

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

学士学位论文

BACHELOR'S THESIS



论文题目： 基于 WebX 框架的自助打印系统的分析和实现

学生姓名： 邵一凡

学生学号： 5120379019

专 业： 软件工程

指导教师： 王赓

学院(系)： 电子信息与电气工程学院

上海交通大学

毕业设计（论文）学术诚信声明

本人郑重声明：所呈交的毕业设计（论文），是本人在导师的指导下，独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者签名：

日期： 年 月 日

上海交通大学

毕业设计（论文）版权使用授权书

本毕业设计（论文）作者同意学校保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权上海交通大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本毕业设计（论文）。

保密☐，在____年解密后适用本授权书。

本论文属于

不保密☐.

（请在以上方框内打“√”）

作者签名：

指导教师签名：

日期： 年 月 日

日期： 年 月 日

基于 WebX 框架的自助打印系统的分析和实现

摘要

随着互联网技术的不断发展,越来越多地应用被设计出用以解决人们日常生活中的问题,方便人们的生活。目前,在拥有员工数众多的互联网公司,HR 的工作经常会被各种琐事充斥,极大地降低了员工的工作效率,尤其是帮员工开具各项证明,每天数百名员工的需求让 HR 焦头烂额,员工们也需要等待较长的时间才能取到证明。

为了解决这个问题,提高员工的工作效率,论文提出了自助打印系统的想法,按照打印证明的流程进行了需求分析,并以此为基础,分析了系统的架构,设计出系统的数据库表、接口与服务。在系统的实现中,论文采用了灵活可扩展的 WebX 框架使用 maven 进行项目管理完成 WEB 层的实现,实现了六个 RPC 接口以用于服务器和打印机软硬件交互,使用高速分布式远程调用中间件 HSF 完成了系统与其他系统之间的交互,从而完成为接口提供各项功能的服务。系统实现后,大大减少员工开具证明的时间,解决了效率低下的问题,并为日常生活中其它问题的解决提供了技术思路。

关键词: 自助打印, 证明, WebX 框架, maven, HSF

SELF-SERVICE CERTIFICATE PRINTING SYSTEM BASED ON WEBX FRAMEWORK

ABSTRACT

With the development of Internet technology, there are more and more applications designed in order to solve problems of daily life and bring people convenience. Nowadays, human resource employees in Internet companies with large amounts of employees are busy with various small things, which causes a huge decrease to their efficiency. Especially the work of issuing certificates for hundreds of employees make them tired and have little time to do other jobs. Also, employees who needs certifications also have to wait long to get them.

In order to solve this problem and raise employee's work efficiency, the thesis points out the idea of self-service printing system. According to the process of printing a certification, the thesis lists the software and hardware demands of the system. On the basis of these, the thesis designs the database tables, the interfaces and services of the system. During the implementation of the system, the thesis uses the flexible and extensive WebX Framework to implement the web layer, uses Maven to manage the project, implements six RPC interfaces for interaction between the server and the printer, uses the High-speed Service Framework to implement the interaction between the system and other systems and thus completes the services which provide functions for the interfaces above. After completing the system, time for employees to issue certifications is largely decreased, which solves the problem of low efficiency and provides technical ideas for solving other problems of daily life.

Key words: self-service print, certification, WebX Framework, HSF

目 录

第一章 绪论	1
1.1 目的和意义	1
1.2 本文组织	1
第二章 研究背景	1
2.1 互联网时代背景	2
2.2 项目中使用的技术	2
2.2.1 Maven	2
2.2.2 WebX 框架	3
2.2.3 HSF	5
2.2.4 TDDL	7
2.3 本章小结	7
第三章 分析和设计	7
3.1 需求和目标分析	8
3.1.1 系统硬件需求	8
3.1.2 系统软件需求	9
3.2 设计与架构	10
3.2.1 系统交互流程	10
3.2.2 系统软硬件交互接口设计	10
3.2.3 系统间交互接口设计	12
3.2.4 服务类设计	13
3.2.5 数据库设计	15
3.2.6 系统架构	17
3.3 开发计划	18
3.4 本章小结	20
第四章 实现	21
4.1 具体实现	21
4.1.1 服务类实现	21
4.1.2 接口实现	24
4.1.3 DAO 层实现	25
4.2 实现中的其它问题	25
4.3 测试	26
4.4 实现结果	27
4.5 本章小结	30
第五章 总结和展望	31
5.1 结论	31
5.2 展望	31
参考文献	32
谢辞	33

第一章 绪论

1.1 目的与意义

由于互联网公司普遍员工人数众多，因此内部员工的管理与服务就显得尤为重要。面对数量如此多的员工的日常需求，仅仅依靠 HR 们的人力已经是远远不够了，需要完善的系统来帮助处理。目前，为了方便员工、提高员工的工作效率，公司已经自主开发了员工入职、离职、异动系统，商业保险系统、体检系统、福利系统、HR 服务中心系统、薪酬核算系统等内部系统，使得各项员工服务流程化、线上化。

除了这些之外，HR 们还有一项工作是经常被大家忽略但也是十分重要的，那就是员工的在职证明、收入证明等各个各项证明的开具，目前此项工作都是由某几位员工手动完成的。而依靠员工手动开具证明有以下问题：

(1) 效率问题。目前，经过 HR 服务中心员工的统计，在公司杭州西溪园区一共有一万五千多名员工工作，每天对收入证明、在职证明等各项证明的打印需求量约为两百份，这就至少需要一名员工整天投入在帮其他员工开具证明这项机械的工作中而无法进行其他工作，使得员工工作简单单调且产出小，不符合公司的基本理念。

(2) 安全问题。员工收入是公司内部一个保密级别非常高的敏感数据，泄露员工薪资会被处以非常严厉的惩罚，而处理开具收入证明的员工可以接触到所有员工的薪资，有十分严重的安全隐患。而这在此时也是一种无奈之举。

因此，如若是有一个完整的系统能够使每个员工自助地完成各项证明的开具，这不仅仅能够使每个员工能够免于各项繁琐的审批，方便快捷地拿到证明，还能大大解放 HR 们的工作量，使他们宝贵的时间能够投入到更加有意义的工作中。同时，在线上完成证明的开具可以使 HR 员工避免看到其他员工的薪资数据，解决了手动开具证明的安全隐患问题。

经过调研，在国内大型公司中尚未有此种系统的开发，开具证明都是通过相关员工手动操作。开具证明这一操作虽然属于较为简单、工作量不是很大的，但是随着数量的增多，依然会花去大量的时间。再者，现在已经进入了互联网时代，人类的生产力已经大大被互联网所解放，有越来越多的工作正在慢慢实现线上化、自助化。自助打印系统也是顺应时代的潮流，从身边的一件小事开始推动时代的进程。

1.2 本文组织

第二章主要介绍自助打印系统的研究背景以及所需要用到的技术的简介，包括 webx 框架、HSF、maven 等。第三章主要阐述系统的软硬件需求、使用流程以及软硬件交互接口的设计、数据库的设计、系统的整体架构。第四章介绍系统的实现，包括功能、交互、界面，以及系统的测试、最终上线投入使用后的成果。最后进行总结，并展望前景。

第二章 研究背景

2.1 互联网时代背景

随着网络信息技术的不断发展, 社会已经迈入了互联网时代。互联网公司应运而生, 诸多互联网应用也随之发展, 人们的生活也已经离不开互联网。以前, 互联网只会让人联想到浏览网站、搜索资料、观看视频等, 而现在, 互联网已经普及到人们生活的方方面面, 给人们带来了极大的便利。现在的人们, 可以利用互联网购物、支付、理财、发微博、聊天、点外卖、订机票订车票订酒店等, 大大提高了我们的效率, 从前我们需要外出逛街吃饭, 去车站买车票, 而现在我们仅需要打开电脑网页或手机 APP 简单地操作几下即可。在享受这些便利的同时, 人们也在追求互联网更加广阔的使用途径, 期望能使互联网在更多的领域解决我们的问题, 让我们有更多的时间来花在有意义的事上。

2.2 项目使用的技术

2.2.1 Maven

Maven 是一种用于构建、管理项目并且能够执行构建任务的工具。Maven 对于使用大量三方库, 并寻求一种方法来管理和构建大量相互依赖的模块和类库的项目不可或缺。它可以减轻维护第三方依赖的负担, 大大减轻构建管理项目的工作量, 构建和维护软件不再会限制软件设计了。

Maven 包含了 POM 也就是项目对象模型, 项目生命周期, 标准集合, 项目依赖管理系统和用来运行在生命周期中定义的插件目标的逻辑。使用 maven 时, 开发者可以用一个明确的 POM 来对项目进行说明, 然后 Maven 可以应用一组来自于共享或自定义的插件的逻辑。^{[1][2]}

(1) Maven 的功能特征:

a. 依赖管理: 一个项目是按照含有组标识符, 构件标识符和坐标定义的。项目间可以使用这些坐标来声明依赖。^[3]

b. 远程仓库: 可以使用项目中 pom.xml 文件中的坐标来创建 Maven 构件的仓库。

c. 全局性构建逻辑重用: 插件没有被设计成对某位置的已知文件进行操作, 而是被和工程中的 pom.xml 文件一起工作。开发者的自定义操作和插件的配置都抽象到模型中, 在模型中进行。

d. 工具可移植性/集成: 类似于 Eclipse, NetBeans 等等工具现在可以使用相同的方法来找到项目的信息。曾经每个开发环境都有各自的方法来存储其实是自定义项目的 pom.xml 文件的信息。Maven 把这个流程标准化了。尽管每个开发环境依然可以接着维护它的自定义项目文件, 但这些文件现在能够直接通过模型生成。

e. 便于搜索和过滤构件: 另外有一些工具可以使开发者利用存储在模型对象中的数据来搜索仓库内容。

(2) Maven 的基本结构

Maven 的基本单元是项目, 每个有 POM 文件的目录就是一个项目, 项目可以对其他很多项目有依赖, 也可以由多个子项目构成。Maven 的基本结构包括仓库和 POM, 仓库分为

本地仓库和远程仓库，可以用生命的形式进行配置。Maven 从远程的仓库将 POM 文件中定义的包下载到本地仓库，所有应用使用同一个本地仓库的包，同时它采用了插件体系架构，保留最小的核心，其余功能都通过插件提供。另外，项目也可以生产自己的 jar 包然后部署到远程仓库中。

（3）Maven 和 Ant 的差别

Ant 拥有十分出色的构建过程，是一个依靠依赖和任务而形成的构建系统。每个任务包含一组 xml 格式的命令。Ant 的任务包括有 copy 任务、javac 任务和 jar 任务。

