<风语>

软件架构文档

版本 <1.2>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 10/7/2019 | 1.0 | 进程试图、开发试图 | 陶伯承 |
| 10/7/2019 | 1.1 | 用例试图、部署试图、实现试图 | 项羽心 |
| 11/7/2019 | 1.2 | 文档整合 | 蔡子诺 |

目录

1. 简介 4

1.1 目的 4

1.2 参考资料 4

2. 用例视图 4

3. 逻辑视图 5

3.1 概述 5

3.2 在构架方面具有重要意义的设计包 5

4. 进程视图 6

5. 部署视图 7

6. 实现视图 8

软件架构文档

# 简介

## 目的

本文档将从构架方面对系统进行综合概述，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。

本文档用于第20组正在开发中的Winsper项目。Winsper项目是用于记录生活分享心情的校园app，将提供书写函、写手账、函的发现、通过函关注其他用户等功能。

## 参考资料

《软件工程原理》 沈备军，陈昊鹏，陈雨亭编著.-北京：高等教育出版社，2013.2（2017.5重印）

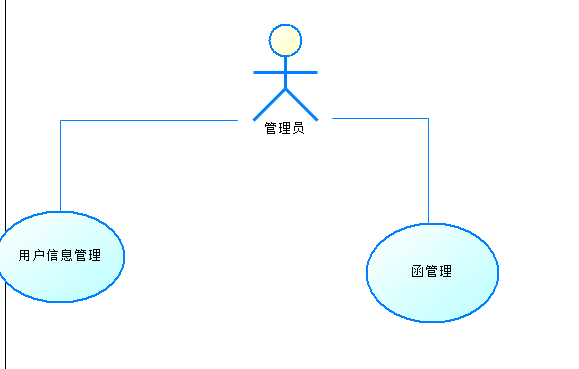
# 用例视图

这里只选取了部分与架构设计相关的用例。其他用例可参考软件需求规约。

选取的用例包含：

用户：

·注册

·个人信息管理

·个人函管理

·个人手账管理

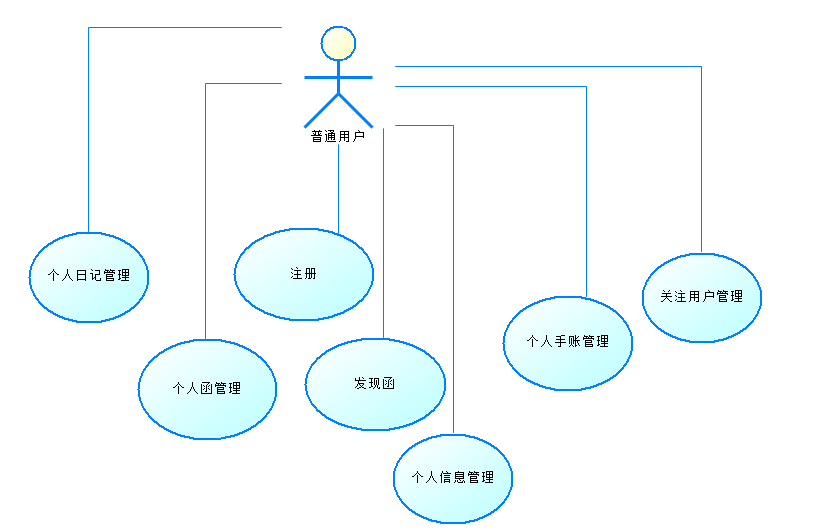
·个人日记管理

·关注用户管理

·发现函

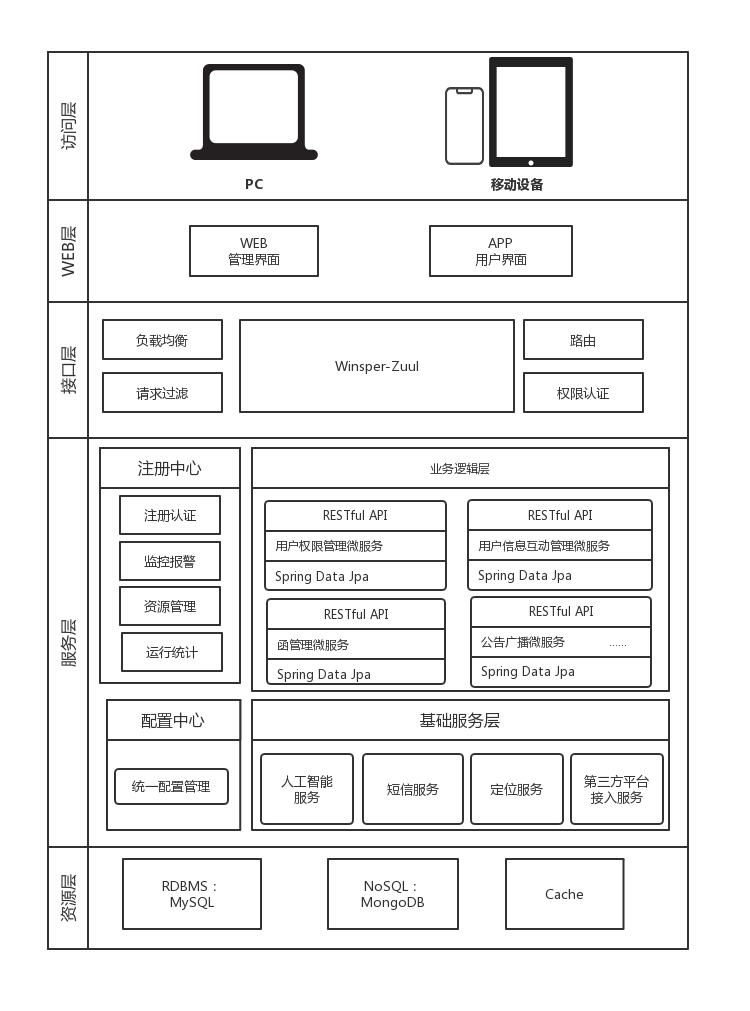
管理员：

·用户权限管理

·函管理

# 逻辑视图

## 概述



Winsper项目的逻辑视图由访问层、WEB层、接口层、服务层、资源层组成。

在访问层和WEB层，管理员通过PC的浏览器浏览用户界面，普通用户通过移动端下载应用进行操作。两者的操作界面各不相同。

在接口层，由一个叫做Winsper-Zuul的微服务构成。它作为面向所有管理员和用户请求的总接口，可进行请求过滤、权限认证、负载均衡、将满足权限认证的请求分发给不同的微服务。

服务层主要完成实际的业务逻辑，它包含业务逻辑层、注册中心、配置中心和基础服务。基础服务层为业务逻辑层提供服务接口。配置中心对所有微服务的配置进行统一管理。所有的微服务注册到注册中心并受注册中心的监控、管理、统计。业务逻辑层包括业务接口、与数据库表对于的实体类，以及访问数据库的Repository类。

