

Unitoto

软件设计文档

编写：王子豪

目录

UNITOTO	1
软件设计文档.....	1
1、导言	3
1.1、编写目的	3
1.2、背景.....	3
2、总体设计	3
2.1、系统概述	3
2.2、技术选型	3
2.3、架构设计	4
2.4、模块划分	5
2.4.1、数据库设计	5
2.4.2、Dao 模块 (Mapper 接口和xml映射)	5
2.4.3、API 模块 (service api 类和entity类)	6
2.4.4、Implement 模块 (service implement 类)	7
2.4.5、Controller 模块 (action 类)	8
2.4.6、视图模块 (Vue、iView 和 Ajax)	8
2.5、软件设计技术	9
2.5.1、面向服务编程 (涉及 API 模块和 Implement 模块)	9
2.5.2、面向接口编程 (涉及 Dao 模块)	11
2.5.3、面向对象编程 (涉及 Controller 模块)	12

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

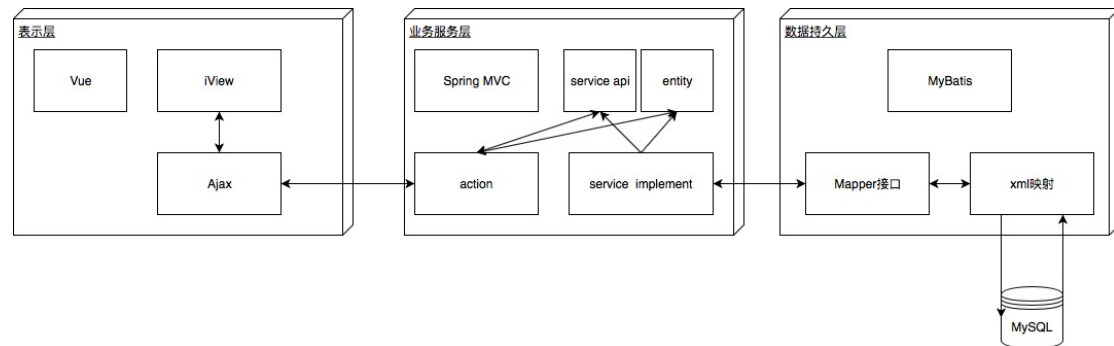
2.1、数据库管理系统选择基于腾讯云的 MySQL 数据库，主要原因在于通过使用腾讯云的 MySQL 数据库，后台开发成员能够随时随地地开发和测试软件项目，而无需受限于本地数据库的放置地点和开启时间。

