软件架构文档

版本<2.0>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| <2014-11-01> | <1.0> | 对美家秀秀的软件架构初步描述 | 杨成、舒弋 |
| <2014-11-01> | <1.1> | 对软件架构中的数据库进行建模 | 尹超 |
| <2014-11-19> | <1.2> | 根据细化的需求更新数据库架构和设计 | 尹超 |
| <2014-11-19> | <1.3> | 用例视图、逻辑视图、开发视图修改 | 杨成 |
| <2014-12-05> | <1.4> | 修改数据库建模为数据视图，详细描述数据库表，以及使用软件生成真实的数据库物理建模图 | 尹超 |
| <2014-12-06> | <1.5> | 细化部署视图、数据库文字说明 | 杨成 |
| <2014-12-13> | <2.0> | 根据中期答辩结果，细化需求 | 杨成 |

软件架构文档

# 简介

## 目的

本文档将从构架方面对美家秀秀系统进行综合概述，其中会使用多种不同的构架视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面作出的重要决策。此文档的特定读者为软件开发人员，他们应该仔细阅读此文档。

## 范围

这一软件架构文档适用于将由舒弋等人的团队开发的美家秀秀室内设计系统。此团队将开发这个基于C/S架构的系统。美家秀秀室内设计系统可以使商家分享成套设计模板信息，并为他们提供宣传通道，允许用户自行下载，进行各种修改与搭配。

本软件架构文档通过用例视图、逻辑视图、进程视图、部署视图、开发视图五个方面（“4+1 个 View”）来描述美家秀秀室内设计系统。

## 定义、首字母缩写词和缩略语

暂无

## 参考资料

《美家秀秀软件开发计划》

《美家秀秀软件需求规约》

《美家秀秀立项建议书》

## 概述

文档以下内容从用例视图、逻辑视图、部署视图、实现视图等方面就软件架构进行了说明。对每个usecase进行了实现分析，并确定了基本技术框架，明晰了系统内部分析类间的关系，在较高的抽象层次描述了整个系统的组成。

# 用例视图

本软件用例视图共分为两大部分，分别是门店用户部分，以及软件服务商部分。具体如图：



图2.1 用例视图

# 逻辑视图

本设计模型的逻辑视图分为客户端的各种功能性需求的关键抽象，以及服务器端的各种功能性需求的关键抽象。具体如下：



图3.1 客户端逻辑视图

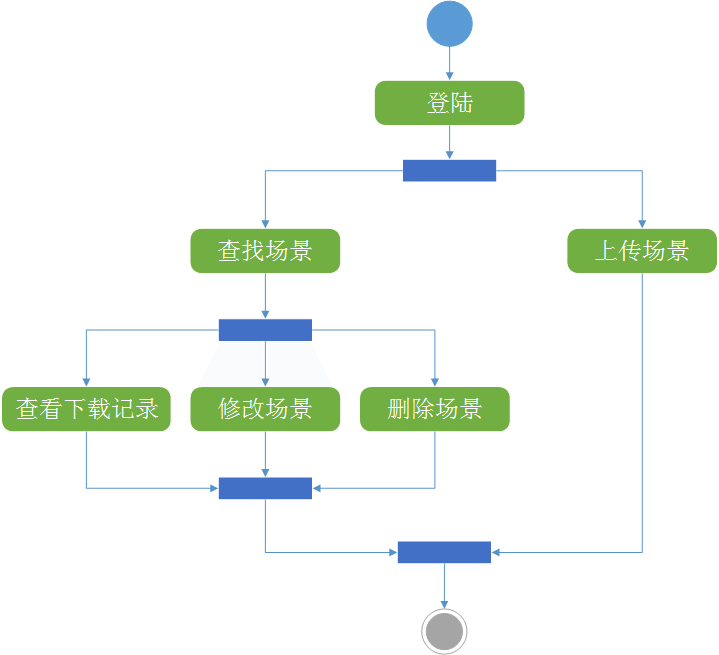
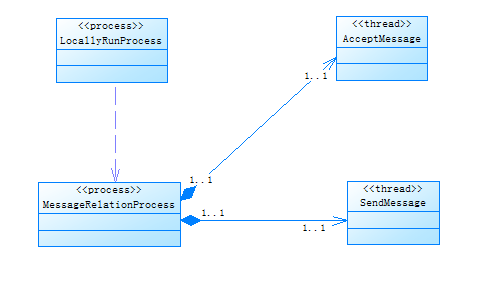


图3.2 服务器端逻辑视图

# 进程视图

每台机器主要分为两个进程同时进行。一个是本地运行进程，负责运行本地室内设计进程的运行，包括模型的显示与操作、分享等相关功能，它依赖于信息传递进程。信息传递进程有两个信息流一个是接受服务器反馈的信息，一个是向服务器发送信息。



# 部署视图

软件采用的是设备之间的点对点连接。每台设备都有进程视图中的两个进程。客户端与服务器端点对点连接，他们分别与服务器交换信息。

客户端与服务器端两者之间通过**Http协议**进行通讯，web服务器软件使用**Apache**，服务器端脚本语言**PHP**。通讯中的关键数据二进制流（账号，密码等），采用 **base64 编码**进行加密。

数据交换格式为 **JSON**。

数据库管理系统为 MySQL，具体建表内容见本文档第 7 部分：数据库建模。



图5.1 部署视图

# 开发视图

## 6.1. 类图

本设计模型的逻辑视图分为以下两种具体的图构成：类图、数据库建模。其中类图分为客户端类图，以及服务器端的类图两部分构成。具体如下：



图6.1 客户端类图



图6.2 服务器端类图

# 数据库建模

## 7.1 概念建模

## 7.2 物理建模

详细描述

|  |  |
| --- | --- |
| 表名 | 描述 |
| user | 存储系统中的用户相关信息，包括id，密码，注册邮箱，显示名称，注册日期。 |
| scene | 存放场景套装信息，包括场景id，品牌id，显示名称，修改日期，设计师，描述文字，下载次数，可用视角数目。 |
| brand | 品牌信息，即场景的制作企业品牌。包括id，显示名称，描述，联系方式，和信息修改日期。 |
| user\_scene | 用户场景信息，存储用户修改后的scene的信息。包括用户场景id，对应scene的id，用户的id，显示名称和修改日期。 |
| tag | 维护预定义的标签，包括id，显示名称和描述 |
| tag\_scene | 存储标签和各场景之间的对应关系 |
| tag\_uscene | 存储标签和各用户场景之间的对应关系 |