随着互联网接入成本日趋降低以及网络增速、4G正式运营等条件的具备，采用数字语音的统一通信方案成为政企客户建设语音门户、跨区域部署企业内部通信网络的首选。企业通过租用运营商语音线路和自行投资小交换机来搭建企业语音门户和内外部通信网络的方式也逐渐呈现外包趋势，由运营商提供含线路、平台在内整体解决方案并由企业通过租用资源来使用的方式已被广泛认识和接受。

云总机正是在这样的背景下引入运营的，是基于云计算而搭建的总机系统，客户无需购买任何软、硬件系统，只需具备人员、场地等基本条件，就可以快速拥有属于自己的总机系统。具有建设周期短、投入少、安全高、部署灵活、系统容量伸缩性强、运营维护成本低等众多特点。云总机运营管理系统，是专门针对企业云总机制定的后台管理系统。便于政企管理其下的分机，以及配置IVR导航，网关，路由等，极大的简化了对云总机各项参数的配置，甚至对于无任何专业知识的人，只需经过短时间培训即可上手操作，对于政企方面来说，这不仅降低了成本，也减少了培训成本。

云总机研究现状：

目前，互联网的接入成本日益降低，移动互联网速度越来越快，政企客户在建设语音门户以及跨区域部署企业内部通信时，更倾向于运用数字语音的统一通信方式。通过租用运营商线路和小交换机，企业搭建语音门户和通信网络的途径也逐渐呈现外包趋势。由运营商提供语音门户和通信网络的整体解决方案，企业通过租用资源来使用的方式已被广泛认识和接受。

当前市场背景下，云总机应运而生。它可以为政企提供统一的语音门户和通信，可随时随地提供内部通信和外部通信服务，具有跨渠道通信能力，是以租赁的方式为租户提供解决方案的。

发展趋势：

以云平台的方式提供服务，具有部署方便、花费少、安全可靠、容量可扩展等多种特色，是目前互联网行业开始普及的一种服务形式。并且绝大多数企业都需要部署内部总机系统，因此云总机服务将有广阔的发展市场。

主要实现目标如下：

1. 系统实现了统一管理租赁云总机服务的政企；
2. 系统可以统一管理IVR分级菜单管理；
3. 系统可以灵活配置系统菜单并且为之匹配相应的角色；
4. 系统的业务伸缩性强，降低公司的开发成本。

本系统采用Spring Boot + MyBatis + Vue.js开发，程序设计语言采用Java语言，采用MySQL作为后台数据库。云总机运营管理系统，主要分为超级管理员、租户管理员、运营管理员和普通分机用户等模块，为客户提供整套的管理云总机的云平台。本系统界面简洁清爽，具有功能丰富，操作简单等特点。

云总机是基于云计算而搭建的总机系统，客户无需购买任何软、硬件系统，只需具备人员、场地等基本条件，就可以快速拥有属于自己的总机系统；通信资源、日常维护与服务由服务商提供。具有建设周期短、投入少、安全高、部署灵活、系统容量伸缩性强、运营维护成本低等众多特点；无论是客户服务中心，产品售后中心，普通的办公总机，客户只需按需租用服务，便可建立一套功能全面、稳定、可靠、分机可分布全国各地，全国呼叫接入的云总机系统。

随着互联网接入成本日趋降低以及网络增速、4G正式运营等条件的具备，采用数字语音的统一通信方案成为政企客户建设语音门户、跨区域部署企业内部通信网络的首选。企业通过租用运营商语音线路和自行投资小交换机来搭建企业语音门户和内外部通信网络的方式也逐渐呈现外包趋势，由运营商提供含线路、平台在内整体解决方案并由企业通过租用资源来使用的方式已被广泛认识和接受。

云总机正是在这样的背景下引入运营的一项面向政企用户的、为用户提供统一语音门户的、可实现内外部统一通信、且具备跨域服务能力的增值产品，可为企业提供虚拟总机、数字分机、虚拟一号通、客服语音IVR及呼叫分组等功能，以租用的方式为政企用户提供统一通信解决方案。

