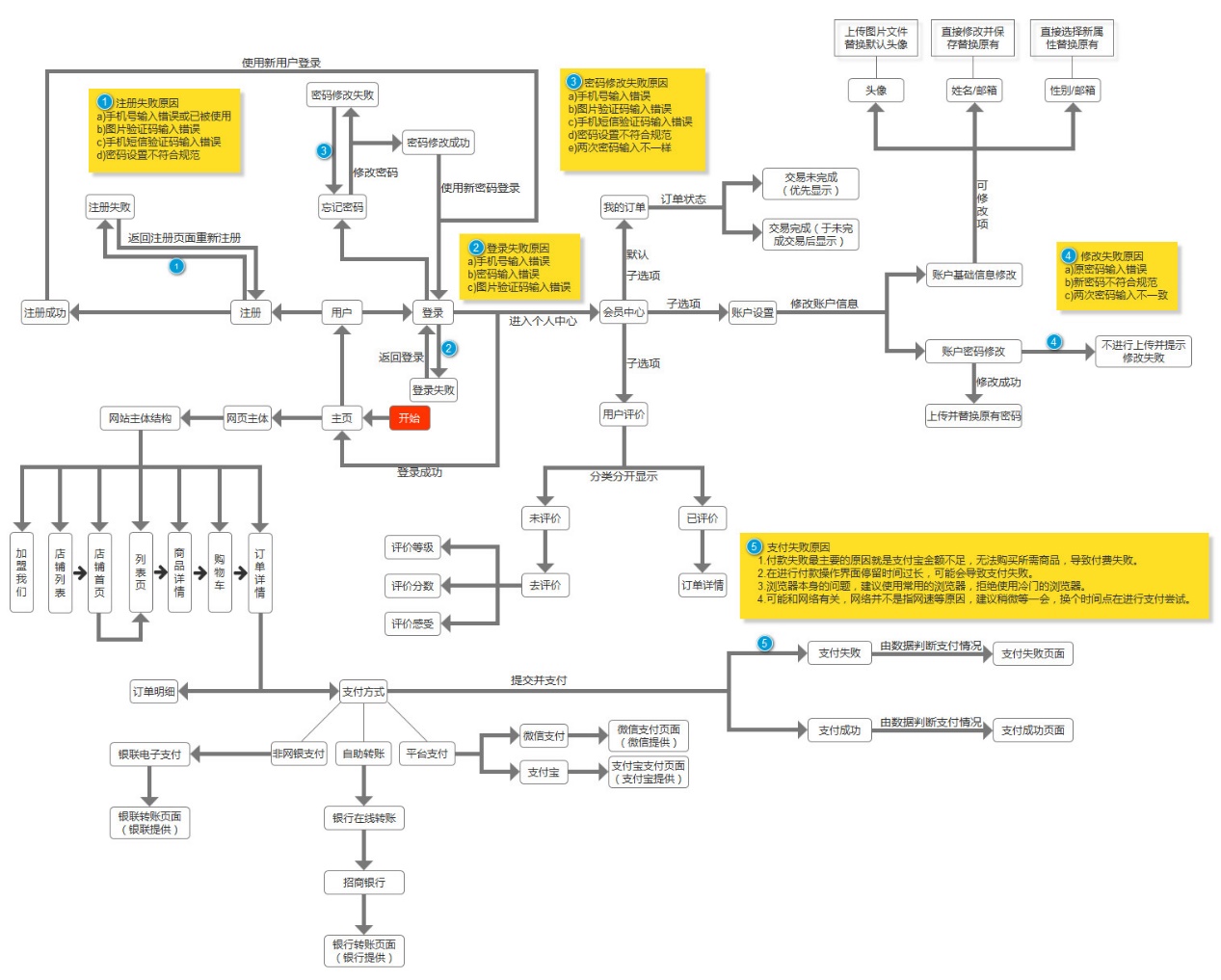


**5. 系统架构图**

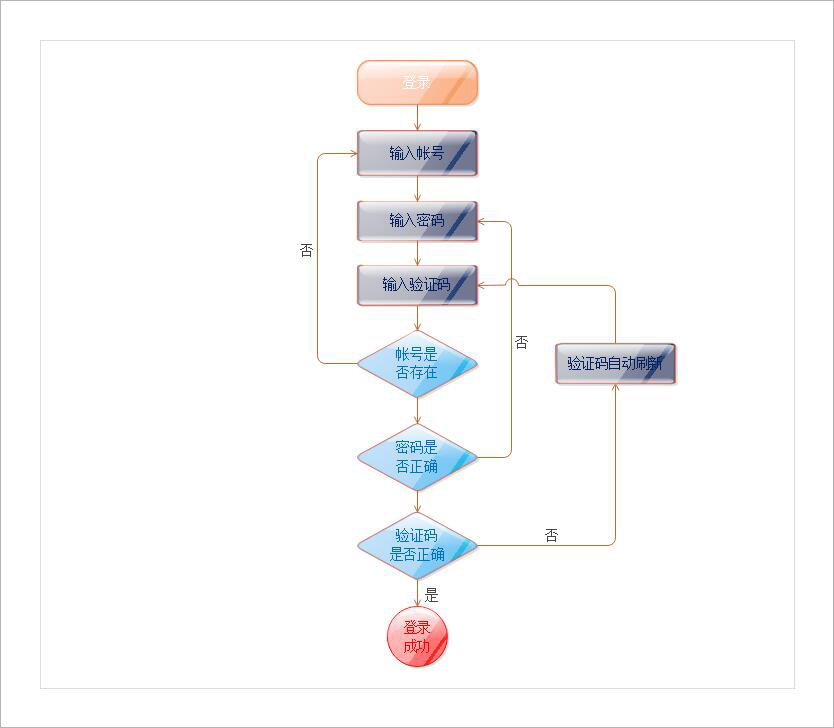


**6. 功能流程图**

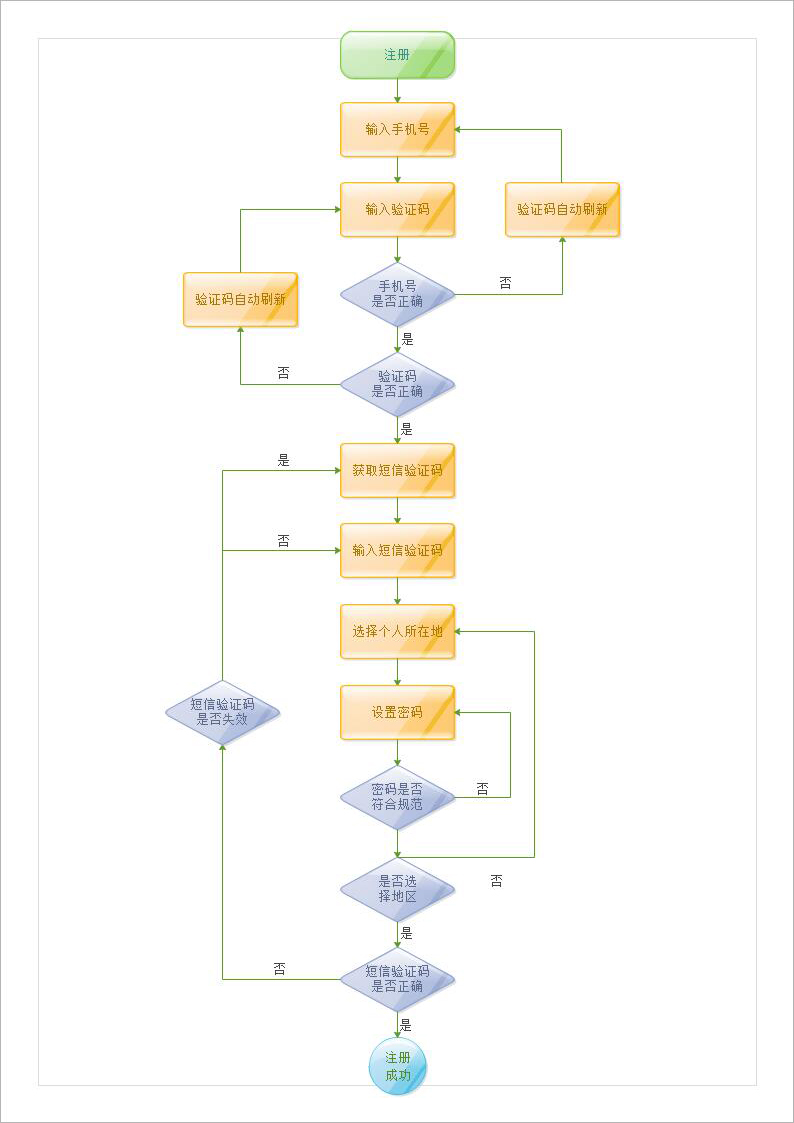
6.1功能总流程图



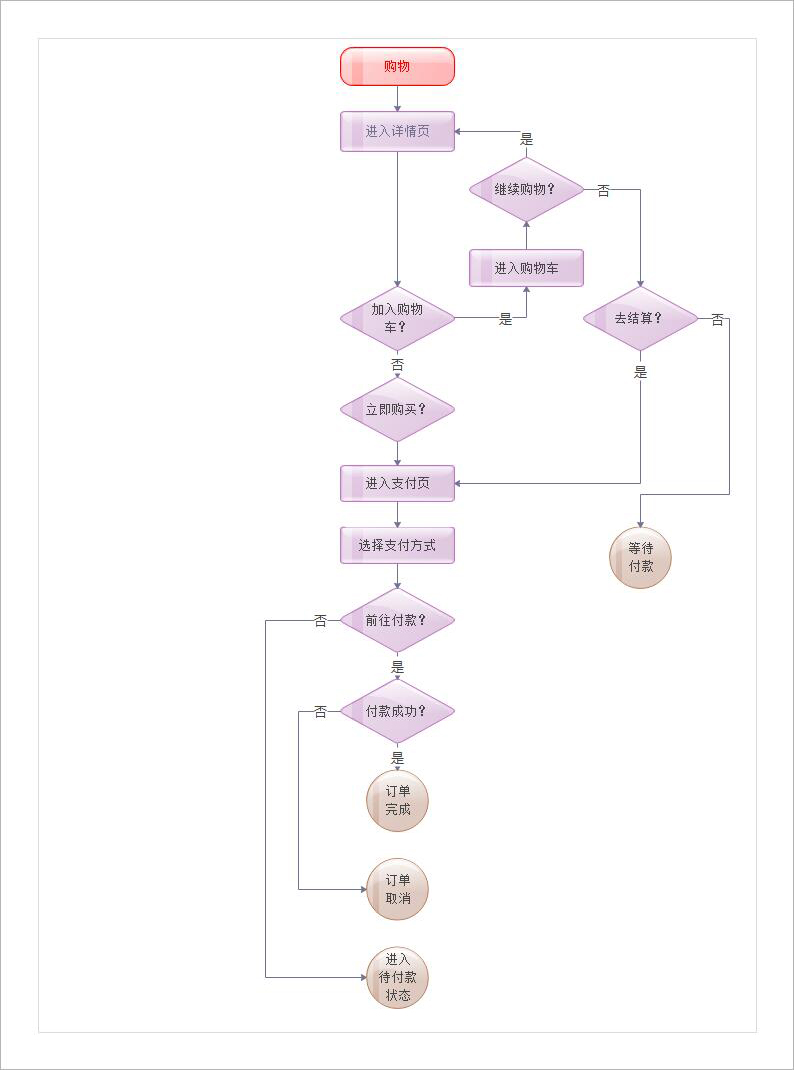
6.2登录流程



6.3注册流程



6.4购物流程



**7. 系统约束**

7.1 通用规则

|  |  |  |
| --- | --- | --- |
| 编号 | 功能 | 描述 |
| 1 | 浏览器兼容 | 网页需兼容的浏览器如下： 1、安卓 4.0 版本及以上系统自带的浏览器； 2、微信内置浏览器、QQ 内置浏览器； 3、小米、华为、三星、苹果、vivo、OPPO 手机自带的浏览器； 4、“uc 浏览器、搜狗浏览器、QQ 浏览器、谷歌浏览器、360 浏览器”的手机版本 |
| 2 | 布局尺寸 | 1、若没有特别说明，原型中的图片、布局、颜色、尺寸均以设计稿为准。 |
