

用户名称：武汉爱约拍科技有限公司

武汉弗特劳尔信息技术有限公司

密级：普通

# Photolover小程序建设项目

## 详细设计说明书

(V1.0)

文档编号：		项目名称：	Photolover小程序建设项目
编写：		编写日期：	2022-7-27
审核：		审核日期：	
批准：		批准日期：	

武汉弗特劳尔信息技术有限公司

**--小组分工记录表--**

阶段	截止日期	负责人	描述
初期准备	2022. 6. 20	张艺凡、赵颖、 魏千懿、陈晨	软件需求说明书 数据库设计
基本搭建	2022. 7. 15	张艺凡	网页端页面搭建
		赵颖、魏千懿	小程序端页面搭建
		陈晨	后端服务架构搭建
前后端交互	2022. 7. 23	张艺凡、赵颖	网页端和小程序端传参交互
		陈晨	生成接口服务
整合测试	2022. 7. 26	张艺凡、赵颖、 陈晨	业务逻辑与功能整合汇总 详细设计说明书 用户操作手册

# 目录

1. 引言 .....	5
1.1 编写目的 .....	5
1.2 背景 .....	5
1.3 术语和缩略语 .....	6
1.4 参考资料 .....	6
1.5 约定 .....	6
2. 架构设计 .....	7
2.1 整体架构 .....	7
2.2 前端架构 .....	8
2.2.1 小程序端 .....	8
2.2.2 网页管理端 .....	9
2.3 后端架构 .....	10
3. 程序设计 .....	11
3.1 模块结构设计 .....	11
3.1.1 角色模块结构设计 .....	11
3.1.2 核心业务流模块详细设计 .....	13
3.2 程序详细结构设计 .....	14
3.2.1 服务设计 .....	14
3.2.2 接口设计 .....	15
4. 数据设计 .....	23
4.1 数据常量和变量设计 .....	23
表 1 Photographers 摄影师基本信息表 .....	23
表 2 Customers 顾客基本信息表 .....	24
表 3 Works 作品基本信息表 .....	25
表 4 Likes 作品收藏信息表 .....	26
表 5 Requests 需求基本信息表 .....	26
表 6 Orders 订单基本信息表 .....	28

表 7 Bills 发票基本信息表 .....	29
表 8 Administrators 管理员基本信息表 .....	29
表 9 Reviews 订单评价信息表 .....	30
表 10 Back 用户系统反馈表 .....	30
4.2 存储配置设计 .....	31

# 1. 引言

在自媒体的时代下，“互联网+”作为一种新的社会形态，能充分发挥互联网在社会资源配置中的优化和集成作用。照片是记录生活的一种方式，在摄影颇为流行的今天，“互联网+摄影”的商业模式将分散的摄影资源整合，为消费者提供一个完整的服务平台，将顾客从线上引流到线下。这样的商业模式近年来在我国急剧兴盛，“互连网+摄影”的产业莫事日渐成熟。

## 1.1 编写目的

本详细设计说明书是针对爱约拍（PhotoLover）小程序的项目编写。目的是对该项目进行详细设计，在概要设计的基础上进一步明确系统结构，详细地介绍系统的各个模块，为进行后面的实现和测试做准备。本详细设计说明书的预期读者为本项目小组的成员以及对该小程序感兴趣，在以后想对系统进行拓展和维护的人员。

## 1.2 背景

由于社交媒体的蓬勃发展，以及人们自身对美好事物的追求使得约拍成为新型的社交流行趋势，在各大平台都有非常大量的约拍需求，小红书、微博、豆瓣等是约拍的主要平台。以微博为例，“北京约拍”话题已有 20.4 亿阅读，144.3 万讨论。类似的“上海约拍”、“南京约拍”等话题的阅读量和讨论量也不相上下。并且根据问卷显示，超过半数以上的人有过或实施过约拍，约拍也成为一些人生重要节点，如成年、毕业等等时间节点的热门选择。

微信是当今用户最多应用最广的平台之一，基于微信平台开发的各种小程序，大大满足和丰富了人们多样化的需求。结合优化推荐算法、PC 端、移动端应用等信息化手段，建立集摄影师身份资历认证、摄影作品宣传、预约摄影师、定制服务、售后评价等功能于一身的爱约拍小程序，为模特预约摄影师和摄影企业的互联网化发展提供一个切实有效的信息化平台。通过项目建设，有

效整合各个平台的约拍需求，可以使摄影师和模特之间得到快速的匹配，同时提供安全可靠的约拍环境，实现多方共赢。

### **1.3 术语和缩略语**

1. 互免：双方都不支付费用，摄影师与模特为了互相提高自己的能力，或者找对方一起创作作品，互相不收取费用的约拍，一般叫做互勉约拍或者互免约拍。

### **1.4 参考资料**

- [1] 窦万峰. 软件工程方法与实践(第三版). 北京：机械工业出版社，2016
- [2] 2022 年中国摄影行业分析报告-市场运营态势与发展趋势预测[R]. 观研天下，2020.
- [3] 秦小波. 设计模式之禅[M]. 第二版. 机械工业出版社.

