****

**《软件体系结构与设计》实习报告**

**题目： 第3次上机实验**

班级序号： 111172

学生姓名： 江悦

任课教师： 尚建嘎

**中国地质大学信息工程学院软件工程系**

**2019年11月**

**一、实验概况**

**操作环境:** 操作系统:Windows 7 及以上

**开发环境:** Visual Studio 2012 或 IDEA

**开发语言:**不限

**建模工具**:RSA 或 Visio

注:以上是机房提供的环境情况，自带笔记本电脑不限。 二、任务要求

现实生活中，存在着大量发送手机短信通知，发送电子邮件消息的系统需求，例如:当 你登录 AWS 或阿里云账号时，可以通过手机验证实现账号登入功能。这类通知服务是可以 在其他应用程序中重用的功能模块。

1. 结合课堂上讲授的 SOA 风格，基于 AWS SES 或阿里云的邮件服务，实现一个能发送电 子邮件消息的Web Service服务，包括如下三个具体服务(开发语言不限，要求提供基于SOAP 协议和 REST 风格的两种接口):  
  sendEmail(String \_url,String \_payload)  
  sendEmailBatch(String[] \_url,String \_payload)  validateEmailAddress(String \_url)

**注:** //邮件地址为\_url，内容为\_payload //批量发送邮件 //验证是否为有效的邮件地址

a. 以上三个服务，返回的值为 Y 或者 N。例如在第一个服务中，发送成功则返回 Y， 发送失败则返回 N。

b. 在你实现这个服务时，可基于 AWS SES 或阿里云的邮件推送服务来实现你的 Web Service，AWS SES 参考链接 https://aws.amazon.com/cn/ses/，阿里云的邮件推送服务地 址 https://dm.console.aliyun.com/。

c. validateEmailAddress 方法尽量使用正则表达式完成。

2. 编写至少两种版本客户端(例如:桌面版、Web 版、Android 版、IOS 版)，分别调用 你编写的邮件推送 Web Service 服务，并分析 Web Service 集中解决远程调用、跨平台调用、 跨语言调用所带来的好处及不足。

