《软件体系结构与设计》实习报告

题目：第 2 次上机实验

班级序号：111171

学生姓名：董安宁

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 10 月

一、实验概况

实验时间：2019 年 9 月 18（周三）晚上 18:30-21:30，共 4 课时

实验地点：未来城校区公教 2-503

实验目的：掌握设计、开发、测试、发布、调试经典三层架构软件的基本方法、工具和流程，理解层次体系结构风格基本原理、结构和特点。掌握设计系统时的“高内聚低耦合”的思想。

背景及要求：

[综述研究背景：概述本项工作的研究或观察的理论基础，给出简明的理论或研究背景，一定要列举重要的相关文献。若可能指出存在问题：说明为什么要做这项工作；阐述研究目的：说明有别于他人的“主意”（此红色字体一条不做强行要求）。]

三层架构就是将整个业务应用划分为：表现层、业务逻辑层、数据访问层。区分层次的目的即为了“高内聚低耦合”的思想，在软件体系架构设计中，分层式结构是最常见，也是最重要的一种结构，三层架构软件系统为用户的数据传输、提取、储存创造了便利条件。在应用数据时，信息划分架构开发项目，对各层次之间的工作职责进行清晰规划，这样就降低了系统的维护风险。

具体任务如下：

结合课堂上讲授的“一个简单的用户信息查询程序三层逻辑架构”原理，参考以下链接中给出的 C#代码，http://www.codeproject.com/Articles/36847/Three-Layer-Architecture-in-C-NET，完成：

1.结合以下示例数据库或自选其他相当规模数据集，使用 Java 设计实现一个三层架构的业务数据分析系统。各逻辑层的功能如下：

表现层：包含输入、查询相关的控件以及数据图表的展示（如百分图，折线图）；

业务逻辑层：数据处理、数据分析（不少于三项统计分析功能）、数据查询；

数据访问层：负责数据库的访问，主要职责为打开、关闭数据库、构建 SQL 查询、返回查询结果。

附：SqlServer 示例数据库 Northwind

https://www.cnblogs.com/mahongbiao/p/3764782.html

附：上证 1999-2016 的某公司股票走势数据，构建其业务数据分析系统。

2. 修改程序以适应三个逻辑层的分布式部署，要求三个逻辑分层分别部署于客户机（本机或手机）、AWS EC2 应用服务器和 AWC RDS 数据库服务器上（即多层 C/S 架构），部署完成后能通过公网 IP 访问该系统。