### **1.5 约定**

无

# 2.架构设计

## 2.1 整体架构

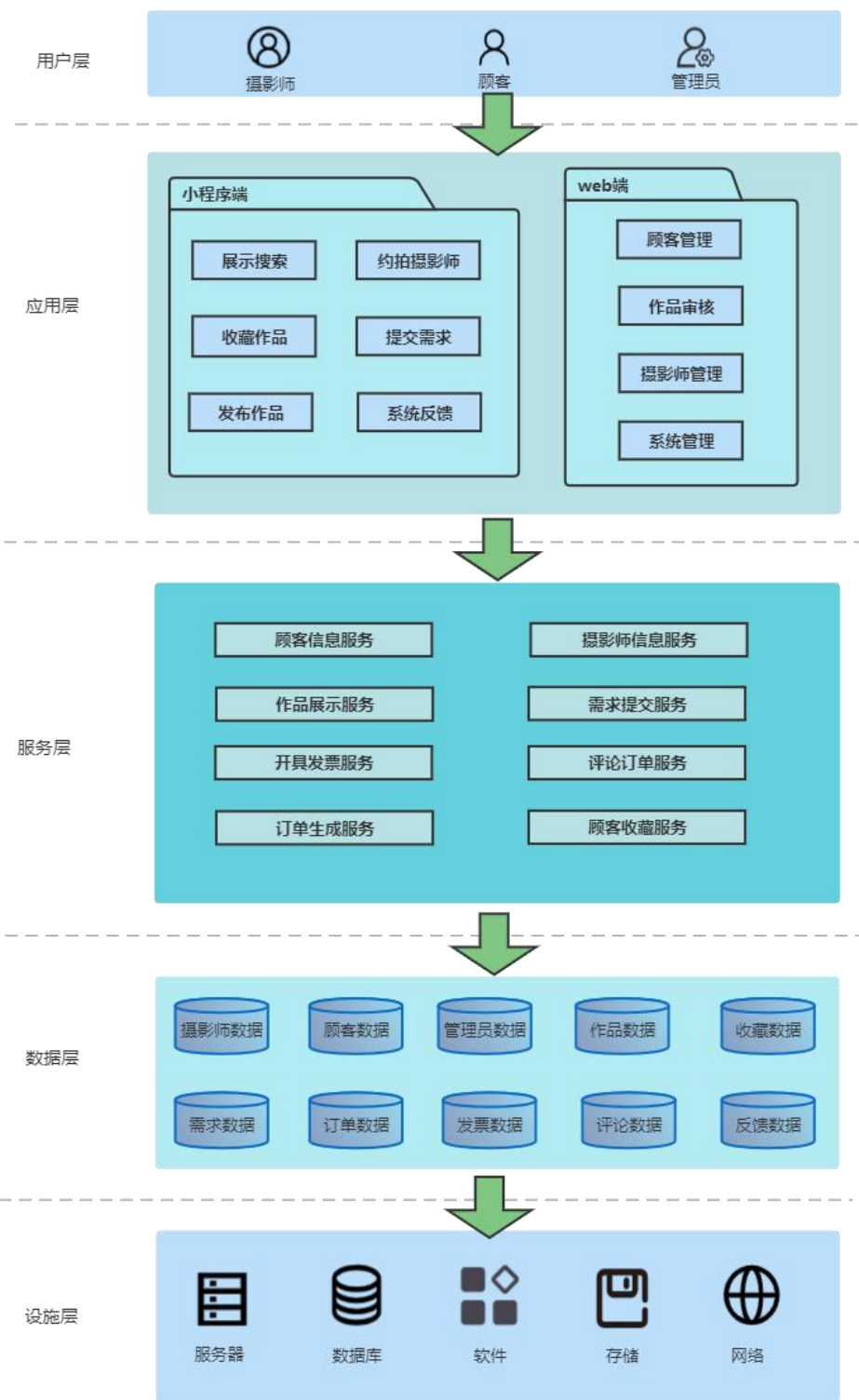


图 1：整体架构图

## 2.2 前端架构

### 2.2.1 小程序端

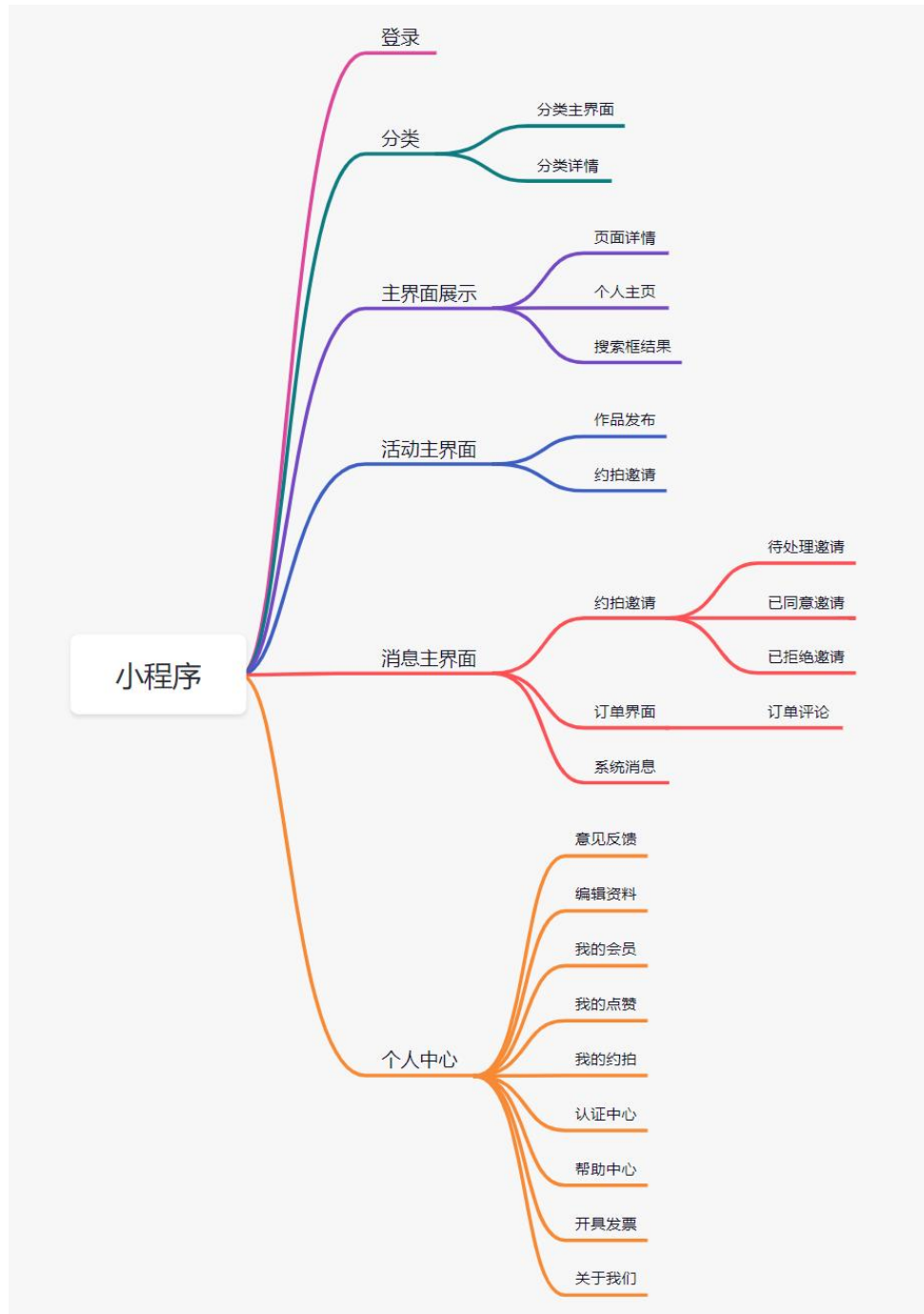


图 2 前端小程序架构图



## 2.2.2 网页管理端

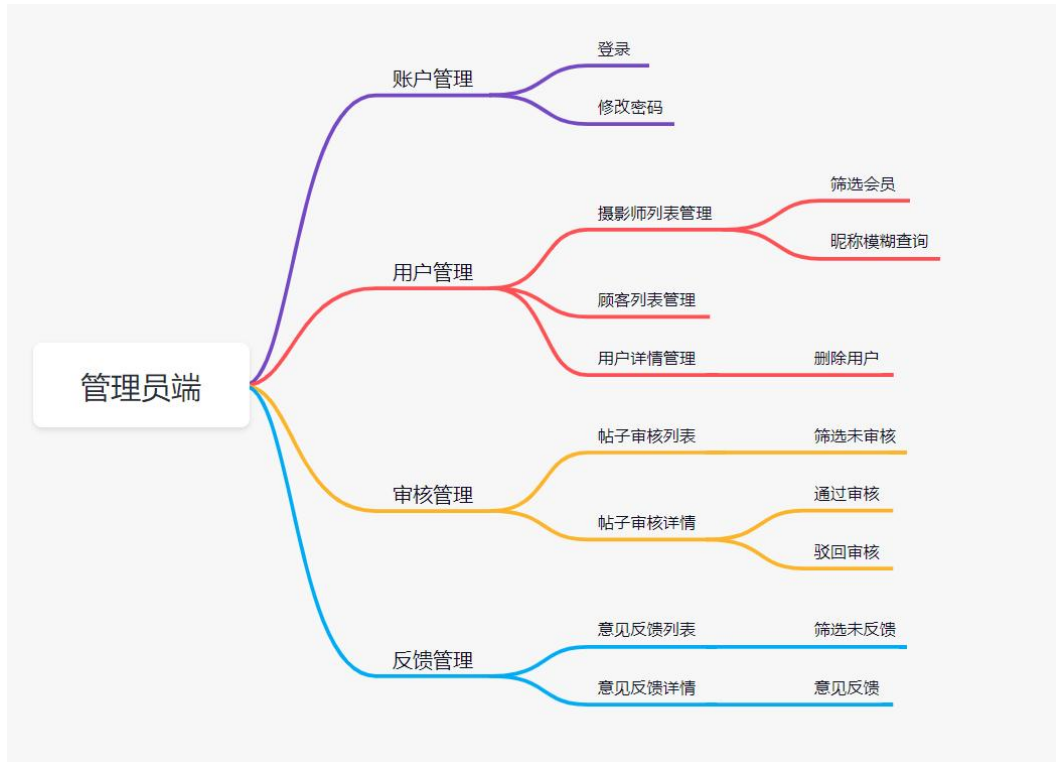


图 3 前端网页端架构图