3. 结合上述实例，从软件体系结构风格的角度分析 Web Service 风格的主要构件和连接件? 提示:可参考课堂上讲授的“两层 C/S 结构”风格

**基本构件:** • 数据库服务器:存放数据的数据库、负责数据处理的业务逻辑; • 客户机应用程序:  
 • GUI:用户界面  
 • 业务逻辑:利用客户机上的应用程序对数据进行处理;

连接件:经由网络的调用-返回机制或事件机制。 •客户机<->服务器:客户机向服务器发送请求，并接收返回结果。

1. 参考 Kruchten 4+1 视图模型，试着给出上述系统的视图模型。

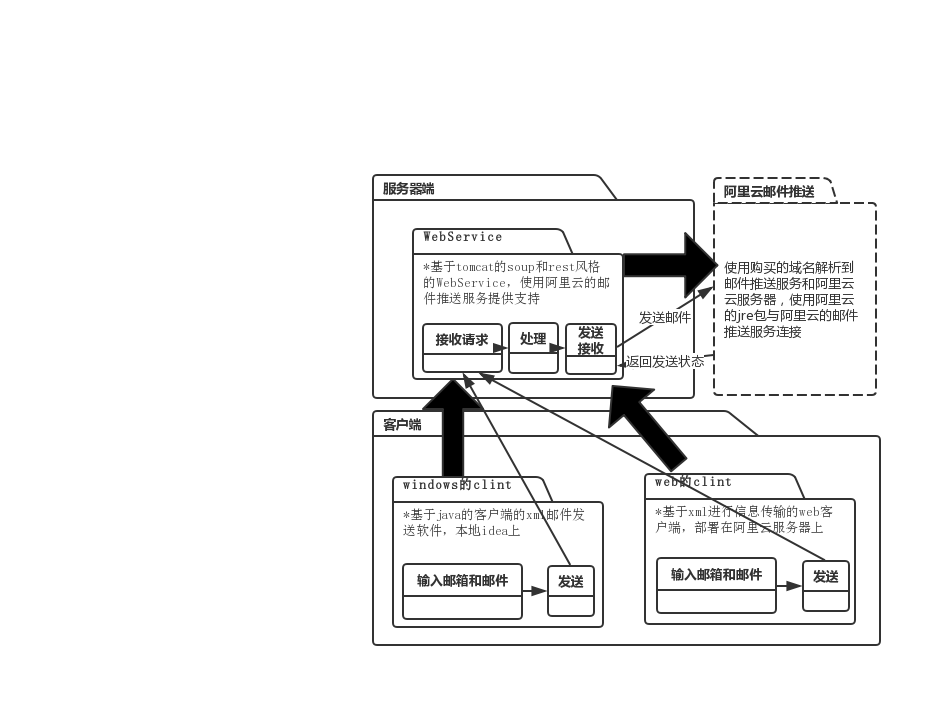
5. 编写实验报告(除个别术语外均应为中文)。

**二、实验设计(给出你的实习内容的设计方案，可根据实际情况调整条目)**

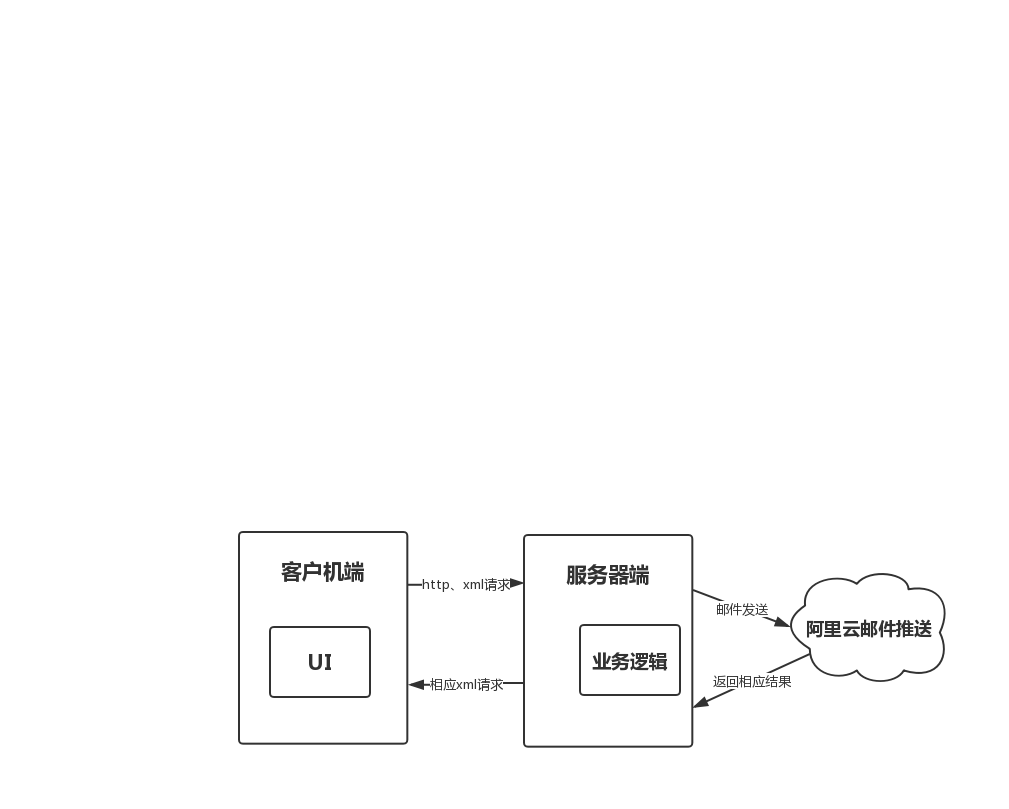