Ant 没有正式规定类似于普通项目的目录结构，程序员必须明确地通知 Ant 源代码在何处，以及输出放置在何处。对这些结构的不正式的约定在 Ant 中也慢慢出现了，但这些约定还没有在产品中模式化。

Ant 高度依赖于程序，它根据开发者所说明的时间和内容来完成工作。比如，在构建项目时，编译、复制、压缩等工作都必须有开发者来指定。

Ant 没有生命周期，目标之间的依赖都需要由程序员来进行约束。每个目标都需要一个任务序列才能够正常运行。

Maven 约定了项目的目录结构，它可以知道源代码的位置。把字节码放到 target/classes，然后在 target 目录下生成一个 JAR 文件。

Maven 是声明式的。开发者只需要创建 POM 文件然后将源码放到约定里的位置，其他的事情都由 Maven 来完成。

Maven 有一个周期，当执行命令 mvn install 的时候生命周期就被调用。Maven 会根据这条命令来执行一系列有序任务，直到到达指定的生命周期。遍历生命周期会让 Maven 运行一些插件目标，编译和创建一个 JAR 文件等工作就是由这些目标实现的。

2.2.2 WebX 框架

WebX 是一套基于 Java Servlet API 的通用 WEB 框架，目前是阿里巴巴使用的主流框架，类似于 SSH (Struts, Spring, Hibernate) 这套经典框架中的 Struts 框架，支持与 Spring 框架的结合。^[4]

（1）框架架构

WebX 分为三个大层次，分别为最基础的 SpringExt 层、WebX Framework 层以及最外层的 WebX Tuebine 层。

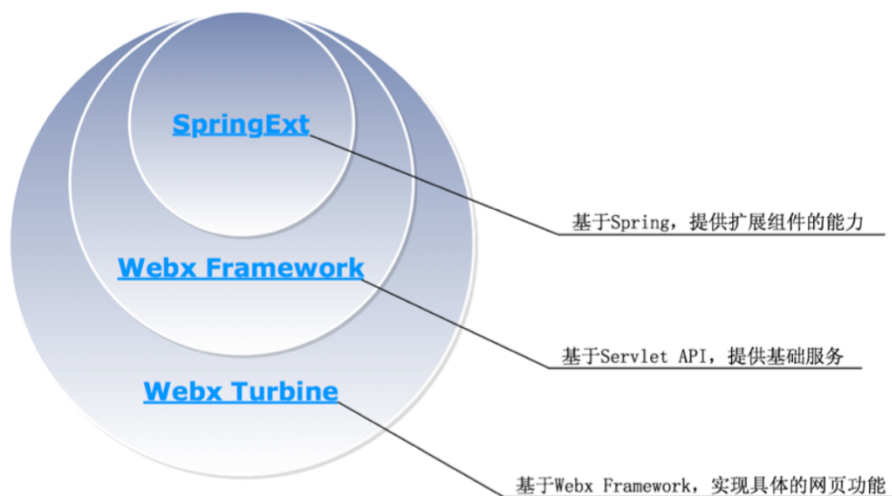


图 2-1 WebX 框架架构

SpringExt 层是以 Spring 为基础发展而来的，它是 webx 框架的基础，为整个框架提供扩展组件的能力。Spring 是适用范围很广的轻量级框架，SpringExt 提供了比 Spring 更加强大的扩展能力，但是没有损失 Spring 的任何功能。在工程中，可以通过 SpringExt 实现资源的装载。在还没有 Spring2.0 时，为了装载资源，必须通过装配 beans 的方法，举个例子，在用 Spring Beans 装配 Resource Loading 服务时，ResourceLoadingServiceSpring 的服务使用者利用 Spring 把 ResourceLoadingService 和 ResourceLoader 等其他服务装配在一起，使之可以协同工作，然而 Spring 的配置文件会依赖于服务实现类的公开 API，装配者要知道 API 的细节必须通过查看源代码或是 API 文档，这可能会导致没有校验机制、无法了解更多约束条件、服务改变时配置文件失效等问题。^[5]在 Spring2.0 中，这种情况得到了改善。Spring2.0 在定义配置文件时候支持使用 XML Schema，这种配置方式比起 Spring Beans 来更加简单并可以验证，不需要等到项目运行就可以验证配置的正确性，另外，XML Schema 包含更多的约束条件，会显示参数是否可选、是否必填以及参数类型。最重要的是，装配者是不知道服务是如何实现的，当服务的实现发生变化时，只要 XML Schema 没有变化那 Spring 的配置就不会受到影响。服务的提供者完成提供类名等等与具体实现有关的事，这样服务的使用者就能用更容易的方式来对服务进行装配，这是 Spring Schema 带来的核心价值。但是它还是有一个问题即它是不可扩展的。同样使用 ResourceLoadingService 为例，在 API 层面，ResourceLoadingService 支持任何对 ResourceLoader 接口的扩展，但在 Spring 配置文件上无法自由添加新元素。因此，SpringExt 改进了 Spring 使得 Spring Schema 可以被扩展。要添加新的 ResourceLoader 扩展，仅需要将扩展所在的 jar 包添加到项目依赖中，然后在 Resource Loading 服务中添加新的装载器，这样就无需通知 ResourceLoadingService 的作者去修改 Schema，一种全新的 ResourceLoader 扩展被注入到 ResourceLoadingService 中。

WebX Framework 开始真正涉及 WEB 方面的技术，负责完成一系列基础性的任务，如系统初始化时初始化 Spring 容器和日志系统，在响应请求时增强 request, response 和 session 功能，提供 pipeline 流程处理机制，异常处理以及开发模式等。在对 Spring 容器进行初始化操作时，WebX Framework 会生成一组级联的 Spring 容器结构，WebX 所创建的 Spring 容器可以被所有以 Spring 为基础的 WEB 框架使用，因为它很好地兼容了 Spring MVC 创建的容器。WebX Framework 将一个网络应用分成很多个更小的应用模块，各个模块都独自有一个 Spring 子容器，两个 sub context 中的 beans 不能互相注入，每个小模块共同拥有一个根容器，它的 beans 可以被注入到子容器中。当遇到请求的时候，WebX Framework 先会对 request, response 和 session 的功能进行增强，然后把他们打包成 RequestContext 对象，这个对象更加易于使用，然后它会调用相对应的子应用的流水线，从而可以做下面的处理。流水线在 WebX Framework 中尤为重要，它可以定制处理请求的流程，它其中定义了许多的阀门，用来控制流程的走向。

WebX Turbine 是在 WebX Framework 的基础上建立起来的，实现了页面的布局和渲染、数据验证和提交等一系列工作。WebX Turbine 遵循页面驱动和约定胜于配置这两个设计理念。页面驱动是指在程序员在还没有参与到开发时，界面的设计师可以直接创建页面模板。这不仅提高了开发效率，也使得设计师在开发前期就可以利用框架所提供的工具完成页面跳转以及表单的提交验证等等操作。约定胜于配置是指开发者只要按照约定的规则开发，不用进行其他的配置操作，而在别的框架中每当需要添加一个页面就要在 config 文件中增加相应的部分。^[6]WebX Turbine 的规则约定了映射方面。

将 URL 映射成 target: target 是一个抽象的概念，说明了这个请求所应该完成的任务。Target 由 pipeline 解释，有可能被解释成模板的名字，也可能被解释成其他。

将 target 转换成模板名：模板用来展现页面的内容。Velocity、Freemarker、JSP 都可以作为模板的格式,但在 Webx 建议使用 velocity 模板。

将 target 转换为 layout 布局：可以为一组页面选择一组布局（包括有导航栏、菜单、页首页尾等），然后为另外一组页面选择另外一种布局。

将 target 转化为 module：在 Webx Turbine 中，module 是指 screen、action、control 等，大致相当于其它框架中的 action 或者 controller。

开发者只要遵从这些规则，将 template 放在特定的目录，然后根据约定中的方法对 screen、control、action 等模板进行命名，就不需要再做其他配置。

（2）工作流程

在 WebX 执行工作时，首先，用户输入 url，webx 拦截到请求，接着在 root-context 中封装 request 和 response，并解析参数，根据 url 分析由哪一个 context 处理该请求，找到具体的 context 后，将 requestContext 交给具体的子 context 处理，进入具体的 sub context 后生成一个 pipeline。

在 webx 执行模板渲染时，首先在执行完 module 类的时候，如果需要渲染模板则查找模板，找到模板后开始渲染，如果在 screen 中有 control 模板，那么执行 control 对应的类，渲染 control，以此类推，并完成 screen 的内容渲染，然后查找 layout。接着，将 screen 渲染生成的字符串，作为 layout 的一个变量，渲染模板，如果在 layout 里面有 control 的话，那么按照片 screen 的处理方式渲染，渲染完成后，将整张页面刷到客户端。

2.2.3 HSF

HSF 的全称是 High-Speed Service Framework，目的是为应用提供高速稳定的分布式服务框架。HSF 从分布式应用层面以及统一的发布与调用方式层面为开发者提供技术支持，开发者可以很容易地开发分布式的应用以及提供公用功能模块，而不用考虑分布式领域中的各种细节技术，例如同步和异步的调用方式的实现、进程之间的通讯、调用是否透明、性能损耗等等问题。HSF 总体上提供了远程调用的服务方式、软负载体系和服务可用性保障三大功能。^{[7][8][9]}

