



**专业综合设计论文——**

品牌广告投放管理系统

学 院 管理学院

专 业 电子商务

年级班别 2016级1班

学 号 3116004308

学生姓名 陈祖桐

指导教师 闵惜琳

**2019年 12 月**

目录

[**1 选题来源与业务背景** 4](#_Toc26027734)

[1.1选题来源 4](#_Toc26027735)

[1.2 业务背景与业务问题 4](#_Toc26027736)

[1.3 具体业务流程 4](#_Toc26027737)

[1.4业务目标 5](#_Toc26027738)

[1.4.1 广告素材管理 5](#_Toc26027739)

[1.4.2 下单管理 5](#_Toc26027740)

[1.4.3 菜单管理 5](#_Toc26027741)

[**2 业务建模** 6](#_Toc26027742)

[2.1 业务分析 6](#_Toc26027743)

[2.2 业务用例 6](#_Toc26027744)

[2.3业务用例场景 7](#_Toc26027745)

[2.3.1 广告素材管理业务用例场景 8](#_Toc26027746)

[2.3.2 下单管理业务用例场景 9](#_Toc26027747)

[2.3.3 菜单管理业务用例场景 9](#_Toc26027748)

[2.4业务用例规约 10](#_Toc26027749)

[**3 需求分析** 13](#_Toc26027750)

[3.1 系统目标 13](#_Toc26027751)

[3.1.1 广告素材管理业务 13](#_Toc26027752)

[3.1.2 下单管理业务 13](#_Toc26027753)

[3.1.3 菜单管理业务 14](#_Toc26027754)

[3.2 核心业务 19](#_Toc26027755)

[3.2.1核心业务场景图 19](#_Toc26027756)

[3.2.2 关键业务用例 21](#_Toc26027757)

[3.3 概念用例与场景图 21](#_Toc26027758)

[3.3.1概念用例 21](#_Toc26027759)

[3.3.2概念用例场景图 22](#_Toc26027760)

[**4 系统分析** 24](#_Toc26027761)

[4.1 软件架构与框架 24](#_Toc26027762)

[4.1.1 开发工具 24](#_Toc26027763)

[4.1.2 工具介绍 24](#_Toc26027764)

[4.1.3软件架构与框架 25](#_Toc26027765)

[4.1.4 小程序安全管理 25](#_Toc26027766)

[4.2确定系统用例 26](#_Toc26027767)

[4.3交互顺序图 29](#_Toc26027768)

[4.3.1 菜单管理 29](#_Toc26027769)

[4.3.2 素材管理 31](#_Toc26027770)

[4.3.3 下单管理 33](#_Toc26027771)

[4.4 分析类图 33](#_Toc26027772)

[**5 系统设计** 35](#_Toc26027773)

[5.1数据库的详细设计 35](#_Toc26027774)

[5.2包设计 39](#_Toc26027775)

[5.2.1 素材管理 40](#_Toc26027776)

[5.2.2 下单管理 40](#_Toc26027777)

[5.2.3 菜单管理 40](#_Toc26027778)

[5.3 详细设计 41](#_Toc26027779)

[5.3.1 素材管理模块 43](#_Toc26027780)

[5.3.2 下单管理模块 49](#_Toc26027781)

[5.3.3 菜单管理模块 56](#_Toc26027782)

[**6 心得体会** 64](#_Toc26027783)

[**参考文献** 65](#_Toc26027784)

[**附录** 65](#_Toc26027785)

# 1 选题来源与业务背景

## 1.1选题来源

选题是同组组员实习公司中真实的业务需求来源，来自北京字节跳动科技有限公司品牌广告业务中针对流程中的广告素材管理，下单管理，后台管理以及菜单管理进行系统设计与开发。

## 1.2 业务背景与业务问题

字节跳动主要业务是互联网产品广告业务，广告业务按广告类型分为品牌广告、竞价广告、搜索广告和各种规模的招商类项目。

根据目前的实际情况，在代理商/广告主、媒介两者实际操作中所存在的情况，在品牌广告业务主要存在以下两个问题。

第一，提交的广告素材的规格要求不统一，提交素材包含文字说明、多种图片规格、多种视频规格、链接等，通过邮件不仅不能检查出规格是否符合标准，也无法明确素材数量，出现代理商/广告主需要重复修改与补交素材，大大降低投放的效率。

第二，各类需求、素材、下单均通过邮件发送，媒介在同时收到大量邮件时，容易混淆与遗漏部分邮件，也容易在与代理商/广告主沟通过程中忽略了某笔广告的投放业务，无法很好地直观地管理各类需求、素材、下单的进度。

## 1.3 具体业务流程

在设计需要负责开发的系统业务模块，需要先具体了解其真实的业务场景，于是先对公司的品牌广告下单流程进行了解与分析，总结出具体业务流程。

广告首先需要经过预审看是否符合要求，其次再通过代理商提交广告需求后，媒介确认需求查询广告库存与排期，再要求代理商上交广告素材进行二次审核，素材审核过关才能进行下单，媒介再在后台对广告进行投放的执行操作。

## 1.4业务目标

针对业务存在的问题，确定开发以下三个业务模块，广告素材管理，下单管理，菜单管理来对业务流程进行流程的优化，统一操作的方式与规格，减少重复的沟通，达到提高整个品牌广告下单流程的工作效率。

### 1.4.1 广告素材管理

广告素材是不同形式和格式内容的物理表示，不同用途的素材需要使用大量的用于不同目的的素材格式。广告素材管理并不仅是一个存储文件的仓库，它还必须理解文件结构、媒体语法、甚至特定部分的语义；提供媒体索引、自动信息检索等功能和服务。

内容管理系统还可以通过自动检索来获得内容的附加信息。

### 1.4.2 下单管理

下单管理是后台系统中较为重要的一部分，它记录了所有的交易数据，下单管理后台的操作用户都是公司内部人员，但同时也需要C端用户的需求。所以在设计时，订单管理系统会包括两部分的内容：一是要能够与C端用户在整个订单流程中各个场景的操作相对应；另一个是要能满足公司内部相关部门的需求。

### 1.4.3 菜单管理

菜单管理是指整体的需求管理、素材管理与下单管理的整体的菜单与进度的显示情况。可以通过菜单管理系统跟进同一个订单的不同情况单的情况，都进行可视化的菜单式管理。或者同个进度的所有表

# 2 业务建模

## 2.1 业务分析

经过调研与了解，总结品牌广告下单流程，主要是通过邮件形式相互沟通与交流进行。而具体的业务流程如下：

代理商先按照【需求邮件模板】填写完相关信息，向媒介同学发送邮件。媒介同学确认好排期会返回【排期表】。代理商充值费用到广告主账户中。广告主账户充值后，媒介同学才能进行操作，没充值则无法进行操作。

代理商按照【排期表】或刊例素材要求（以媒介排期表的素材要求为准），准备好投放的广告素材后，邮件发送素材给媒介同学。按照【下单邮件模板】填写完相关信息，发送邮件给媒介同学，媒介负责执行。

## 2.2 业务用例

业务用例是用来捕获功能性需求的，功能性需求是业务主角的业务目标来体现的。对于业务主角来说，他所负责的业务需要由一系列的业务目标组成。业务目标构成业务主角职责的全部。业务用例体现了需求。

经过分析与总结，参与此业务过程的业务主角有：媒介，代理商，银行。同时，也得出四个主要的业务用例为：提交素材，确认下单，查询素材进度，查询下单。其中，提交素材与查询素材进度属于素材管理，确认下单与查询下单属于下单管理。综合整个品牌广告下单系统，作出以下业务用例视图（如图2.1）。

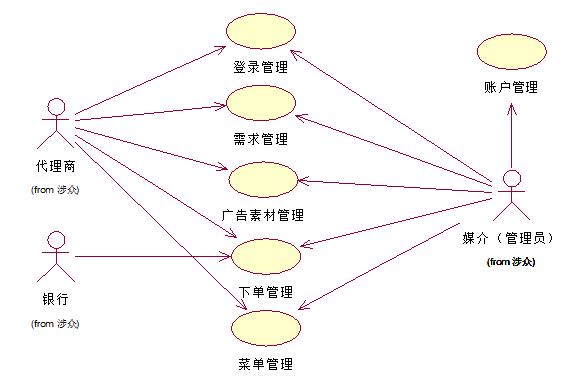


图2.1 业务用例图

## 2.3业务用例场景

以业务目标为边界的业务模型中，业务场景图描绘的是贡献于这个业务目标的什么人及其做的什么事，这些人和事的交互过程和完成顺序就是完成整个业务目标的流程。而这些人往往是业务主角，而他们所做的事是业务用例。