## 2.3 后端架构

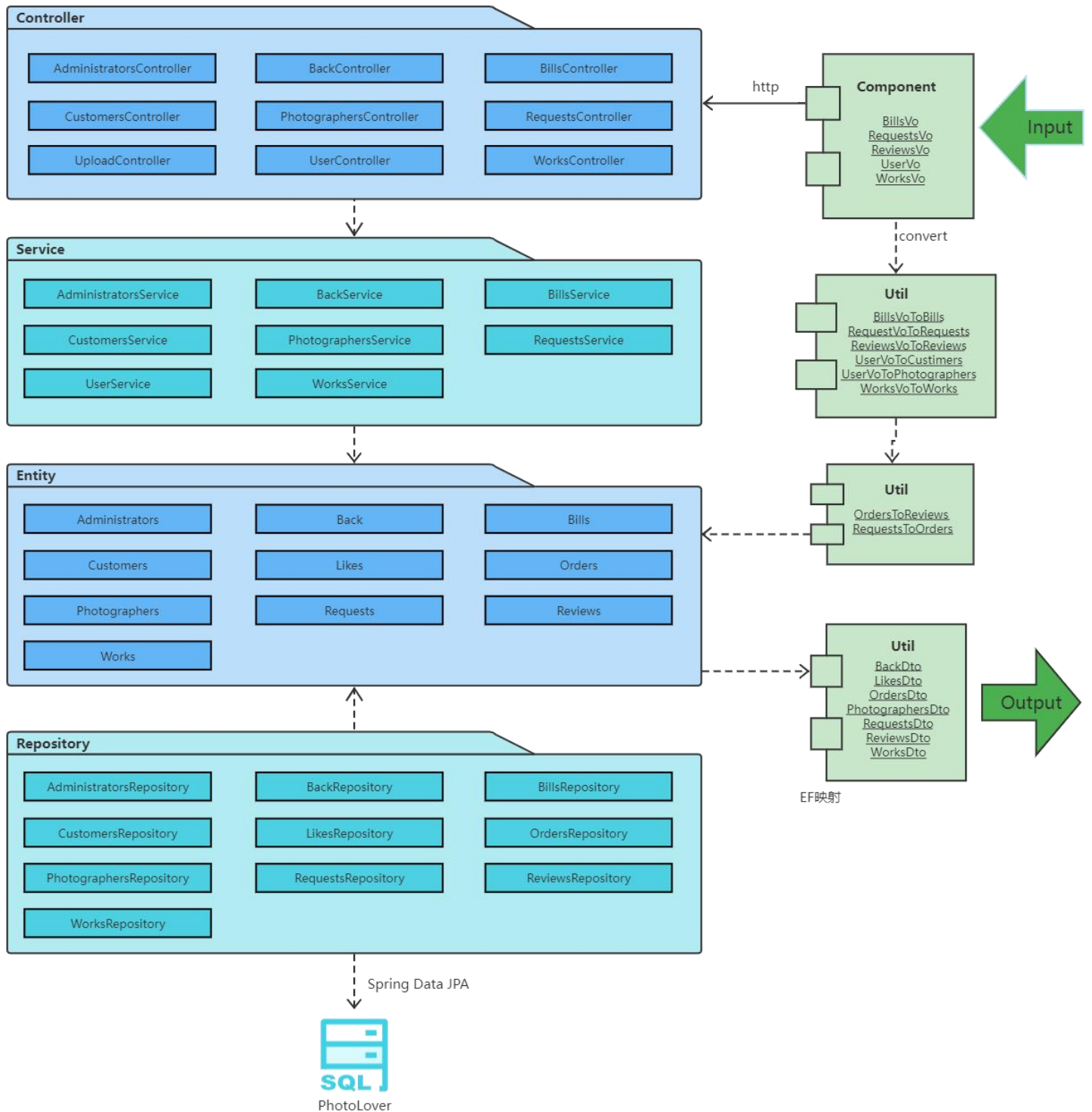


图 4 后端架构图

# 3.程序设计

## 3.1 模块结构设计

### 3.1.1 角色模块结构设计

#### (1) 管理员模块

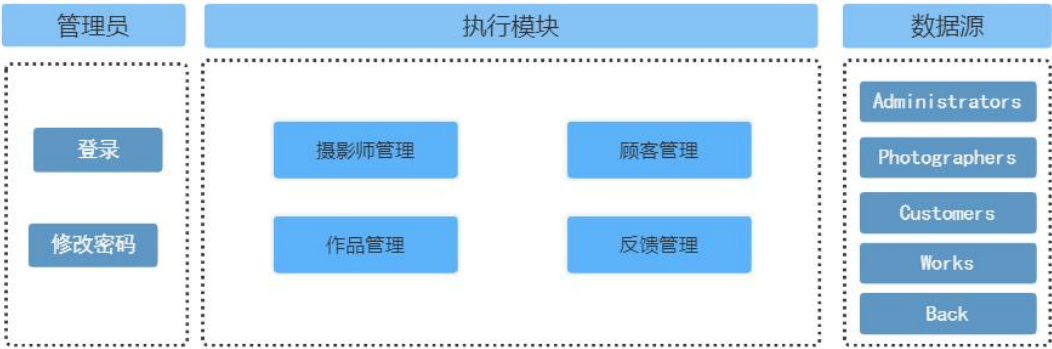


图 5 管理员模块图

管理员在系统中的功能模块包含：摄影师管理、顾客管理、作品管理和反馈管理四个部分，并在 web 端实现其功能。

对摄影师和顾客的管理主要是对其信息的查看与删除；作品管理是在摄影师上传作品后，由系统管理人员对作品内容进行审核的过程；反馈管理中，管理员应该对系统使用者对该系统的评价反馈进行查看及再反馈。

#### (2) 摄影师模块



图 6 摄影师模块图

摄影师在系统中的角色是内容的主要生产者，其功能模块包括：上传作品、处理约拍邀请、开通会员以及系统反馈四个部分。

摄影师在小程序端进行授权登录之后进行作品上传，包括图片、作品价格、风格等主要内容；在顾客对某一摄影师的作品发出约拍邀请之后，摄影师会受到顾客的需求，决定是否接受，接受成功则生成订单，拒绝则需要给出拒绝理由；摄影师还有开通会员的操作，开通会员之后有优先审核和优先展示等特权；摄影师也可以在使用系统后对系统进行评价反馈，以帮助系统更好地升级。

