钉钉OKR管理系统

项目开发说明书

[V1.0(版本号)]

拟 制 人\_\_\_\_\_\_\_赵旭东\_\_\_\_\_\_\_\_\_

批 准 人\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

[二零一七年一月六日]

目录

[1.需求说明 2](#_Toc472342352)

[1.1目的 2](#_Toc472342353)

[1.3定义 2](#_Toc472342354)

[2.系统分析 2](#_Toc472342355)

[2.1需求收集 2](#_Toc472342356)

[2.2数据模型分析 3](#_Toc472342357)

[3.系统的结构 4](#_Toc472342358)

[3.1服务器架构图 4](#_Toc472342359)

[3.2软件架构 4](#_Toc472342360)

[3.3系统质量 6](#_Toc472342361)

[3.4系统结构 6](#_Toc472342362)

[3.5系统功能列表 7](#_Toc472342363)

[4.模块设计 8](#_Toc472342364)

[4.1 OKR首页 8](#_Toc472342365)

[4.2目标列表 9](#_Toc472342366)

[4.3目标管理 10](#_Toc472342367)

[4.4关键结果管理 11](#_Toc472342368)

[4.5个人统计 12](#_Toc472342369)

[4.6团队统计 13](#_Toc472342370)

[4.7员工详情 14](#_Toc472342371)

[5.项目部署 15](#_Toc472342372)

[5.1项目下载 15](#_Toc472342373)

[5.2项目部署 15](#_Toc472342374)

## 1.需求说明

### 1.1目的

在钉钉中接入OKR管理系统，使目标管理与绩效评估体系有机融，并能得到够快速有效沟通反馈。

目标管理确保员工、团队、公司聚焦核心目标，同心协力，而目标执行过程中，组织要敏捷相应外部变化，保证目标适应性，同事还要通过快速有效的沟通反馈，激发员工潜能；

绩效评估则着力于客观评价员工的贡献和能力成长情况，并专注员工的未来绩效提升。 目标管理与绩效评估的结合旨在实现人才成功的前提下帮助企业成功，最终实现员工企业的一体共赢。

### 1.3定义

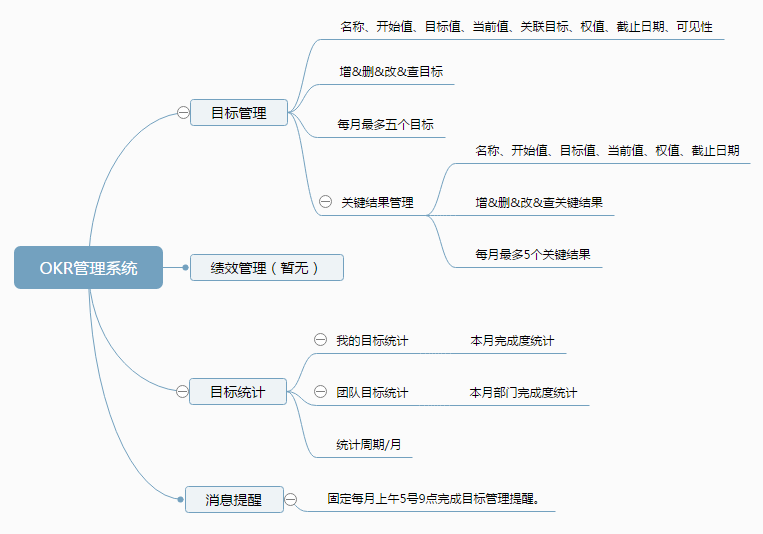
OKR全称是Objectives and Key Results即目标与关键成果法，OKR是一套定义和跟踪目标及其完成情况的管理工具和方法。

钉钉（DingTalk）是阿里巴巴集团专为中小企业打造的沟通和协同的多端平台。钉钉因中小企业而生，帮助中小企业通过系统化的解决方案，全方位提升中小企业沟通和协同效率。