依照上文中分析的业务用例，分别包含广告素材管理业务、下单管理业务、菜单管理业务，以下使用活动图来绘制业务用例场景。

### 2.3.1 广告素材管理业务用例场景

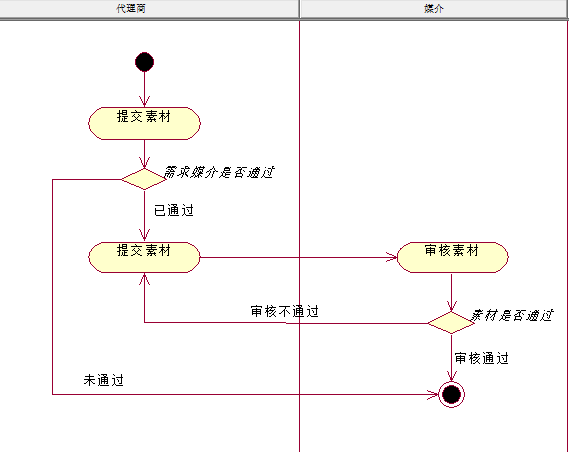


图2.2 广告素材管理业务用例场景图

### 2.3.2 下单管理业务用例场景

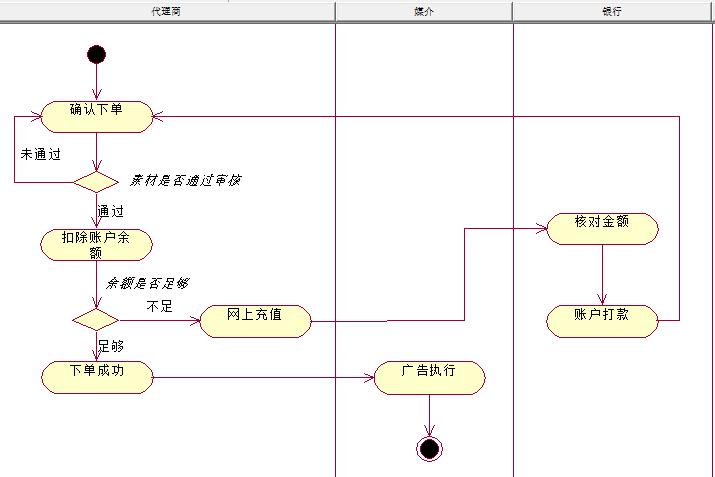


图2.3下单管理业务用例场景图

### 2.3.3 菜单管理业务用例场景

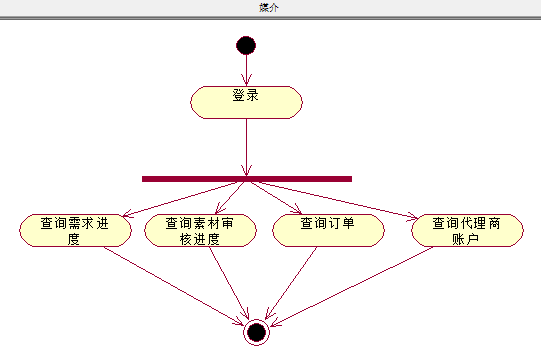


图2.4 菜单管理业务用例场景图

## 2.4业务用例规约

通过业务用例场景图已经可以得知每个业务的实际执行过程，以及参与者的实际职责，下面再通过业务用例规约对素材管理业务、下单管理业务以及菜单管理业务用例进行描述说明与规范。

表2-1 素材管理业务用例规约

|  |  |
| --- | --- |
| 用例名称 | 素材管理 |
| 用例描述 | 代理商提交广告素材，媒介进行后台审核 |
| 执行者 | 代理商、媒介 |
| 前置条件 | 需求表已通过 |
| 后置条件 | 广告下单 |
| 主过程描述 | 1.代理商按照表单提交相应的广告素材  2.媒介在后台审核代理商提交的素材  3.媒介确认素材通过 |
| 分支过程描述 | 素材不符合要求代理商需重新提交素材 |
| 异常过程描述 | 无 |
| 业务规则 | 代理商提交广告素材，媒介审核广告素材 |
| 涉及的业务实体 | 素材提交表 |

表2-2 下单管理业务用例规约

|  |  |
| --- | --- |
| 用例名称 | 下单管理 |
| 用例描述 | 代理商确认广告下单，媒介执行广告 |
| 执行者 | 代理商、媒介 |
| 前置条件 | 素材已通过审核 |
| 后置条件 | 无 |
| 主过程描述 | 1.代理商填写并确认下单信息  2.扣除代理商账户余额  3. 媒介执行广告 |
| 分支过程描述 | 1.账户余额不足，返回订单页面  2.素材审核不通过，无法下单，返回下单页面 |
| 异常过程描述 | 无 |
| 业务规则 | 代理商确认下单，媒介执行 |
| 涉及的业务实体 | 下单提交表 |

表2-3 菜单管理业务用例规约

|  |  |
| --- | --- |
| 用例名称 | 菜单管理 |
| 用例描述 | 代理商查看品牌广告的需求、素材与订单信息；媒介查看品牌广告的需求、素材、订单与代理商账户信息 |
| 执行者 | 代理商、媒介 |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 主过程描述 | 1.代理商查询需求进度、查询素材审核进度，查询订单  2.媒介查询需求进度、查询素材审核进度，查询订单 ，查询代理商账户信息 |
| 分支过程描述 | 无 |
| 异常过程描述 | 无 |
| 业务规则 | 进行查询操作 |
| 涉及的业务实体 | 需求信息表、素材提交表、下单提交表、代理商用户信息表 |

# 3 需求分析

## 3.1 系统目标

根据对现状业务当前存在问题的分析，以满足用户业务目标为基础，进行细化、具体化，最终得出本系统的开发目标。本系统针对品牌广告下单业务流程设计登录管理、用户管理、广告需求管理、广告素材管理、下单管理、菜单管理以及后台管理等七大业务功能模块。本人负责开发后四个业务板块。

### 3.1.1 广告素材管理业务

针对代理商，在需求表审批通过后，代理商提交符合规格的图片，文字，标题，视频等广告素材进行审核。

同时，广告素材管理还提供审核进度的分类查看，分别分类为“已提交”、“已通过”、“未通过”。已提交显示已经提交的广告素材表单；已通过显示媒介已经审核通过的广告素材表表单；未通过显示审核不通过的广告素材表单，原因可能有：广告素材不符合规格、视频内容不符合平台调性、视频/图片内容涉及政治敏感内容或者行业敏感字词等，此情况需要代理商重新提交广告素材表单。

针对媒介（管理员）广告素材管理，分类为“全部”、“未审核”、“已审核”。全部显示代理商提交的所有状态的广告素材表单；未审核显示还未处理审核的广告素材表单；已审批显示已经审批通过提交的广告素材表单。

### 3.1.2 下单管理业务

针对代理商，素材审核通过后，代理商可进行下单操作。

下单管理提供审核进度的分类查看，分别分类为“未下单”、“已下单”。未下单显示尚未确认的下单表单；已下单显示代理商已确认的下单表单。

针对媒介（管理员）下单管理，仅显示代理商已确认下单的表单。

### 3.1.3 菜单管理业务

针对代理商的页面，显示公司最新的品牌广告信息以及活动热点，同时也显示所有表单的最新进度。媒介（管理员）显示所有表单的最新进度，同时统计媒介人员的工作进度，待处理表单，已处理表单等工作数据。

总结整个品牌广告下单系统用例图如下：

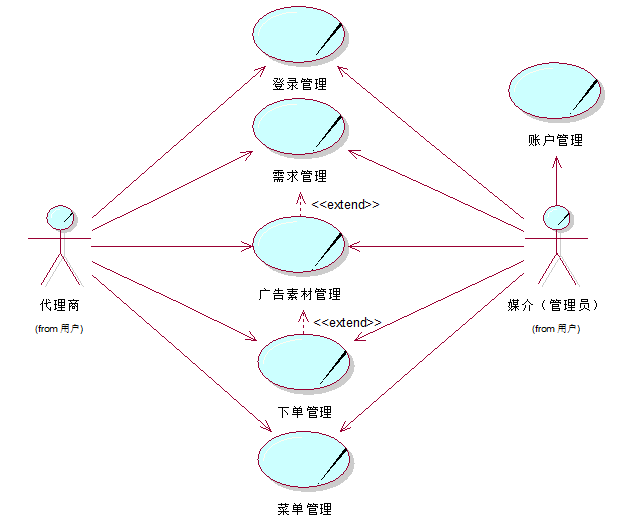


图3.1 系统用例图

下面对系统用例图中的广告素材管理，下单管理以及菜单管理的每个用例考虑它的细化工作，并为每个细化后的用例添加描述，即给出用例规约。

（1）素材管理



图3.2 素材管理用例细化

表3-1 提交素材用例的描述

|  |  |
| --- | --- |
| 用例名称 | 提交素材用例 |
| 用例标识符 | DM01 |
| 用例描述 | 代理商提交广告素材 |
| 执行者 | 代理商 |
| 前置条件 | 需求表已通过 |
| 后置条件 | 广告下单 |
| 基本流程 | 代理商按照表单提交相应的广告素材 |