| 3 | 微信适配 | 1、微信规则与 pc 相同，仅页面大小及布局有所调整，具体请参考原型和设计稿 |
| 4 | 访问限制 | 1、某些特定的页面，需区分“登录用户”和“非登录用户”。 |

7.2 信息验证规则

|  |  |  |
| --- | --- | --- |
| 编号 | 项目 | 描述 |
| 1 | 用户名 | 1、4到16位，以字母开头，只允许英文字母、数字、下划线、中划线。 |
| 2 | 密码 | 1、6-20位，只能使用字母、数字、下划线。 |
| 3 | 邮箱 | 1、只允许英文字母、数字、下划线、英文句号、以及中划线组成，且前后不能为特殊字符。  2、名称与域名之间用@连接，域名后面包含英文句号。 |
| 4 | 手机号 | 1、长度为11位纯数字，并且以1为开头；  2、第二位可为3,4,5,7,8,中的任意一位；最后以0-9的9个整数结尾。 |
| 5 | 电话号码 | 1、区号+号码，区号以0开头，3位或4位；号码由7位或8位数字组成；  2、区号与号码之间可以无连接符，也可以“-”连接。 |

7.3 文件规范

1. html, css, js, images文件均归档至<系统开发规范>约定的目录中;

2. html文件命名: 英文命名, 后缀.htm. 同时将对应界面稿放于同目录中, 若界面稿命名为中文, 请重命名与html文件同名, 以方便后端添加功能时查找对应页面;

3. css文件命名: 英文命名, 后缀.css. 共用base.css, 首页index.css, 其他页面依实际模块需求命名.;

4. Js文件命名: 英文命名, 后缀.js. 共用common.js, 其他依实际模块需求命名.

7.4 html书写规范

7.4.1. 文档类型声明及编码: 统一为html5声明类型; 编码统一为 , 书写时利用IDE实现层次分明的缩进;

7.4.2. 非特殊情况下样式文件必须外链至...之间;非特殊情况下JavaScript文件必须外链至页面底部;

7.4.3. 引入样式文件或JavaScript文件时, 须略去默认类型声明, 写法如下:

<script src="..."></script>

7.4.4. 引入JS库文件, 文件名须包含库名称及版本号及是否为压缩版, 比如jquery-1.4.1.min.js; 引入插件, 文件名格式为库名称+插件名称, 比如jQuery.cookie.js;

7.4.5. 所有编码均遵循xhtml标准, 标签 & 属性 & 属性命名 必须由小写字母及下划线数字组成, 且所有标签必须闭合, 包括 br (), hr()等; 属性值必须用双引号包括;

7.4.6. 充分利用无兼容性问题的html自身标签, 比如span, em, strong, optgroup, label,等等; 需要为html元素添加自定义属性的时候, 首先要考虑下有没有默认的已有的合适标签去设置, 如果没有, 可以使用须以"data-"为前缀来添加自定义属性，避免使用"data:"等其他命名方式;

7.4.7. 语义化html, 如 标题根据重要性用h\*(同一页面只能有一个h1), 段落标记用p, 列表用ul, 内联元素中不可嵌套块级元素;

7.4.8. 尽可能减少div嵌套, 如  
欢迎访问XXX, 您的用户名是  
用户名

完全可以用以下代码替代:  
欢迎访问XXX, 您的用户名是用户名;  
7.4.9. 书写链接地址时, 必须避免重定向，例如：href="http://itaolun.com/", 即须在URL地址后面加上“/”；

