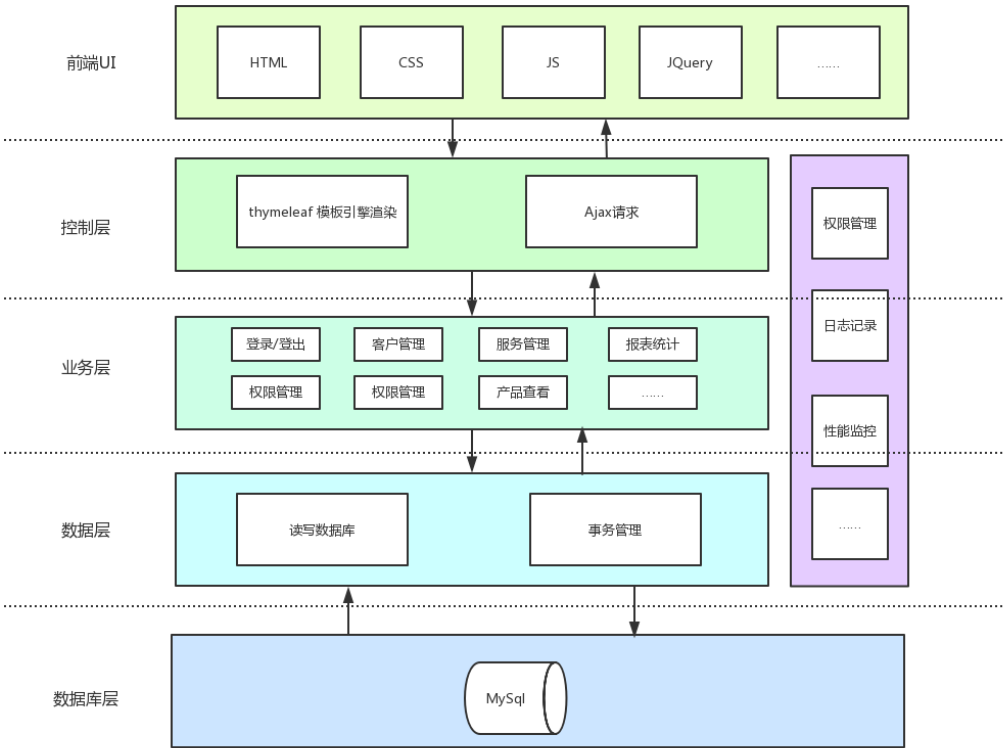


# 详细设计文档

版本	修订人	AMD	说明
0.1	杨晓辉	A	
0.2	高凯	A	

## 架构设计

架构设计图



## 开发工具/环境说明

开发语言 java1.8+ / kotlin1.20+

使用框架 SpringBoot 2.0+  
构建工具 Gradle 4.8.0+  
数据库 Mysql 5.7+

## 接口设计

### 概要说明

基础接口地址：

URL: <http://localhost:8080/>

### 接口详细说明

#### 主页

- URL: /
- HTTP请求方式: GET /POST
- 是否需要登录: 是
- 请求参数: 无
- 返回结果: index页面
- 返回参数说明: 无

#### 登录

- URL: /login
- HTTP请求方式: POST
- 是否需要登录: 否
- 请求参数

参数	类型	是否可为空	说明
username	String	否	用户登录账号
password	String	否	用户登录密码
remember-me	String	是	用户登录时记住密码选项

- 返回结果

结果	说明
成功	返回index页面
错误	返回login页面重新登录

- 返回参数说明: 暂时无

## 登出

- URL: /logout
- HTTP请求方式: POST
- 是否需要登录: 是
- 请求参数: 无
- 返回结果

结果	说明
成功	返回login页面，重新登录
错误	当前页面

- 返回参数说明: 暂时无

## 营销管理

### 1. 创建营销机会

- URL:/sale/createOpportunity
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:无
- 返回结果

结果	说明
int	创建成功或失败

### 2. 修改营销机会

- URL:/sale/modifyOpportunity
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:

结果	说明
int	要修改机会的id

- 返回结果

结果	说明
int	修改成功或失败

3. 展示未分配的营销机会

- URL:/sale/showOpportunity
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:无
- 返回结果

结果	说明
list<Opportunity>	返回的未分配的营销机会的集合

4. 查询单个营销机会

- URL:/sale/findOpportunityById
- HTTP请求: POST
- 是否需要登录:是
- 请求参数

结果	说明
int	要查询机会的id

- 返回结果

结果	说明
list<Opportunity>	返回的未分配的营销机会的集合

5. 删除营销机会

- URL:/sale/findOpportunityById
- HTTP请求: POST
- 是否需要登录:是
- 请求参数

结果	说明
int	要删除机会的id

- 返回结果

结果	说明
int	删除成功与否

6. 新建销售计划

- URL:/sale/createPlan
- HTTP请求: POST
- 是否需要登录:是

- 请求参数：无
- 返回结果

结果	说明
int	新建成功与否

## 7.修改销售计划

- URL:/sale/modifyPlan
- HTTP请求：POST
- 是否需要登录:是
- 请求参数:

结果	说明
int	要修改计划的id

- 返回结果

结果	说明
int	修改成功或失败

## 8. 展示未分配的营销机会

- URL:/sale/showOpportunity
- HTTP请求：POST
- 是否需要登录:是
- 请求参数:无
- 返回结果

结果	说明
list<Plan>	返回的计划的集合

## 客户管理

### 1. 客户信息查询

- URL:/customer/findById
- HTTP请求：POST
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
id	客户经理id

- 返回结果:

结果	说明
List<Customer>	

2. 客户详细信息

- URL:/customer/findByCid
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
c_id	根据客户id查询客户信息