资源层由保存数据的关系型数据库mysql、非关系型数据库mongodb、移动端与PC端的cache。

## 在构架方面具有重要意义的设计包

### Controller

Mvc架构中的控制层，用于与前端进行对接。

#### UserController

用户控制层，用于对用户服务的控制，包括对增加用户（注册），登录，发送验证码，更改个人信 息的控制。

#### AdminController

管理员控制层，用于对管理员服务的控制，包括对管理用户（禁用和解禁用户），管理函的控制。

#### HeanController

函控制层，用于对函服务的控制，包括对新增函，删除函，查看函，查看图片的控制。

### Service

服务层，用于实现业务逻辑。

#### UserService

#### 用户服务层，用于对用户服务逻辑的实现，包括对增加用户，登录，发送验证码，更改用户的逻辑 实现。

#### AdminService

管理员服务层，用于对管理员服务逻辑的实现，包括对更改用户状态，查找函的逻辑实现。

#### HeanService

函服务层，用于对函服务逻辑的实现，包括对上传函，查找函，删除函的逻辑实现。

#### PictureService

图片服务层，用于对图片服务的逻辑实现，包括上传图片，查看图片的逻辑实现。

#### TokenService

Token服务层，用于对权限认证服务的逻辑实现，包括生成Token，验证Token的逻辑实现。

### Repository

数据接口层，用于与数据库的交互，对数据库进行增删改查操作。

#### UserRepository

用户数据接口层，包括增加，修改，查询用户数据。

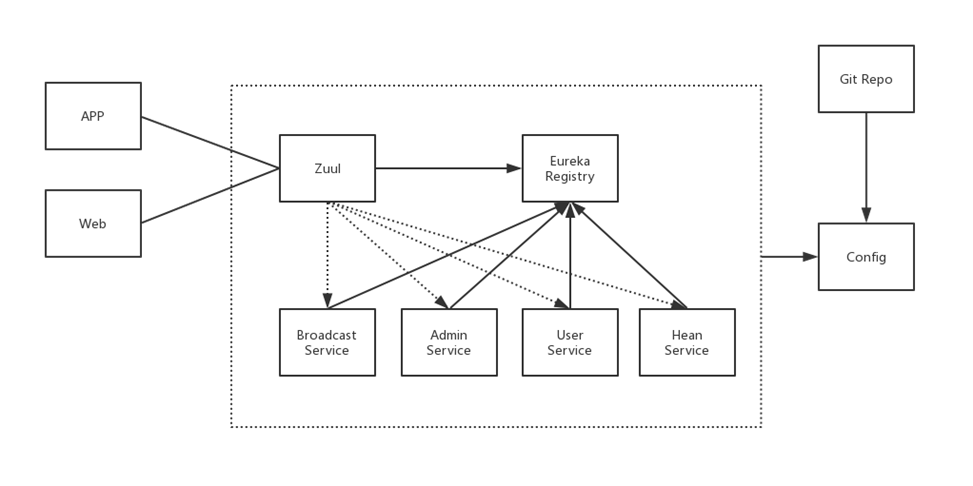
#### HeanRepository

函数据接口层，包括增加，修改，查询函数据。

#### PictureRepository

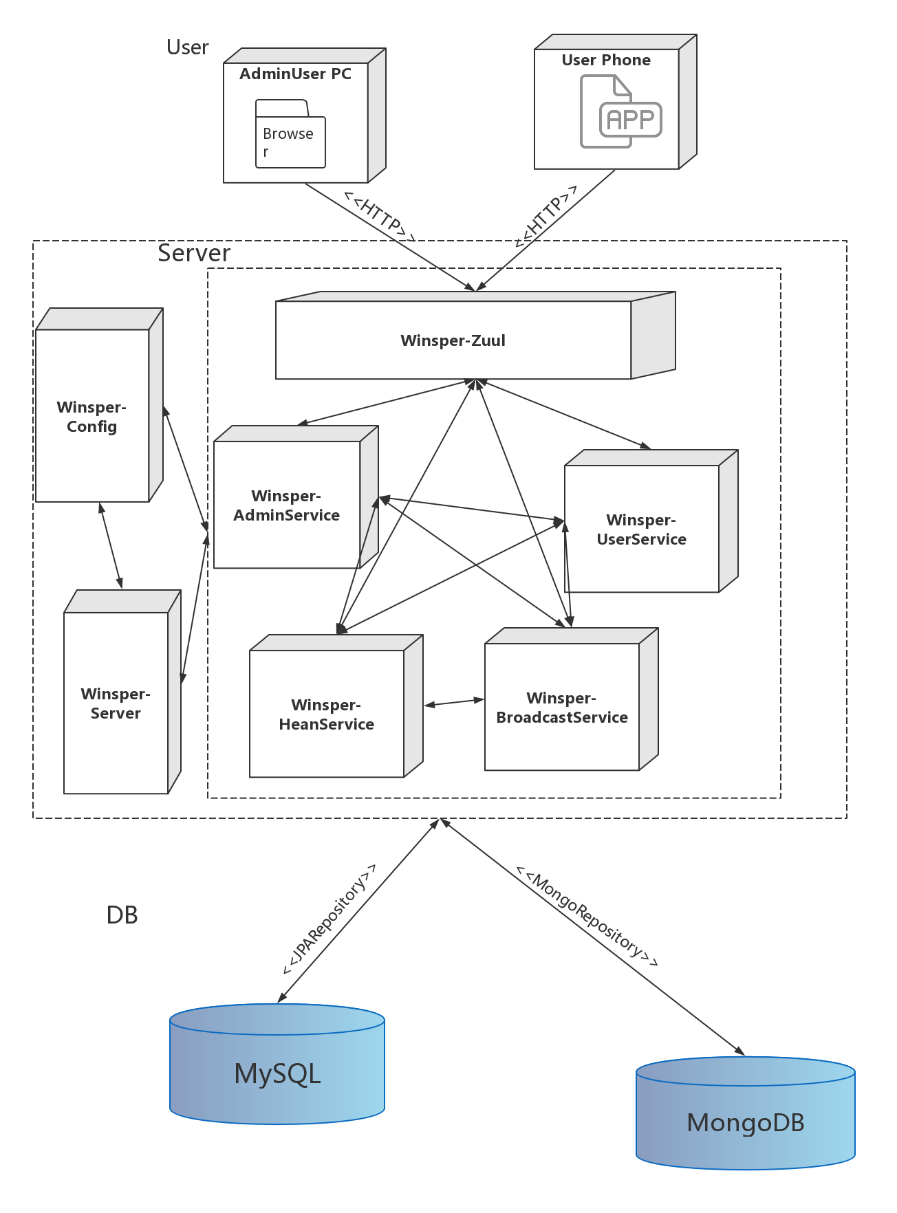
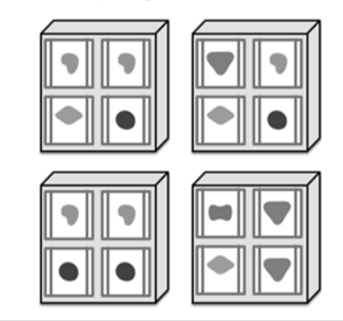
图片数据接口层，包括增加，查询图片数据。