**2.1 系统需求**

需要构件一个基于两种以上终端的基于阿里云的邮件发送应用，需要使用soup和rest两种风格的接口。

**2.2 架构设计**

****

**2.3 接口设计**

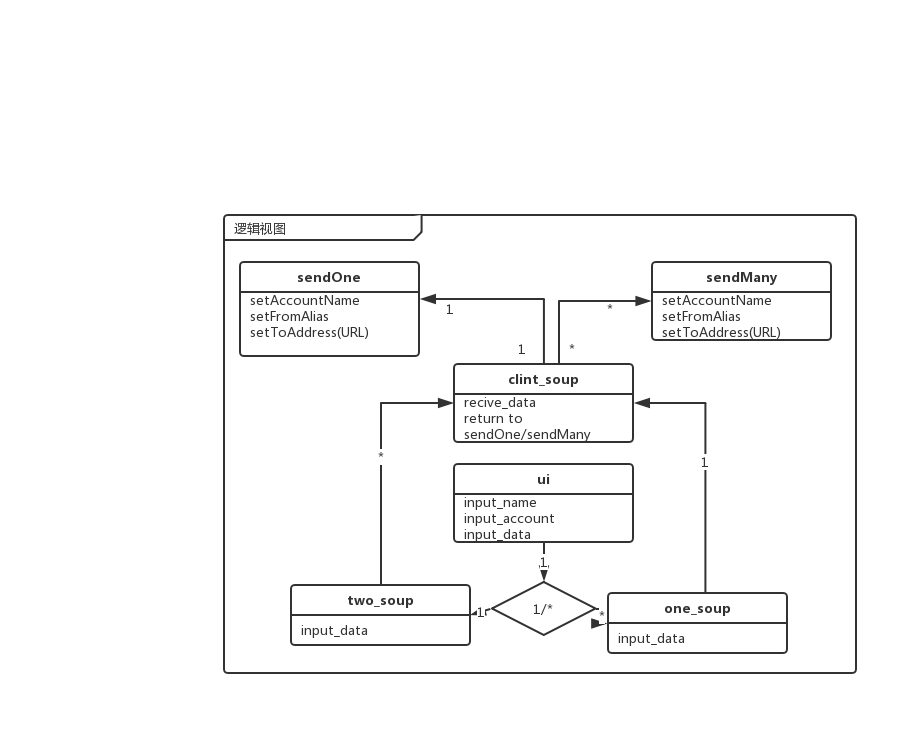
****

**2.4Kruchten 4+1 视图模型**

**<1>逻辑视图**

**视角：用户**

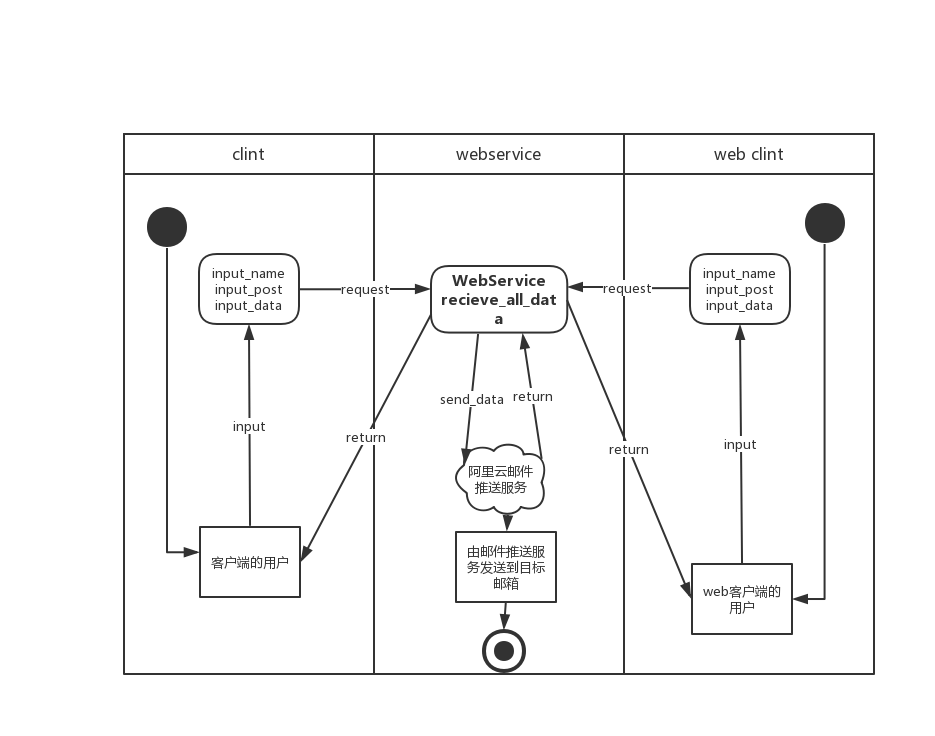
**关注点：功能性需求**

****

**<2>进程视图**

**视角：系统集成者**

**关注点：非功能需求**

****

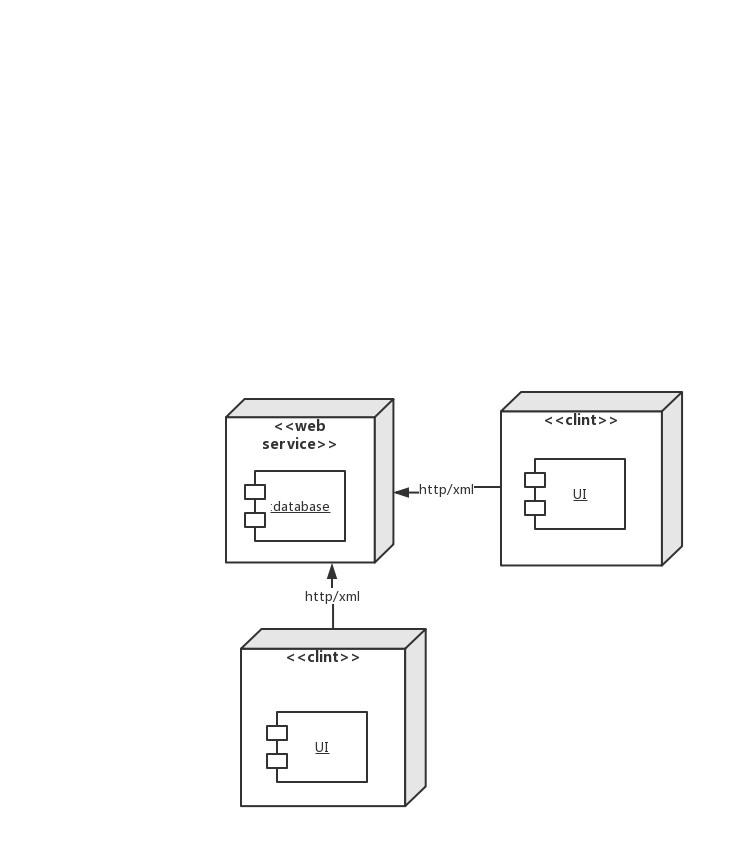
**<3>物理视图**

**（详见2.2架构设计）**

**<4>开发视图**

**视角：系统工程师**

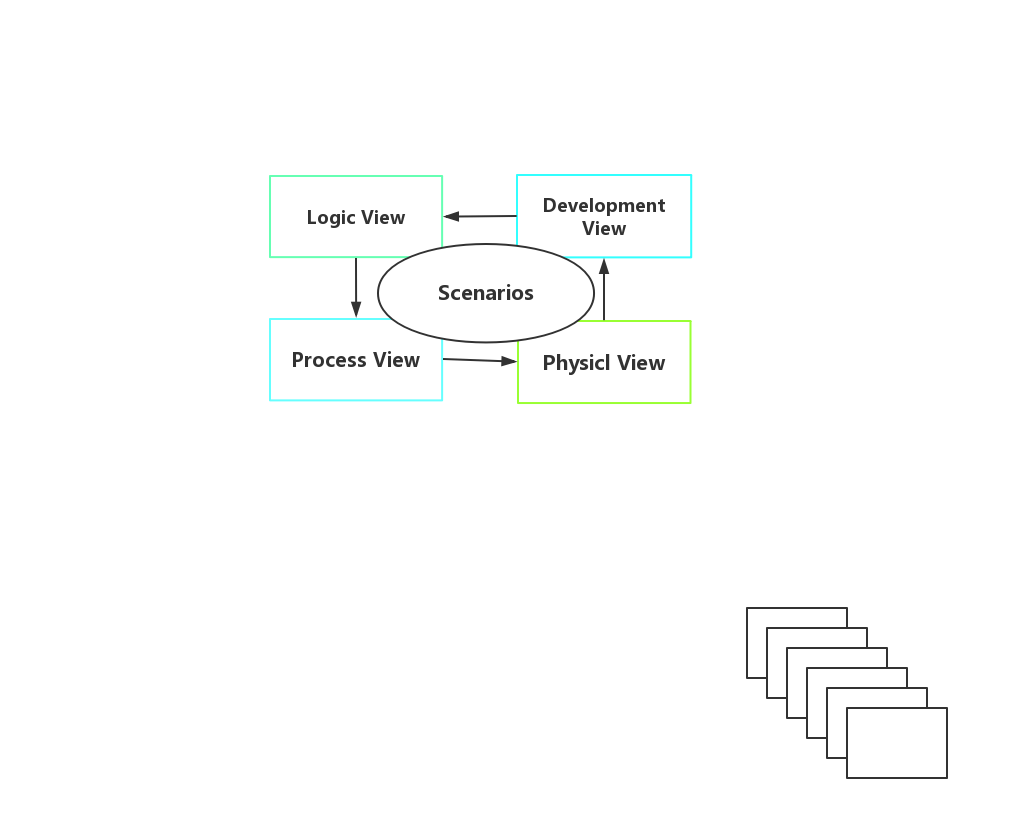