（1）工作原理

HSF 的部署模型如图 2-2 所示，在 JBoss 中，HSF 与应用一起存在于 JBoss 的 deploy 目录中，并且与应用相互独立。

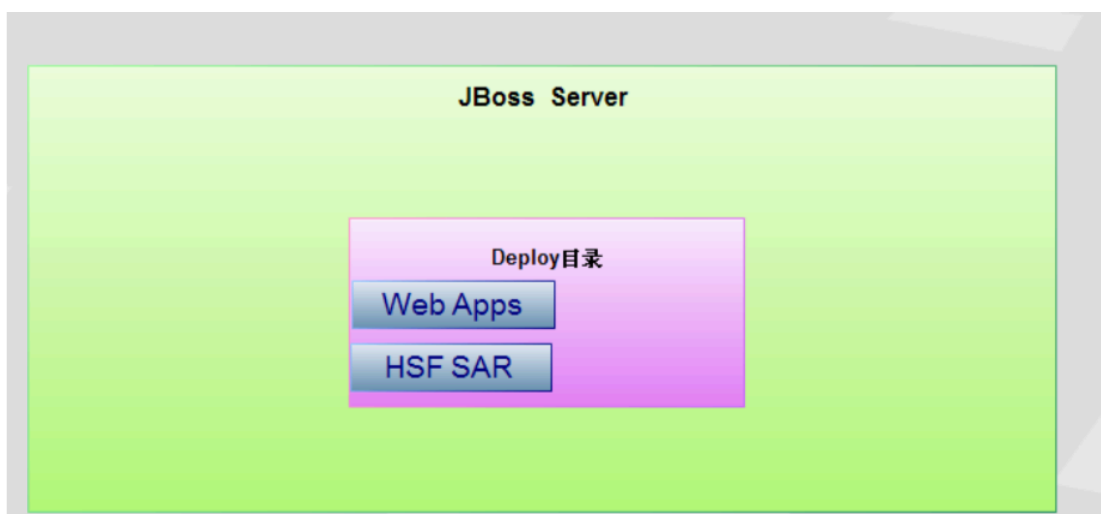


图 2-2 HSF 部署模型

HSF 由三个部分组成,分别是服务的提供者、服务的调用者以及居中调度的 config server。

在 HSF 发布服务时 JBoss 服务器启动后,会启动 HSF 的 sar 应用,接着应用自身启动, Spring 容器初始化。这时 HSFSpringProviderBean 会进行初始化,在初始化的过程中将向 config server 注册要这个 bean 服务,这一注册过程是服务的提供者对 config server 说 IP 为某某的机器提供了某服务, config server 就可以根据服务调用者的请求中的服务名来转发推送服务地址。

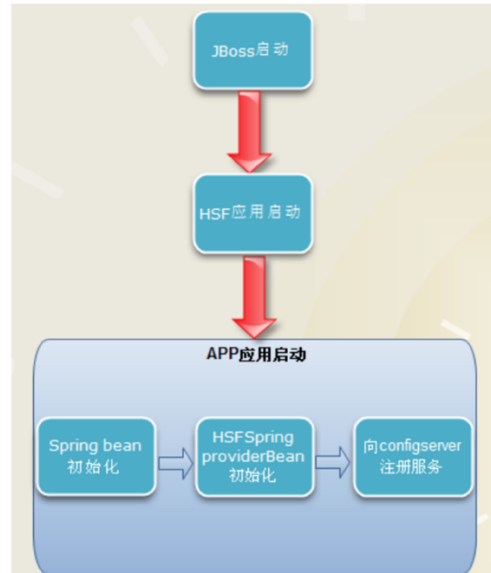


图 2-3 HSF 服务发布过程

HSF 的服务调用时,服务器、应用和 Spring 容器的启动类似,不同之处在于 HSFSpringConsumerBean 的初始化。与 HSFSpringProviderBean 向 config server 注册服务不同,调用服务是向 config server 订阅服务地址,简而言之就是服务的调用方告诉 config server 需要某服务的地址, config server 就会根据这一服务名将相应的服务地址返回给这个应用。

服务的调用者获取到服务的地址后,可以使用获取到的服务地址调用服务。一般来说,一个 HSF 服务是由很多台机器来提供的,因此调用方经常会订阅到一个地址列表,这个地址列表里有所有提供这个服务的地址, HSF 会任意选择其中一个来调用。^[10]

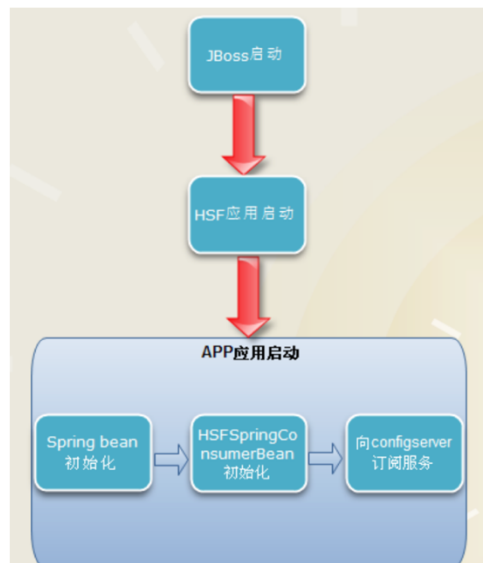


图 2-4 HSF 服务调用过程

2.2.4 TDDL

TDDL 是用于管理数据库的分布式数据库中间件。

数据库存储一般分为三个部分，顶层的用户 API、中间层的关系代数的事物引擎以及底层的 K-V 存储。

K-V 存储本质来说就是映射，按照 key 找到 value，是所有数据存储最基本和最底层的结构。关系代数指的是如何按照 key 找到对应的数据。^[11]在 TDDL 中采用多机 Key-Value 存储，多机存储的特点是可运维、高性能并且可以比较容易地扩容。比如 MongoDB 采用的是 mongos 服务器，Hbase 采用的是 region server + client jar 包，而 TDDL 采用的都是 tddl-rule 组件+数据迁移。在路由方面规则引擎采用 Groovy 脚本实现，本质为了实现多版本的规则推送，内奸对 3 种 hash 模式的支持，并且允许从简单的 hash 平滑的过渡到一致性哈希和虚拟节点哈希。

TDDL 的结构分为三层如图所示，分别是 matrix 层，group 层和 atom 层，matrix 层的核心是规则引擎，可以单独抽取出来放在其它实现里，并且该层通过规则引擎实现了动态扩容，主要路径为 sql 解析-规则引擎计算-数据执行-合并结果。Group 层主要实现了读写分离，并且允许动态添加新的 slave 节点。Atom 层是单个数据库的抽象，拥有动态化的 Jboss 数据源，ip port 用户名密码都可以动态修改，拥有 Thread count 模式，保护业务的处理线程，超过指定值时保护启动，另外它可以动态阻止某个 sql 的执行以及执行次数统计和限制。^[12]

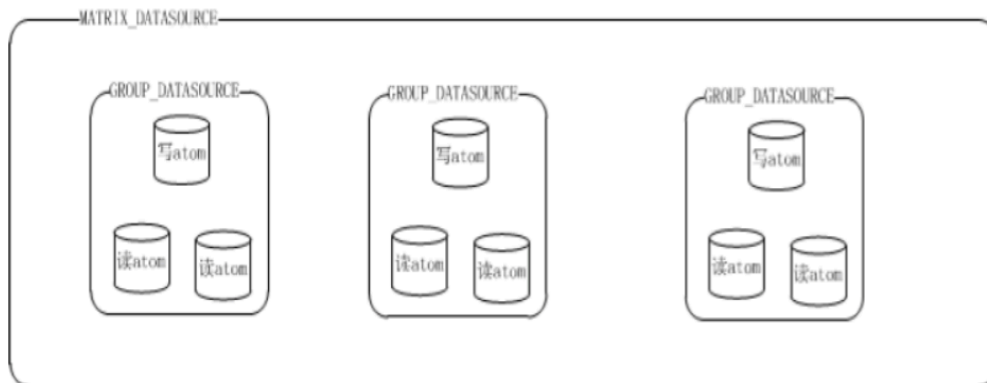


图 2-5 TDDL 的架构

2.3 本章小结

本章介绍了自助打印系统的背景以及运用到的技术。首先提到了互联网应用的时代背景，也就是自助打印系统需求提出的大环境，接着介绍了系统实现中所需要用到的几种技术，先介绍了项目管理工具 maven，包括它的特征、结构以及 maven 和 ant 的区别，再介绍了自助打印系统使用的架构 webx，主要详细介绍了其框架结构中的三个部分以及 webx 的工作流程，然后是关于远程过程调用服务的 HSF，主要介绍了其工作原理，最后介绍了管理分布式数据库的中间件 TDDL 的相关信息。

第三章 分析和设计

3.1 自助打印系统的需求和目标分析

整个系统的业务需求为员工在机器上刷身份证或者手动输入身份证号进行登录，与此同时系统向员工的手机发送验证码，员工在屏幕上输入验证码登录进入。登陆后打印机屏幕上显示该员工可打印的所有证明类型，根据类别以树状形式展现，员工点击选择某一种证明后，跳转至下一级页面，页面显示该证明缩略图（其中已经包含该员工信息）、打印预览按钮、选择打印份数按钮以及打印按钮，员工选择份数并点击打印后，打印机判断该员工所在公司并确定纸盒后从该纸盒打印出已经加盖好公章的员工所需证明，如若员工打印的是收入证明，则在员工打印完成后将该员工打印的证明的电子版发送到系统邮箱以便银行核查。

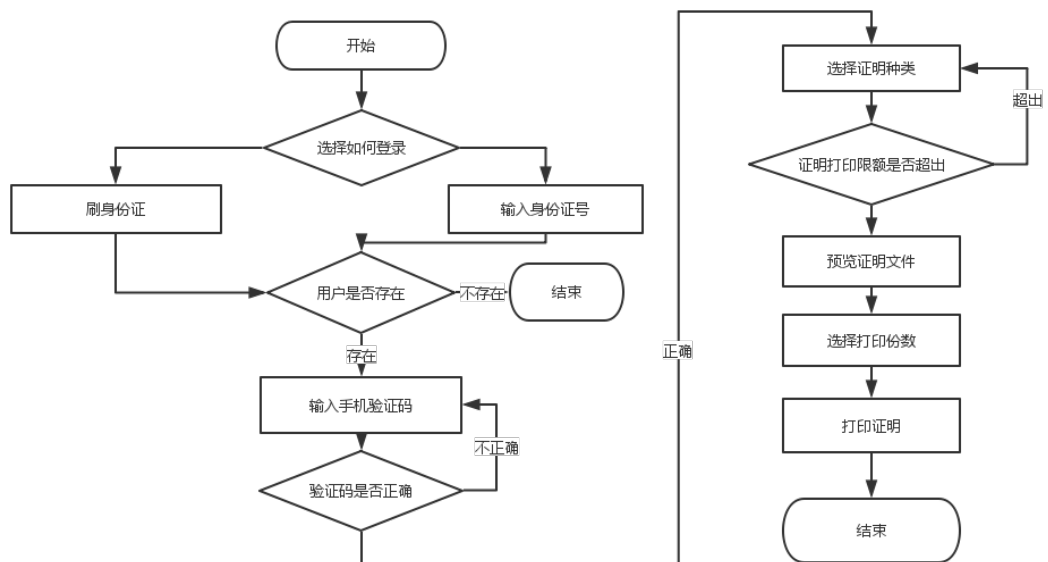


图 3-1 系统使用流程

3.1.1 系统硬件需求

表 3-1 自助打印系统硬件需求

项目	要求
设备尺寸(长*宽*高, 单位: mm)	850* (612--900) * (1500-1763)
打印盒	6 个, 至少每个 250 张 A4 纸, 需带锁管理纸张数量
打印盒设置	自助设置打印盒对应的章
设备本身是否自带监控	要求设备有摄像头, 在有人靠近时录像并存储