(3) 顾客模块

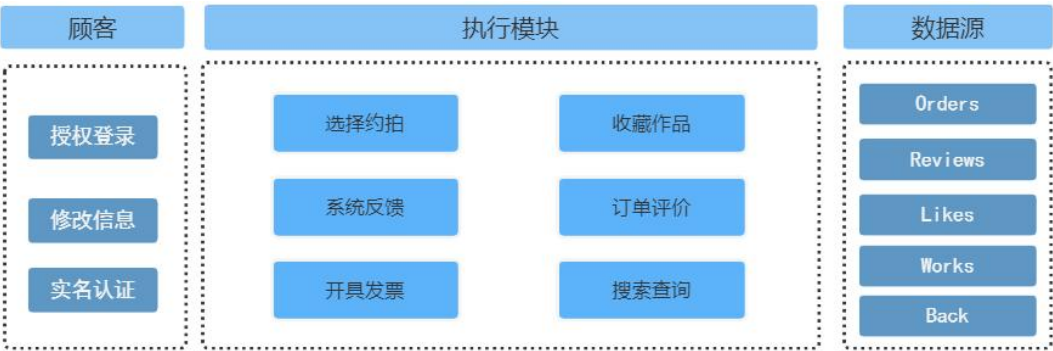


图 7 顾客模块图

顾客在小程序端授权登录后，通过平台实现自己的约拍需求，其功能模块包括：选择约拍、收藏作品、搜索查询、订单评价、开具发票和系统反馈六个部分。

顾客进入约拍广场进行浏览，可以按照作品名称和作品风格进行作品搜索；在选定心仪的作品后，对该作品的摄影师发出约拍邀请，并提交需求表单；在订单完成后，顾客可以对订单进行评价和评级；系统还提供用户对作品的收藏功能，用户可以在自己的收藏列表中便捷找到心仪作品；用户在使用系统的过程中，可以提交自己的使用反馈；此外，在用户有需要的情况下，可以根据订单开具发票。

### 3.1.2 核心业务流模块详细设计

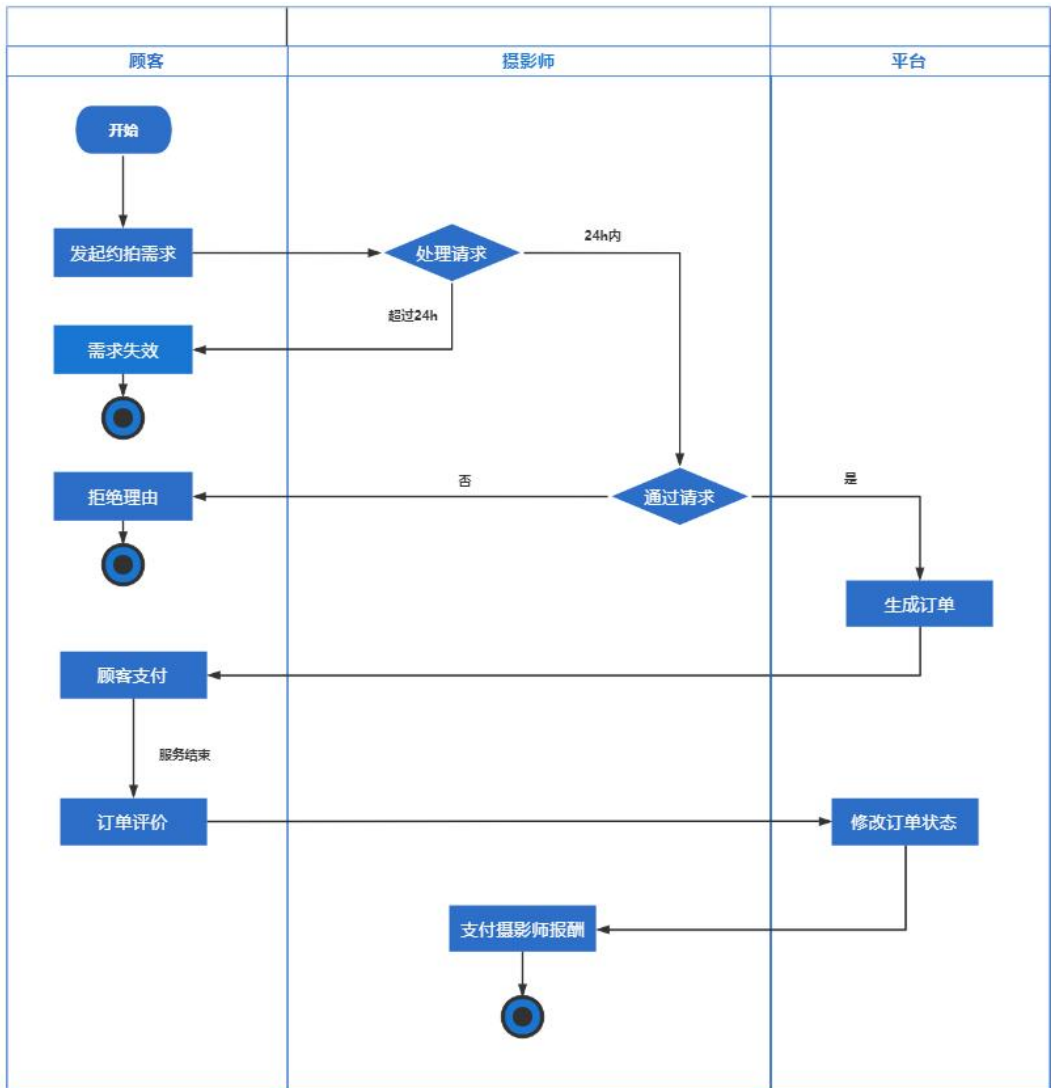


图 8 核心业务泳道图

本系统核心业务围绕顾客的约拍行为展开，通过顾客需求表单的提交实现顾客与摄影师之间的交互。具体流程如下：

- (1) 顾客选中摄影师作品后进入详情页约拍摄影师。
- (2) 顾客填写需求订单，包括预约时间、地点、服务时长、是否勾选增值服务等信息，之后提交付款。
- (3) 对应摄影师收到顾客需求订单，选择是否接单。拒绝接单和在规定时间内不答复系统均会取消该订单，并告知顾客原因。
- (4) 成功接单后订单生成，约拍成功，系统返回给顾客和摄影师对方的联系方式。