# 进程视图



APP 端和 Web 端发送 HTTP 请求给 Zuul组件，然后转发到具体的微服务。Zuul组件和四个服务注册到注册中心，并从配置中心获取配置信息。

# 部署视图

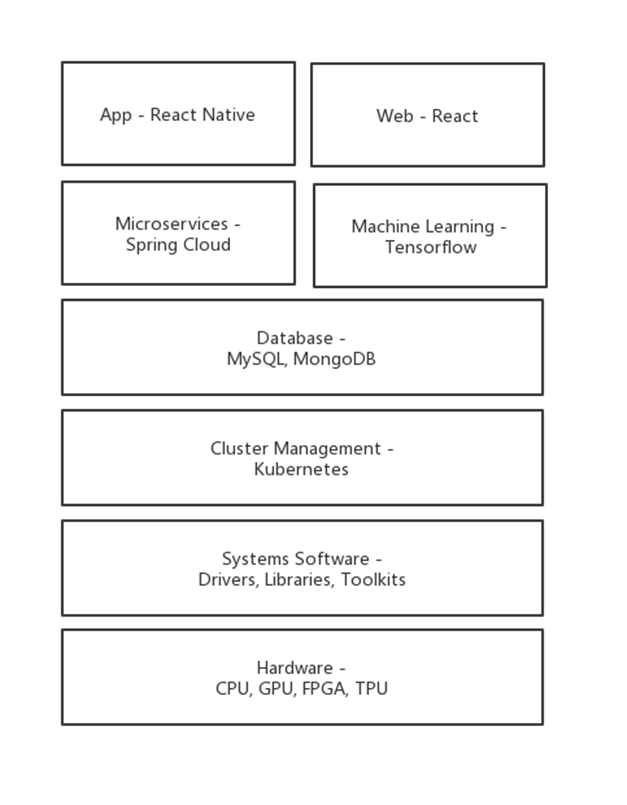


管理员通过浏览器访问，普通用户通过移动端上的app来访问，支持的浏览器版本为chrome50+，支持的andriod版本为4.3+，支持的iOS版本为9+,客户端和服务端的连接可以是局域网或互联网。

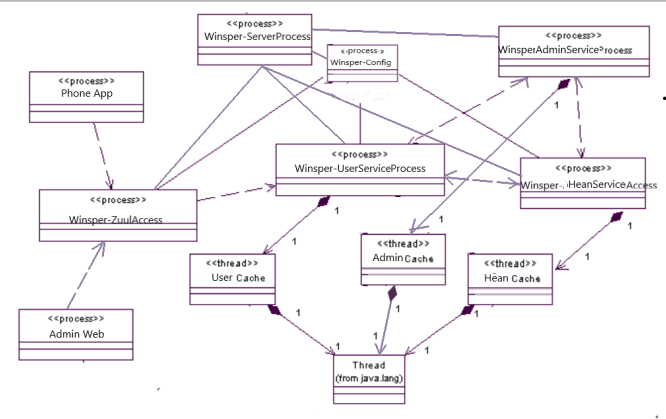
我们有1个主服务器和3个从服务器。每种微服务有多个，各自部署在3个从服务器上，注册中心和配置中心部署于主服务器上。

数据库跟随对应的微服务部署于各服务器上，为使各数据库保持数据一致，设置主从服务器。

# 实现视图



分为六层，前端的移动端和web端，后端的微服务和机器学习，数据库层，集群管理层，系统软件层，以及硬件层。



\*配置中心和注册中心

配置中心对所有微服务的配置进行统一管理。所有的微服务注册到注册中心并受注册中心的监控、管理、统计。

\*Zuul

它作为面向所有管理员和用户请求的总接口，可进行请求过滤、权限认证、负载均衡、将满足权限认证的请求分发给不同的微服务。此接口旨在为希望在活动时执行代码的对象提供通用协议。

\*线程

线程是程序中执行的线程。Java虚拟机允许应用程序同时运行多个执行线程。

每个线程都有优先权。具有较高优先级的线程优先于具有较低优先级的线程执行。每个线程可能也可能不会被标记为守护进程。当在某个线程中运行的代码创建一个新的Thread对象时，新线程的优先级最初设置为等于创建线程的优先级，并且当且仅当创建线程是守护进程时才是守护进程线程。