

WEB 票务在线商城 需求文档

作者	更新日期	版本号	更新说明
邓乃文	2020-09	1.0	

目录

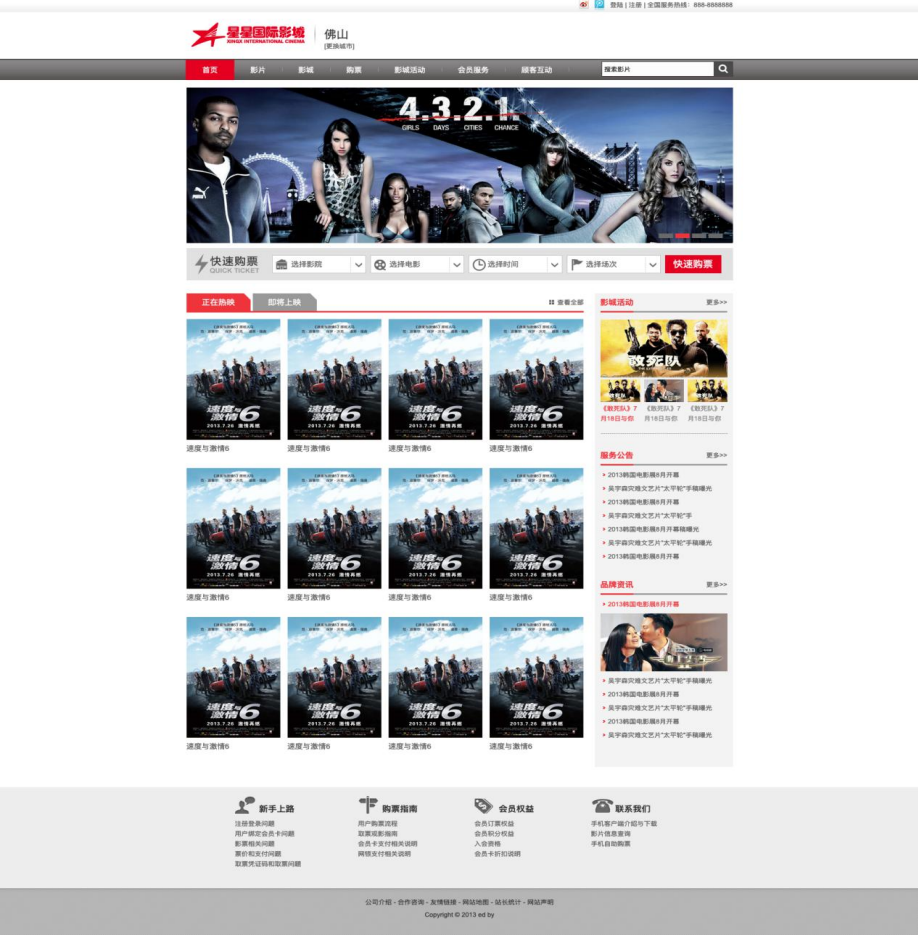
一、 引言.....	2
二、 页面需求.....	4
三、 功能需求.....	5
四、 项目规范.....	6
五、 技术栈.....	7

一、引言

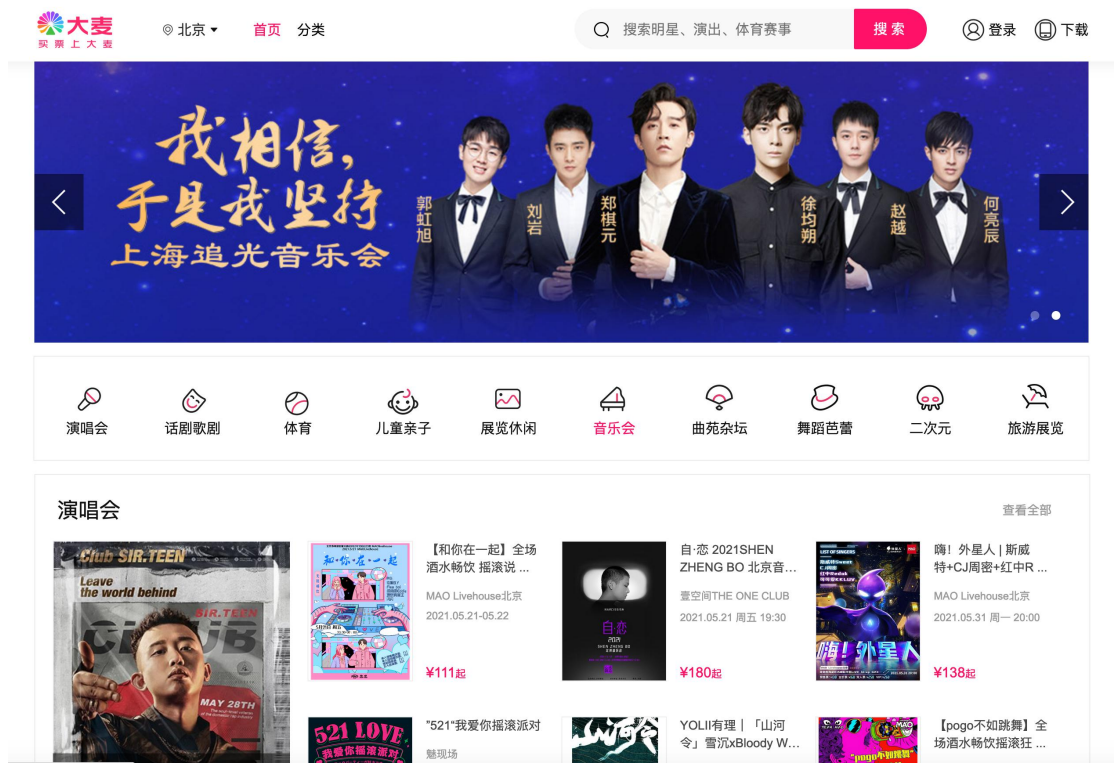
- 1. 编写目的
详细说明本项目的页面需求和功能需求点、代码规范和技术要求
- 2. 项目简介