- 返回结果:

结果	说明
Customer	

3. 修改客户信息

- URL:/customer/updateByCid
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
c_id	根据客户id修改

- 返回结果:

结果	说明
int	修改的记录数

4. 查询某客户联系人

- URL:/customer/findLinkmanByCid
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
------	------

c_id	根据客户id查询联系人信息
------	---------------

- 返回结果:

结果	说明
List<Linkman>	

#### 5. 新增某客户的联系人

- URL:/customer/saveLinkman
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
Linkman	

- 返回结果

结果	说明
int	创建成功或失败

#### 6. 修改联系人信息

- URL:/customer/updateByLid
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
l_id	根据联系人id修改

- 返回结果:

结果	说明
int	修改的记录数

#### 7. 删除联系人信息

- URL:/basic/deleteLinkmanByLid
- HTTP请求: POST
- 是否需要登录:是
- 请求参数:

请求参数	说明
l_id	根据联系人id删除联系人信息

- 返回结果:

结果	说明
int	对数据库修改数据条数

## 服务管理

### 1.服务创建

- URL:/statistic/createService
- HTTP请求: GET
- 是否需要登录:是
- 请求参数:
- 返回结果:

结果	说明
int	

### 2.服务分配

- URL:/statistic/findByState?search=&
- HTTP请求: GET
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
search	根据某个或多个字段进行模糊查询

- 返回结果:

结果	说明
List<Customer>	返回客户信息

### 3.服务处理

- URL:/statistic/handlesService?&id=&
- HTTP请求: GET
- 是否需要登录:是



- 请求参数:

请求参数	参数说明
id	处理的服务编号

- 返回结果:

结果	说明
int	处理成功或失败

#### 4.服务反馈

- URL:/statistic/feedbackService?&id=&
- HTTP请求: GET
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
id	反馈的服务编号

- 返回结果:

结果	说明
int	反馈成功或失败

#### 5.服务归档

- URL:/statistic/findByState?&state=&
- HTTP请求: GET
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
state	状态为已归档的服务

### 统计报表

#### 1. 客户构成

- URL:/statistic/findByStructure?search=&
  - HTTP请求: GET
  - 是否需要登录:是
  - 请求参数:
-

请求参数	参数说明
search	searchclassify: 根据客户类型查询 satisfaction: 根据客户满意度查询 creditation: 根据信誉度分类

- 返回结果:

结果	说明
List<Customer>	

## 2. 查询流失客户

- URL:/statistic/findLost?year=&customerName=&managerName=
- HTTP请求: GET
- 
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
year	
customerName	根据客户姓名模糊查询
managerName	根据客户经理名字模糊查询

- 返回结果:

结果	说明
List<Customer>	

## 3. 根据服务类型查询

- URL:/statistic/findByService?year=
- HTTP请求: GET
- 
- 是否需要登录:是
- 请求参数:

请求参数	参数说明
year	根据某一年查询服务的条数

## 4.客户贡献分析

- URL:/statistic/findByContribute

- HTTP请求：GET
- 
- 是否需要登录:是
- 请求参数：无
- 返回结果：

结果	说明
List<Customer>	

## 基础数据

### 1. 查询产品信息

- URL:/basic/findAllProduct
- HTTP请求：POST
- 是否需要登录:是
- 请求参数：无
- 返回结果：

结果	说明
List<Product>	所有产品信息

### 2. 查询库存信息

- URL:/basic/findAllStock
- HTTP请求：POST
- 是否需要登录:是
- 请求参数：无
- 返回结果：

结果	说明
List<Stock>	所有库存信息

### 3. 查询数据字典信息

- URL:/basic/findAllDictionary
- HTTP请求：POST
- 是否需要登录:是
- 请求参数：无
- 返回结果：

结果	说明
List<Dictionary>	所以数据字典信息

### 4. 根据id查询数据字典信息

- URL:/basic/findById
- HTTP请求： POST
- 是否需要登录:是
- 请求参数：

请求参数	说明
id	根据数据字典id查询单条数据

- 返回结果：

结果	说明
List<Dictionary>	该条数据字典的所有信息

5. 修改数据字典信息

- URL:/basic/updateDictionary
- HTTP请求： POST
- 是否需要登录:是
- 请求参数：

请求参数	说明
id	数据id
typeKey	字典类型
typeValue	数据名称
parentId	字典类型的id
editable	可否编辑

- 返回结果：

结果	说明
int	对数据库修改数据条数

6. 删除数据字典信息

- URL:/basic/deleteDictionary
- HTTP请求： POST
- 是否需要登录:是
- 请求参数：

请求参数	说明
id	根据数据字典id查询单条数据

- 返回结果：

结果	说明
int	对数据库修改数据条数

7. 增加数据字典信息

- URL:/basic/addDictionary
- HTTP请求：POST
- 是否需要登录:是
- 请求参数：

请求参数	说明
id	数据id
typeKey	字典类型
typeValue	数据名称
parentId	字典类型的id
editable	可否编辑

- 返回结果：

结果	说明
int	对数据库修改数据条数

## 安全说明

1. 用户敏感数据禁止直接展示，必须对展示数据进行脱敏。例如手机号 182\*\*\*\*6071
2. 数据库中不允许直接存储密码明文，加密后存储。

## 应用系统设计

### 设计原则

一个对象的基本要素包括四大要素：

对象标识：给对象一个可以辨识的名称，要求具有唯一性。对象的标识符全为自增并且名为id。

主实体：对象主要操作的数据库表结构。

动作过程：即对象的输入输出和对数据的处理过程。

I/O关系：即对象的动作过程中所牵涉到的其他数据实体。

## 逻辑结构分析

本系统采用 MVC进行开发。