表3-2审核素材用例的描述

|  |  |
| --- | --- |
| 用例名称 | 审核素材用例 |
| 用例标识符 | DM02 |
| 用例描述 | 媒介进行后台审核 |
| 执行者 | 媒介 |
| 前置条件 | 需求表已通过 |
| 后置条件 | 广告下单 |
| 基本流程 | 媒介在后台审核代理商提交的素材 |

表3-3查询素材用例的描述

|  |  |
| --- | --- |
| 用例名称 | 查询素材用例 |
| 用例标识符 | DM03 |
| 用例描述 | 代理商、媒介可以进行查询素材状态 |
| 执行者 | 媒介、代理商 |
| 前置条件 | 需求表已通过 |
| 后置条件 | 广告下单 |
| 基本流程 | 媒介、代理商查询素材审核状况 |

（2）下单管理



图3.3 下单管理用例细化

表3-4 确认下单的描述

|  |  |
| --- | --- |
| 用例名称 | 确认下单 |
| 用例标识符 | OD01 |
| 用例描述 | 代理商确认广告下单 |
| 执行者 | 代理商 |
| 前置条件 | 素材已通过审核 |
| 后置条件 | 无 |
| 基本流程 | 代理商填写并确认下单信息 |

表3-5 查询下单的描述

|  |  |
| --- | --- |
| 用例名称 | 查询下单 |
| 用例标识符 | OD02 |
| 用例描述 | 代理商、媒介查询下单 |
| 执行者 | 代理商、媒介 |
| 前置条件 | 素材已通过审核 |
| 后置条件 | 无 |
| 基本流程 | 代理商、媒介查询下单情况与信息 |

（3）菜单管理



图3.4 菜单管理用例细化

表3-6 查询代理商账户

|  |  |
| --- | --- |
| 用例名称 | 查询代理商账户 |
| 用例标识符 | DM01 |
| 用例描述 | 代理商、媒介可以进行查询代理商账户信息 |
| 执行者 | 媒介、代理商 |
| 前置条件 | 无 |
| 后置条件 | 无 |
| 基本流程 | 媒介、代理商查询代理商账户信息 |

表3-7查询需求描述

|  |  |
| --- | --- |
| 用例名称 | 查询需求 |
| 用例标识符 | DM02 |
| 用例描述 | 代理商、媒介可以进行查询需求 |
| 执行者 | 媒介、代理商 |
| 前置条件 | 需求表已提交 |
| 后置条件 | 提交素材 |
| 基本流程 | 媒介、代理商查询需求 |

表3-8查询素材用例的描述

|  |  |
| --- | --- |
| 用例名称 | 查询素材用例 |
| 用例标识符 | DM03 |
| 用例描述 | 代理商、媒介可以进行查询素材状态 |
| 执行者 | 媒介、代理商 |
| 前置条件 | 需求表已通过 |
| 后置条件 | 广告下单 |
| 基本流程 | 媒介、代理商查询素材审核状况 |

表3-9查询下单的描述

|  |  |
| --- | --- |
| 用例名称 | 查询下单 |
| 用例标识符 | MU04 |
| 用例描述 | 代理商、媒介查询下单 |
| 执行者 | 代理商、媒介 |
| 前置条件 | 已下单 |
| 后置条件 | 无 |
| 基本流程 | 代理商、媒介查询下单情况与信息 |

## 3.2 核心业务

### 3.2.1核心业务场景图

针对整个品牌广告下单。从代理商用户登录，新增需求，填写需求表然后提交需求表。媒介在后台对需求表进行审核。审核通过后，代理商提交广告素材。媒介在后台对素材进行审核。审核通过后，代理商补充下单信息，确认下单。下单成功，媒介负责执行广告。由此，得出核心业务的场景图，如下：

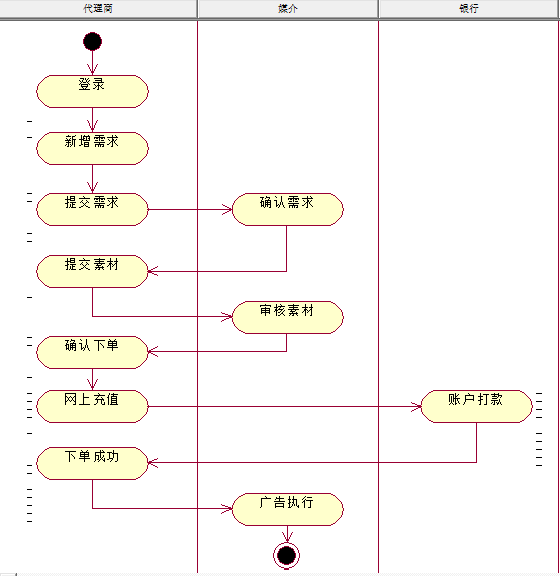
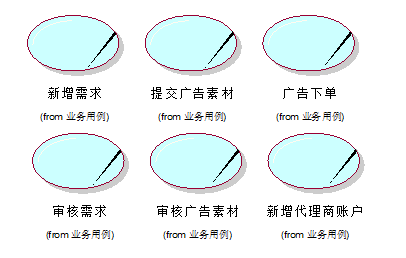
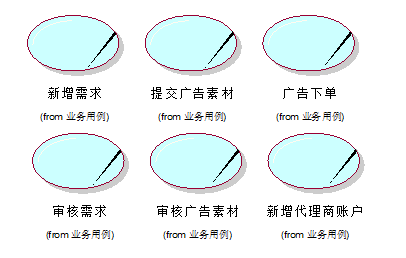


图3.5 核心业务的场景图

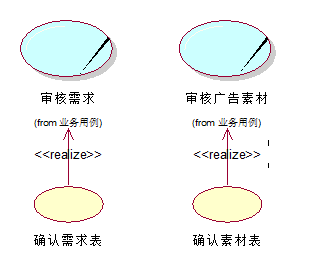
### 3.2.2 关键业务用例



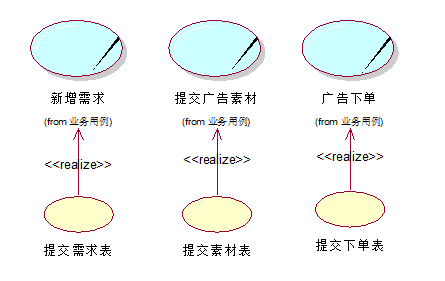
## 3.3 概念用例与场景图

### 3.3.1概念用例

（1）媒介



（2）代理商



### 3.3.2概念用例场景图

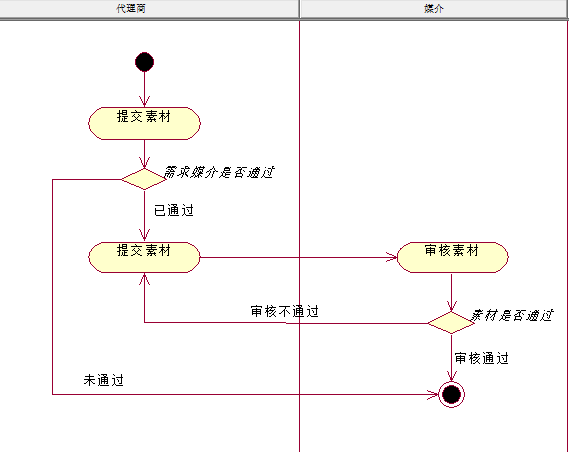


图3.6 提交素材用例场景图

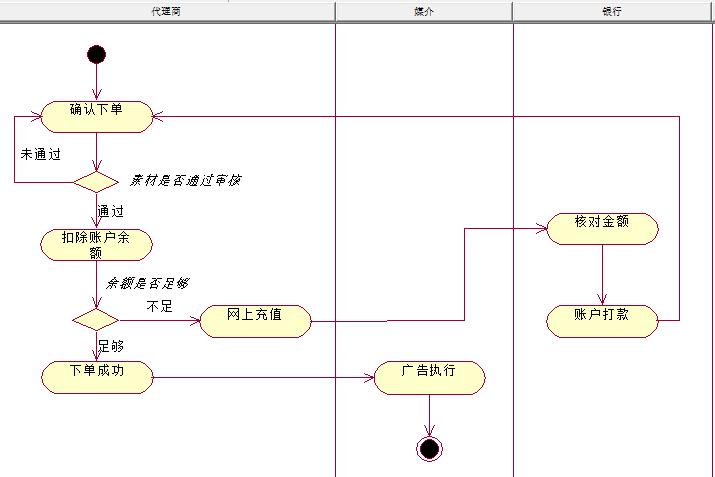


图3.7 确认下单用例场景图

# 4 系统分析

## 4.1 软件架构与框架

### 4.1.1 开发工具

选择微信小程序来实现此次系统设计的原因之一是之前课堂老师有教过，并且在课后对微信小程序也比较感兴趣，使用微信开发者工具开发过小程序，对其有一定的经验。

同时，此开发工具面对用户与开发者具有不同的优点：

（1）对于普通用户，小程序实现了应用的触手可及，只需要通过扫描二维码、搜索或者是朋友的分享就可以直接打开，加上优秀的体验，小程序使得服务提供者的触达能力变得更强。