续表 3-1

项目	要求
有友好界面显示屏	可设定选择键进行组合选择，能呈现预览状态，能完成打印输出确认
设备外形图片	可根据需求选择最佳外形
机器自带按钮	数字+确认 OR 删除 按钮
UPS 支持	设备支持 UPS 保护装置；
纸张预警	设备支持纸张预警保护装置；支持纸张打印统计

3.1.2 系统软件需求

表 3-2 自助打印系统软件需求

项目	要求
登录方式	手机验证码+身份证原件扫描或身份证号码手动输入。
界面内容显示	可支持 3 重以上需求选择（证明类型+模板+公司），直接生成文档后，可预览，确认后可选择打印输出，打印完毕后系统自动删除文档。
	打印文档份数可根据证明类型的不同系统有设置。
	证明可以由 HR 服务中心常用证明处申请（有假期要求的），也可以是在自助打印机上直接申请打印的，登录后，界面需要直接显示需求状态
	从选择类型到打印完成时间大概在 10 秒内（内网）
	展示类目、模板由网络交互传递给设备。
	身份证+验证码登录服务器，自动退出的功能：所有的用户 40 秒没有操作自动退出。自动退出后文档删除，再预览需要重新刷验证码。
	后台确认每个证明的打印分数，超过后，告知今日不能再行操作（提供展示文档）。
系统内置接口	可开发完成与阿里假期系统、薪酬系统、主数据系统数据对接
文档模板及公司名称设置	由管理员可随时增加或修改证明类型文档，文档模板，及纸盒对应公司的设定修改
报错功能	系统有卡纸等故障，自动报警
	系统有纸张数量使用预警功能，纸张还剩*% 以下，触发邮件或者短信给管理员提醒加纸
安全性	打印完成后证明文件不在打印机内留存；员工各项数据不在系统内留存；收入证明中的薪资信息应经过严格的加密解密验证；
备份功能	每天设定一个时间点，系统将当天所有打印并加密后上传到阿里内部云端备份并在系统内部删除这些文档；

续表 3-2

项目	要求
要求能够支持的证明类型	在职证明、实习证明、收入证明（包括本年度收入证明（税前、税后）、上年度收入证明（税前、税后）、近六个月收入证明（税前、税后）、近十二个月收入证明（税前、税后）、合同工资收入证明）、介绍信、要事先经过线上审批才能打印的包含员工请假记录的签证信息证明；
报表功能	每天设定一个标准时间点，系统自动按照需要的报表，发邮件给管理员，记录整体的使用情况：明细时间、打印人员姓名、纸张的使用情况，公司分布情况，证明类别、打印份数等相应记录
	周、月、年都可要求设置自动打印总体情况、分类证明的数据报表

3.2 自助打印系统的设计与架构

3.2.1 系统交互流程

由于在该项目中，供应商提供硬件设备，因此我只需要提供向设备传输数据的接口即可。根据需求，设计了如下流程：

（1）用户使用

a. 用户在打印机上刷身份证或手动输入身份证号后，打印机调取系统接口，将身份证号发送给系统，系统接收到身份证号之后获取到该员工的信息，并向员工手机发送验证码，同时将验证码以及该员工的个人信息发送到打印机端。

b. 打印机端收到系统发的验证码以及员工个人信息后，匹配员工在屏幕上输入的验证码是否与系统发送的一致，若一致则登陆成功，将员工工号发送到系统服务器以请求接下来的操作。

c. 系统服务器收到登录成功的员工工号后，根据内部逻辑计算出员工可打印的证明列表，并以树形结构发送到打印机端（证明类型结构[type_id, type_name, parent_type_id]）。

d. 打印机收到服务器端传来的证明列表后展示在屏幕上供用户选择，用户选择后打印机向服务器发送员工工号以及员工选择的证明 id，系统收到信息后从数据库中获取该证明每周以及每日可以打印的份数限制，然后从打印日志中获取该员工在时间段内打印成功的份数，计算出该员工本次可打印的份数，生成本次打印请求的唯一验证码，并将该员工的个人信息填至证明模板内生成预览文件，将可打印份数、验证码以及预览文件发送到打印机端。

e. 打印机收到可打印的份数和预览文件后展示在屏幕上供用户检查并选择，然后根据员工签约公司进行打印操作，打印成功后，打印机将该用户的打印份数连同之前服务器发过来的验证码一同发给服务器，服务器根据验证码识别此次打印的相关信息后在日志中更新状态为打印成功。

（2）管理员使用

a. 在打印机出现问题，如纸盒缺纸、墨盒缺墨等时，打印机将报错信息发送到系统服务器，系统服务器整合报错信息后自动发送邮件给管理员。

b. 每天固定一个时间点，系统将当天的打印日志发送到每个管理员的邮箱，包括明细时间、打印人员姓名、纸张的使用情况，公司分布情况，证明类别、打印份数等相应记录。

c. 每天一个固定时间点，系统将当天打印过的所有文档压缩加密并上传到阿里内部云端，并在系统内把这些文档删除。

3.2.2 系统软硬件交互接口设计

根据系统的使用流程，设计了六个接口来完成打印机与系统服务器之间的交互。

(1) getSealType()

接口说明：获取所有公章类型。

返回的参数为：

- a. 查询到的公章类型。
- b. 异常返回错误信息。

(2) getWorkerInfo(String creditId)

接口说明：自助打印机通过传入身份证 Id 获取员工的个人信息，向员工手机发送验证码，将验证码存到数据库中，将验证码发送到打印机。

传入的参数为：

- a. creditId 身份证号码。

返回的参数为：

- a. 查询到员工的信息，返回验证码、工号、姓名、公司名称以及员工可使用的公章。
- b. 异常返回错误信息。

(3) getTypeList(String workNo)

接口说明：若打印机返回登录成功的工号，则向打印机返回可以打印的证明列表。

传入的参数为：

- a. workNo 登录成功的工号。

返回的参数为：

- a. 验证成功，返回员工可打印的证明列表（证明类型结构 [typeId, typeName, parentId]）。
- b. 异常返回错误信息。

(4) checkTypeLimit(String workNo, String typeId)

接口说明：自助打印机传入员工工号和要打印的证明 id，获取可以打印次模板的份数，若可打印的份数为零返回报错信息。生成验证码以及证明文档并返回。

传入的参数为

- a. workNo 员工工号。
- b. typeId 要打印的证明类型 id。

返回的参数为

- a. 正常则返回可以打印的份数（报错信息）以及生成的验证码、证明文档。
- b. 异常返回错误信息。

(5) closeApply(String verifyCode, Integer printNumber)

接口说明：打印机打印成功，将校验打印份数时生成的验证码返回给服务器端。服务器获取此次打印的信息，在数据库中更新状态为打印成功。

传入的参数为：

- a. verifyCode 校验打印份数时生成的验证码。
- b. printNumber 此次打印的份数。

返回的参数为：

- a. 正常。
- b. 异常返回错误信息。

(6) alert(String type, String msg)

接口说明：打印机纸张存量不足或有其他问题，向平台发送报警信息，平台向管理员发送短信和邮件。

传入的参数为

- a. type 报警信息的类型（供应商约定后告知）。
- b. msg 打印机向平台发送的报警信息。

返回的参数为：

- a. 正常。
- b. 异常返回报错信息。

以上所有接口都需要通过签名验证来保证接口的安全性，调用方使用私钥在参数中添加签名，接收方使用私钥验证签名。

3.2.3 系统间交互

由于在系统需求中提到需要支持收入证明以及要事先经过线上审批才能打印的包含员工请假记录的签证信息证明，因此系统需要与公司其他内部系统进行交互来获取员工相关信息。需要参与交互的系统有主数据系统（记录员工的基本信息）、假期系统（记录员工已提交的请假记录）、薪酬系统（记录员工的薪资信息）、证明申请系统（记录员工证明申请信息）。

交互接口：

(1) Employee360Service.getWorkerBaseInfo

接口说明：根据工号获取员工基本信息，如员工姓名、签约公司、工作地等。

对接系统：主数据系统。

传入的参数为：

- a. workNoList 要查询的所有工号。
- b. clientId 服务调用方的名字。
- c. secret 服务调用方密钥。

返回的参数为：

- a. 正常则返回查询员工基本信息类。
- b. 异常返回错误信息如在公司数据库内查询不到该工号。

(2) EmployeeLicenseService.getWorkerLicenseInfo

接口说明：根据身份证号查询员工信息（工号）。

对接系统：主数据系统。

传入的参数为：

- a. licenseNumber 要查询的身份证号。
- b. clientId 服务调用方的名字。
- c. secret 服务调用方密钥。

返回的参数为：

- a. 正常返回员工工号等基本信息。
- b. 异常返回错误信息如在公司数据库中查询不到该身份证号。

(3) EmployeeVacationInfoService.getVacationListByBeginTime

接口说明：根据员工工号以及员工请假时的假期开始时间获取假期结束时间。

对接系统：假期系统。

传入的参数为：

- a. workNo 要查询的员工工号。
- b. startDate 员工请假的假期开始日期以 yyyy-MM-dd 格式。
- c. appName 服务调用方名字。
- d. authkey 服务调用方密钥。

返回的参数为：

- a. 正常则返回员工请假记录中查询条目的请假结束时间。
- b. 异常则返回错误信息如在系统中找不到该条假期记录。

(4) IncomeCertificationService.queryContractIncomeInfo

(5) IncomeCertificationService.querySixMonthIncomeInfo

(6) IncomeCertificationService.queryTwelveMonthIncomeInfo

(7) IncomeCertificationService.queryThisYearIncomeInfo

(8) IncomeCertificationService.queryLastYearIncomeInfo

接口说明：根据员工工号获取员工薪资数据。

对接系统：薪资系统。

传入的参数为：

- a. workNo 员工工号。
- b. clientId 服务调用方名字。
- c. secret 服务调用方密钥。

返回的参数为：

- a. 正常则返回相应的加密过后的薪资数据。
- b. 异常返回错误信息如工号错误等

(9) BaseInfoService.queryIncomeApplyBaseInfoByWorkNo

接口说明：根据员工工号获取员工申请收入证明表单。

对接系统：员工证明申请系统。

传入的参数为：

- a. 员工工号。
- b. clientId 服务调用方名字。
- c. secret 服务调用方密钥。

返回的参数为：

- a. 正常则返回最近一次员工申请的收入证明的表单相关信息。
- b. 异常返回报错信息。

3.2.4 service 类设计

根据前面提到的接口以及需求，设计了如下 service 类以便使用。

(1) SignUtilService 在打印机端调用这些 RPC 接口时提供签名验证服务，以判断这次调用是否有效。

传入的参数为：

- a. linkString 调用 PRC 接口时传参连接成的字符串。
- b. sign 打印机端生成的签名。
- c. publicKey 签名公钥。
- d. inputCharset 编码格式，一般为 utf-8。