web 在线票务商城是在符合目前移动互联网大环境下而提出的演艺票务在线商城 app 解决方案。在当前国情下，对于在线购物已然是全民日常活动，其日常用品、吃、住、行等方面行业里有很多成熟的项目解决方案，如京东、天猫、携程、大众点评、美团等。但对于演艺圈方面的票务在线商城 app，行业的成品并不多。所以本项目旨于向用户提供用户体验极佳的演艺票务 app。为行业注入新鲜活力，为用户购票提供更加便捷的服务。

- 3. 参考资料
 - 1. 蜗牛电影院（截图）

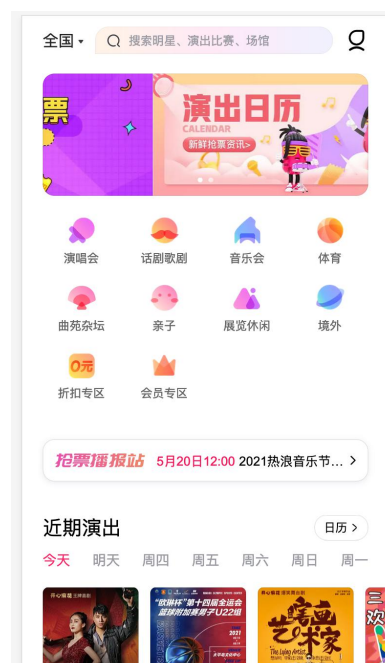


2. 大麦网 (https://www.damai.cn/)

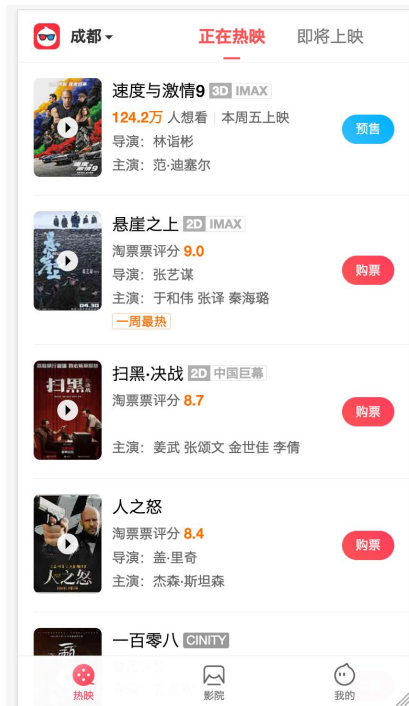


App:

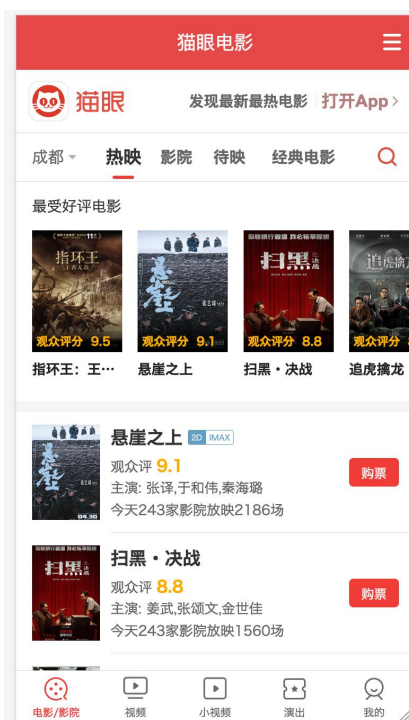
1. 大麦网



2. 淘票票



3. 猫眼



二、页面需求

1. 必需的部分(pc 和移动端是单独的),其他扩展的可视情况加分:

- ◆ 首页 (参考: 头部、导航栏、轮播图、一个或多个商品列表数据模块)
- ◆ 商品详情页面(参考: 商品的轮播图、数量加减、加入购物车、商品详情、商品评论)
- ◆ 影院、剧院详情页面 (需包含简介, 及旗下的上线的影片或上演的演

演唱会之类的信息)