（2）对于开发者而言，小程序框架本身所具有的快速加载和快速渲染能力，加之配套的云能力、运维能力和数据汇总能力，使得开发者不需要去处理琐碎的工作，可以把精力放置在具体的业务逻辑的开发上。

### 4.1.2 工具介绍

对于此次的系统设计，我们使用开发工具的小程序调试来对系统的具体实现。微信开发者工具有模拟器、调试器和编辑器三个窗口，具有视觉、交互、性能等方面的优势。

在编辑代码的时候，可以随时使用模拟器模拟小程序在微信客户端的表现，选择不同的设备，也可以添加自定义设备来调试小程序在不同尺寸机型上的适配问题。使用调试窗口可以查看编辑代码的错误和后台数据。

与普通网页开发的对比。小程序的主要开发语言是 JavaScript ，小程序的开发同普通的网页开发相比有很大的相似性。对于前端开发者而言，从网页开发迁移到小程序的开发成本并不高，但是二者还是有些许区别的。

网页开发需要面对的环境是各式各样的浏览器，PC 端需要面对 IE、Chrome、QQ浏览器等，在移动端需要面对Safari、Chrome以及 iOS、Android 系统中的各式 WebView 。而小程序开发过程中需要面对的是两大操作系统 iOS 和 Android 的微信客户端，以及用于辅助开发的小程序开发者工具

### 4.1.3软件架构与框架

架构分为视图层，逻辑层，组件，API几个部分。

（1）视图层负责页面结构、样式和数据展示，用wxml、wxss语言编写。

（2）逻辑层负责业务逻辑，调用API等，由js编写。

视图层和逻辑层类似MVVM模式，逻辑层只需对数据对象更新，就可以改变视图层的数据显示。

（3）组件是视图层封装好的基础组件，如按钮、输入框等。

（4）API提供了访问手机设备、网络、服务器、微信平台接口等能力。

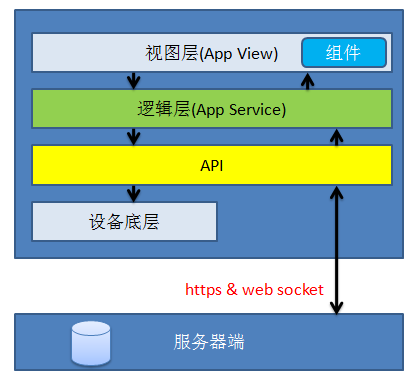


图4.1 小程序架构图

### 4.1.4 小程序安全管理

微信小程序在产品功能设计上给用户更多控制力。在小程序的设置页,为用户提供了数据权限开关,一旦用户授权之后又关闭,小程序再次使用该用户数据时需要重新获得授权,为用户提供更方便的数据控制权。用户在小程序的资料页还可以看到隐私数据保护的提示以及投诉入口。

小程序在收集、获取用户数据上,小程序坚持“必要+合理”原则。必要是指只有在小程序的具体业务中,确实有场景需要获取用户数据的情况下,开发者才能去获取用户的同意授权；合理是指开发者获取数据的范围不应该超出具体场景所需要的数据范围。例如一个提供外卖服务的小程序,可能需要获得电话、地址等数据,但没有必要获取性别、年龄等数据,否则平台会认为小程序收集用户数据的行为违反了“必要并且合理”的原则,会对这类小程序进行处理。

## 4.2确定系统用例

系统用例，系统能够为执行者提供的、涉众可以接受的价值。系统用例属于系统范围的概念。也可理解为要实现某个业务用例的系统级实现。其要建设的系统功能性需求由这些系统用例构成。根据前两章对业务进行分析与设计，此次针对品牌下单系统进行系统用例分析。

(1)媒介

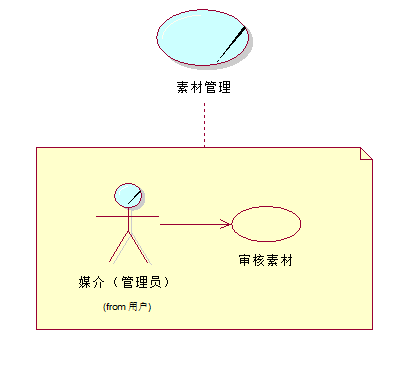


图4.2 审核素材（媒介）系统用例图

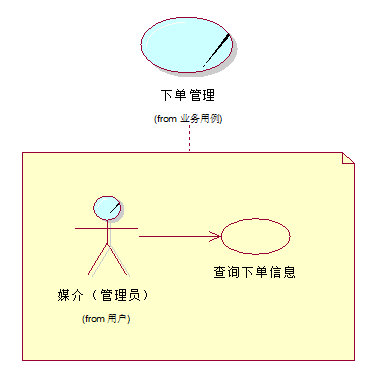


图4.3 查询下单（媒介）信息系统用例图

（2）代理商

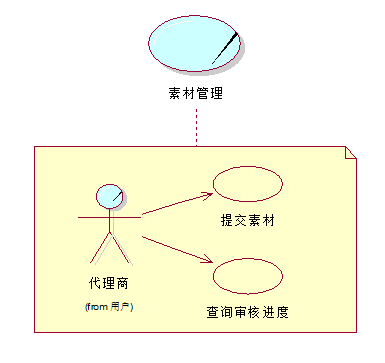


图4.4 素材管理（代理商）系统用例图

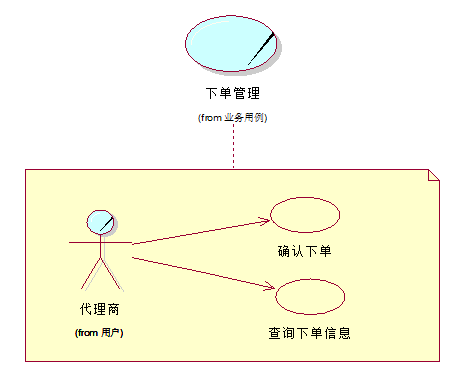


图4.5下单管理（代理商）系统用例图

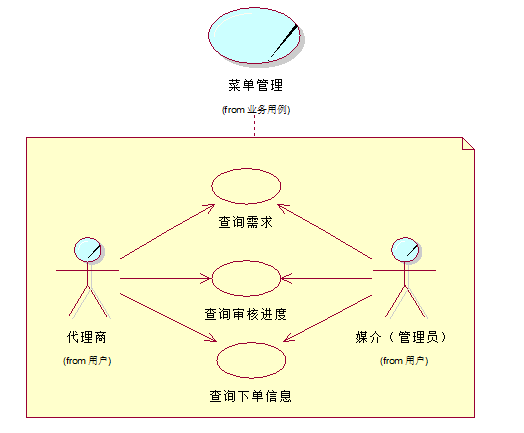


图4.6菜单管理系统用例图

## 4.3交互顺序图

交互顺序图是描述对象之间的关系以及对象之间的信息传递的图

通过对业务流程的分析，确定以下菜单管理的查询需求对象交互顺序图、查询素材审核对象交互顺序图与查询下单对象交互顺序图，素材管理的提交素材对象交互顺序图、素材审核对象交互顺序图，下单管理的下单确认对象交互顺序图。