7.4.10. 在页面中尽量避免使用style属性,即style="…";

7.4.11. 必须为含有描述性表单元素(input, textarea)添加label；

7.4.12. 能以背景形式呈现的图片, 尽量写入css样式中;

7.4.13. 重要图片必须加上alt属性; 给重要的元素和截断的元素加上title;

7.4.14. 给区块代码及重要功能(比如循环)加上注释, 方便后台添加功能;

7.4.15. 特殊符号使用: 尽可能使用代码替代: 比如 <(<) & >(>) & 空格( ) & »(») 等等;

7.4.16. 书写页面过程中, 请考虑向后扩展性;

7.4.17. class & id 参见 css书写规范.

7.5 css书写规范

7.5.1. 编码统一为utf-8;

7.5.2. 协作开发及分工: i会根据各个模块, 同时根据页面相似程序, 事先写好大体框架文件, 分配给前端人员实现内部结构&表现&行为;  共用css文件base.css由i书写, 协作开发过程中, 每个页面请务必都要引入, 此文件包含reset及头部底部样式, 此文件不可随意修改;

7.5.3. class与id的使用: id是唯一的并是父级的, class是可以重复的并是子级的, 所以id仅使用在大的模块上, class可用在重复使用率高及子级中; id原则上都是由我分发框架文件时命名的, 为JavaScript预留钩子的除外;

7.5.4. 为JavaScript预留钩子的命名, 请以 js\_ 起始, 比如: js\_hide, js\_show;

7.5.5. class与id命名: 大的框架命名比如header/footer/wrapper/left/right之类的在2中由i统一命名.其他样式名称由 小写英文 & 数字 & \_ 来组合命名, 如i\_comment, fontred, width200; 避免使用中文拼音, 尽量使用简易的单词组合; 总之, 命名要语义化, 简明化.

7.5.6. 规避class与id命名(此条重要, 若有不明白请及时与i沟通):

    a） 通过从属写法规避, 示例见d;

    b）取父级元素id/class命名部分命名, 示例见d;

    c）重复使用率高的命名, 请以自己代号加下划线起始, 比如i\_clear;

    d）a,b两条, 适用于在2中已建好框架的页面, 如, 要在2中已建好框架的页面代码  
中加入新的div元素,  
按a命名法则:...,  
样式写法:  #mainnav  .firstnav{.......}  
按b命名法则:...,  
样式写法:  .main\_firstnav{.......}

7.5.7. css属性书写顺序, 建议遵循:  布局定位属性-->自身属性-->文本属性-->其他属性. 此条可根据自身习惯书写, 但尽量保证同类属性写在一起. 属性列举: 布局定位属性主要包括: display & list-style & position（相应的 top,right,bottom,left） ＆ float & clear ＆ visibility ＆ overflow； 自身属性主要包括: width & height & margin & padding & border & background; 文本属性主要包括：color & font & text-decoration & text-align & vertical-align & white- space & 其他 & content; 我所列出的这些属性只是最常用到的, 并不代表全部;

7.5.8. 书写代码前, 考虑并提高样式重复使用率;

7.5.9. 充分利用html自身属性及样式继承原理减少代码量, 比如:  
这儿是标题列表2010-09- 15  
定义  
ul.list li{position:relative}  ul.list li span{position:absolute; right:0}  
即可实现日期居右显示

7.5.10. 样式表中中文字体名, 请务必转码成unicode码, 以避免编码错误时乱码;

7.5.11. 背景图片请尽可能使用sprite技术, 减小http请求, 考虑到多人协作开发, sprite按模块制作;

7.5.12. 使用table标签时(尽量避免使用table标签), 请不要用width/ height/cellspacing/cellpadding等table属性直接定义表现, 应尽可能的利用table自身私有属性分离结构与表现 , 如thead,tr,th,td,tbody,tfoot,colgroup,scope; (cellspaing及cellpadding的css控制方法:table{border:0;margin:0;border-collapse:collapse;} table th, table td{padding:0;}, base.css文件中我会初始化表格样式)