## 3.2 程序详细结构设计

### 3.2.1 服务设计

服务	具体操作描述
管理员登录	根据管理员账号索引数据表，核验密码
管理员修改密码	根据管理员账号索引数据表，重置密码
系统反馈回复	根据反馈表 id 判断反馈状态 对未反馈状态：进行反馈值设置后修改反馈状态 对反馈过的状态：返回已反馈过的信息
顾客订单查找	在订单表中根据外键顾客 id 进行索引，筛选所有满足项，即该顾客所有订单
顾客收藏查找	在收藏表中根据外键顾客 id 进行索引，筛选所有满足项，即该顾客所有收藏
摄影师姓名模糊查询	根据输入字段在摄影师表中进行姓名列的模糊查询，返回所有包含该字段的项
摄影师变身会员	根据摄影啥 id 查询其身份状态 对未开通 VIP 的状态进行修改 对已开通过 VIP 的状态进行返回说明
摄影师同意约拍需求后生成订单	修改需求表中的需求审核状态为已审核，需求同意状态为已同意，之后将需求类型转为订单类型，生成新订单
摄影师拒绝需求	修改需求表中的审核状态为已审核，并填入需求拒绝原因，保存
用户身份鉴别	根据用户的 OpenId 对分别在摄影师表和顾客表中查询记录，确定用户身份
作品审核	根据作品 Id 进行索引，修改作品的审核状态为审核通过或审核不通过
作品标题模糊查询	根据输入字段在作品中进行作品标题列的模糊查询，返回所有包含该字段的项

作品风格模糊查询	根据输入字段在作品表中进行作品风格列的模糊查询，返回所有包含该字段的项
作品评论生成	输入订单评论内容后生成订单评论，之后根据外键订单号索引，设置订单表中订单的评论状态为已评论
收藏作品	对于前端传回的标识值 judge 进行收藏表中记录的添加或删除，之后在作品表中更新作品的总收藏量

### 3.2.2 接口设计

输入接口	说明	输出输入	输出	格式/请求
/login	管理员登录	String AId, String password,	"success"; "登录失败，请重新登录！";	json/POST
/repass	管理员修改密码	String AId, String repass,	"密码修改完成！"	json/POST
/PhInfo	显示摄影师列表		List <PhotographerDto>	json/GET
/aPhInfo	显示一个摄影师信息	String POpenId,	Photographers;	json/POST
/queryName	摄影师姓名模糊查询	String PName,	List <PhotographerDto>	json/POST
/ToVip	摄影师开通会员	String POpenId,	"VIP 开通成功！"; "你已经是会员！";	json/POST

/UserInfo	鉴别用户身份	String OpenId,	Photographers; Customers; "查询用户不存在!";	json/POST
/UserDel	删除用户	String OpenId,	"删除摄影师成功"; "删除顾客成功"; "操作用户不存在!";	json/POST
/Register	用户注册	UserVo userVo, Integer judge,	If judge==1: "摄影师注册成功"; If judge==2: "顾客注册成功"; Else: "error";	json/POST
/Identify	用户实名认证	String OpenId,	"客户实名认证成功"; "摄影师实名认证成功"; "查无此人";	json/POST
/WallInfo	显示所有作品列表		List<WorkDto>;	json/GET
/aWInfo	显示一个作品信息	Integer WId,	WorkDto;	json/POST
/CheckWork	管理员审核作品	Integer WId, Integer mes,	if mes==1: "审核通过!"; if mes==2: "审核不通过!"; else: "你没事吧?!";	json/POST



/WallInfo	审核后作品列表		List<WorkDto>;	json/GET
/LoadWork	摄影师上传作品	WorksVo workVo,	"提交成功, 待审核!";	json/POST
/queryTitle	作品标题模糊查询	String WTitle,	List<WorkDto>;	json/POST
/queryTag	作品风格查询	String WTag,	List<WorkDto>;	json/POST
/makeReview	作品评论生成	ReviewsVo reviewsVo,	True;	json/POST
/queryReviews	一个作品的所有评论	Integer WId,	List<ReviewsDto>;	json/POST
/like	收藏作品	String COpenId, Integer WId, Integer judge,	True;	json/POST
/likeState	作品初始收藏状态	String COpenId, Integer WId,	True/False;	json/POST
/BackInfo	显示反馈列表		List<BackDto>;	json/GET

/CBackInfo	一个顾客的反馈列表	String COpenId,	BackDto;	json/POST
/aBackInfo	显示一个反馈信息	Integer BackId,	BackDto;	json/POST
/makeBack	生成反馈	String COpenId , String text,	True;	json/POST
/fillBack	反馈回复	Integer BackId, String text,	If BackCheck=0: "反馈成功! "; If BackCheck=1: "已反馈过! "; Else: "bug";	json/POST
/CInfo	显示所有顾客列表		List<Customers>;	json/GET
/aCInfo	显示单个顾客信息	String COpenId,	Customers;	json/POST
/myOrders	根据顾客查找订单	String COpenId,	List<OrderDto>;	json/POST
/myLikes	我的收藏	String COpenId,	List<Likes>;	json/POST
/LoadRequest	顾客生成约拍请求	RequestsVo requestsVo,	"提交成功, 待接单! ";	json/POST
/AgreeRequest	顾客同意	IntegerRId,	True;	json/POST

	约拍请求 并生成订 单			
/RefuseReques t	摄影师拒 绝约拍请 求	Integer RId,  Integer reason,	True;	json/POST
/UncheckedReq uests	摄影师待 处理的约 拍需求	String POpenId,	List<RequestsDto>;	json/POST
/AgreedReques ts	摄影师已 同意的约 拍需求	String POpenId,	List<RequestsDto>;	json/POST
/RefusedReque sts	摄影师已 拒绝的约 拍需求	String POpenId,	List<RequestsDto>;	json/POST
/makeBills	生成发票	BillsVo  billsVo,	True;	json/POST