### 4.3.1 菜单管理

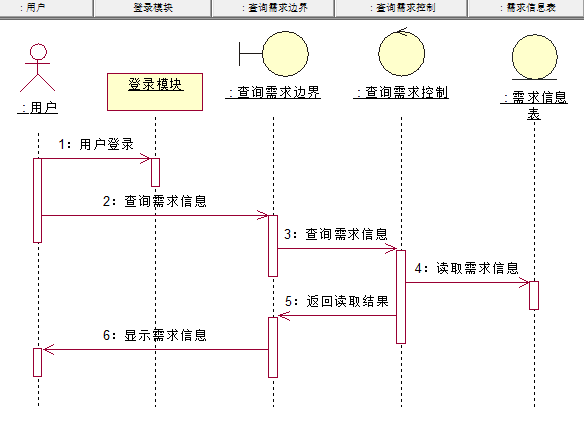


图4.7查询需求对象交互顺序图

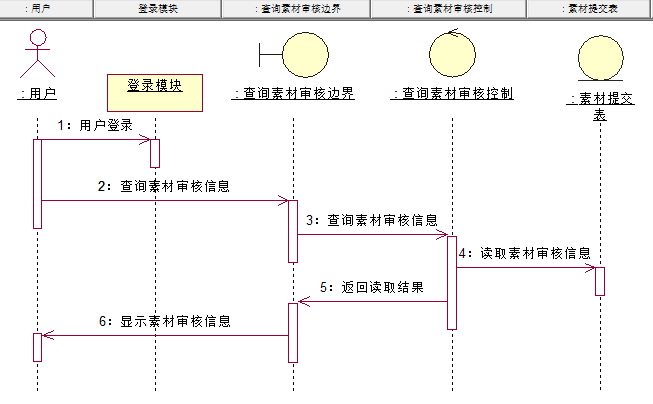


图4.8 查询素材审核对象交互顺序图

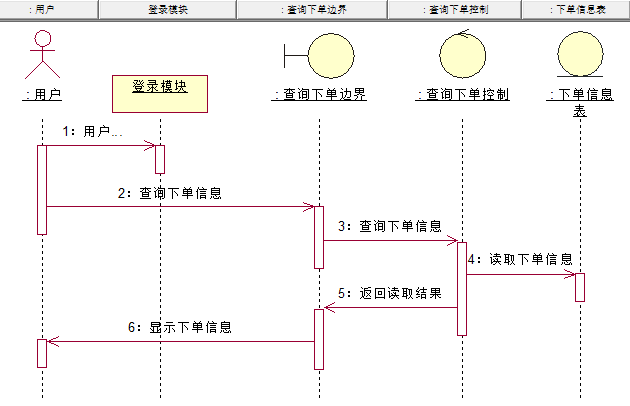


图4.9查询下单对象交互顺序图

### 4.3.2 素材管理

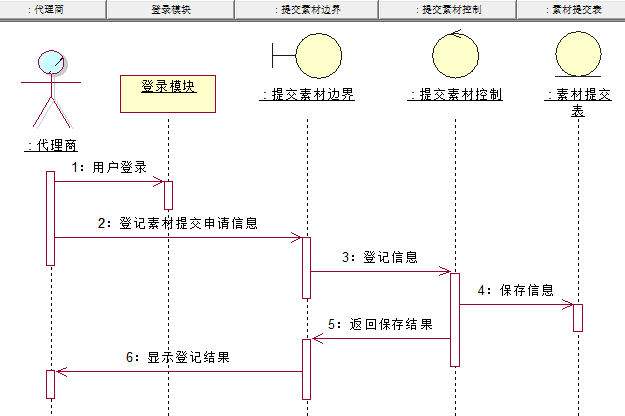


图4.10提交素材对象交互顺序图



图4.11 素材审核对象交互顺序图

### 4.3.3 下单管理

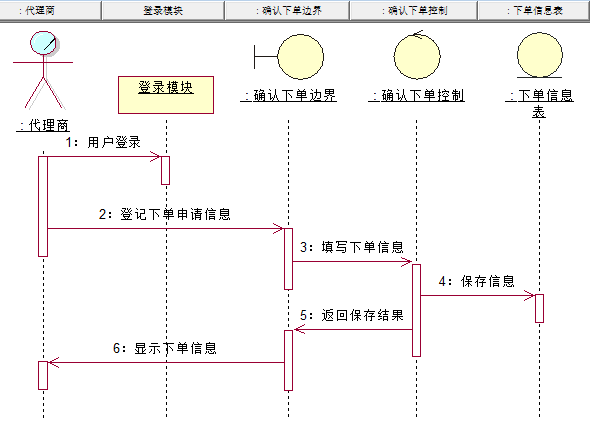


图4.12下单确认对象交互顺序图

## 4.4 分析类图

类图(Class diagram)是显示了模型的静态结构，特别是模型中存在的类、类的内部结构以及它们与其他类的关系等。类图(Class diagram)是最常用的UML图，显示出类、接口以及它们之间的静态结构和关系，它用于描述系统的结构化设计。

通过以上对象间交互的分析，可以明白新系统将如何使用这些对象来实现用例。那么，根据这些对象的特征，以及调查中的数据分析，可以初步得出相关的类。其中，本系统定义了如下实体类：

|  |  |  |
| --- | --- | --- |
| 序号 | 类名 | 中文含义 |
| 4 | Demo\_Info | 素材信息表 |
| 5 | Order\_Info | 下单信息表 |

# 5 系统设计

## 5.1数据库的详细设计

数据库是“按照数据结构来组织、存储和管理数据的仓库”。是一个长期存储在计算机内的、有组织的、有共享的、统一管理的数据集合。数据库是以一定方式储存在一起、能与多个用户共享、具有尽可能小的冗余度、与应用程序彼此独立的数据集合，可视为电子化的文件柜，存储电子文件的处所，用户可以对文件中的数据进行新增、查询、更新、删除等操作。

根据对业务场景的分析，收集与总结系统所需要建立的数据的表单信息，因此构建一下数据库的素材提交表格，下单提交表格，同时结合其他子系统所需要的需求提交表格，代理商信息表，媒介用户信息表。以下是设计的两个数据库表格，如下表：

表5-1 素材提交表格

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 说明 | 可见性 | 数据类型 | 描述 |
| demoID | 素材单编号 | private | varchar | （主键） |
| managerID | 管理员账号 | private | char | （外键） |
| userID | 代理商账号 | private | char | （外键） |
| screen-style | 开屏样式 | 选“开屏广告”可见 | char | “静态”，“动态，“视频”” |
| feed-style | 信息流样式 | 选“今日头条-信息流广告”可见 | char | “小图”，“大图”，“组图”，“视频” |
| 选“抖音-信息流广告”可见 | char | 仅“视频” |
| link | 落地页链接/文章链接 | private | varchar |  |
| source | 广告来源 | private | varchar |  |
| title | 广告标题 | private | varchar |  |
| describe | 广告描述 | private | varchar |  |
| screen-photo | 素材 | 选“静态”可见 | 图片 |  |
| screen-gif | 选“动态”可见 | GIF |  |
| video | 选“视频”可见 | 视频 |  |
| Demo-state | 素材审核状态 | private | char | “未审核”，“审核通过”“未通过” |
| Submitdemo() | 提交需求 |  | | |
| Enquirydemo() | 查询需求 |  | | |

表5-2 下单提交表格

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 字段名 | 说明 | 可见性 | 数据类型 | 描述 |
| orderID | 下单编号 | private | varchar | （主键） |
| managerID | 管理员账号 | private | char | （外键） |
| userID | 代理商账号 | private | char | （外键） |
| managerName | 渠道经理姓名 | private | varchar |  |
| total | 投放金额 | private | varchar |  |
| Payment | 付款方式 | private | char | “网上银行”，“现金结算”，“账户余额” |
| AccountPeriod | 账期 | private | char | “无”， “10个工作日”，“15个工作日”，“20个工作日” |
| order-state | 下单处理状态 | private | char | “已提交”，“已处理”，“撤改单” |
| Submitorder() | 提交下单表 |  | | |
| Enquiryorder() | 查询下单表 |  | | |

通过以上对类的设计与规范，进一步明确了系统实体类的属性与操作，结合其他子系统所建表格，从而得到以下品牌广告下单系统实体类图：

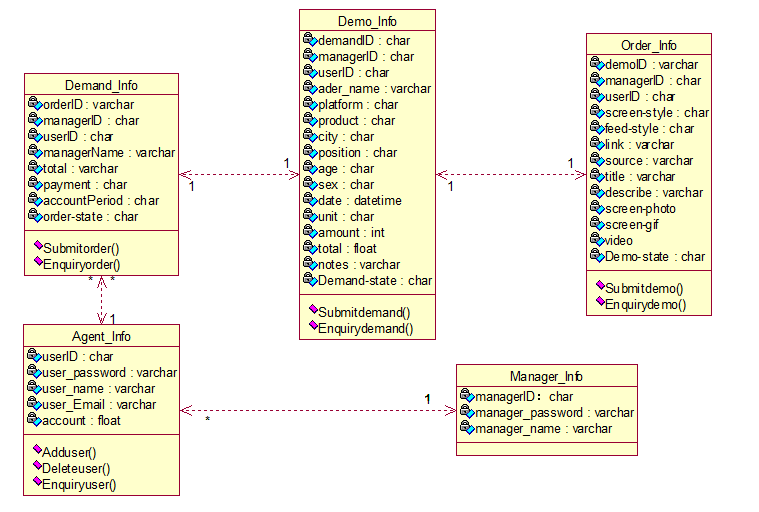


图5.1 系统实体类图

## 5.2包设计

包图是用来描述模型中的包和所包含元素的组织方式的图，是维护和控制系统总体结构的重要内容。包图能够组织许多UML中的元素，不过其最常用的用途是用来组织用例图和类图。包图中包含包元素以及包之间的关系。 与其他图类似，包图中可以创建注解和约束。