返回的参数为：

- a. 正常则返回布尔类型的验证结果。
- b. 异常返回报错信息。

(2) EmailService 提供邮件发送服务。

传入的参数为

- a. receiver 邮件接收人列表。
- b. title 邮件标题。
- c. content 邮件正文。
- d. file 邮件附件（可选）。
- e. filename 邮件附件名（可选）。

返回的参数为：

- a. 邮件发送结果。

(3) ModuleInfoService 获取到证明模板内所需要的所有变量信息的服务。

传入的参数为：

- a. workNo 员工工号。
- b. typeId 证明类型。

返回的参数为：

- a. 正常则返回该证明类型所需要的该员工的所有信息。
- b. 异常返回报错信息。

(4) MasterDataInvocationService 调用主数据接口获取员工基本信息的服务。

传入的参数为：

- a. 员工工号。

返回的参数为：

- a. 正常则返回该员工的所有基本信息。
- b. 异常返回报错信息。

(5) PDFService 根据证明模板和参数信息生成证明的 PDF 文档的服务。

传入的参数为：

- a. template 证明模板。
- b. parameter 参数列表。

返回的参数为：

- a. 正常则返回生成 pdf 文档文件流。
- b. 异常返回报错信息。

(6) CheckPrintLimitService 校验本次能够打印的数量限制的服务。

传入的参数为：

- a. typeId 需要检查的证明类型。
- b. workNo 需要检查的员工工号。

返回的参数为：

- a. 正常则返回可打印的最大数量。

(7) TypeListService 获取该员工能够打印的证明列表的服务。

传入的参数为：

- a. workNo 员工工号。

返回的参数为：

- a. 正常则返回员工可打印的证明列表。
- b. 异常则返回报错信息。

(8) MessengerService 提供发送短信功能的服务。

传入的参数为：

- a. receiver 短信的收件人列表。
- b. parameter 短信模板中变量的值。

返回的参数为：

- a. 短信发送的状态。

(9) VacationService 调用假期系统接口获取员工请假信息的服务。

传入的参数为：

- a. workNo 需要查询的员工工号。
- b. startDate 需要查询的假期条目开始时间。

返回的参数为：

- a. 正常返回需要查询的假期条目结束时间。
- b. 异常返回报错信息。

(10) IncomingCertificationService 调用薪酬系统接口获取员工薪资信息的服务。

传入的参数为：

- a. workNo 需要查询的员工工号。

返回的参数为：

- a. 正常则返回需要查询的员工薪资。
- b. 异常返回报错信息。

(11) BaseInfoService 调用证明申请系统获取员工证明申请信息的服务。

传入的参数为：

- a. workNo 需要查询的员工工号。

返回的参数为：

- a. 正常则返回需要查询的员工证明申请表单信息。
- b. 异常返回报错信息。

同时还有两个定时任务

UploadGiantFileJob 向云端发送加密大文件的任务。

LogSendJob 每天向管理员发送打印日志的任务。

3.2.5 数据库设计

根据上文提到的需求，一共设计了五张数据库表以便使用。

表 3-3 证明类型表 auto_print_type

字段名	字段类型	字段介绍
id	integer	主键，在表内唯一 id
gmt_create	date	该字段创建时间
gmt_modified	date	该字段最近一次修改时间
creator	varchar[32]	创建人
modifier	varchar[32]	修改人
is_deleted	varchar[8]	是否删除
type_id	integer	类型编号
parent_type_id	integer	父节点类型编号

续表 3-3

字段名	字段类型	字段介绍
daily_limit	integer	该证明每日打印限制
monthly_limit	integer	该证明每月打印限制
limit_comment	varchar[64]	该证明打印限制说明

以 type_id 为索引。

表 3-4 打印记录表 auto_print_log

字段名	字段类型	字段介绍
id	integer	主键，在表内唯一 id
gmt_create	date	该字段创建时间
gmt_modified	date	该字段最近一次修改时间
creator	varchar[32]	创建人
modifier	varchar[32]	修改人
is_deleted	varchar[8]	是否删除
work_no	integer	员工工号
print_time	date	打印时间
type_id	integer	打印的证明类型
print_number	integer	打印份数
type_name	varchar[64]	打印的证明名称
status	varchar[16]	打印状态
verify_code	varchar[32]	验证码

以 verify_code 为索引。

表 3-5 证明模板表 auto_print_module

字段名	字段类型	字段介绍
id	integer	主键，在表内唯一 id
gmt_create	date	该字段创建时间
gmt_modified	date	该字段最近一次修改时间
creator	varchar[32]	创建人
modifier	varchar[32]	修改人
is_deleted	varchar[8]	是否删除
type_id	integer	证明类型编号
detail	varchar[4096]	证明模板内容
param	varchar[256]	证明模板中的参数

以 type_id 为索引。

表 3-6 报警信息表 auto_print_alert

字段名	字段类型	字段介绍
id	integer	主键，在表内唯一 id
gmt_create	date	该字段创建时间
gmt_modified	date	该字段最近一次修改时间
creator	varchar[32]	创建人
modifier	varchar[32]	修改人

续表 3-6

字段名	字段列表	字段介绍
is_deleted	varchar[8]	是否删除
address	varchar[64]	打印机位置
time	date	出错时间
description	varchar[64]	错误介绍
name	varchar[32]	错误名称
code	varchar[32]	错误代码
solution	varchar[64]	解决方案

表 3-7 公章信息表 auto_print_seal_type

字段名	字段类型	字段介绍
id	integer	主键，在表内唯一 id
gmt_create	date	该字段创建时间
gmt_modified	date	该字段最近一次修改时间
creator	varchar[32]	创建人
modifier	varchar[32]	修改人
is_deleted	varchar[8]	是否删除
seal_id	integer	公章编号
seal_name	varchar[32]	公章名称
company_no	varchar[16]	公章所对应的公司编号

3.2.6 系统架构

本系统采用 webx 作为基本架构，使用 maven 进行项目管理，六个接口分别由一个 RPC 类构成，在 RPC 类中完成信息的传输，RPC 类通过调用服务类来实现功能，每个服务类通过与其他系统交互以及一些运算逻辑来实现一部分功能，从而为接口类提供服务，同时，服务类需要通过 DAO 层来与数据库进行连接，从而实现与数据库的数据交互。

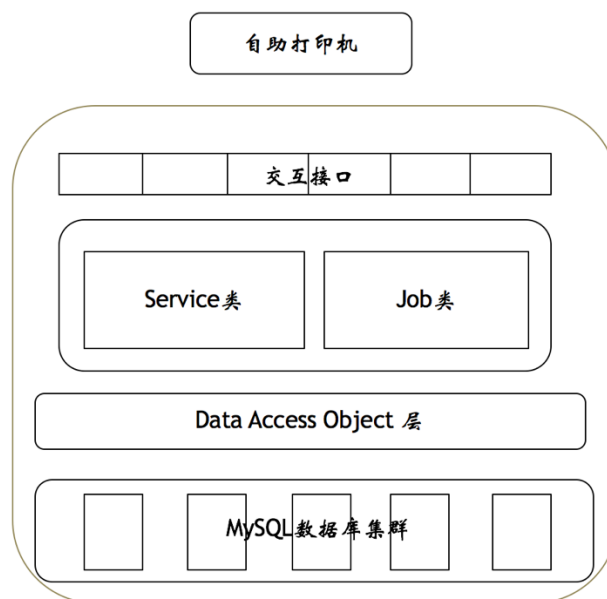


图 3-2 自助打印系统架构

3.3 开发计划

在完成所有类的设计、数据库的设计以及架构设计后，制定出软硬件具体的开发计划表。

表 3-8 服务器系统开发计划

编号	样例		说明	时间（天）
	模块	名称		
1	架构	服务器搭建	搭建 tomcat 服务器	2
2		框架搭建	使用 webx 框架加上 maven 完成系统大体架构搭建	3
3		数据库搭建	使用 MySQL 进行数据库搭建并建表	1
4		DAO 层实现	实现项目和数据库表之间的连接	1
5	服务类	发送邮件	向外发送能够带附件邮件的功能	2
6		发送短信	向外发送短信的功能	2
7		员工信息	获取员工基本信息	1
8		假期信息	获取员工请假记录信息	2
9		薪资信息	获取员工薪资收入数据	3
10		申请表单信息	获取员工申请证明时填写的表单信息	2
11		PDF 转换	将文档模板和参数组合起来并转化为 PDF 文档	2
12		模板参数	获取员工要打印的模板的所有参数	4
13		证明列表	获取员工可以打印的证明列表	2
14		打印限制	获取员工选中的证明在当日的打印份数限制	1
15		签名验证	校验打印机发来的签名正确性	2
16	定时任务	报表	每天向管理员发送报表信息	3
17		备份	每天向云端上传备份打印文档	5
18	接口	公章信息	发送所有的公章信息	1
19		登录	向打印机发送员工信息使员工登录系统	1
20		证明列表	发送员工可打印的证明列表	1
21		打印文件	发送员工可打印此证明的份数并发送文档	2
22		打印结束	打印完成，关闭此次任务	1
23		报警	收到打印机发来的报警信息	1

表 3-9 打印机硬件开发计划

编号	样例		说明	时 间 (天)
	角色	名称		
1	普通用户	欢迎	显示欢迎	1
2		登录	身份证原件/身份证号码输入+获取验证码登录	2
3		选择打印	显示可打印文档列表	3
4		预览	选定可打印文档进行预览，可以设置成按缩放比例进行预览	3
5		报警信息	当机器有异常，显示异常信息(异常信息获取)，提示用户操作	6
6		打印	从打印机输出文档	5
7		退出登录	退出界面，并清理用户数据	2
8	管理员	管理员登录	输入密码直接登录界面	2
9		统计	按照给定条件进行报表输出	8
10		印章设置	把印章对应到输出窗口，其中一个打印出口对应两个不同的纸盒以及印章	2
11		机头设置	设置机头是否可用	3
12		报警信息处理	打印机发生故障，或者缺纸，缺墨等发送信息	3
13		退出	保存好设置并退出登录	2
14		管理员纸张放置操作	管理员放置了纸张，计数器重置	1
15	系统	自动退出	用户登录超时直接退出	1
16		清理用户数据	在用户退出时进行数据清理	1
17		结果上传	将打印结果上传到服务器	1
18		报警信息上传	打印机故障信息上传	2
19		发送消息给管理员	纸张数量低于报警下限，预警；故障的提醒	1
20		停电处理	停电的时候不运行打印，停电需要关闭系统	2
21		机头计数	对打印的纸张进行计数	1
22		摄像头监控	当有用户操作的时候，记录视频数据	4
23	研发设计跟测试方案	设计方案	需求分析	10
24			数据库设计	5
25			系统设计	5
26			接口核对	8
27			测试用例设计以及测试方案	5
28			测试执行	20