## 2.系统分析

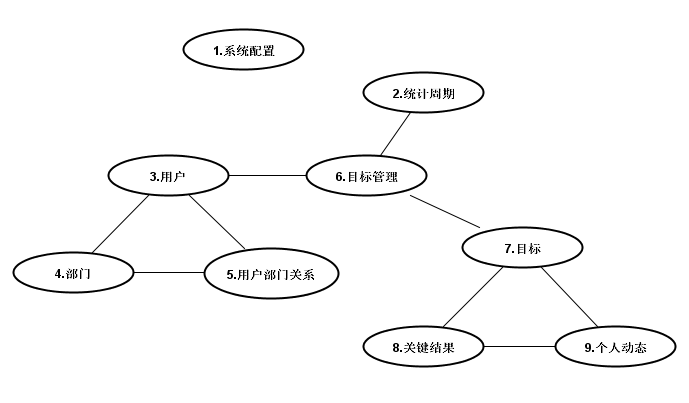
### 2.1需求收集

OKR管理系统包含以下功能，包含目标与关键结果管理、绩效管理、目标统计等。详细请看下图：



### 2.2数据模型分析

数据模型（Data Model）是现实世界数据特征的抽象，它是现实世界与计算机世界的桥梁。下面请看OKR管理系统的数据模型：



1、系统配置：保证系统可正常运行的参数。

2、统计周期：目标统计是按月来统计的，每个月一个周期。

3、用户：企业内用户数据，来自钉钉。

4、部门：企业内部门数据，来自钉钉。

5、用户部门关系：用户与部门关系，来自钉钉。

6、目标管理：用于记录一个月用户目标的管理。

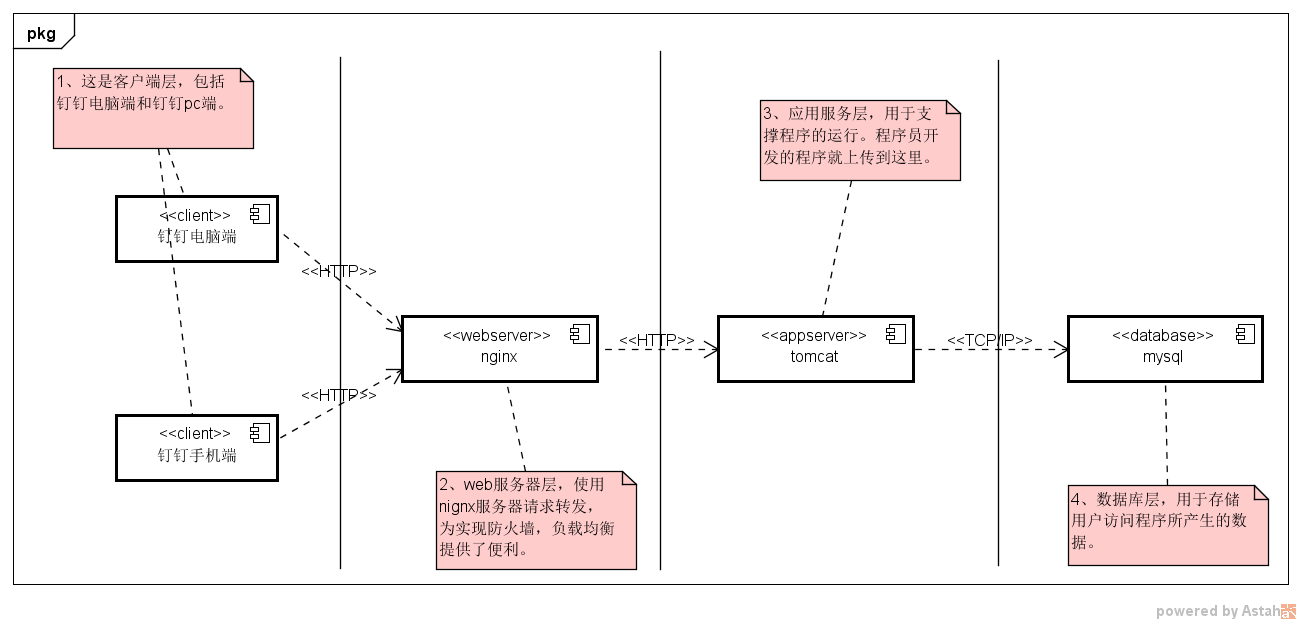
7、目标：用户记录创建的目标。

8、关键结果：用户记录创建的关键结果。

9、个人动态：用于记录目标或关键结果处理的动态。

## 3.系统的结构

### 3.1服务器架构图



图示说明：

* 最左边是客户端层，包括钉钉电脑端和钉钉PC端；
* 左中侧是web服务器层，使用nginx服务请求转发，为实现防火墙，负载均衡提供打下基础；
* 右中侧是应用服务器层，用于支撑程序的运行，程序员开发的程序就上传到这里；
* 最右侧是数据库层，用于存储用户访问程序所产生的数据。