- ◆ 选座: (参考: 当前商品信息简介、选座)
- ◆ 支付: (参考: 订单信息、支付方式、收货地址选择)
- ◆ 搜索: 可以按照各种指标进行搜索。比如名字、分类、剧院、影城、导演、演员等
- ◆ 商品分类页面: 显示该分类下的所有商品 (商品可以采取瀑布流方式, 具体参考百度图片)
- ◆ 个人中心: 至少包含修改密码、查看、删除个人订单
- ◆ 登录\注册

三、功能需求

1. 数据渲染:

- i. 所有页面的数据都应该是动态渲染出来, 且是通过 **ajax** 请求获取 (除本地存储(**localStorage**)数据以外)。包括但不限于 用户数据、商品数据、订单数据、购物车数据、影院、影城数据。
- ii. 用户数据: 保存到 **localStorage**
- iii. 商品数据: 可保存到 **json** 文件或 **fastmock** 中
- iv. 订单数据: 保存到 **localStorage**
- v. 购物车数据: 保存到 **localStorage**
- vi. 影院、影城数据: **json** 文件或 **fastmock**

2. 用户数据:

- i. 所有用户相关数据都保存到本地存储中
- ii. 每个用户包含: 账号、密码、昵称、性别、年龄、相关订单数据、购物车数据、账号需保证唯一
- iii. 建议: 每个用户数据都是单独的一个键值对, 以账号名为键值, 其相关数据为键值。

3. 商品数据:

- i. 电影至少包含: 商品编号(唯一)、名称、价格、导演、上映时间、电影类型、时长、主要演员、简介、预告视频(可选)、所属国家地区、海报
- ii. 演艺圈票 (比如演唱会、戏剧、歌剧等票): 商品编号(唯一)、类别、名称、价格、库存、所属剧院、海报、表演者、地点、开场时间、简介

4. 评论数据:

- i. 用户昵称、评论时间、评分、内容、具体的场次和所属剧院\影城

5. 订单数据:

- i. 需包含: 订单号(唯一)、商品名称、数量、支付总价、所属用户
- ii. 订单页面需提供评论功能

6. 影城数据:

- i. 需包含: 名称、地点、简介、海报、上映影片和排场信息

7. 剧院数据:

- i. 需包含: 名称、简介、地点、海报、旗下的演艺场次信息

8. 首页轮播图: 提供手动前后图片切换按钮,

9. 搜索: 提供完整的搜索功能

- i. 要求至少可以通过票的名称、剧院、影城、地点、分类、时间其中一个指标来进行搜索

- ii. 全做完可酌情加分
- 10. 登录/注册：需保存用户数据。在进行登录或注册时，需提供以下检测手段：
 - i. 数据合法性正则表达式判断：（需使用断言）
 - ii. 用户名是否已存在检测(推荐：用户输入数据失去焦点后发送 **ajax** 请求进行验证，并通过更改边框颜色以及提示来给用户反馈)
 - iii. 验证码判断：随机 5 个字母或数字并进行输入是否正确验证
- 11. 购物车数据操作：
 - i. 能够实现购物车单个数据的删除，修改，多个数据选择或全选删除。并根据商品所在影城或剧院进行分类展示、总价
- 12. 选座&支付
 - i. 选座需实现，数据保存。即该场次的选座数据需保存，只需保存已售即可。选座后加入购物车
 - ii. 支付需跟选座数据挂钩。并实现订单 15 分钟失效(开发中可以先 10s 失效)，具体可以设置倒计时，如果还没支付则回到购物车页面，不保存数据
- 13. 地图：
 - i. 影城、剧院、收货地址需使用百度地图进行展示
- 14. 商品评论
 - i. 用户评论需到对应商品详情中登录后进行评论，
 - ii. 评论数据需保存到本地，每个商品为一个键值对
 - iii. 评论数据跟商品要一一对应。

四、项目规范

1. 合理项目目录架构

- 根目录 // 一般是项目名
- src // 表示源代码目录
 - css // 存放 css 文件
 - sass // 存放 sass 文件
 - html //存放 html 文件
 - js //存放页面 js 文件
 - util //存放第三方库,比如 jQuery、bootstrap 等
 - images //存放所需图片
 - data //存放数据文件, 比如 json 文件,没有则可以省略

2. 接口规范

a. 请求命名规范

- a. 所有请求路径都需加上项目基础路径，一般为/项目名
- b. 每个具体的请求在基础路径后加入该请求的功能
- c. 例子：
 - 1. 发送登录请求： /WoniuTickets/login
 - 2. 获取电影数据： /WoniuTickets/getMovies

3. 数据规范

- a) 数据保存到本地： 将对象转为一定格式的字符串并保存到本地,拿数据时直接利用 **split** 和正则表达式转换为对象，比如保存多个订单数据：程序里

是一个对象数组，那么可以指定分隔符（比如;）将多个订单数据进行拼接处理，如下：

订单编号=001&商品编号=123&数量=4&金额=400&用户名=zs;订单编号=002&商品编号=345&数量=3&金额=100&用户名=ls;

即： 订单 1;订单 2;订单 3;订单 n;

五、技术栈

1. 页面技术栈:

- i. HTML
- ii. css
- iii. sass
- iv. bootstrap

2. 数据交互技术栈

- i. JavaScript-ES5
- ii. JavaScript DOM 和 jQuery