云总机运营管理系统，是专门针对政企云总机呼叫系统开发的云平台，便于政企管理其下的分机，以及配置IVR导航，网关，路由等，极大的简化了对云总机各项参数的配置，甚至对于无任何专业知识的人，只需经过短时间培训即可上手操作，对于政企方面来说，这不仅降低了成本，也减少了培训成本。

因此，想要为企业提供这样的云服务，必定需要超级管理员作为服务提供商登录系统的角色，以管理包括本系统所有参数，以及所有租户，分机等。而作为使用服务的企业，则需要有企业管理员以租户管理员身份登录系统，以管理自己企业云总机系统中的分机，以及IVR菜单，黑名单，工作日设置等系统参数的设置。而对于普通分机用户来说，只需要查看到自己账户的详细信息，以及与自己分机相关的话单等信息，并且应该能够设置分机免打扰，修改密码等基础功能。最后，服务提供商还可以以运营管理员的身份登录，以生成运营报表，管理租户，开户等操作。

从开发人员角度思考，系统应当具有相当的可扩展性，稳定性等，并且应当尽量采用简单的开发流程，因此最终采用Vue.js + Spring Boot + MyBatis的开发模式。对于本系统的登录角色权限控制部分，采用Spring Security框架，对所有http请求过滤并且通过请求中的cookie认证角色信息，在认证通过后，返回相应数据，若认证不通过，则跳转到登录页面。从而实现了整个系统的角色权限控制。

六、特色与创新

1. 系统实现了菜单的灵活配置，并且与角色关联，形成用户-角色-菜单三重权限控制系统
2. 采用Vue.js + Spring + MyBatis的开发模式，实现了前后端分离
3. 本系统以租赁的形式提供服务于政企，客户无需安装任何设备和软件，通过浏览器即可登录本系统

详见《需求说明书》

Spring Boot

Spring Boot是由Pivotal团队提供的全新框架，其设计目的是用来简化新Spring应用的初始搭建以及开发过程。该框架使用了特定的方式来进行配置，从而使开发人员不再需要定义样板化的配置。通过这种方式，Boot致力于在蓬勃发展的快速应用开发领域（rapid application development）成为领导者。

Spring boot的特点

1. 创建独立的Spring应用程序

2. 嵌入的Tomcat，无需部署WAR文件

3. 简化Maven配置

4. 自动配置Spring

5. 提供生产就绪型功能，如指标，健康检查和外部配置

6. 绝对没有代码生成和对XML没有要求配置

Spring boot的优点

spring boot 可以支持你快速的开发出 restful 风格的微服务架构

自动化确实方便，做微服务再合适不过了，单一jar包部署和管理都非常方便。只要系统架构设计合理，大型项目也能用，加上nginx负载均衡，轻松实现横向扩展

spring boot 要解决的问题, 精简配置是一方面, 另外一方面是如何方便的让spring生态圈和其他工具链整合(比如redis, email, elasticsearch)

Spring Security是一个能够为基于Spring的企业应用系统提供声明式的安全访问控制解决方案的安全框架。它提供了一组可以在Spring应用上下文中配置的Bean，充分利用了Spring IoC，DI（控制反转Inversion of Control ,DI:Dependency Injection 依赖注入）和AOP（面向切面编程）功能，为应用系统提供声明式的安全访问控制功能，减少了为企业系统安全控制编写大量重复代码的工作。

Vue 是一套用于构建用户界面的渐进式框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与现代化的工具链以及各种支持类库结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