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。

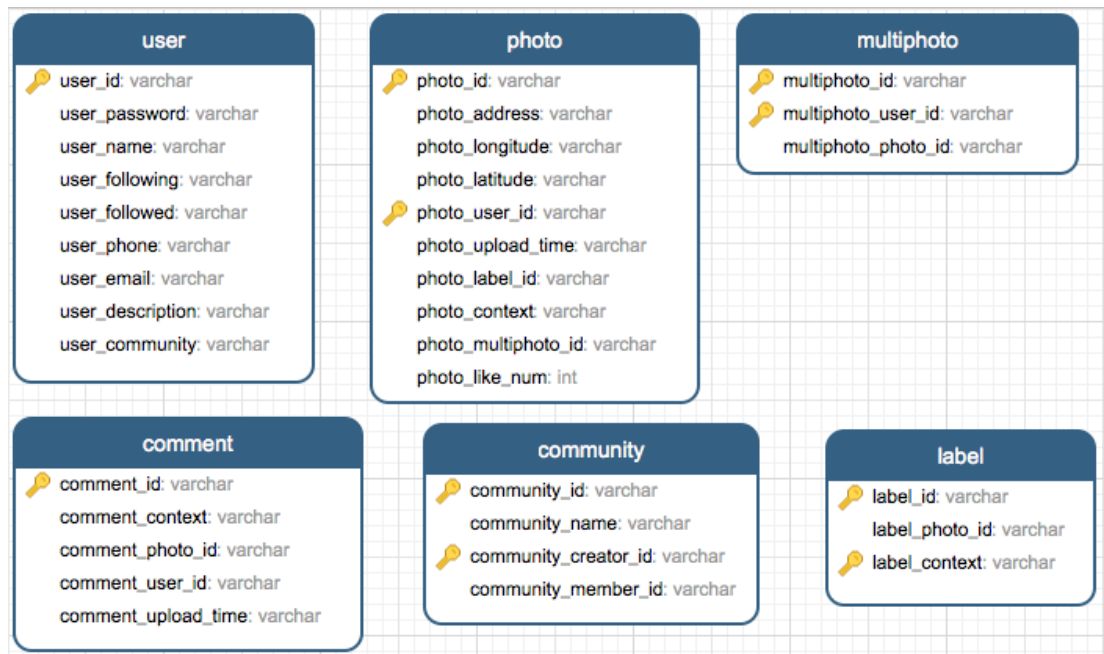
2.3、架构设计



2.4、模块划分

2.4.1、数据库设计

2.4.1.1、表和字段

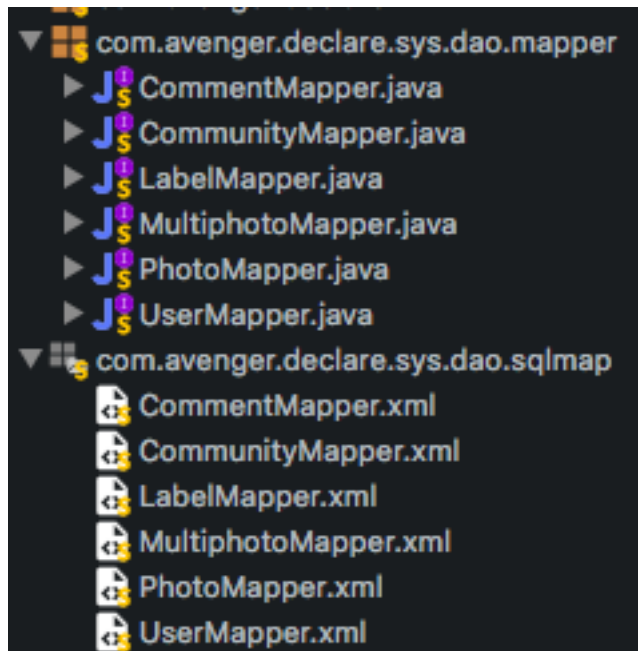


2.4.2、Dao 模块（Mapper 接口和 xml 映射）

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

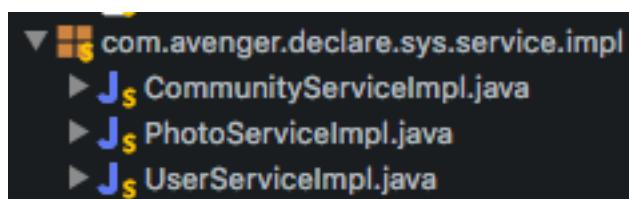


2.4.4、Implement 模块 (service implement 类)