3.4 本章小结

本章分为三个部分，分别是自助打印系统的需求分析、系统的设计以及开发计划。在需求分析中分析了系统的使用流程，然后具体罗列出系统的硬件需求和软件需求。根据需求完成了整个系统的设计，分别有软硬件交互接口的设计、为接口提供功能服务的服务类设计、定时任务的设计、系统与其他系统交互接口设计以及数据库的设计，同时将所有的接口与类所要完成的功能、输入输出参数进行了限定。最后，根据需求与设计，完成了对整个系统软硬件开发的进度计划。

第四章 实现

自助打印系统服务器是使用 java 进行开发，使用 eclipse 作为开发 IDE，打印机端使用 c# 进行开发，visual studio 作为开发 IDE。实现时，每个接口作为一个 RPC 类，每个 service 抽象为一个功能类，每个数据库在 DAO 层都有对应的实体 DO 类以及 service 类来进行操作。

4.1 具体实现

4.1.1 服务类实现

在上文提到的 13 个服务类中，MasterDataInvocationService, VacationService, BaseInfoService 和 IncomingCertificationService 四个类的作用都是与其他系统进行交互获取数据。这些系统为了提供对外服务，都封装了一个 jar 包来提供可调用的接口，我的工程在 pom.xml 中导入了这些 jar 包后 maven 会自动将它们装载到项目中。我们的程序在调用这些接口后，系统通过背景中所提到的 HSF (High-Speed Framework) 提供的 RPC 服务与提供数据的这些系统产生交互。在这之前，HSF 要求每个服务的 consumer 即调用方在服务的 provider 即提供方申请所需要调用的接口，provider 会给调用方一个唯一的 key 与 secret，在调用时需要把 key 和 secret 作为参数传递到服务的提供方，经过校验后 provider 才会提供相应的数据。薪资数据的接口较为复杂，由于薪资是敏感数据，provider 在发送数据前对数据进行了 DESede 加密，调用方需要经过揭秘才能使用这些数据。另外，由于模板中需要的薪资数据为大写，而从系统中获得的数字是阿拉伯数字，因此还需要编写一个函数将阿拉伯小写数字转化为中文汉字。算法如下：

```
private String convert(String origin) {
    char[] d = { '零', '壹', '贰', '叁', '肆', '伍', '陆', '柒', '捌', '玖' };
    char[] d2 = { '元', '拾', '佰', '仟', '万', '拾', '佰', '仟', '亿', '拾', '佰', '仟' };

    String intString[] = origin.split("\\.");
    String str = intString[0];
    StringBuilder sb = new StringBuilder();

    for (int i = 0; i < str.length(); i++) {
        sb.append(d[str.charAt(i) - '0']);
        sb.append(d2[str.length() - i - 1]);
    }
    str = sb.toString();
    str = str.replaceAll("零[拾佰仟]", "零");
    str = str.replaceAll("零{2,}", "零");
    str = str.replaceAll("零([元万亿])", "$1");
    str = str.replaceAll("亿万", "亿");
```

```
if (str.equals("元")) str = "零";  
str=str.replace("元", "");  
return str;  
}
```

EmailService 和 MessengerService 是向外发送邮件和短信的服务。为了向外发送短信，阿里内部有一个短信平台，申请了预算之后，在平台上申请短信相应的模板，成功后会获得通道号、模板号等等参数，在调用时把这些参数加入即可成功发送短信。在发送邮件时，EmailService 在数据库中建了一张表，里面记录了发件人、收件人、邮件内容以及发送状态，每次有发送邮件的请求时就在表中新建一条条目，并标志发送状态为未发送。同时编写了一个任务类，使用一个定时器定时从数据库中获取状态为未发送或是发送失败的条目，再把每个条目创建为一封邮件以 smtp 协议发送出去，发送成功后再在数据库中更新该条目的状态为发送成功。使用定时任务并以数据库中的表为参考可以保证邮件发送的成功率，由于系统发送邮件数量的不确定性，在高峰期可能会遇到一些不稳定的情况导致发送失败，直接发送会导致一些用户收不到邮件的情况发生。

CheckTypeLimitService 是检查该用户当时能打多少份该证明的服务，较为简单，首先从 auto_print_type 中取到该证明每天和每月的打印限额，然后从 auto_print_log 中获取当天和当月该员工打印该证明的份数，限额与已打印份数相减后取最小值即为可打份数。

GetTypeListService 是获取该名员工可打印的证明列表的服务。由于不同层级正式员工、实习生、外包员工可以打印的证明都不相同，所以需要加以区分，这一点可以通过在 auto_print_type 表中的 limit_comment 字段来获知。另外，由于需求中要求能够打印要事先经过线上审批才能打印的包含员工请假记录的签证信息证明，这需要通过调用 BaseInfoService 类来获取该员工申请的记录。由于该种证明是员工在收入证明申请页面中在证明用途上选择签证并且审批状态为可领取之后才能开具的，因此要对请求到的数据进行筛选并分析其中的 json 字符串才能判断能否打印。由于模板数量众多，所以需要分级存储，如收入证明、在职证明为顶层节点，则本年度、上年度、近十二个月、近六个月、合同工资收入证明为收入证明子节点，税前、税后本年度收入证明又为本年度收入证明子节点。

ModuleInfoService 是获取模板中的变量值的服务。经过对二十多个模板的整理，共整理出这些模板一共需要 56 个变量，在表 4-1 中显示。将这些变量囊括到一个大类中，当要打印某份证明则从数据库中取出其模板以及这份模板需要的变量，然后从各个服务类中取到其值。若是要事先经过线上审批才能打印的包含员工请假记录的签证信息证明，则和 GetTypeListService 一样需要解析 json 字符串判断出是可领取的用途为签证的收入证明并解析出表单中的假期时间。另外，关于证明人的这三个字段，由于公司内部不同层级员工的证明人不同，因此首先需要获得要打印证明员工的层级，经过判断后取到其对应层级证明人的相关信息。

表 4-1 模板所有字段

员工姓名	员工工号	员工性别	护照号码	入职年	入职月	入职日
入职年月日	员工职位	员工公司	员工部门	打印年	打印月	打印日
打印年月日	身份证号	证明人姓名	证明人电话	证明人工作地	英文名	英文公司名
英文公司地址	社保缴纳地	英文职务名	假期开始年	假期开始月	假期开始日	假期开始年月日
假期结束年	假期结束月	假期结束日	假期结束年月日	工作地点	本年度税后工资	本年度税后月均工资
本年度税前工资	本年度税前月均工资	合同月工资	近六个月税后工资	近六个月税后月均工资	近六个月税前工资	近六个月税前月均工资
近十二个月税后工资	近十二个月税后月均工资	近十二个月税前工资	近十二个月税前月均工资	上年度税后工资	上年度税后月均工资	上年度税前工资
上年度税前月均工资	当前月公积金个人缴纳部分	当前月公积金公司缴纳部分				

PDFService 是将获取到模板和其中变量的值后生成证明文档并转化为 pdf 格式文件的服务。首先编写一个 velocityUtil，传入数据库中获取到的证明模板以及 ModuleInfoService 中获取到的变量值，将变量填入到模板中，生成 html 文件存放在本地，然后导入 ITextRenderer 类库将其转化为 pdf 格式。但是在这之中遇到了几个问题。

(1) ITextRenderer 原生库不支持中文，需要改写。首先在其中导入中文字体文件，这样就可以显示中文而不是乱码，其次，因为英文的换行形式与中文不同，英文换行都是在空格处而中文不是，所以需要在源码中增加对中文的判断并修改换行条件为中文换行条件。

(2) ITextRenderer 仅仅支持 html 文件转化为 pdf 文件，并且对 html 文件中的代码规范要求极高，在尝试所能找到的所有 plain text 转化为 HTML 的工具生成的文件都不能被该类库识别后，只得采取手动编写 css 的方式，将每个模板手动编写 css 代码，转化为 html 格式之后使用类库进行转化。

SignUtilService 是每个接口都要使用的进行签名验证的服务。由于自助打印系统中包含大量敏感信息，因此要保证只有打印机且在其正常工作时才能获取到我们服务器的数据，所以需要在调用接口时进行验证。在这里我采用了 RSA 算法进行加密。初始化时随机生成一对公钥和私钥存在数据库中，将私钥交给打印机端留存，在打印机端调用服务器端的接口前，将本次调用的参数组合成一串字符串，根据私钥和 RSA 算法生成一个签名字符串，并使用 Base64 算法进行加密，再以参数的方式传递到服务器。服务器收到请求后首先取到签名，用 Base64 算法进行解密，再从数据库中取到公钥，使用 RSA 算法进行验签，如若通过则进行下一步操作。

UploadGiantFileService 和 LogSendService 是两个为满足管理员需求设计的定时任务。LogSendService 较为简单，设置一个定时器，每天在一个时间点触发，从 auto_print_log 中获取当天所有打印成功的信息，进行计数和汇总成一段报告，然后调用邮件发送的服务发送到管理员的邮箱。UploadGiantFileService 比较复杂。

首先把系统本地存储的当天的所有文件打包，然后使用 DES 算法进行加密，然后准备上传到云端。这里提供云服务的是阿里云旗下的 OSS 系统，要接入到 OSS 则需要在申请空间后获取到密钥，在工程中引用 OSS 的 jar 包，然后调用其接口。由于每天需要备份的文件可能会很大，大文件上传如若遇到网络故障很容易出问题，因此在这里采用分块上传策略。首先将大文件拆分成一个个大小为 20MB 的部分并标上号，开启五个线程同时进行上传操作，等待所有部分上传完成后把所有部分按照标号拼接，检查文件完整性，若是无误则视为上传成功。上传成功后需将本地服务器存储的所有打印文档全部删除。