7.5.13. 杜绝使用 兼容 ie8;

7.5.14. 用png图片做图片时, 要求图片格式为png-8格式,若png-8实在影响图片质量或其中有半透明效果, 请为ie6单独定义背景:  
\_background:none;\_filter:progid:DXImageTransform.Microsoft.AlphaImageLoader (sizingMethod=crop, src=’img/bg.png’);

7.5.15. 避免兼容性属性的使用, 比如text-shadow || css3的相关属性;

7.5.16. 减少使用影响性能的属性, 比如position:absolute || float ;

7.5.17. 必须为大区块样式添加注释, 小区块适量注释;

7.5.18. 代码缩进与格式: 建议单行书写, 可根据自身习惯, 后期优化i会统一处理;

7.6 JavaScript书写规范

7.6.1. 文件编码统一为utf-8, 书写过程过, 每行代码结束必须有分号; 原则上所有功能均根据XXX项目需求原生开发, 以避免网上down下来的代码造成的代码污染(沉冗代码 || 与现有代码冲突 || ...);

7.6.2. 库引入: 原则上仅引入jQuery库, 若需引入第三方库, 须与团队其他人员讨论决定;

7.6.3. 变量命名: 驼峰式命名. 原生JavaScript变量要求是纯英文字母, 首字母须小写, 如iTaoLun;  
jQuery变量要求首字符为'\_', 其他与原生JavaScript 规则相同, 如: \_iTaoLun;  
另, 要求变量集中声明, 避免全局变量.

7.6.4. 类命名: 首字母大写, 驼峰式命名. 如 ITaoLun;

7.6.5. 函数命名: 首字母小写驼峰式命名. 如iTaoLun();

7.6.6. 命名语义化, 尽可能利用英文单词或其缩写;

7.6.7. 尽量避免使用存在兼容性及消耗资源的方法或属性, 比如eval\_r() & innerText;

7.6.8. 后期优化中, JavaScript非注释类中文字符须转换成unicode编码使用, 以避免编码错误时乱码显示;

7.6.9. 代码结构明了, 加适量注释. 提高函数重用率;

7.6.10. 注重与html分离, 减小reflow, 注重性能.

7.7 图片规范

7.7.1. 所有页面元素类图片均放入img文件夹, 测试用图片放于img/demoimg文件夹;

7.7.2. 图片格式仅限于gif || png || jpg;

7.7.3. 命名全部用小写英文字母 || 数字 || \_ 的组合，其中不得包含汉字 || 空格 || 特殊字符；尽量用易懂的词汇, 便于团队其他成员理解; 另, 命名分头尾两部分, 用下划线隔开, 比如ad\_left01.gif || btn\_submit.gif;

7.7.4. 在保证视觉效果的情况下选择最小的图片格式与图片质量, 以减少加载时间;

7.7.5. 尽量避免使用半透明的png图片(若使用, 请参考css规范相关说明);

7.7.6. 运用css sprite技术集中小的背景图或图标, 减小页面http请求, 但注意, 请务必在对应的sprite psd源图中划参考线, 并保存至img目录下

7.8 注释规范

7.8.1. html注释: 注释格式 , '--'只能在注释的始末位置,不可置入注释文字区域;

7.8.2. css注释: 注释格式 ;

7.8.3. JavaScript注释, 单行注释使用'//这儿是单行注释' ,多行注释使用 ;

7.9 开发及测试工具约定

建议使用Aptana || Dw || Vim , 亦可根据自己喜好选择, 但须遵循如下原则:

7.9.1. 不可利用IDE的视图模式'画'代码;

7.9.2. 不可利用IDE生成相关功能代码, 比如Dw内置的一些功能js;

7.9.3. 编码必须格式化, 比如缩进;  
测试工具: 前期开发仅测试Chrome, 后期优化时加入Opera & FireFox & Safari;  
建议测试顺序: Chrome-->FireFox-->Opera--> Safari, 建议安装firebug插件.