# 《游趣》 软件架构设计说明书

Ver 1.0

目录

[《游趣》 软件架构设计说明书 1](#_Toc502305907)

[1. 引言 3](#_Toc502305908)

[1.1 目标 3](#_Toc502305909)

[1.2 文档范围 3](#_Toc502305910)

[1.3 术语和缩略语 3](#_Toc502305911)

[1.4 参考资料 3](#_Toc502305912)

[1.5 系统目标和约束 3](#_Toc502305913)

[2. 系统设计 3](#_Toc502305914)

[2.1 系统架构概述 3](#_Toc502305917)

[2.2 对象模型 4](#_Toc502305918)

[2.3 接口 4](#_Toc502305919)

[2.4 特征实现 4](#_Toc502305920)

[2.5 错误代码 4](#_Toc502305921)

[2.6 错误日志 4](#_Toc502305922)

[2.7 部署视图 4](#_Toc502305923)

[3. 数据库设计 5](#_Toc502305924)

[3.1 概念模型 5](#_Toc502305928)

[3.2 物理模型 5](#_Toc502305929)

[4. 质量及其他方面 5](#_Toc502305930)

[4.1 可维护性 5](#_Toc502305935)

[4.2 安全性 5](#_Toc502305936)

[4.3 可扩展性 5](#_Toc502305937)

[4.4 可靠性 5](#_Toc502305938)

[4.5 可用性 6](#_Toc502305939)

[4.6 性能设计 6](#_Toc502305940)

[5. 附录 6](#_Toc502305941)

[5.1 附件 6](#_Toc502305947)

[5.2 修改记录 6](#_Toc502305948)

## 引言

### 目标

在完成软件开发前期的准备工作如项目需求等，结合《需求确认书》，项目小组提出了这份软件设计说明书。

此概要设计说明书对游趣APP的功能分配，模块划分，程序的总体结构，输入输出和接口设计，运行设计，数据结构设计及出错设计等方面作了全面的概括性的说明，为软件详细设计奠定了基础，同时作为系统分析员工作的阶段性总结和程序员进行开发及未来测试的重要文档资料。

### 文档范围

本文档包括系统的架构设计、数据库设计、安全性、可靠性、可用性等方面的规划和设计。

### 术语和缩略语

### 参考资料

部分内容参考了《游趣\_需求规格说明书》

### 系统目标和约束

系统目标：项目需要完成用户发表、收藏、评论、点赞消息（其中消息为文字、图片、视频），创建问题，参与问题回答与讨论，以及把一些建议反馈给系统。

系统的约束：因为时间有限，该系统只实现最基本的功能。

## 系统设计



### 系统架构概述

该系统的架构采用三层架构（MVC）的模式：展示层、逻辑层和数据层。采用面向对象的设计方法。

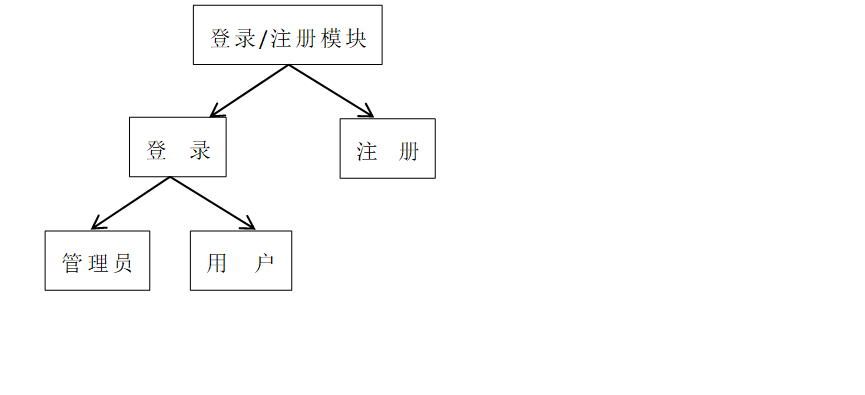
### 对象模型

该系统采用面向对象的设计方法。

### 接口

### 特征实现

**用户信息管理**

****

1. 用户注册
   1. 用户注册页面

用户注册项包括账号（用户名）、注册密码、重复密码，其中，账号、注册密码、重复密码为必填项。

通过检测账号（用户名）是否唯一决定是否注册成功。

* 1. 注册成功，则跳转到首页。

1. 用户登录

注册成功即可使用登录功能，未登录的用户无法进入首页。

* 1. 输入登录名和密码，对应成功即可登录成功
  2. 登录成功，可发表文字、图片、视频、问题，也可以回答问题，查看他人主页以及游记，评论其他人的新鲜事或文章，以及进入个人主页完成相应的扩展功能。

### 错误代码

### 错误日志

### 部署视图

* + 1. **硬件环境：**

1. 需要互联网
2. 需要服务器
   * 1. **软件环境：**
3. 需部署一台Linux操作系统的服务器，计划采用CentOS7.264位系统。
4. 需要Apache2.2作为WEB服务器。
5. 需要安装PHP解释器
   * 1. **配置：**
6. 由于CentOS自带MariaDB数据库管理系统，所以不再做更换MySQL的配置。
7. 配置MariaDB
8. 安装phpMyadmin
   * 1. **部署：**
9. 配置PHP.ini,关闭错误提示。
10. 配置PHP连接MariaDB的文件，修改IP地址、数据库的用户名和密码。
11. 将程序包放在Apache安装目录中得htdocs目录下。

## 数据库设计



### 概念模型

### 物理模型

## 质量及其他方面



### 可维护性

1、 通过程序注释等方式增加代码的可读性和可维护性。

2、 将数据访问层分离，做成一个个函数，由其它层调用，以增加代码的可维护性。

3、 数据库有日志记录，系统一旦出现故障有恢复到故障之前的信息和数据的能力

### 安全性

1、 密码使用md5加密

2、 对用户的输入进行验证

3、 对用户的输入的特殊字符进行转义，防止SQL注入攻击

### 可扩展性

1、 可以通过增加硬件资源的方式提高系统的响应速度。

2、 可以通过修改代码，扩充系统的功能。

### 可靠性

1、 数据库按时备份，有日志记录

### 可用性

做好充分的前期工作，系统一旦投入使用，尽量减少宕机的次数和时间

### 性能设计

## 附录



### 附件

### 修改记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 版本 | 修改人 | 修改时间 | 修改内容 | 核准 |
| 1.0 | 薛佳炜 | 2017-12-29 |  |  |