3.使用 RSA 或 Visio 等建模工具构建软件架构模型（UML 图），要求：

（1）画出逻辑分层结构图；

（2）画出每个逻辑层中所包含的核心构件（此处为类）；

（3）画出每个逻辑层中构件（类）之间的关系，且要细化到聚合(Aggregation)、组成（Composition）关系并给出重数（如 1:1,1:*）；

（4）画出系统部署结构图。

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

环境需求：

① 用户需要在 C:\Users\用户名\.aws 目录下设置 credentials 和 config 文件，用以保存自己的 AWS 账号信息。

② 用户只需要在一个终端进行操作，不需要管剩下几层是怎么运作的。

功能需求：

系统需要分为三层，每一层的进程分别在不同的主机上运行，因此需要完成：

① 客户端明确的操作界面

② 业务逻辑层完成数据处理

③ 数据访问层完成数据库的访问

④ 完成三层之间的通信过程

⑤ 各个程序需要部署在不同的主机

质量需求：

在遇到错误指令或者系统内部发生错误后可以显示出来，不会因此导致程序崩溃。

2.2 架构设计

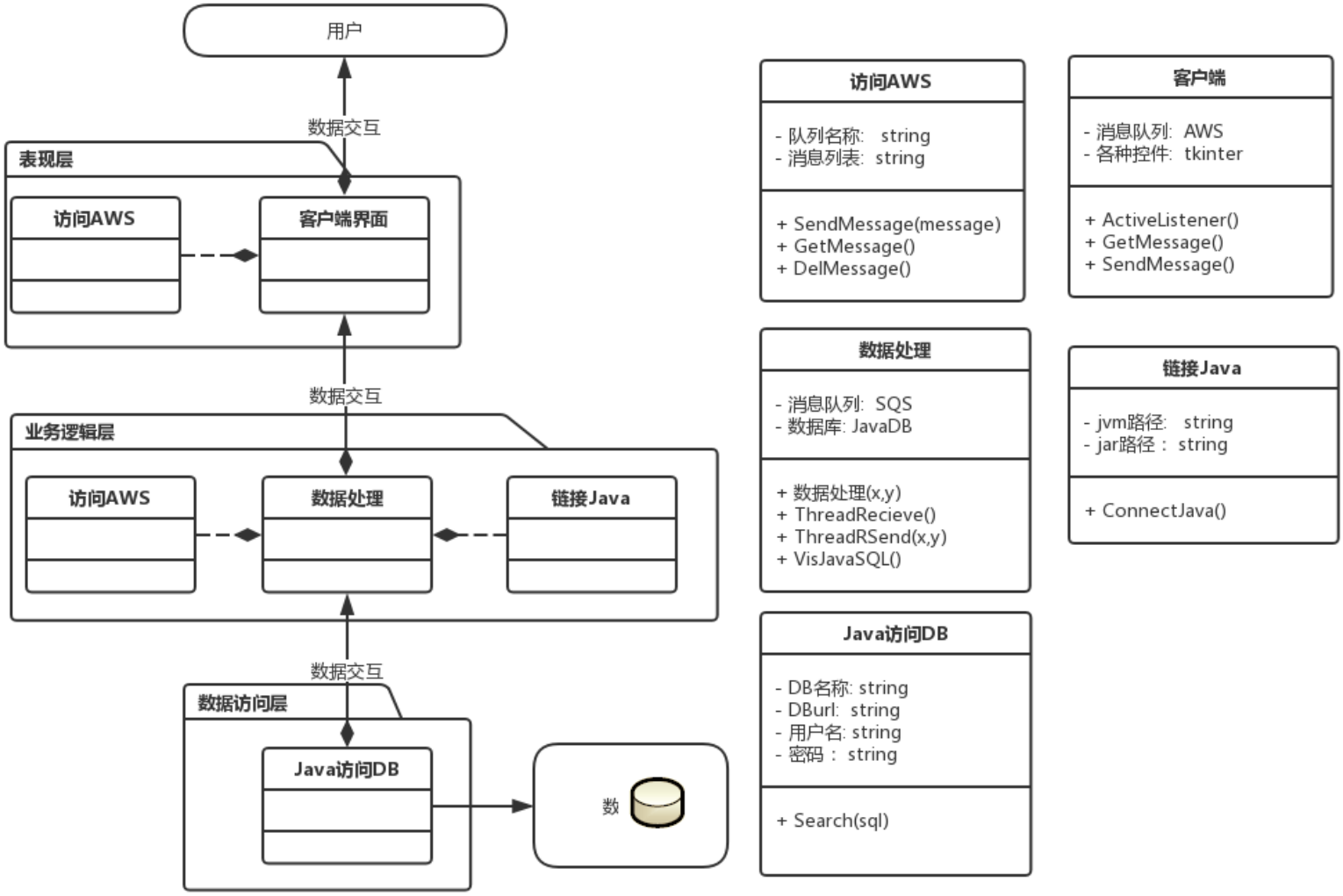


图 1 架构图

如图所示，我设置了 5 个构件，分别是

访问 AWS：

该构件的主要功能是链接到 AWS 的 SQS，完成消息的发送和接收。方法是通过 python 的 boto3 库访问 SQS，然后通过队列名称获得队列，将结果保存在消息列表中。在上一次实习的时候我也用到了这个功能，但是当时没有将他们封装到一个类中，这次就把他们封装了起来，调用起来异常舒适。

客户端：

该构件的主要功能是向用户展示界面。由于本次实习我主要把精力投入在了业务的处理中，所以没有过多的展示界面…界面的逻辑很简单，就是三个按钮分别向业务处理层发送不同的消息。发送完消息后，就持续处于接收状态，接收数据处理层处理好的数据。

数据处理：

数据处理层的功能有接收客户端的消息，访问数据库，处理来自数据库的数据，向客户端发送消息。在启动之后，数据处理层会持续地接收消息，只要收到消息就开始分析消息是要进行什么操作，然后据此选择要向数据库查询什么信息，也根据这个，对于数据库返回的信息也进行相应的处理，最终发送消息回到客户端。

链接 Java

由于不会直接通过 python 链接 AWS 的数据库，因此选择了曲线救国的方式…先用 java 写好访问 DB 的 jar 包，然后 python 再去调用该 jar 包，通过调用 jar 包里面的函数，完成查询 sql 语句的目的。