**关注点：依赖于硬件的非功能性需求**

****

**<5>场景视图**

**视角：所有视图的用户，评估人**

**关注点：系统一致性**

****

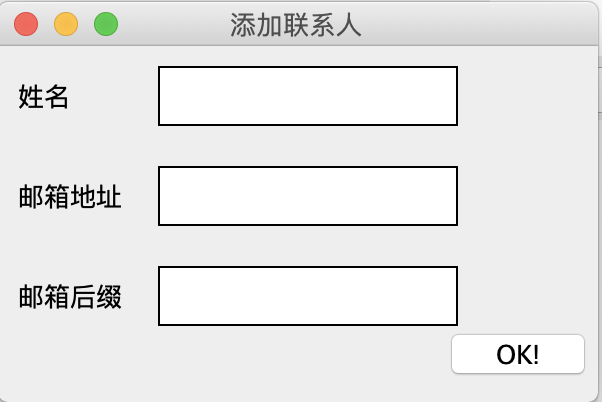
**三、实验过程**

**3.1 软件实现**

**使用idea的swing和awt库制作了客户端UI界面**

****

**可以添加联系人，输入邮箱地址和联系人id**

****

使用html，css，js，ajax，制作了web端UI界面



可以单发或者群发邮件，只需要中间用逗号分隔开就可以

使用idea制作了WebService并使用scp指令发送到基于阿里云的linux服务器的toncat目录下，几经调试，已经上传了多个war包，可以通过下载客户端的应用，发送邮件，也可以直接使用8080端口访问服务器47.97.184.36下myPost3的web端