按职能集中分包的原则，对本系统进行包设计，以下为实际模型图：

### 5.2.1 素材管理

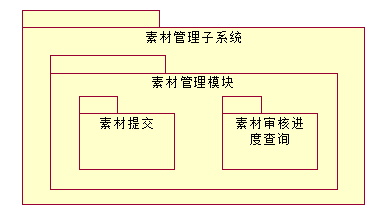


图5.2 素材管理包设计图

### 5.2.2 下单管理

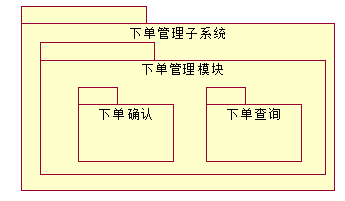


图5.3下单管理包设计图

### 5.2.3 菜单管理

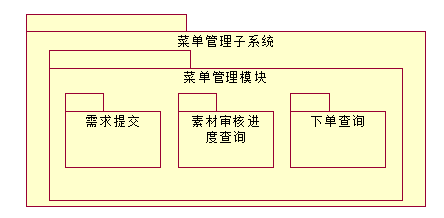


图5.4 菜单管理包设计图

## 5.3 详细设计

品牌广告投放管理系统共有六个管理业务模块，包括登录管理，账户管理，需求管理，素材管理，下单管理和菜单管理模块，以下是系统的总览图。本报告只展示其中的素材管理，下单管理和菜单管理模块。

（1）登录管理

登录模块有主要功能为验证用户身份，分为管理员账号与广告主账号。验证为媒介（管理员），拥有管理用户权限，可以新增、删除、查询代理商的账号。同时，确认与审批代理商提交的广告需求，审核提交的广告素材，查看代理商已确认的下单信息进行广告执行等。

验证为代理商，拥有需求创建与提交、素材提交、下单等操作权限。

（2）用户管理

用户管理是媒介（管理者）管理代理商账号，可以增加、删除与查询代理商用户的账号，同时也显示各广告主账号品牌广告下单业务进度的最新的状态。

（3）需求管理

针对代理商，需求管理是提供给广告主新增与提交广告需求，需要填写需求表单，填写相关信息：广告主选择广告投放APP平台、广告位置、定向、转化组件、投放日期、投放数量，查询与预定库存，并且确认投放金额等，填写完成并提交需求。

同时，需求管理还提供需求审批的分类查看，分别分类为“已提交”、“已通过”、“未通过”。已提交显示已经提交的需求表表单；已通过显示媒介已经审核通过的需求表表单；未通过显示审核不通过的需求表单，原因可能有：需求不明确，广告库存不足等，此情况需要代理商重新提交需求表单。

针对媒介（管理员），需求管理，分类为“全部”、“未审批”、“已审批”。全部显示代理商提交的所有状态的需求表单；未审核显示还未处理审核的需求表单；已审批显示已经审批通过提交的需求表表单。

（4）广告素材管理

广告素材是不同形式和格式内容的物理表示，不同用途的素材需要使用大量的用于不同目的的素材格式。广告素材管理并不仅是一个存储文件的仓库，它还必须理解文件结构、媒体语法、甚至特定部分的语义；提供媒体索引、自动信息检索等功能和服务。

内容管理系统还可以通过自动检索来获得内容的附加信息。

（5）下单管理

下单管理是后台系统中较为重要的一部分，它记录了所有的交易数据，下单管理后台的操作用户都是公司内部人员，但同时也需要C端用户的需求。所以在设计时，订单管理系统会包括两部分的内容：一是要能够与C端用户在整个订单流程中各个场景的操作相对应；另一个是要能满足公司内部相关部门的需求。

(6)菜单管理

菜单管理是指整体的需求管理、素材管理与下单管理的整体的菜单与进度的显示情况。可以通过菜单管理系统跟进同一个订单的不同情况或者同个进度的所有表单的情况，都进行可视化的菜单式管理。

图5.5品牌广告投放管理系统功能总览图

### 5.3.1 素材管理模块

（1）功能描述

功能类型：素材管理

功能描述：素材提交与素材审核进度查询

前提业务：需求表审核通过后

后继业务：系统功能操作

功能约束：权限约束

操作权限：代理商

（2）界面设计

****

 图5.6 上传素材页面

主要代码：

<form bindsubmit='submitForm'>

<view class='row box'>

<label>开屏样式：</label>

<picker name='screen' range='{{screenItems}}' bindchange='screenChange' value='{{screen}}'>

<view>{{screen}}</view>

</picker>

</view>

<view class='row box'>

<label>信息流样式：</label>

<picker name='feed' range='{{feedItems}}' bindchange='feedChange' value='{{feed}}'>

<view>{{feed}}</view>

</picker>

</view>

<view class='col box'>

<label>落地页链接/文章链接：</label>

<textarea name='link' auto-height value='{{link}}'></textarea>

</view>

<view class='col box'>

<label>广告来源：</label>

<textarea name='source' auto-height value='{{source}}'></textarea>

</view>

<view class='col box'>

<label>广告标题：</label>

<textarea name='title' auto-height value='{{title}}'></textarea>

</view>

<view class='col box'>

<label>广告描述：</label>

<textarea name='describe' auto-height value='{{describe}}'></textarea>

</view>

<view class='col box'>

<view class="weui-cells">

<view class="weui-cell">

<view class="weui-cell\_\_bd">

<view class="weui-uploader">

<view class="weui-uploader\_\_hd">

<view class="weui-uploader\_\_overview">

<view class="weui-uploader\_\_title">图片上传</view>

<view class="weui-uploader\_\_info">{{files.length}}/9</view>

</view>

<view class="weui-uploader\_\_tips">

上传提示

</view>

</view>

<view class="weui-uploader\_\_bd">

<view class="weui-uploader\_\_files" id="uploaderFiles">

<block wx:for="{{files}}" wx:key="\*this">

<view class="weui-uploader\_\_file" bindtap="previewImage" id="{{item}}">

<image class="weui-uploader\_\_img" src="{{item}}" mode="aspectFill" />

</view>

</block>

</view>

<view class="weui-uploader\_\_input-box">

<view class="weui-uploader\_\_input" bindtap="chooseImage"></view>

</view>

</view>

</view>

</view>

</view>

</view>

</view>

<button form-type='submit'>提交</button>

</form>



图5.7 素材审核页面

主要代码：

<view>

<view class="weui-search-bar">

<view class="weui-search-bar\_\_form">

<view class="weui-search-bar\_\_box">

<icon class="weui-icon-search\_in-box" type="search" size="14"></icon>

<input type="text" class="weui-search-bar\_\_input" placeholder="搜索" value="{{inputVal}}" focus="{{inputShowed}}" bindinput="inputTyping" />

<view class="weui-icon-clear" wx:if="{{inputVal.length > 0}}" bindtap="clearInput">

<icon type="clear" size="14"></icon>

</view>

</view>

<label class="weui-search-bar\_\_label" hidden="{{inputShowed}}" bindtap="showInput">

<icon class="weui-icon-search" type="search" size="14"></icon>

<view class="weui-search-bar\_\_text">搜索</view>

</label>

</view>

<view class="weui-search-bar\_\_cancel-btn" hidden="{{!inputShowed}}" bindtap="hideInput">取消</view>

</view>

<!-- 导航栏 -->

<view class="nav-scroll">

<text wx:for="{{section}}" wx:key="id" id="{{item.id}}"

class="nav-name {{item.id == currentId ? 'nav-hover' : ''}}"

catchtap="handleTap" >{{item.name}}</text>

</view>

<block wx:if="{{currentId == 1001}}">

<view class="col box" wx:for='{{demolist\_submitted}}' wx:key="unique">

<text>待处理{{index+1}}\n</text>

<text>对应需求编号:{{item.demandID}}\n</text>

<text>代理商:{{item.username}}</text>

<form bindsubmit="lookDemo">

<textarea class="content4" hidden="ture" name='indextest' auto-height value='{{index}}'></textarea>

<button class="button4" form-type='submit'>查看</button></form>

</view>

</block>

<block wx:if="{{currentId == 1002}}">

<view class="col box" wx:for='{{demolist\_passed}}' wx:key="unique">

<text>待处理{{index+1}}\n</text>

<text>对应需求编号:{{item.demandID}}\n</text>

<text>代理商:{{item.username}}</text>

</view>

</block>

<block wx:if="{{currentId == 1003}}">

<view class="col box" wx:for='{{demolist\_failed}}' wx:key="unique">

<text>待处理{{index+1}}\n</text>

<text>对应需求编号:{{item.demandID}}\n</text>

<text>代理商:{{item.username}}</text>

</view>

</block>

</view>

（3）逻辑层实现代码

 1、提交素材

const db = wx.cloud.database();

db.collection('material').doc(demo.\_id).update({

data: {

screen: screen,

feed: feed,

link: link,

source: source,

title: title,

describe: describe,

state: 'submitted',

},

success: function (res) {

// res 是一个对象，其中有 \_id 字段标记刚创建的记录的 id

// console.log(res)

wx.showToast({

title: '提交成功！！！',

icon: 'success',

duration: 2000

})

