Unitoto

软件设计文档

**编写：王子豪**

**目录**

[Unitoto 1](#_Toc516758047)

[软件设计文档 1](#_Toc516758048)

[1、导言 3](#_Toc516758049)

[1.1、编写目的 3](#_Toc516758050)

[1.2、背景 3](#_Toc516758051)

[2、总体设计 3](#_Toc516758052)

[2.1、系统概述 3](#_Toc516758053)

[2.2、技术选型 3](#_Toc516758054)

[2.3、架构设计 4](#_Toc516758055)

[2.4、模块划分 5](#_Toc516758056)

[2.4.1、数据库设计 5](#_Toc516758057)

[2.4.2、Dao模块（Mapper接口和xml映射） 5](#_Toc516758058)

[2.4.3、API模块（service api类和entity类） 6](#_Toc516758059)

[2.4.4、Implement模块（service implement类） 7](#_Toc516758060)

[2.4.5、Controller模块（action类） 8](#_Toc516758061)

[2.4.6、视图模块（Vue、iView和Ajax） 8](#_Toc516758062)

[2.5、软件设计技术 9](#_Toc516758063)

[2.5.1、面向服务编程（涉及API模块和Implement模块） 9](#_Toc516758064)

[2.5.2、面向接口编程（涉及Dao模块） 11](#_Toc516758065)

[2.5.3、面向对象编程（涉及Controller模块） 12](#_Toc516758066)

# 1、导言

## 1.1、编写目的

本设计文档编写的目的是说明程序模块的设计考虑，包括程序描述、输入/输出和流程逻辑等，为软件编程和系统维护提供基础

## 1.2、背景

该项目基于2018年中山大学软件创意创新大赛的比赛项目，主要功能是实现一个基于照片分享的社交软件，用户可以登录注册、上传一张或多张图片、评论、添加标签或者加入自己喜欢的社区，该软件是一个运行在网页端的Web应用。

# 2、总体设计

## 2.1、系统概述

该项目总体上分为后台和前端，其中后台分为api、service和web。

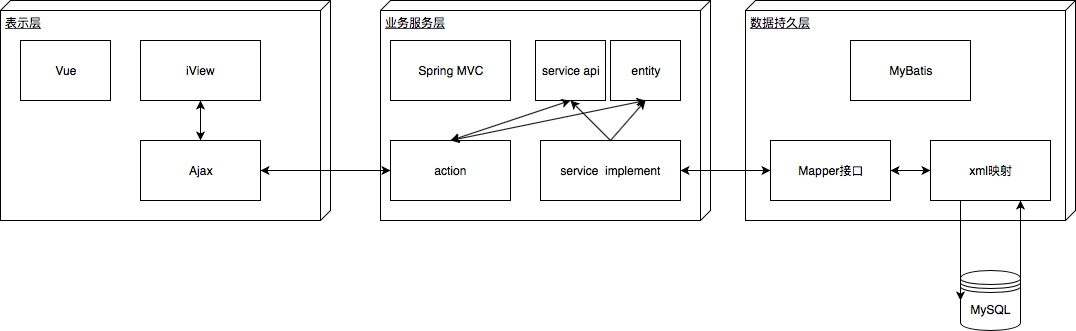
## 2.2、技术选型

2.1、数据库管理系统选择基于腾讯云的MySQL数据库，主要原因在于通过使用腾讯云的MySQL数据库，后台开发成员能够随时随地地开发和测试软件项目，而无需受限于本地数据库的放置地点和开启时间。

2.2、后台选用的技术包括Java、Spring MVC、MyBatis、Maven、Tomcat、Zookeeper和Dubbo。选用Java的原因在于后台开发成员使用的操作系统均不一样，Java的跨平台特性可以很好地解决这个问题，此外Java还有着丰富的框架、软件和社区支持，可以为项目开发提供帮助。选用Spring MVC的原因在于我们的项目属于Web应用，MVC可以对项目进行解耦合，把项目分成三个部分，这有利于项目开发的分工，从而实现并行开发，而Spring MVC作为Java平台最流行的MVC框架，其丰富的软件和社区支持可以帮助我们开发。选用MyBatis的原因在于该框架专注于SQL语句，而对象映射和输入输出则由框架来完成，从而减少后台开发成员的负担。选用Maven的原因在于这是一个集验证、编译、测试、打包、安装和部署为一体的项目管理工具，可以更好地管理我们的项目。选用Zookeeper和Dubbo的原因在于使用分布式服务架构可以让网站承载更大的流量，实现面向服务编程，从而提高Spring MVC的可用性。