**3.2 实验环境**

**硬件：**MacBook pro一台，基于Macos 10.15bata，

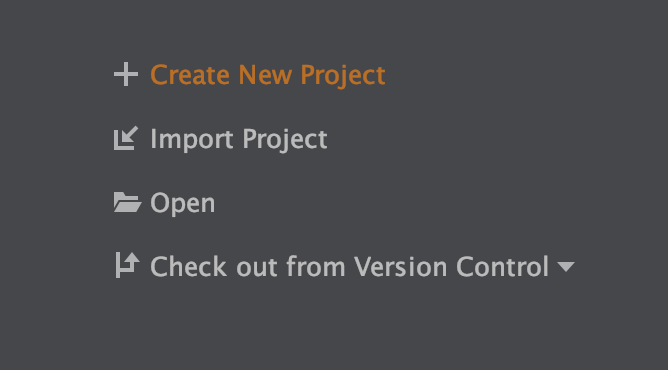
**软件**：Jetbrains的idea来做客户端用户界面和生成tomcat服务器端的war包

Sublime text3写web端的界面

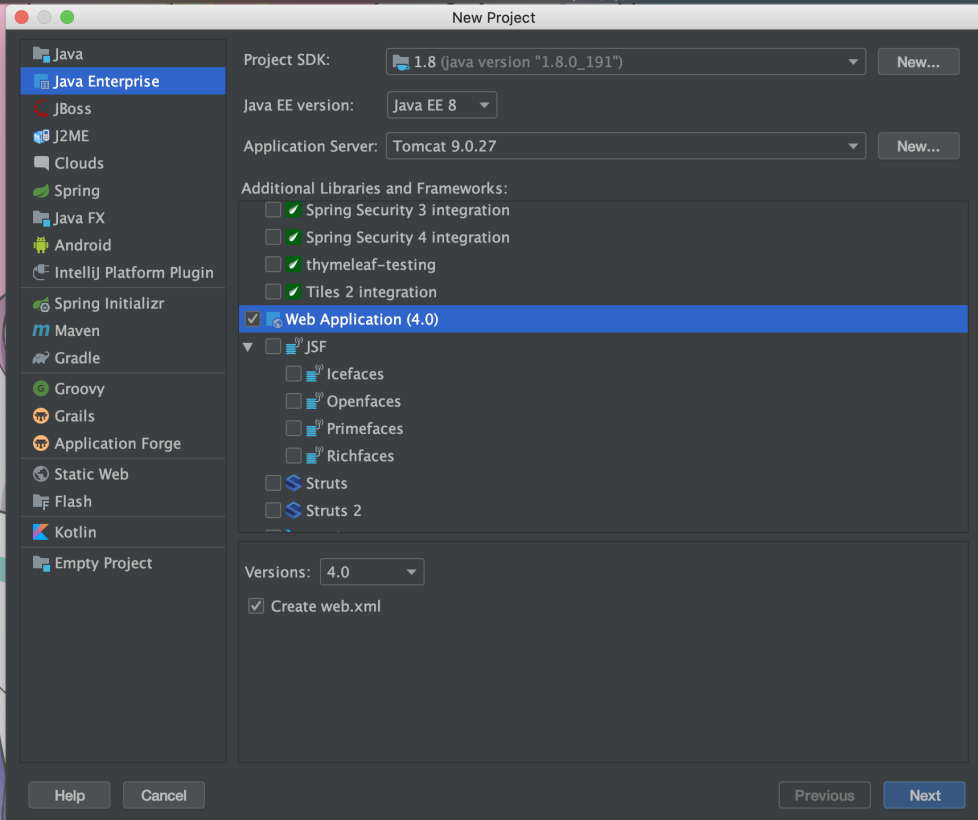
**服务器端**：使用了阿里云的ECS，基于windows server的学生服务器，使用的是ubantu18.04