### 3.2软件架构

#### 3.2.1技术架构

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **层次结构** | **客户端层** | **表示层** | **业务层** | **整合层** | **资源层** |
| **语言规范** | html/js/css | Java | | | Sql |
| **技术框架** | **saltui** | SpringMVC | Spring | Jdbctemplate | Mysql sql |
| **运行引擎** | Browser | Tomcat | | | Mysql服务 |

软件技术说明：

1）Html是超级文本标记语言，通过与js,css等web技术结合，实现功能强大的网页；

2）Java编写web应用，具有简单性、面向对象、分布式、健壮性、安全性、平台独立性与可移植性、多线程、动态性等优点；

3）Sql（Structured Query Language）是一种特殊目的的编程语言，是一种数据库查询和程序设计语言，用于存取数据以及查询，更新和管理关系型数据系统。

4）SaltUI 作为钉钉官方微应用组件库，提供了微应用开发中常用设计元素的实现，旨在带给开发者极速、便捷、一致的体验。在 Salt 的整体解决方案中，使用 Nowa 作为工具层面的支持。Nowa 是基于 webpack 的前端开发解决方案集合。包含一系列前端开发生命周期中可能使用到的工具。

5）Spring是一个开源框架，Spring是于2003 年兴起的一个轻量级的Java 开发框架，由Rod Johnson创建。简单来说，Spring是一个分层的JavaSE/EEfull-stack(一站式) 轻量级开源框架。

6） Spring MVC属于SpringFrameWork的后续产品，已经融合在Spring Web Flow里面。Spring 框架提供了构建 Web 应用程序的全功能 MVC 模块。与Spring框架天生整合，无框架兼容问题，与Struts2相比安全性高，配置量小、开发效率高。

7） Spring JDBC模块是Spring框架的基础模块之一。相对hibernate可以写出足够灵活的sql，相对较好的查询效率，使用极其方便，不需要引入第三方框架，不用配置xml。

8）Browser浏览器是指可以显示网页服务器或者文件系统的HTML文件（标准通用标记语言的一个应用）内容，并让用户与这些文件交互的一种软件。

9）Tomcat是由Apache软件基金会下属的Jakarta项目开发的一个Servlet容器，按照Sun Microsystems提供的技术规范，实现了对Servlet和JavaServer Page（JSP）的支持，并提供了作为Web服务器的一些特有功能，如Tomcat管理和控制平台、安全域管理和Tomcat阀等。由于Tomcat本身也内含了一个HTTP服务器，它也可以被视作一个单独的Web服务器。

10）MySQL是一个关系型数据库管理系统，由瑞典MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 最流行的关系型数据库管理系统，在 WEB 应用方面MySQL是最好的 RDBMS (Relational Database Management System，关系数据库管理系统) 应用软件之一。MySQL所使用的 SQL 语言是用于访问数据库的最常用标准化语言。MySQL 软件采用了双授权政策，它分为社区版和商业版，由于其体积小、速度快、总体拥有成本低，尤其是开放源码这一特点，一般中小型网站的开发都选择 MySQL 作为网站数据库。

#### 3.2.2开发环境

1、数据库环境（mysql）

mysql节点：测试服务器192.168.40.211-> mysql -> 数据库（dingding）

连接工具：使用Navicat 连接服务器，进行开发。

2、服务器端（eclipse）

eclipse集成开发环境，使用maven 构建java项目。

eclipse-jetty 插件启动项目，提供后台接口服务。

3、客户端（nowa）

nowa 是基于webpack的前端开发解决方案的集合，它是阿里内外前端组的静态资源构建工具。

nowa 的核心仅仅是个插件的脚手架、安装器和加载器，但它为基于它开发的插件提供了诸多便利，而这些插件共同构成了 nowa 的生态体系。

nowa 插件：

* nowa init 初始化工具
* nowa server 开发服务器
* nowa build 打包工具
* nowa lib 第三方组件定制构建工具

详情： <http://nowa-webpack.github.io/docs/jian_jie.html>

### 3.3系统质量

1、性能、可升级性：前端服务器使用高性能nginx，支持多台tomcat配置。

2、安全性：前端参数传递进行DES加密，防破解。

3、可移植性：后端使用java技术，天生支持window,linux 等多平台。

4、易用性：前端页面使用saltui ，其ui具有与钉钉界面一致性，简洁，易使用。

5、可维护性：后端代码使用maven 模块化构建，前端使用nowa组件化构建生成。使得代码易扩展，易修改。

### 3.4系统结构

#### 3.4.1后台目录结构

**## 目录结构**

-- + dingding

-- — com.dyd.dingding

-- — api........................内部接口，对接钉钉接口的地方

-- — client.....................客户端接口，为钉钉前端提供接口

-- — common.....................公用程序目录

-- — filter...................过滤器

-- — sysconfig................配置文件管理

-- — timer....................定时器

-- — util.....................工具类(\*不得添加任何业务逻辑)