同时为了方便管理员检查备份文件，还开发了下载的接口。同样，下载也需要使用并发下载的方式来处理大文件，在 OSS 服务器端进行文件分块后开启 5 个线程同时下载，每个线程的下载任务完成后根据每部分的编号拼接成原始大文件，再使用 DES 算法解密后即可获得源文件。

4.1.2 接口实现

接口是为了打印机端和服务端交互而设计的，打印机端通过 http 请求调用接口，参数以 get 方式传递。传输文件时使用 httprequest 和 httpresponse 进行传输。

返回值为 json 格式，具体为

ResultModel<T>，其中 ResultModel.hasError==true，表示网络没有问题，ResultModel.content.isSuccess == true，表示请求成功，ResultModel.content.returnValue, T，表示返回结果，ResultModel.content.exceptionDesc，是请求失败的原因描述。

根据分析时所设计的六个接口，获取公章列表，只需要将记录公章列表的数据库所有未删除的条目取出，封装为 List<Map<String, Object>>格式，然后返回给打印机段即可。

获取员工信息的接口，打印机在员工刷身份证后将身份证号发送到服务器，服务器获取到身份证号之后，调用主数据的接口，根据身份证号获取到该员工的工号，如果查询失败则直接返回报错信息给打印机表示该名用户不是公司员工。获取到员工工号之后调用主数据的另一个接口获取员工的基本信息，如姓名、签约公司、手机号等。然后随机生成四位数字验证码，调用发送短信服务发送到用户手机，将验证码、员工工号、员工姓名、员工手机号、员工签约公司名封装为 Map<String, Object>格式返回到打印机端。

获取员工可打印证明列表只需调用上文所述的相应服务，封装为 List<Map<String, Object>>格式返回。

获取员工本次可打印份数和文件也需要先调用上文所述的获取限额的服务，获取到打印份数限额。若是限额不为 0，则从模板数据库中取出对应模板和变量，调用生成 pdf 的服务，生成证明的 pdf 文档。同时在打印日志数据库内生成一个新的条目，将本次请求的时间、工号、证明类型等存入，另外，生成本次请求的唯一验证码，并且不能和数据库中所有已有验证码重复，这时候采用随机字母数字加上本条目在数据库中的主键 id 的形式作为验证码，将打印限额、员工工号、验证码以 Map<String, Object>结构传回打印机，将生成的 pdf 文档以文件流形式发送给打印机，在打印日志的数据库表中将此次打印的状态设置为已发送。

打印结束的接口中，打印机将打印成功的请求的验证码发送到服务器后，根据此验证码在打印日志数据库内将这次打印的状态设置为打印成功。到此为止，打印机端一次完整的打印过程结束。服务器端还需要做一件事，根据需求要将已经打印的收入证明文档发送到系统

邮箱。因此要从日志表中查询到本次打印的证明类型 id，然后根据类型 id 去证明类型的数据库表中遍历查询其顶层节点，判断是否为收入证明，如果是，则将本地存储的本次打印的文档以邮件附件的形式发送到系统公共邮箱。

在打印机发生故障报警的接口中，打印机将相关报警信息发送到服务器后，服务器从 diamond 系统中取出已经预设好的邮件模板，将报警信息填入到模板中，组织成一封邮件发送给各位管理员。

在这些接口中，多次实现了邮件、短信发送功能，由于系统管理员以及邮件模板经常会变更，所以讲管理员邮箱等配置在代码中是效率十分低下的一种方法，因此在这种情况下使用公司内部 diamond 系统来对这些常量进行管理，在要使用这些常量时，仅需要根据 dataId 来调用 diamond 系统的接口来获取。Diamond 系统是一个方便快捷的常量管理系统，并且能对日常测试环境、预发布环境以及正式环境进行隔离。

4.1.3 DAO 层实现

DAO 层（Data Access Objects）是用来连接工程和数据库的桥梁。首先针对每个数据库表有一个 DO 类与之对应，DO 类对表的每个字段进行初始化，并声明了对表操作的各个函数包括增删改查等等，这些函数的实现在每个 DO 类对应的 xml 格式的 mapper 文件中通过编写 sql 语句来实现。在每个 DO 类之外都封装了一个专门的 service 类，将其其他服务和数据库隔离，其他类想要对数据库进行操作只需要调用其服务类提供的接口即可。

4.2 实现中的其他问题

在自助打印系统的项目中，作者不仅是技术开发还是项目经理，这就意味着工作远远不只是写代码完成这个项目。在项目立项时，接到需求方也就是公司 HR 部门提交的需求概要和上线时间要求后，首先与需求方开会讨论，由于需求方对实现技术并不了解，因此需要对需求进行逐条讨论分析其可行性以及开发时间评估，经过数次讨论之后才慢慢罗列出了前文详细的需求表。确定系统整体需求之后开始与供应商联系，供应商负责提供硬件设备。由于硬件设备成本较高，而且是只能针对自助打印系统新开发的设备，因此只有等到制造出之后才能进行验收。在这种情况下，硬件具体需求的确定就显得尤为重要，需求的改变可能会导致制造时间延后一个周期，制造成本也会翻倍。因此花了两周时间与供应商商讨硬件需求、使用流程、软件接口细节以及 UI 界面等，其中硬件需求包括打印机的外形尺寸、屏幕大小、触摸屏原理、纸盒数量等等，软件接口细节包括接口调用方式、接口规格、参数、返回值以及其格式、调用顺序等，系统在打印机屏幕上展示的 UI 界面也是由打印机端实现的，但是其样式是由作者和需求方以及供应商一同商讨绘制而成，在确定这些需求之后，和供应商根据需求中的上线时间制定开发时间表以及测试计划。将这些全都约定完成后，才开始了正式的开发工作。同时，作为产品经理兼开发，在完成自己开发任务的同时，还需要随时根据开发计划跟进项目进度，与供应商不断沟通调整接口细节。

另外，在开发过程中需要大量用到公司内部其他系统与技术，这些也需要和其他部门的开发员工进行沟通。比如自助打印系统需要与假期系统进行对接获取员工的假期数据，但是假期系统目前并没有提供根据员工工号和员工请假开始时间查询假期结束时间的相关接口，因此需要向假期系统的开发员工提出需求，给出接口的规格参数返回值后，由假期系统的开发评估开发时间，等待排期开发测试上线后方能使用。在开发时遇到了许多这种问题，等待其他系统开发完成接口会对原先的开发计划造成很大的影响，在这种情况下，为了按时完成开发任务，必须对开发计划进行微调，在等待接口开发的同时完成其余任务。

4.3 测试

自助打印系统的测试分为软件接口调试、硬件设备调试和软硬件联调三部分。同时，系统有三个环境可供测试，分别为日常环境、预发布环境以及正式环境。日常环境为测试环境，使用的数据均为测试数据，用于测试系统能否正确运行。预发布环境数据均为线上数据，与正式环境的区别为预发布环境只实现功能，并不对外界提供服务。正式环境即为系统正式发布后所处的环境。软件接口测试在日常环境进行，对六个负责软硬件交互的接口进行测试，通过编写测试脚本使用各种不同参数调用六个接口，测试其是否有正确的输出并且有正确的错误处理。硬件设备调试包括打印机纸盒墨盒调试、各个机头文件打印测试、打印机触摸屏调试、身份证刷卡调试等。软硬件联调最为重要，首先在预发布环境进行测试，先要调试打印机硬件能否成功调用系统接口、签名验证是否正确、能否正确处理返回值。接口测试无误后，进行黑盒测试。用户进行完整的打印操作，测试各个环节的结果是否符合预期，选取不同公司的员工对所有证明种类进行打印测试，然后进行错误操作判断错误处理是否正确，管理员对报表功能、备份功能、报警功能进行测试。预发布环境测试无误后正式上线进入正式环境。

4.4 实现结果

在完成服务器端的所有接口后，经过初步调试即可与硬件连接进行联调。经过验证，打印机能够实现需求中的所有条目。

整个系统共有 21 个页面，分为用户页面和管理员页面。用户页面：刷卡页面、登录页面、证明目录页面、打印设置、打印预览、等待打印、打印完成页面。管理员页面：登录页面、总览页面、统计报表、印章配置、打印机设置、报警信息、打印机操作、版本信息、网络设置、纸盒墨盒、管理设置、待机图片配置页面。



图 4-1 刷卡页面

<

欢迎使用自助证明开具



430321198809299014



9646

26秒

手机尾号: 8254

登录

QWERTYUIOP

ASDFGHJKL

↑ZXCVBNM↵

123😊SpaceSearch

图 4-2 登录页面

证明目录



在职证明

• 在职证明-中文

其他证明

本年度收入证明

近六个月收入证明

近十二个月收入证明

上年度收入证明

合同工资收入证明

图 4-3 证明列表页面



图 4-4 打印设置页面



图 4-5 打印成功页面



图 4-6 管理员操作页面



图 4-7 统计页面

**图 4-8 自助打印机外观**

在经过一个月的测试和运营后，自助打印项目正式上线。在上线试运营的一个多月间，放置于阿里巴巴西溪园区 HR 自助服务台的自助打印机一共打印了两千三百余份证明，并且经过大力推广正在日益增多，方便了员工的证明打印，也大大提高了 HR 员工的工作效率，于内网上发布的帖子也受到了员工们的点赞好评与各种意见。

鉴于在西溪园区的打印机试运营效果良好，正在逐步增加打印机的数量，放置在阿里巴巴其他园区，截至目前，已经有七台打印机投入使用，为阿里巴巴的员工们带来便利。对其他公司提供服务的商业化行动也已经开始准备。

4.5 本章小结

本章分为四个部分，分别介绍了自助打印系统的各个服务类以及接口实现中的技术细节、除了技术方面以外实现中遇到的一些问题、系统的测试以及最终系统完成的结果。

第五章 总结和展望

5.1 结论

本论文主要针对公司 HR 部门为员工开具各项证明在员工数量大、证明需求量大的情况下效率低下，使得 HR 员工把时间都用在机械的任务上，没有足够的时间完成更为有意义的工作这一情况，提出了自助打印系统的想法。根据自助打印系统的使用流程，提取出整个系统的所有需求。根据需求设计了系统的架构、软硬件交互的几个接口、为接口提供服务的 service、连接整个项目和数据库的 DAO 层以及数据库的五张表。之后根据设计的架构、接口、服务等类首先使用 webx 架构搭建了服务器，然后实现需求中相应的功能。