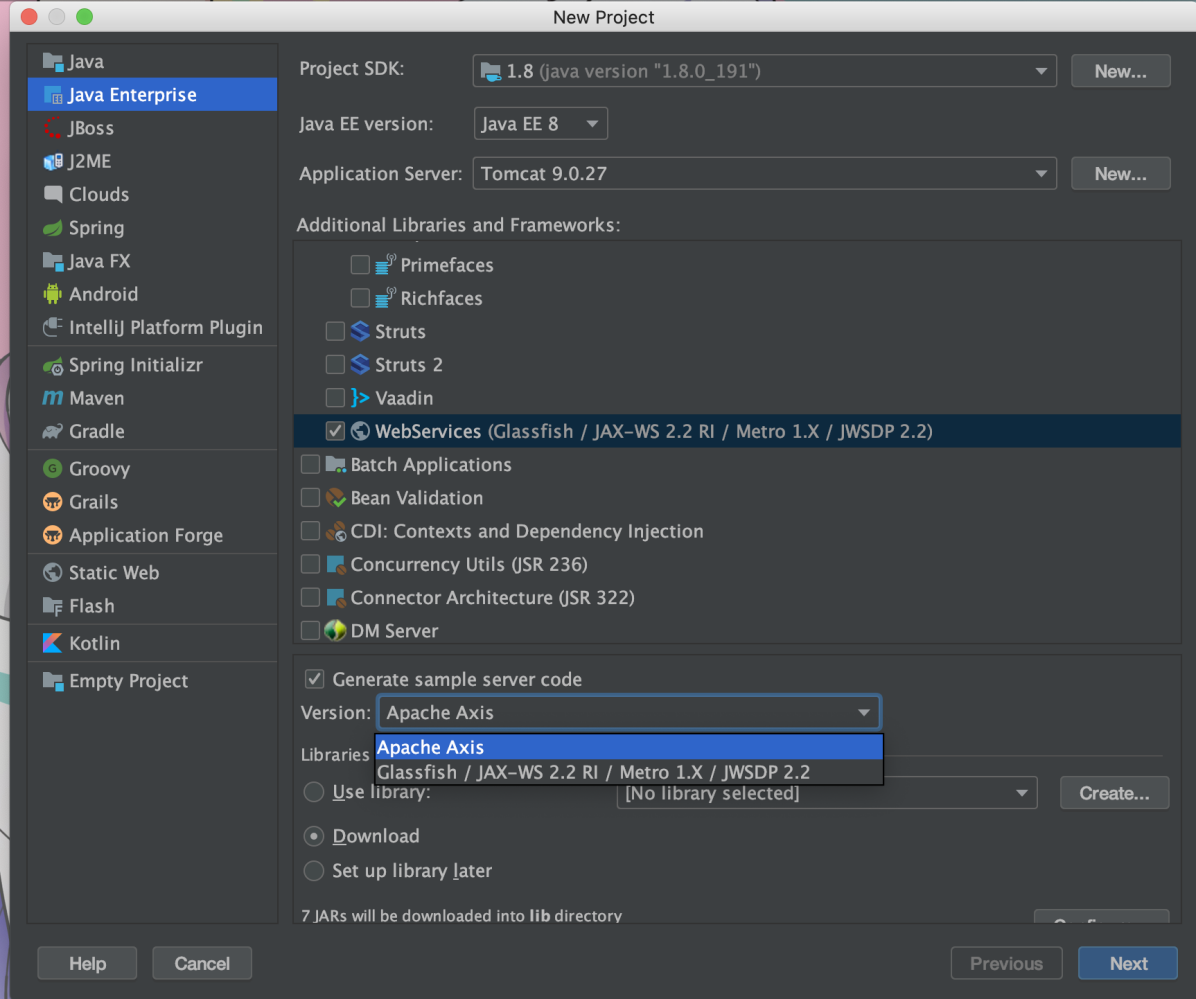
**3.3 实验步骤**

1.首先需要配置webservice服务器，首先从官网下载tomcat9.0.27（越新越好），再在idea新建java enterorise项目

****

2.选择web application和 webservice 并把webservce改成apache axis，进入，

****

****

3.进入后，修改url，改为只剩一个/（以便与之后访问方便），之后有小错误可以fix一下，在者就是需要在sever-config.wsdd里面配置一下soup和rest函数的信息，以便与外界访问

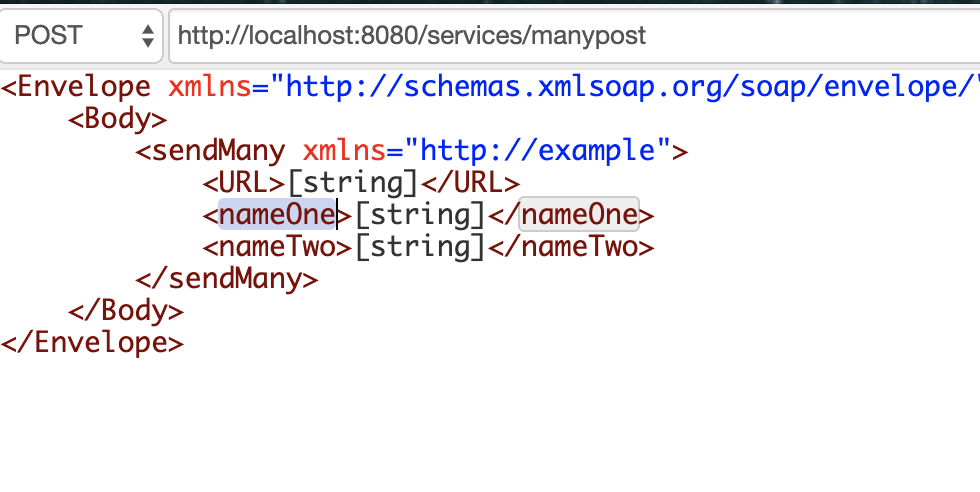
4.然后就可以直接运行服务器，服务器会在浏览器自动打开一个End界面，这就是我们服务器的主界面了，然后我们需要通过新建包和class，来制作我们的发送邮件的服务器端。下载阿里云的邮件支持的包，还有jersey（restful风格的框架），**截屏2019-10-28上午9.22.35**