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

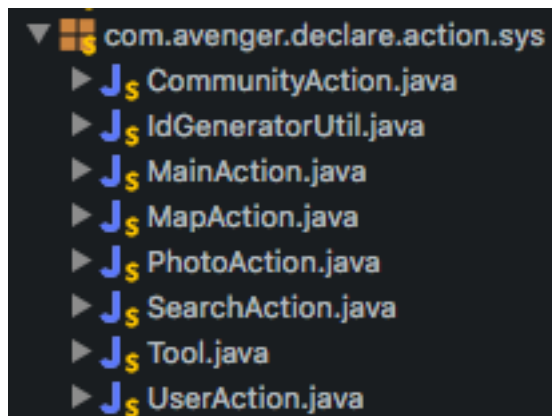


2.4.5、Controller 模块（action 类）

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）

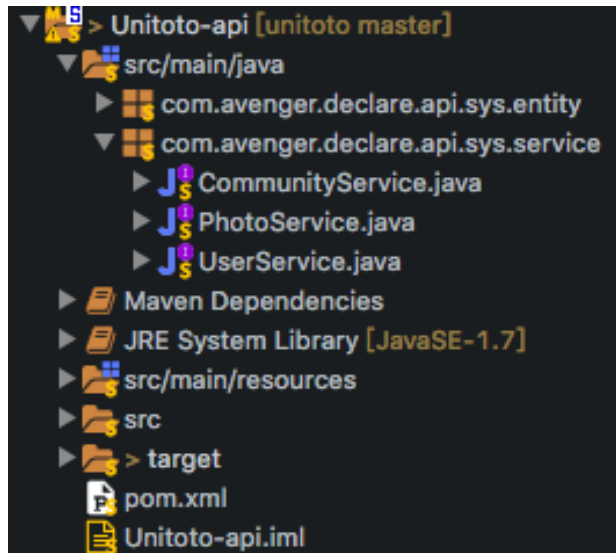
2.4.6.1、功能描述

使用 Vue、iView 和 Ajax 实现前端的视图展示。

2.5、软件设计技术

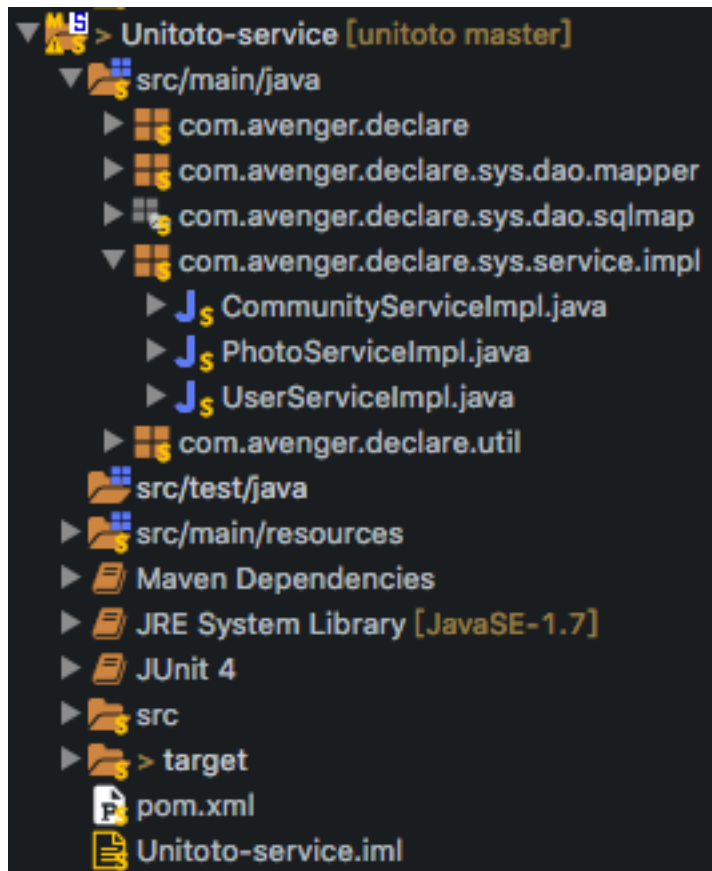
2.5.1、面向服务编程（涉及 API 模块和 Implement 模块）

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

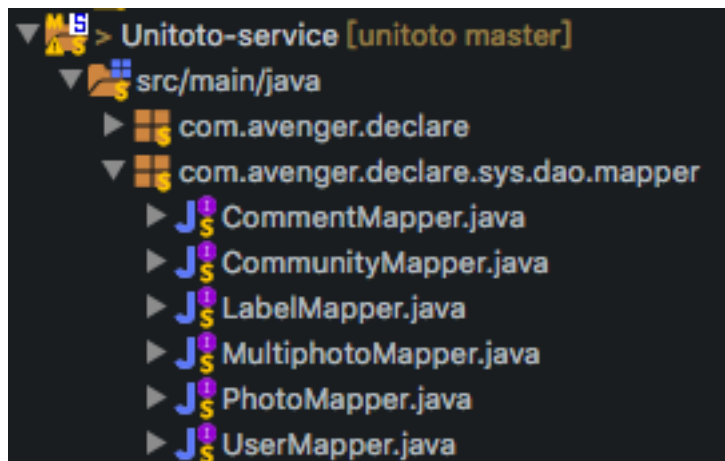
2.5.1.2、实现服务



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。

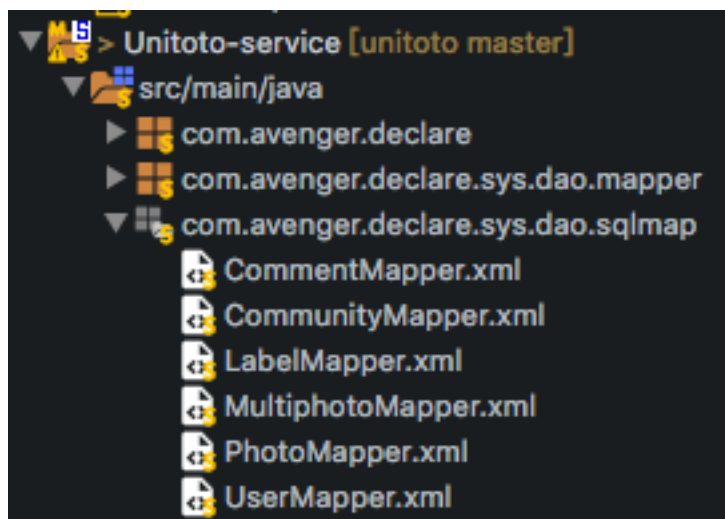
2.5.2、面向接口编程（涉及 Dao 模块）

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