Vue.js是数据驱动的，你无需手动操作DOM。它通过一些特殊的HTML语法，将DOM和数据绑定起来。一旦你创建了绑定，DOM将和数据保持同步，每当变更了数据，DOM也会相应地更新。

springMVC

**为什么要使用springMVC？**

很多应用程序的问题在于处理业务数据和显示业务数据的视图的对象之间存在紧密耦合。通常，更新业务对象的命令都是从视图本身发起的，使视图对任何业务对象更改都有高度敏感性。而且，当多个视图依赖于同一个业务对象时是没有灵活性的。

Spring Web MVC是一种基于Java的实现了Web MVC设计模式的请求驱动类型的轻量级Web框架，即使用了MVC架构模式的思想，将web层进行职责解耦，基于请求驱动指的就是使用请求-响应模型，框架的目的就是帮助我们简化开发，Spring Web MVC也是要简化我们日常Web开发的。

**MVC设计模型**

MVC 是一种著名的设计模式，特别是在 Web 应用程序领域。模式全都是关于将包含业务数据的模块与显示模块的视图解耦的。这是怎样发生的？视图（例如，JSP 页面）怎样能够与其模型（例如，包含数据的 JavaBean）解耦？记得这句格言么？一个层次的重定向几乎可以解决计算机业中的所有问题。确实，在模型和视图之间引入重定向层可以解决问题。此重定向层是控制器。控制器将接收请求，执行更新模型的操作，然后通知视图关于模型更改的消息。依赖于模型的状态并且依赖于请求的控制器可以决定要显示哪个视图。

**springMVC的强大之处**

1.Spring MVC 实现了即用的 MVC 的核心概念。它为控制器和处理程序提供了大量与此模式相关的功能。并且当向 MVC 添加反转控制（Inversion of Control，IoC）时，它使应用程序高度解耦，提供了通过简单的配置更改即可动态更改组件的灵活性。Spring MVC 为您提供了完全控制应用程序的各个方面的力量。

2.Spring 的 Web MVC 模块是围绕 DispatcherServlet 而设计的。DispatcherServlet 给处理程序分派请求，执行视图解析，并且处理语言环境和主题解析，此外还为上传文件提供支持。

3.DispatcherServlet 通过使用处理程序映射来决定哪一个处理程序应当处理传入的请求。处理程序映射只是用于标识使用哪一个处理程序来处理特定 URL 模式的映射。处理程序是只有一种方法 ModelAndView handleRequest(request,response) 的控制器接口的实现。Spring 还有一些可用的高级处理程序实现；其中一个重要的高级处理程序实现是 SimpleFormController，它提供了将命令对象绑定到表单、对其执行验证等功能。

4.您已经在本系列教程的先前教程中使用了 DispatcherServlet 和简单的处理程序。在下一个部分中，将使用 SimpleFormController 并说明 Spring MVC 提供的各种即用功能。

**springMVC优势**

1、清晰的角色划分：前端控制器（DispatcherServlet）、请求到处理器映射（HandlerMapping）、处理器适配器（HandlerAdapter）、视图解析器（ViewResolver）、处理器或页面控制器（Controller）、验证器（ Validator）、命令对象（Command  请求参数绑定到的对象就叫命令对象）、表单对象（Form Object 提供给表单展示和提交到的对象就叫表单对象）。

2、分工明确，而且扩展点相当灵活，可以很容易扩展，虽然几乎不需要；

3、由于命令对象就是一个POJO，无需继承框架特定API，可以使用命令对象直接作为业务对象；

4、和Spring 其他框架无缝集成，是其它Web框架所不具备的；

5、可适配，通过HandlerAdapter可以支持任意的类作为处理器；

6、可定制性，HandlerMapping、ViewResolver等能够非常简单的定制；

7、功能强大的数据验证、格式化、绑定机制；

8、利用Spring提供的Mock对象能够非常简单的进行Web层单元测试；

9、本地化、主题的解析的支持，使我们更容易进行国际化和主题的切换。

10、强大的JSP标签库，使JSP编写更容易。

………………还有比如RESTful风格的支持、简单的文件上传、约定大于配置的契约式编程支持、基于注解的零配置支持等等。