}

})

2、素材审核实现代码

db.collection('material').doc(lookDemo.\_id).update({

data: {

state: state

},

success: function (res) {

wx.showToast({

title: '审核完成！',

icon: 'success',

duration: 2000

})

}

})

wx.reLaunch({

url: '../demo\_manage/demo\_manage'

})

},

（4）数据库实现

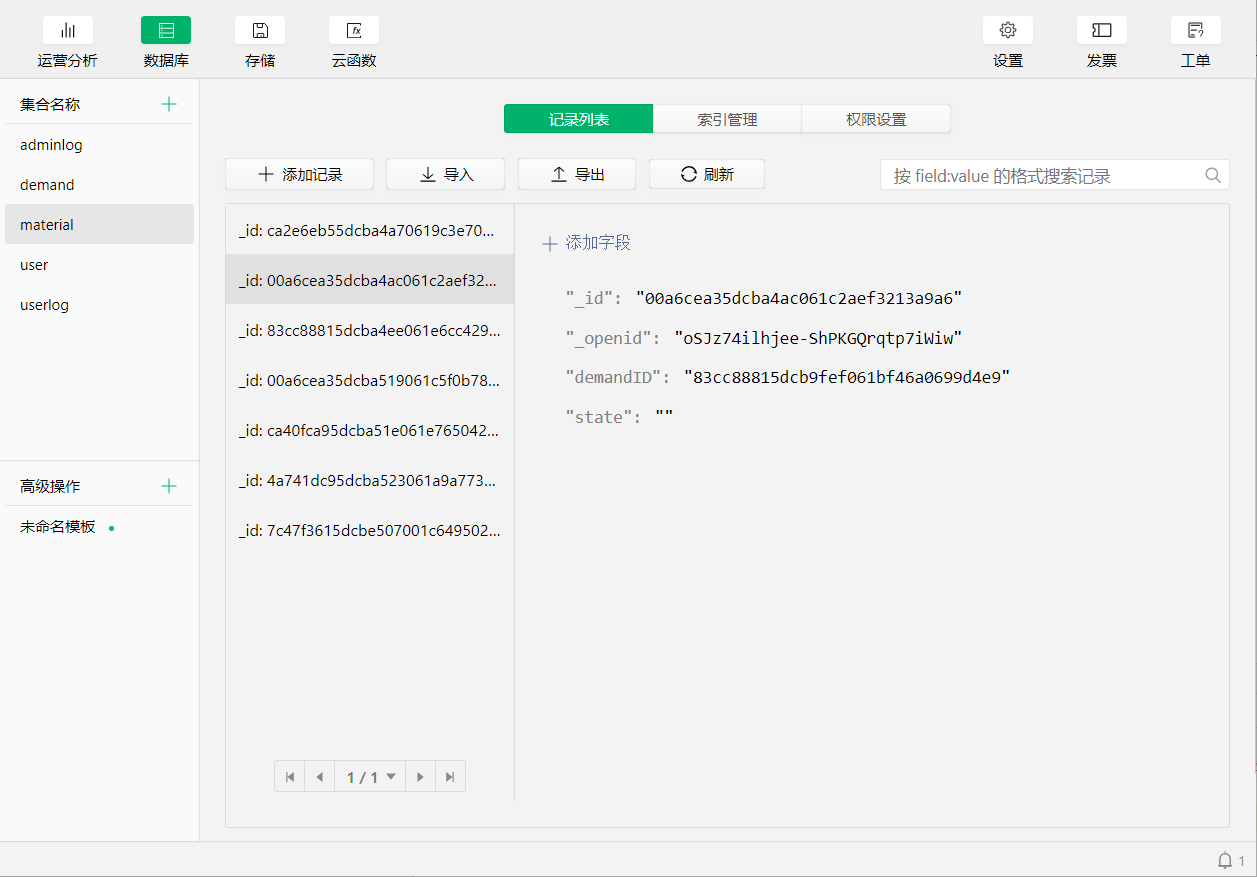


图5.8 素材数据库图

### 5.3.2 下单管理模块

（1）功能描述

功能类型：下单管理

功能描述：提交下单信息，查询下单历史信息

前提业务：素材已通过审核

后继业务：系统功能操作

功能约束：权限约束

操作权限：代理商

（2）界面设计



图5.9下单页面

主要代码：

<view>

<view class="weui-search-bar">

<view class="weui-search-bar\_\_form">

<view class="weui-search-bar\_\_box">

<icon class="weui-icon-search\_in-box" type="search" size="14"></icon>

<input type="text" class="weui-search-bar\_\_input" placeholder="搜索" value="{{inputVal}}" focus="{{inputShowed}}" bindinput="inputTyping" />

<view class="weui-icon-clear" wx:if="{{inputVal.length > 0}}" bindtap="clearInput">

<icon type="clear" size="14"></icon>

</view>

</view>

<label class="weui-search-bar\_\_label" hidden="{{inputShowed}}" bindtap="showInput">

<icon class="weui-icon-search" type="search" size="14"></icon>

<view class="weui-search-bar\_\_text">搜索</view>

</label>

</view>

<view class="weui-search-bar\_\_cancel-btn" hidden="{{!inputShowed}}" bindtap="hideInput">取消</view>

</view>

<!-- 导航栏 -->

<view class="nav-scroll">

<text wx:for="{{section}}" wx:key="id" id="{{item.id}}"

class="nav-name {{item.id == currentId ? 'nav-hover' : ''}}"

catchtap="handleTap" >{{item.name}}</text>

</view>

<block wx:if="{{currentId == 1001}}">

<view class="col box" wx:for='{{orderlist\_submitted}}' wx:key="unique">

<text>对应需求编号：{{item.demandID}}\n</text>

<text>广告主名称：{{item.ader\_name}}\n</text>

<text>投放平台：{{item.platform}}\n</text>

<text>投放时间：{{item.date}}\n</text>

<form bindsubmit="addOrder">

<textarea hidden="ture" name='indextest' auto-height value='{{index}}'></textarea>

<button form-type='submit'>下单</button></form>

</view>

</block>

<block wx:if="{{currentId == 1002}}">

<view class="col box" wx:for='{{ orderlist\_passed}}' wx:key="unique">

<text>对应需求编号：{{item.demandID}}\n</text>

<text>广告主名称：{{item.ader\_name}}\n</text>

<text>投放平台：{{item.platform}}\n</text>

<text>投放时间：{{item.date}}\n</text>

</view>

</block>

</view>



 图5.10 下单表单页面

主要代码：

<view>

<text class='row box'>广告主名称：{{lookOrder.ader\_name}}</text>

<text class='row box'>投放平台：{{lookOrder.platform}}</text>

<text class='row box'>产品名称：{{looOrder.product}}</text>

<text class='row box'>投放城市：{{lookOrder.city}}</text>

<text class='row box'>广告位置：{{lookOrder.position}}</text>

<text class='row box'>定向年龄人群：{{lookOrder.age}}</text>

<text class='row box'>定向性别：{{lookOrder.sex}}</text>

<text class='row box'>投放时间：{{lookOrder.date}}</text>

<text class='row box'>购买方式：{{lookOrder.unit}}</text>

<text class='row box'>购买单位：{{lookOrder.amount}}</text>

<text class='row box'>投放总额：{{lookOrder.total}}</text>

<text class='row box'>开屏样式：{{lookOrder.screen}}</text>

<text class='row box'>信息流样式：{{lookOrder.feed}}</text>

<text class='row box'>落地页链接/文章链接：{{looOrder.link}}</text>

<text class='row box'>广告来源：{{lookOrder.source}}</text>

<text class='row box'>广告标题：{{lookOrder.title}}</text>

<text class='row box'>广告描述：{{lookOrder.describe}}</text>

</view>

<view>

<form bindsubmit="stateOrder">

<textarea hidden="ture" name='state' auto-height value='passed'></textarea>

<button class="button-blue" hover-class='button-white' form-type='submit'>确认下单</button>

</form>

</view>

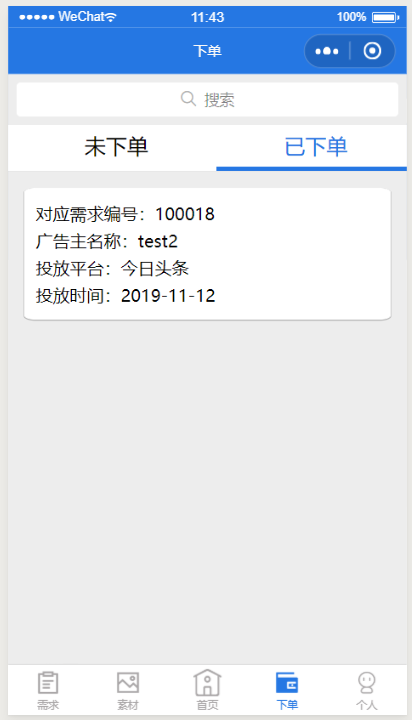


图5.11 下单显示页面

（3）逻辑层实现代码

1、提交下单代码

//点击下单

stateOrder: function (e) {

let state = e.detail.value.state

console.log(state)

let lookOrder = wx.getStorageSync('ORDER')

const db = wx.cloud.database();

db.collection('order').doc(lookOrder.\_id).update({

data: {

state: state

},

success: function (res) {

wx.showToast({

title: '下单成功！',

icon: 'success',

duration: 2000

})