实现阶段以服务器搭建、服务类和接口类的实现为主要工作。在搭建 webx 服务器前，以前仅仅有过 SSH（Struts, Spring, Hibernate）框架服务器的搭建，对 webx 框架不甚熟悉，因此翻阅各种文档和现成应用进行学习。搭建好服务器之后进行 service 类的编写，编写过程中完成了本系统和公司内部其他若干系统的交互并学习使用了 HSF、TDDL 等核心中间件。在不断学习中实现了获取员工基本信息、薪资数据、假期信息，发送邮件短信，生成 pdf 文档，对文件、请求进行加密解密加签验签，通过定时器定时发送邮件、整理文档压缩加密并分块并发上传等服务。有了这些服务的支撑才能完成打印机端与服务器之前软硬件交互的接口。六个接口分别实现了打印流程中登录、获取证明列表、校验打印份数、获取打印文件、打印成功端口关闭以及故障报警的功能。

在经过接口自身调试、硬件本身调试以及软硬件交互联调后，系统正式在公司内部上线，完整地实现了所有需求并以较高的稳定性、友好的人机界面、方便的操作和高效的服务获得了员工们的一致好评，并逐渐推广到其他园区和外部市场。

5.2 展望

系统本身也存在一些不足之处。比如在后期维护时若是想要修改或新增模板，则需要修改数据库中存储的 html 文件，此 html 文件对非技术人员来说可读性不高难以操作。因此计划在接下来的迭代中为了方便模板修改编写一个网页页面，在页面中嵌入富文本编辑器，后台自动将模板转化为系统可用的 html 文件。这样可以方便后期维护人员的操作。

整个系统虽然只是解决了一个公司 HR 部门工作的问题，但是这种产品和设计思路可以运用到实际生活中的方方面面。在互联网不断给人们的生活带来种种便利的今天，我们可以运用这种技术来实现更多系统来解决一些妨碍人们生产效率的小问题。积少成多，当更多的问题被解决，人们就有更多的时间放在有意义的事上，加速社会的发展从而能够推动时代的进程。

参考文献

- [1] 李俊杰. Maven在企业Java软件产品中的应用[J]. 电脑知识与技术,2011-7.
- [2] 徐立新,雷相波. 应用 Maven 管理项目[J]. 电脑知识与技术,2010-4.
- [3] Sonatype. Maven Guide[C]. O'Reilly Media, Inc. 2008:4-9.
- [4] Michael Zhou. Webx3 Guide Book[C/OL]. 2010-11. http://www.openwebx.org/docs/Webx3_Guide_Book.html.
- [5] 王君. 基于 Struts+Spring+Hibernate 的企业级 WEB 应用框架的研究[D].合肥工业大学硕士学位论文. 2007-05-01.
- [6] Della Mea Vincenzo, Bin Irene, Pandolfi Maura et al.. A web-based system for tissue microarray data management.[J]. Diagnostic Pathology, 2006, 1.
- [7] 梁乾宇. 分布式共享技术的研究与实现[D]. 华南理工大学硕士学位论文. 2010-11-23.
- [8] 赵璐. 阿里巴巴广告应用质量平台的设计与实现[D]. 南京大学硕士学位论文. 2014-05-01.
- [9] 一江 . HSF 新人用户手册 [C/OL]. 2011-10. <http://wenku.baidu.com/view/41d8f76baf1ffc4ffe47ace4.html>.
- [10] 郎中锋 . 淘宝技术架构分享 [C/OL]. 2012-05. <http://wenku.baidu.com/view/46834f2be2bd960590c67770.html>
- [11] 何振兴. 分布式数据库中间件 server 层的设计与实现[D]. 哈尔滨工业大学硕士学位论文. 2014.
- [12] 屈华平. 基于 Webx 框架的三方应用平台的研究与实现[D]. 暨南大学硕士学位论文. 2012-05-12.

谢 辞

首先我要感谢我的导师王赓老师，无论是在项目的开题立项还是在开发过程中，都给予了我大力的支持，当思路或是代码遇到瓶颈时，老师总是耐心解答我的问题。同时我还要感谢我在阿里巴巴公司的主管秦国亮，他在我的开发过程中给了我很大的帮助，带我熟悉了阿里巴巴内部各项系统的使用。最后，我要感谢软件学院所有老师和同学朋友在学习和生活上给我的支持，是我顺利完成毕业设计的开发。

SELF-SERVICE CERTIFICATE PRINTING SYSTEM BASED ON WEBX FRAMEWORK

With the development of Internet technology, more and more applications are designed in order to solve problems of people's daily life and bring people convenience. Because of the large amount of employees in Internet corporations, management and service for employees becomes increasingly important. It's impossible for human resource managers to deal with large numbers of daily requirements of workers only by themselves. So it's necessary to develop a sound system to help with these work. At present, there have been systems like vacation system, health system, commercial system, salary system providing convenience for companies. However, there's another work which is often overlooked but is also very important. That is issuing proof of service to staff, proof of income and so on. At present, this work is done by a handful of employees manually. But rely on employees to manually attestation has the problem of low efficiency. According to the statics of HR service center employees, a total of more than fifteen thousand employees work in one park of their company, and daily demand of all kinds of proves is about two hundred pieces. This makes at least one employee invested in this attestation mechanical work and have no time to do other work, which does not meet the company's basic principle. In this condition, a self-service proof printing system can do a lot of help. The thesis mainly discusses the design and implement of the self-service printing system.

The thesis analyzes the using flow of the printing system. Firstly, user puts his identity card on the machine or input his identity number on the screen. Then the system will send a mobile message to user, containing a piece of verify code in it. User input the verify code on the printer's screen in order to log in. After user successfully logs into the system, the screen on the printer will show all the certificate types he can print in the form of a tree. At this time, user can choose any type on the screen, after that, the screen will show another page, containing the preview of certificate and some buttons. User can check the limit he can print this time and choose how many he will print. After choosing and pressing the print button, the printer will check user's company can print out certification with his company's seal on it. If the certificate printed is certificate of salary, an email attached with e-edition of the certificate printed should be sent to a system manager's mailbox. With this using flow, the thesis designs six interfaces used for software and hardware interaction, nine interfaced used for interaction with other systems and thirteen services used to provide services. The six interfaces for hardware and software interaction are getSealType, getWorkerInfo, getTypeList, checkTypeLimit, closeApply and alert. The getSealType interface is to convey all the seal types that can be printed to the printer. The getWorkerInfo interface is referred by the printer with credit ID as parameter to ask for user's information like work number, name, company name and so on. Also, in this interface the system will send a piece of verify code to user's mobile phone and tell the printer what the code is. In the getTypeList interface, printer sends user's work number to system, and system will return with all the types that the user can print in the form of [typeId,

typeName, parentId]. In the checkTypeLimit interface, print sends user's work number and the type he chooses to system, system checks the limit he can print and then creates a PDF file of the certificate, then send back to printer. In the closeApply interface, printer tells system that the user has finished printing, and then system record this printing action in the database. When the printer faces an error, it will refer the alert interface, and send all the information about this error to the system in order to let the system send alert emails to system managers. All these interfaces should be signed by printer before referred and system will check the sign before returning the results. Because some certificates need information like salary, vacation information and so on, the system has to make interactions with other systems like main information system, vacation system, salary system and so on. When in need of such data, the printing system will call interfaces of other systems, using the technology of HSF (High-speed Service Framework), which is a system that provides services of remote procedure call. With these information, the thirteen service classes can do their jobs to provide services for interfaces. For example, Sign Utilization Service will provide sign check service in order to enhance the safety of PRC calls. Email Service can make the system able to send emails with attachments to others. Module Info Service lets interfaces know all the information needed in the certificate type the user chooses. Master Data Invocation Service calls interface of Master Data system and gets worker's basic information. PDF Service can create a printable PDF file with a template and parameters. Check Print Limit Service can check the limit that this user can print this certificate this time. The Type List Service can return all the certificate types this user can print. Messenger Service provides the service of sending mobile messengers to users. Vacation Service, Incoming Certification Service and Base Info Service can gather vacation information, salary information and some other information. Upload Giant File Job sets a timer so that system can upload the zip file of that day's printing file to cloud everyday. Log Send Job also sets a timer so the system can send printing logs to all the system managers everyday. After designing these classes, the thesis designs five database tables. Auto Print Type table records all the types of certifications that can be printed. It also includes its parent's type id and print limit. Auto Print Log table records the work number of user, printing time, type id, print number and print status of each single print action. Auto Print Module table records type id, detail of the template and parameters of each type. Auto Print Alert table records the address, time, description, name, and solution of the error. Auto Print Seal Type table records the id, name and company name of the seal.

After designing the classes and database tables, the thesis starts to implement the system. The self-service printing system is implemented by Java with Eclipse as IDE. The printer is implemented by C# with Visual Studio as IDE. All the classes are implemented with the help of detailed designed above. During the implement of the system, the author is not only the developer, but also the project manager. It means the author's work is far more than just writing codes to complete the program. After the requirement of such printing system, it's significant to communicate with human resource managers who raise this requirement about the details of this system. When everything has been checked, the author has to get contact with hardware provider and discuss with them about how to cooperate and what they should do. The developing process can only begin after all these have been done. When developing, keeping up with hardware providers is also of great significance. Testing has been divided into three parts. The first one is testing the software interfaces. They are tested by writing scripts to call the interfaces and check whether the result is correct. The second one is

hardware testing. It includes printing test of all printers, test of the screen, test of carton box and so on. The third one is the united testing of software and hardware. Users operate the whole process of the printing action and test all the function listed in the requirement.

After all-round testing of the system, the self-service printing system is completed. It meets all the requirements listed above. The whole system has twenty-one pages, including user pages and system manager pages. User pages includes the ID card identify page, the log in page, the type list page, the preview page, the waiting for print page and print complete page. System manager pages include the log in page, the overview page, the printing record page, the seal config page, the printer setting page, the alert message page, the printer operating page, the version information page, the Internet setting page, the paper box setting page, the setting page and the background picture setting page. After one month's testing, the self-service printer formally gets in use. Within one month, it has printed over two thousand and three hundred certifications. And the number is getting larger with the help of vigorous promote. The printer has brought great convenience to employees who want to get their certifications, and also it has risen the work efficiency of human resource managers.

However, the system also has some short comes. For example, when system managers want to edit or add a new template of certification, they have to modify the HTML format files stored in the database. This HTML file is hard to read and modify for them.

Although this system only solves a problem of human resource manager in a company, the idea and design can be applied to all rounds of life. Nowadays, the Internet is bringing people all kinds of convenience. We can use this technology to solve more problems that decreases people's work efficiency. With more and more problems solved, people can have more time to spend on meaningful things, and that will accelerate the progress of technology.