5.接下来就是漫长的配置之旅，我jersey没有配置好，所以现在rest风格还无法实现，但是基于阿里云的soup风格已经可以完美运行了。

6.在配置web客户端的时候，使用了layui的框架，传输xml使用的是ajax

7.tomcat配置服务器ID原理就是可以发布服务的wsdl，其他接口可以通过访问需要服务的wsdl获取可以使用的函数信息，通过设计接口与函数对接，就可以通过函数使用服务器的功能了

比如我的

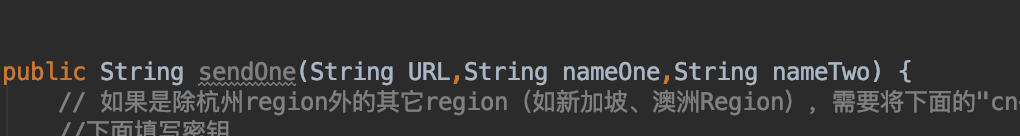
****

**8.这个是我的soup风格的群发的xml的请求头，ajax使用这个请求头，把html页面上form表单的数据通过js对应到请求头的匹配项上，再配置ajax的请求头，**

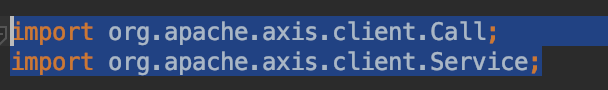
****

9.url是已经配置好的我的服务器里的对应的接口函数地址

data是要传输给该函数的数据，从我的form表单来看，需要传输，邮箱地址，邮件标题和邮件正文，正好匹配我的调用阿里云服务的函数的参数

****

10.客户端向服务器端发送请求使用apache的call和service

****

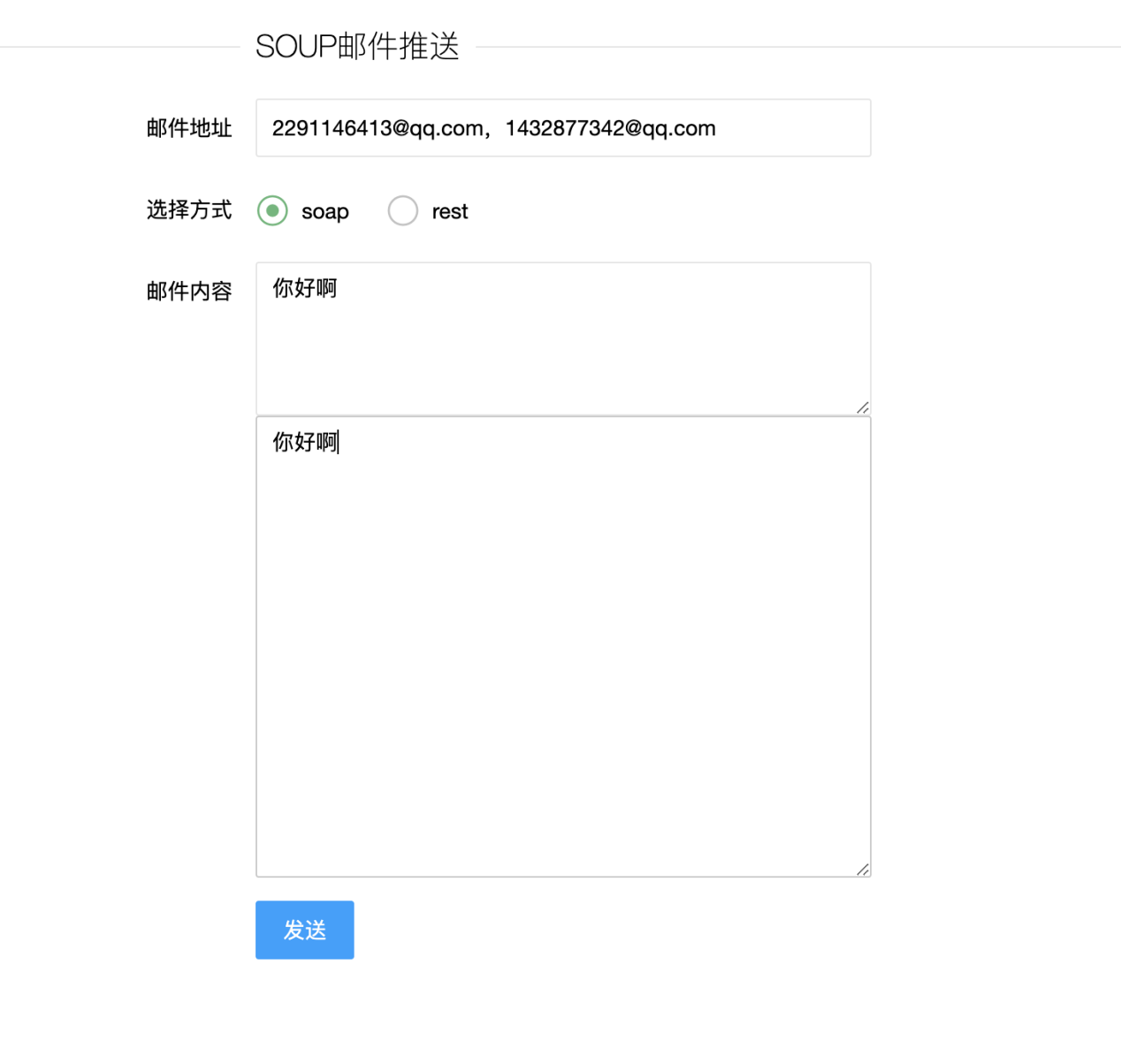
11.以上是soup风格，rest风格我的服务器端现在还不能支持，需要下载jersey包，然后配置tomcat.rest需要使用路径访问，具体实现跟soup差别不太大