-- — manage.....................项目管理自身程序（\*所有的数据库操作必须在此目录下）

-- — base.....................父级继承类目录

-- — 其它.....................各模块处理

-- + doc............................说明文档目录

-- + webapp.........................配置文件及静态文件目录

#### 4.4.2前台目录结构

**## 目录结构**

├── abc.json ------------------------- 项目构建配置文件

├── favicon.ico ---------------------- 页面图标

├── html ----------------------------- html目录

│ └── index.html ------------------- 入口页面

├── package.json --------------------- 项目配置

├── README.md ------------------------ 说明文件

└── src ------------------------------ 源码目录

├── app -------------------------- 项目级代码

│   ├── app.js ------------------- 项目级脚本逻辑

│   ├── app.styl ----------------- 全局样式

├── components ------------------- 业务模块集合目录

├── images ----------------------- 图片资源目录

└── pages ------------------------ 页面集合目录

   └── demo --------------------- 某一个页面

      ├── index.js ------------- 页面入口文件

      ├── actions.js ----------- 事件列表

      ├── store.js ------------- 存储器

      ├── PageDemo.js ---------- 页面视图逻辑

      └── PageDemo.styl -------- 页面样式

### 3.5系统功能列表

|  |  |  |  |
| --- | --- | --- | --- |
| OKR首页 | **首页** | | 显示目标列表和目标统计菜单 |
| 目标制定 | **目标列表** | | 展示个人信息，以及个人目标列表和目标下关键结果列表，提交目标。展示目标和关键结果操作动态。 |
| **目标管理** | 目标创建 | 创建个人当月目标，最多5个，记录创建动态。 |
| 目标编辑 | 编辑个人某个目标，记录修改动态。 |
| 目标删除 | 删除个人某个目标，记录删除动态。 |
| **关键结果管理** | 关键结果创建 | 创建某个目标下关键结果，最多5个，记录创建动态 |
| 关键结果编辑 | 编辑某个目标下关键结果，最多5个，记录修改动态。 |
| 关键结果删除 | 删除某个目标下关键结果，记录删除动态。 |
| 目标统计 | **个人统计** | | 统计个人目标完成度，可切换月份。 |
| **团队统计** | | 统计所在部门团队目标进度，可切换部门，切换月份。 |
| **员工详情** | | 点击统计中员工，展示目标列表与目标列表模块展示相同，只是不可编辑。 |

## 4.模块设计

### 4.1 OKR首页

#### 4.1.1 模块描述

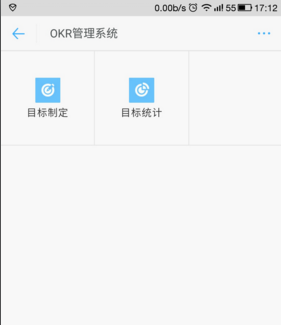
目标管理的首页，显示目标列表和目标统计菜单。此页面静态页面数据交互。

#### 4.1.2 流程逻辑

1、点击目标制定，跳到目标列表页。

2、点击目标统计，跳到目标统计页。

#### 4.1.3 功能界面



### 4.2目标列表

#### 4.2.1 模块描述

展示个人目标列表和目标下关键结果列表，提交目标。并展示目标和关键结果操作动态。

#### 4.2.2 流程逻辑

1、展示个人姓名、部门、目标周期、上次更新时间等信息；

2、展示个人目标列表，包含目标名称、完成度、截止日期、权值、可见人等信息。提供创建&编辑目标入口。

3、展示目标下关键结果列表，包含关键结果名称、完成度、截止日期、权值等信息。提供创建&编辑关键结果入口。

4、目标权值之和等于100时候，可以对其进行提交。

5、展示目标&关键结果操作动态，增删改查等。

#### 4.2.7 功能界面

### 4.3目标管理

#### 4.3.1 模块描述

用于管理个人当月目标，最多5个，记录创建操作动态。

#### 4.3.2 流程逻辑

1、目标信息包括：目标名称、根据关键结果自动计算、完成度衡量方式、单位、开始值、目标值、当前完成值、关联目标、权重、截止日期、全员可见性、可见人。

2、表单字段/验证：

* 目标名称必填，且最大长度为100。
* 权重值必填，且为0~100之间。
* 关联母目标选填，可选择公司所有目标。
* 截止日期默认为当月最后一天，可编辑。
* 可见性默认全员可见，指定人不显示。
* 指定人可见：搜索可见人，可增加多个。