详细接口的输入及输出设计如下：

为简化后端参数使用，构建一些提交类接收前端传入参数，再通过提交类到实体类强制转化完成数据的存储。

附：输入数据提交格式表

提交格式	说明	详细内容
WorksVo	作品生成时的提交表单	String POpen_id;  String W_Title;  String W_TextContent;  String W_Pics;  Double W_Price_Per_Hour;  String WTags;

UserVo	用户注册时的提交表单	String OpenId; String Name; String Pic; String Location; String Phone; Integer Sex;
ReviewsVo	生成评论时的提交表单	Integer OId; String COpenId; Integer WId; Integer grade; String content;
RequestsVo	生成顾客需求时的提交表单	String COpen_id; String POpen_id; Integer WId; Integer+ RIf_cloth; Integer RIf_make_up; String RTime_start; Double RTimeScale; Double WPricePerHour;
BillsVo	生成发票时的提交表单	String BName; String BNumber; String BAddress; String BPhone; String BBank; String BAccount;

根据前端需要的数据内容需求，对输出数据进行封装，封装同时，也可以通过实体类之间的外键关联，获取到更多的信息。

**附：输出数据封装格式表**

封装格式	说明	详细内容
BackDto	反馈信息封装	Integer BackId; String COpenId; String CPhone; String BackContent; Date BackTime; Integer BackCheck; String Backback;
LikesDto	收藏信息封装	String PName; String PPic; String PLocation; Integer PIdentify; String WTitle; String WTags; String WPic; Date time;
OrdersDto	订单信息封装	Integer OId; String POpenId; String PName; Integer WId; Double PriceAll; Date OTimeStamp; Integer OIfReviewed;
PhotographerDto	摄影师信息封装	String POpenId; String PName; Integer PIsVip; Integer PIdentify; Integer PQualify;

RequestsDto	需求信息封装	Integer RId; String CName; String CLocation; Double WPriceAll; String RTime_start; Double RTimeScale; Integer RIf_cloth; Integer RIf_make_up;
ReviewsDto	评论信息封装	String CName; String CPic; Integer RevGrade; String RevContent; Date RevTimeStamp;
WorkDto	作品信息封装	Integer WId; String POpenId; String PPic; String PLocation; Integer PSex; Integer PIdentify; String PName; Date W_TimeStamp; String W_Title; String WTags; String W_Text_Content; Double W_Price_Per_Hour; String W_Pics; Integer W_Pass; Integer W_likes;

## 4. 数据设计

本系统一共有 10 个数据表，摄影师基本信息表、顾客基本信息表、作品基本信息表、作品收藏信息表、需求基本信息表、订单基本信息表、发票基本信息表、管理员基本信息表、订单评价信息表、用户系统反馈表。

摄影师基本信息表、顾客基本信息表和管理员信息表属于基本人员表，包含不同身份的相关信息。其中，摄影师和顾客表的主键从微信小程序的 OpenId 中获得；管理员表的主键 Id 自增，管理员的 Id 由系统分配。

作品表中包含作品信息，还有如下几个表与之业务逻辑相关：顾客收藏作品进入作品收藏信息表；顾客发起约拍请求进入需求基本信息表；摄影师统一顾客需求后生成订单进入订单基本信息表；订单完成后顾客填写订单评价进入订单评价信息表。

此外，本系统提供开具发票服务和用户对系统的反馈服务，相关信息分别存储在数据库的发票基本信息表和用户系统反馈表中。

### 4.1 数据常量和变量设计

表 1 Photographers 摄影师基本信息表

列名	数据类型	是否可空	说明
POpen_id	varchar(50)	否	主键 微信 id 唯一标识
PName	varchar(20)	是	摄影师昵称
PPic	longtext	是	摄影师头像
PLocation	varchar(30)	是	摄影师所属地区
PPhone	varchar(20)	是	摄影师电话号码

PIs_vip	Integer(11)	是	摄影师是否开通vip (0 否 1 是)
PIdentify	Integer(11)	是	摄影师是否实名认证 (0 否 1 是)
PQualify	Integer(11)	是	摄影师是否资质审核 (0 否 1 是)
PSex	Integer(11)	是	摄影师性别 (0 男 1 女)

表 2 Customers 顾客基本信息表

列名	数据类型	是否可空	说明
COpen_id	varchar(50)	否	主键 微信 id 唯一标识
CName	varchar(20)	是	顾客昵称
CPic	longtext	是	顾客头像
CLocation	varchar(30)	是	顾客所属地区
CPhone	varchar(20)	是	顾客电话号码
CIdentify	Integer(11)	是	顾客是否实名认证 (0 否 1 是)
CSEX	Integer(11)	是	顾客性别 (0 男 1 女)



表 3 Works 作品基本信息表

列名	数据类型	是否可空	说明
WId	Integer(11)	否	主键 作品 id
POpen_id	varchar(50)	否	外键 摄影师 openId
WTitle	varchar(30)	是	作品标题
W_text_content	longtext	是	作品内容
W_pics	longtext	是	作品图片
W_price_per_hour	double	是	作品每小时单价
W_time_stamp	timestamp	是	作品发布时间
W_likes	Integer(11)	是	作品点赞次数
WTags	varchar(50)	是	作品标签
W_pass	Integer(11)	是	作品审核状态 (0: 未审核 1: 审核通过 2: 审核未通过)

表 4 Likes 作品收藏信息表

列名	数据类型	是否可空	说明
LId	Integer(11)	否	主键 收藏列表 id
COpen_id	varchar(50)	否	外键 顾客 openid
WId	Integer(11)	否	外键 作品 id
Ltime_stamp	timestamp	是	收藏时间

表 5 Requests 需求基本信息表

列名	数据类型	是否可空	说明
RId	Integer(11)	否	主键 需求表 id
POpen_id	varchar(50)	否	外键 摄影师 openId
COpen_id	varchar(50)	否	外键 顾客 openid
WId	Integer(11)	否	外键 作品 id
Wprice_per_hour	double	是	作品每小时单价,
RIf_cloth	Integer(11)	是	是否需要服装 (0 否; 1 是)