}

})

},

2、获取下单信息代码

onLoad: function (options) {

console.log('test')

// 获取需求表所有信息

const db = wx.cloud.database();

const order = db.collection('order');

order.get({

success: (res) => {

let order = res.data;

console.log(order)

for (let i = 0; i < order.length; i++) {

if ('' == order[i].state) {

orderdict\_submitted = order[i];

orderlist\_submitted.push(orderdict\_submitted);

this.setData({

orderlist\_submitted: orderlist\_submitted

})

}else {

orderdict\_failed = order[i];

orderlist\_failed.push(orderdict\_failed);

this.setData({

orderlist\_failed: orderlist\_failed

})

}

}

}

（4）数据库代码与实现

**//**点击选择

stateDemo: function (e) {

let state = e.detail.value.state

console.log(state)

let lookDemo = wx.getStorageSync('lookdemo')

// 如果在素材点通过，则添加下单表

const db = wx.cloud.database();

if (state == 'passed') {

console.log(lookDemo.\_id)

db.collection('order').add({

data: {

demandID: lookDemo.demandID,

userID: lookDemo.userID,

username: lookDemo.username,

ader\_name: lookDemo.ader\_name,

age: lookDemo.age,

amount: lookDemo.amount,

city: lookDemo.city,

date: lookDemo.date,

describe: lookDemo.describe,

feed: lookDemo.feed,

link: lookDemo.link,

sex: lookDemo.sex,

platform: lookDemo.platform,

product: lookDemo.product,

screen: lookDemo.screen,

source: lookDemo.source,

title: lookDemo.title,

total: lookDemo.total,

unit: lookDemo.unit,

state: ''

}

})

}

### 5.3.3 菜单管理模块

（1）功能描述

功能类型：菜单管理

功能描述：查询与显示需求、素材、下单信息

前提业务：无

后继业务：系统功能操作

功能约束：权限约束

操作权限：媒介、代理商

（2）界面设计



图5.12 代理商菜单显示页面

主要代码：

<view>

<!-- 轮播图 -->

<view class="slider-wrapper">

<swiper indicator-dots="{{indicatorDots}}" autoplay="{{autoplay}}"

interval="{{interval}}" duration="{{duration}}" indicator-active-color="#2577e3">

<block wx:for="{{imgUrls}}" wx:key="unique">

<swiper-item hover-class="navigator-hover">

<image src="{{item.url}}" class="slide-image" width="355" height="150" />

</swiper-item>

</block>

</swiper>

</view>

<view class="content1">

<view class="left">

<image class="tz" src="/images/tz.jpg"></image>

<text decode="{{true}}" class="tzt">&ensp;&ensp;&ensp;最新消息\n</text>

</view>

<view class="content2">

<text class="title1">需求表消息</text>

<view wx:for='{{demandlist}}' wx:key="unique">

<view wx:if="{{item.demandID}}" class="content3">

<text>需求单编号：{{item.demandID}}\n</text>

<text>状态：{{item.state}}\n</text>

</view>

</view>

</view>

<view class="content2">

<text class="title1">素材表消息</text>

<view wx:for='{{demolist}}' wx:key="unique">

<view wx:if="{{item.demandID}}" class="content3">

<text>对应需求：{{item.demandID}}\n</text>

<text>状态：{{item.state}}\n</text>

</view>

</view>

</view>

<view class="content2">

<text class="title1">下单表消息</text>

<view wx:for='{{orderlist}}' wx:key="unique">

<view wx:if="{{item.demandID}}" class="content3">

<text>对应需求：{{item.demandID}}\n</text>

<text>状态：{{item.state}}\n</text>

</view>

</view>

</view>

</view>

</view>



图5.13 媒介菜单显示页面

主要代码：

<view>

<!-- 轮播图 -->

<view class="slider-wrapper">

<swiper indicator-dots="{{indicatorDots}}" autoplay="{{autoplay}}"

interval="{{interval}}" duration="{{duration}}" indicator-active-color="#2577e3">

<block wx:for="{{imgUrls}}" wx:key="unique">

<swiper-item hover-class="navigator-hover">

<image src="{{item.url}}" class="slide-image" width="355" height="150" />

</swiper-item>

</block>

</swiper>

</view>

<view class="libin">

<view class="item" catchtap="toDemand">

<image class="imgfenbu" src="/images/demand.jpg"></image>

<text class='ltext'>需求管理</text>

</view>

<view class="item" catchtap="toDemo">

<image class="imgfenbu" src="/images/demo.jpg"></image>

<text class='ltext'>素材管理</text>

</view>

<view class="item" catchtap="toOrder">

<image class="imgfenbu" src="/images/order.jpg"></image>

<text class='ltext'>下单管理</text>

</view>

<view class="item" catchtap="toUser">

<image class="imgfenbu" src="/images/user.jpg"></image>

<text class='ltext'>用户管理</text>

</view>

</view>

<view class='tuichu' bindtap='exit'>

<text class='right'>退出： >>></text>

</view>

</view>



图5.14 个人中心页面

主要代码：</view>

<view class='item'>

<text class='left'>账号：</text><text class='right'>{{user.username}}</text>

</view>

<view class='item'>

<text class='left'>代理商：</text><text class='right'>{{user.name}}</text>

</view>

<view class='item'>

<text class='left'> 邮箱：</text><text class='right'>{{user.newemail}}</text>

</view>

<view class='item' bindtap='exit'>

<text class='left'>退出：</text><text class='right'>>>></text>

</view>

</view>

（3）逻辑层实现代码

1、获取代理商菜单页面所需信息代码

// 获取云数据库中的需求表数据

const db = wx.cloud.database();

const material = db.collection('material');

const order = db.collection('order');

console.log(demolist)

console.log(user.\_id)

db.collection('demand').where({ userID: user.\_id }).get({

success: (res) => {

let demand = res.data;

demandlist.push(demand[demand.length - 1]);

demandlist.push(demand[demand.length - 2]);

demandlist.push(demand[demand.length - 3]);

this.setData({

demandlist: demandlist

})

}

})

material.where({ userID: user.\_id }).get({

success: (res) => {

let material = res.data;

demolist.push(material[material.length - 1]);

demolist.push(material[material.length - 2]);

demolist.push(material[material.length - 3]);

this.setData({

demolist: demolist

})

}

})

order.where({ userID: user.\_id }).get({

success: (res) => {

let order = res.data;

orderlist.push(order[order.length - 1]);

orderlist.push(order[order.length - 2]);

orderlist.push(order[order.length - 3]);

// console.log(orderlist)

this.setData({

orderlist: orderlist

})

}

})

2、获取媒介菜单页面信息代码：

toUser: function (e) {

wx.navigateTo({

url: '../usermanage/usermanage'

})

},

toDemand: function (e) {

wx.navigateTo({

url: '../demand\_manage/demand\_manage'

})

},

toDemo: function (e) {

wx.navigateTo({

url: '../demo\_manage/demo\_manage'

})

},

toOrder: function (e) {

wx.navigateTo({

url: '../order\_manage/order\_manage'

})

},

exit: function (e) {

wx.reLaunch({

url: '/pages/login/login'

})

# 6 心得体会

做系统设计，全局观很重要，当一个问题条件很多，我无从下手时，要保持理性，从全局观察，找到突破点。 要先有一个总体的架构，明确自己最终要做什么出来，在此基础上去具体实现。

总体的框架做好后，需要对问题的细节方面进行探讨，考虑问题要全面，要把一些概率不大但可能出现的问题考虑进去，尽量做到全面，完美。

如果细节方面处理的不好，很可能在某些情况下造成系统崩溃，所以要在细节方面做好。

  要有自己的思想，想法，一个好的系统要有自己的独到之处，体现出设计者的灵魂。否则就没有任何亮点。一步一步来，一个功能一个功能的实现，一个功能一个功能的调试，步步为营。如果全部堆砌完再去调试，那工作量将是无法想象的，所以要写一个功能调试一个功能。

# 参考文献

[1]大象 Thinking in UML. 谭云杰著.中国水利水电出版社.2009.

# 附录

**测试账号：**

管理员账号：123456

管理员密码：123456

*代理商账号可在管理员后台中用户管理注册一个新的代理商账号，或直接使用以下测试账号登录。*

代理商账号：456

代理商密码：456