3、完成度计算：

* 完成度衡量方式有：百分比/金额/自定义
* 完成度统一计算公式：（当前完成值-开始值）/（目标值-开始值）\*100%；
* 如果勾选根据关键结果自动计算：【单位】【开始值、目标值】【当前完成值]】项将不显示,目标完成进度=关键结果1的完成进度\*（关键结果1的权重值/100）+关键结果2的完成进度\*（关键结果2的权重值/100）+...

4、业务牵涉：

* 若勾选根据关键结果自动计算，则当前目标完成度将会根据关联关键结果自动计算。
* 目标完成度修改，将直接影响目标统计的结果。
* 设置可见人，将只有这些人能看见它的目标统计信息。
* 目标创建&编辑&删除，都会更新目标上次更新时间，并会记录其操作动态。

#### 4.3.3 功能界面

### 4.4关键结果管理

#### 4.4.1 模块描述

用于管理个人当月某目标的关键结果，最多5个，记录创建操作动态。

#### 4.4.2 流程逻辑

1、关键结果信息包括：关键结果名称、完成度衡量方式、单位、开始值、目标值、当前完成值、权重、截止日期。

2、表单字段/验证：

* 关键结果名称必填，且最大长度为100。
* 权重值必填，且为0~100之间，所有权重值之和不能超过100。
* 截止日期默认为当月最后一天，可编辑。

3、完成度计算：

* 完成度衡量方式有：百分比/金额/自定义
* 完成度统一计算公式：（当前完成值-开始值）/（目标值-开始值）\*100%；

4、业务牵涉：

* 若相关目标勾选根据关键结果自动计算，该关键结果添加，修改，删除都会启动更改其完成度。
* 关键结果创建&编辑&删除，都会更新目标上次更新时间，并会记录其操作动态。

#### 4.4.3 功能界面

### 4.5个人统计

#### 4.5.1 模块描述

统计当月个人目标完成度，可切换月份查看。

#### 4.5.2 流程逻辑

1、页面信息：统计人姓名，统计周期。包含提交状态（正常提交，延期提交，未提交）。

2、统计维度：

* 提交状态（正常提交，延期提交，未提交）。
* 当月周期每个O的进度。

#### 4.5.3 功能界面



### 4.6团队统计

#### 4.6.1 模块描述

统计所在部门团队目标进度，可切换部门，切换月份。

#### 4.6.2 流程逻辑

1、页面信息：所在部门、部门按人员目标统计。

2、统计维度：

* 提交状态（正常提交，延期提交，未提交）。
* 当月每个人周期每个O的进度。
* 可选择部门查看其它部门完成度，可选月份

#### 4.6.3 功能界面



### 4.7员工详情

#### 4.7.1 模块描述

点击统计中员工姓名，展示目标列表与目标列表模块展示相同，只是不可编辑。

#### 4.7.2 流程逻辑

1、展示个人姓名、部门、目标周期、上次更新时间等信息；

2、展示个人目标列表，与目标列表模块展示相同，只是不可操作，只可查看。

5、展示目标&关键结果操作动态，增删改查等。

#### 4.7.3 功能界面



## 5.项目部署

### 5.1项目下载

后端代码svn地址：

<https://192.168.40.10:13231/svn/dingding/branches/201612/dingding>

前端代码svn地址：

<https://192.168.40.10:13231/svn/dingding/branches/201612/dingding-web>

### 5.2项目部署

1、后端

首先，下载项目到eclipse；

然后，打包项目，点击右键，导出（export）选择war file，点击下一步；

然后，设置destination路径， 点击finished,将导出项目工程；

最后，将war上传至服务器即可。

2、前端

首先，下载项目到本地，通过cmd命令进入该项目根目录；

然后，输入nowa build，回车，将会在生成目录 dist，该目录为前端编译后的程序；

最后，将dist上传至服务器。