RIf_make_up	Integer(11)	是	是否需要化妆 (0 否; 1 是)
RTime_start	varchar(50)	是	需求起始时间说明
RTime_scale	double	是	拍摄持续时长 (以小时为单位)
RPrice_all	double	是	总价 (时长*每小时单价 + if 服装*50 + if 化妆*50)
RIf_check	Integer(11)	是	摄影师是否反馈 (0: 未反馈, 1: 已反馈 )
RIf_agree	Integer(11)	是	摄影师是否同意 (0: 拒绝 1: 同意)
RRefuse_reason	Integer(11)	是	拒绝的理由 (0: 时间不合适, 1: 距离不合适; 2: 其他原因)
RTime_stamp	timestamp	是	需求发生时间

表 6 Orders 订单基本信息表

列名	数据类型	是否可空	说明
OId	Integer(11)	否	主键 订单表 id
POpen_id	varchar(50)	否	外键 摄影师 openId
COpen_id	varchar(50)	否	外键 顾客 openid
WId	Integer(11)	否	外键 作品 id
PriceAll	double	是	总价
OIf_paid	Integer(11)	是	订单是否支付 (0 是 1 否)
OIf_reviewed	Integer(11)	是	订单是否评论 (0 否 1 是)
OTime_stamp	timestamp	是	订单生成时间

表 7 Bills 发票基本信息表

列名	数据类型	是否可空	说明
BId	Integer(11)	否	主键 发票 id
BName	varchar(50)	是	抬头名称
BNumber	varchar(50)	是	纳税人识别号
BAddress	varchar(50)	是	公司地址
BPhone	varchar(20)	是	公司电话
BBank	varchar(30)	是	开户银行
BAccount	varchar(50)	是	银行账户

表 8 Administrators 管理员基本信息表

列名	数据类型	是否可空	说明
AId	varchar(10)	否	主键 管理员 id
APassword	varchar(20)	否	管理员密码
APhone	varchar(20)	是	管理员电话
AName	varchar(10)	是	管理员姓名
ASex	Integer(11)	是	管理员性别

**表 9 Reviews 订单评价信息表**

列名	数据类型	是否可空	说明
Rev_id	Integer(11)	否	主键 评论 id
OId	Integer(11)	否	外键 订单 id
WId	Integer(11)	否	外键 作品 id
COpen_id	varchar(50)	否	外键 顾客 id
Rev_grade	Integer(11)	是	评价等级 (0——5)
Rev_content	longtext	是	评价内容
Rev_time_stamp	timestamp	是	评价时间戳

**表 10 Back 用户系统反馈表**

列名	数据类型	是否可空	说明
Back_id	Integer(11)	否	主键 反馈 id
COpen_id	varchar(50)	否	外键 顾客 id
Back_content	varchar(50)	否	反馈内容
Back_time	timestamp	是	反馈时间戳
Back_check	Integer(11)	是	反馈是否接收 (0: 否; 1: 是)
Back_back	longtext	是	反馈回复

## 4.2 存储配置设计

系统的重点在于图片的上传和展示，考虑一次上传展示的图片量较大，采用本地存储的形式。将图片内容上传到本地存储并生成唯一 UUID，如下图所示，图片访问开放端口下的“/upload”接口，实现向后端的传输，并写入本地路径：C:/Users/DELL/Desktop/uploadFile/

```
1. @RestController
2. @RequestMapping("/upload")
3. public class UploadController {
4.     static final Logger logger = LoggerFactory.getLogger(photo.demo.controller.UploadController.class);
5.
6.     @RequestMapping("/picture")
7.     public String uploadPicture(HttpServletRequest request, HttpServletResponse response) throws Exception {
8.         String filePath = "";
9.         request.setCharacterEncoding("utf-8"); //设置编码
10.
11.         String realPath = "C:/Users/DELL/Desktop/uploadFile/";
12.
13.         File dir = new File(realPath);
14.         //文件目录不存在，就创建一个
15.         if (!dir.isDirectory()) {
16.             dir.mkdirs();
17.         }
18.         try {
19.             StandardMultipartHttpServletRequest req = (StandardMultipartHttpServletRequest) request;
20.             //获取 formdata 的值
21.             Iterator<String> iterator = req.getFileNames();
22.
23.             while (iterator.hasNext()) {
24.                 MultipartFile file = req.getFile(iterator.next());
25.                 String fileName = file.getOriginalFilename();
26.                 //真正写到磁盘上
27.                 String uuid = UUID.randomUUID().toString().replace("-", "");
28.                 String kzm = fileName.substring(fileName.lastIndexOf("."));
29.                 String filename = uuid + kzm;
30.                 File file1 = new File(realPath + filename);
31.                 OutputStream out = new FileOutputStream(file1);
32.                 out.write(file.getBytes());
```

```

33.         out.close();
34.         filePath = filename;
35.
36.         System.out.println("访问图片名称:" + filePath );
37.     }
38.     } catch (Exception e) {
39.         logger.error("", e);
40.     }
41.     return filePath;
42. }
43. }

```

数据库中存储拼接起来的图片名，通过资源静态映射使得前端可以通过开放的服务端口下的：“/images/图片名”，实现静态图片资源的访问

```

1.     @Configuration
2.     public class PicConfig implements WebMvcConfigurer {
3.         @Override
4.         public void addResourceHandlers(ResourceHandlerRegistry registry) {
5.             // /images/**是静态映射, file:/root/images/是文件在服务器的路径
6.             registry.addResourceHandler("/images/**")
7.                 .addResourceLocations("file:C:/Users/DELL/Desktop/uploadFile/");
8.         }
9.     }

```