2.3、前端使用的框架是Vue，原因在于Vue易于学习，并且是现在比较流行的前端框架，有很好的文档、软件和社区支持。

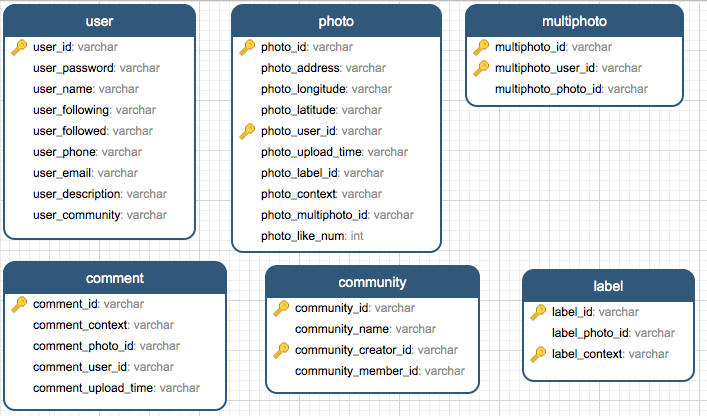
## 2.3、架构设计



## 2.4、模块划分

### 2.4.1、数据库设计

#### 2.4.1.1、表和字段

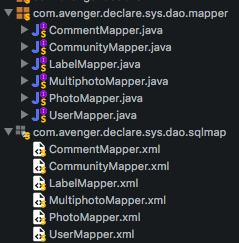


### 2.4.2、Dao模块（Mapper接口和xml映射）

#### 2.4.2.1、功能描述

实现对数据库的增删改查

#### 2.4.2.2、具体实现文件



### 2.4.3、API模块（service api类和entity类）

#### 2.4.3.1、功能描述

service api类定义了该项目需要实现的服务的接口，并且供action类使用。entity类针对数据库中不同的表生成具体的类，并且实现了针对字段的get、set操作。

#### 2.4.3.2、具体实现文件

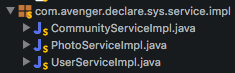


### 2.4.4、Implement模块（service implement类）

#### 2.4.4.1、功能描述

根据API模块中定义的接口，尽可能使用高效的算法准确无误地实现这些接口。

#### 2.4.4.2、具体实现文件

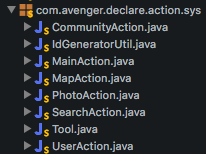


### 2.4.5、Controller模块（action类）

#### 2.4.5.1、功能描述

定义与前端交互的URL，根据前端的需求实现指定的功能，从而实现与前端的数据交互。

#### 2.4.5.2、具体实现文件



### 2.4.6、视图模块（Vue、iView和Ajax）

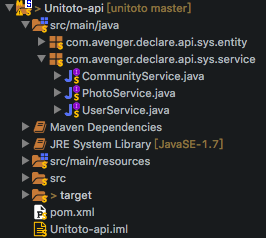
#### 2.4.6.1、功能描述

使用Vue、iView和Ajax实现前端的视图展示。

## 2.5、软件设计技术

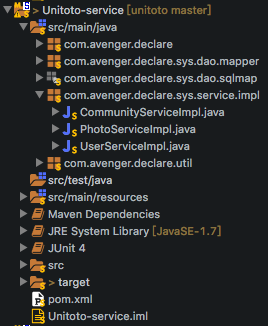
### 2.5.1、面向服务编程（涉及API模块和Implement模块）

#### 2.5.1.1、定义服务



CommunityService.java、PhotoService.java和UserService.java这3个文件定义了需要实现的服务

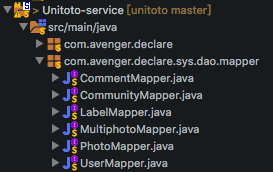
#### 2.5.1.2、实现服务



CommunityServiceImpl.java、PhotoServiceImpl.java和UserServiceImpl.java分别实现了CommunityService.java、PhotoService.java和UserService.java这3个文件中定义的服务。

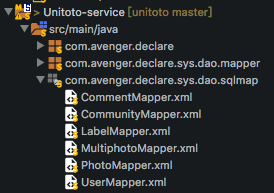
### 2.5.2、面向接口编程（涉及Dao模块）

#### 2.5.2.1、定义接口



CommentMapper.java、CommunityMapper.java、LabelMapper.java、MultiphotoMapper.java、PhotoMapper.java和UserMapper.java这6个文件定义了对数据库进行增删改查的接口

#### 2.5.2.2、实现接口



CommentMapper.xml、CommunityMapper.xml、LabelMapper.xml、MultiphotoMapper.xml、PhotoMapper.xml和UserMapper.xml使用SQL语言分别实现了CommentMapper.java、CommunityMapper.java、LabelMapper.java、MultiphotoMapper.java、PhotoMapper.java和UserMapper.java这6个文件中定义的接口

### 2.5.3、面向对象编程（涉及Controller模块）

#### 2.5.3.1、根据前端的需求定义并实现不同对象需要实现的功能