****

**四、实验评价**

**4.1 实验结果**

**我现在已经可以使用soup风格发送邮件信息了**

****

可以完美的实现使用客户端和web客户端单发群发邮件，rest风格的结果还有些小问题，后续会继续改进

**4.2 结果分析**

完成了基于soup和rest风格的邮件发送webservice，我的作业还有很多不足的地方，希望可以改进

1. **总结**
2. 基于soup和rest风格的两种接口，这次实习设计是比较综合的考虑了客户端和服务器的设计，可以通过客户端和web端访问邮件发送服务器
3. 本次实习我觉得收获最多的是，掌握了很多新知识，包括，tomcat的使用，wsdd，wsdl，ajax，jquery，xml，html，css，js，json，的含义和使用还有linuxtomcat部署，linux的常用指令，linux和本机的文件传输，还有域名的解析，很多的只剩都是之前没有接触到的。这两层架构的设计，可以充分体现这个项目从构思到筹备到设计到实现到维护等方方面面的综合，需要考虑很多的点，让我真正体会到了自己制作一个网站的乐趣。
4. 这种技术呢，我觉得适合我自己，可以做一些web项目，发到服务器上，可以做成一套很多端都能访问的比较成熟的设计流程，对于我自己的程序设计实力有很大的提升

### **参考文献：**

### **【1】**<https://www.iteye.com/blog/201404282204-2118417> 用wsdd发布WebService

【2】<https://www.cnblogs.com/frozenzhang/p/3454024.html> ajax调用webservice

【3】<https://www.cnblogs.com/ciscolee/p/7533445.html> 阿里云的ubantu上的tomcat配置

【4】<https://blog.csdn.net/qq_23878541/article/details/90158167>使用idea把项目打成war，部署到tomcat

【5】www.runoob.com 菜鸟教程

【6】<https://www.cnblogs.com/bettyling/p/4366421.html> java完成webservice项目

【7】<https://blog.csdn.net/zolalad/article/details/31424831>java基于restful风格实现webservice

****附录：****

**什么是SOAP?  
SOAP (Simple Object Access Protocol) 顾名思义，是一个严格定义的信息交换协议，用于在Web Service中把远程调用和返回封装成机器可读的格式化数据。事实上SOAP数据使用XML数据格式，定义了一整套复杂的标签，以描述调用的远程过程、参数、返回值和出错信息等等。而且随着需要的增长，又不得增加协议以支持安全性，这使SOAP变得异常庞大，背离了简单的初衷。另一方面，各个服务器都可以基于这个协议推出自己的API，即使它们提供的服务及其相似，定义的API也不尽相同，这又导致了WSDL的诞生。WSDL (Web Service Description Language) 也遵循XML格式，用来描述哪个服务器提供什么服务，怎样找到它，以及该服务使用怎样的接口规范，简言之，服务发现。现在，使用Web Service的过程变成，获得该服务的WSDL描述，根据WSDL构造一条格式化的SOAP请求发送给服务器，然后接收一条同样SOAP格式的应答，最后根据先前的WSDL解码数据。绝大多数情况下，请求和应答使用HTTP协议传输，那么发送请求就使用HTTP的POST方法。  
  
什么是REST？  
  
REST (REpresentational State Transfort) 形式上应该表述为客户端通过申请资源来实现状态的转换，在这个角度系统可以看成一台虚拟的状态机。抛开R. T. Fielding博士论文里晦涩的理论不说，REST应该满足这样的特点：1)客户端和服务器结构；2)连接协议具有无状态性；3)能够利用Cache机制增进性能；4)层次化的系统；5)按需代码。说到底，REST只是一种架构风格，而不是协议或标准。但这种新的风格（也许已经历史悠久？）对现有的以SOAP为代表的Web Service造成的冲击也是革命性的，因为它面向资源，甚至连服务也抽象成资源，因为它和HTTP紧密结合，因为它服务器无状态。**