Java 访问 DB：

通过输入数据库的 url 和登录的用户名和密码，链接数据库，并且提供数据查询的服务。

2.3 接口设计

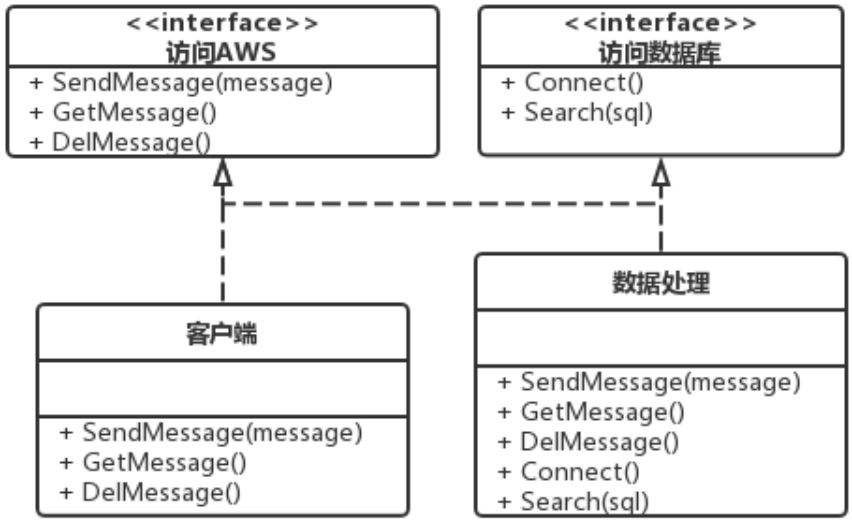


图 2 接口设计图

系统的主要接口是两部分，一部分是访问 AWS 的，一部分是访问数据库的。其中

访问 AWS：

提供向消息队列发送消息，得到消息，删除消息的功能。

访问数据库：

提供链接数据库，查询数据库语句的功能。

这些接口的作用分别是链接客户端和业务逻辑层之间的消息交互，以及业务逻辑层和数据访问层的消息交互。

三、实验过程

3.1 软件实现

本系统使用了三层架构的框架，系统开发的过程是先设计好框架，然后再根据框架所需要实现的类，功能，和接口，一步步地完成的。在开发的时候使用了迭代递增的方式，每次都只完成特定的构件，调试的没有问题，以及完成了一定的扩展性后，再向后进行开发，或者将该构件添加到模块中进行测试，最终一步步地由一个个小的模块搭建出总体的模块。

3.2 实验环境

处理器： i7-7700HQ　　操作系统：windows10
开发语言：python，java　　服务器：AWS
实验场景：宿舍

3.3 实验步骤

① 封装访问 SQS 和访问数据库的功能，分别由 python 和 java 的语言完成。如上面所说，这两个接口都要提供访问和返回数据的功能，经过测试无误后进行下一步，如下图所示：

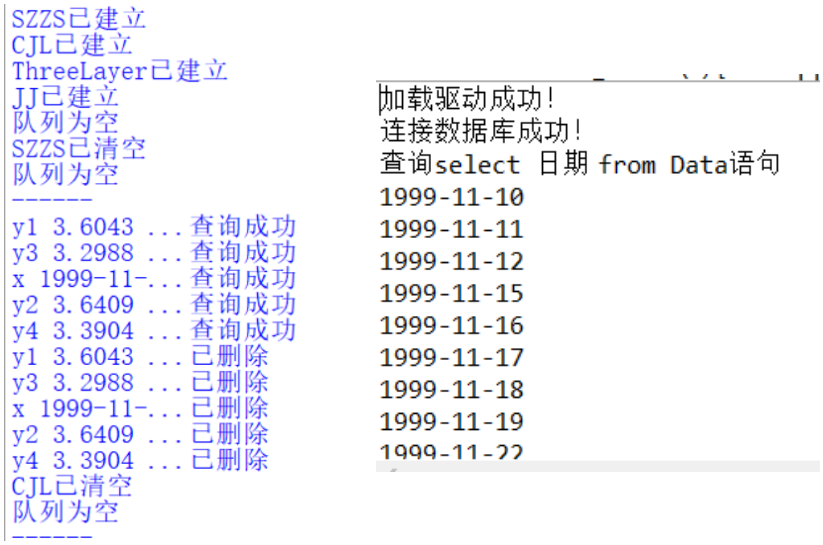
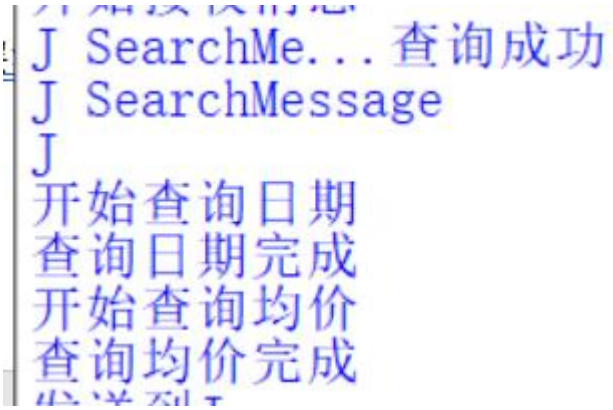


图 3 链接 SQS 和链接数据库接口功能的验证

② 实现客户端和数据访问层的第一层链接，即在这两层之间使用 SQS 发送和接收消息。在这里的时候就要明确客户端的界面发送方式，以及数据处理层的接收方式，并且要预先想好可扩展的结构，方便之后添加功能时的数据调用。测试好的结果如下所示：

图 4 服务层消息发送和逻辑层消息接收

③ 然后是实现业务处理层和数据库的链接,这里就要调用 JDBC 链接数据库了,首先要打包 java 的代码成为 jar 包，然后使用 python 动态调用 jar 包，使用其根据 sql 查询的功能，将结果返



回到 python 的代码形成列表，再对其进行下一步的处理，结果如下图所示：

图 5 数据库查询完成

④ 将从数据库查询到的消息经过一定的运算处理后，使用和①一样的方法，将数据发送回客户端，客户端接收到消息后，将数据绘制成图像，如下图所示

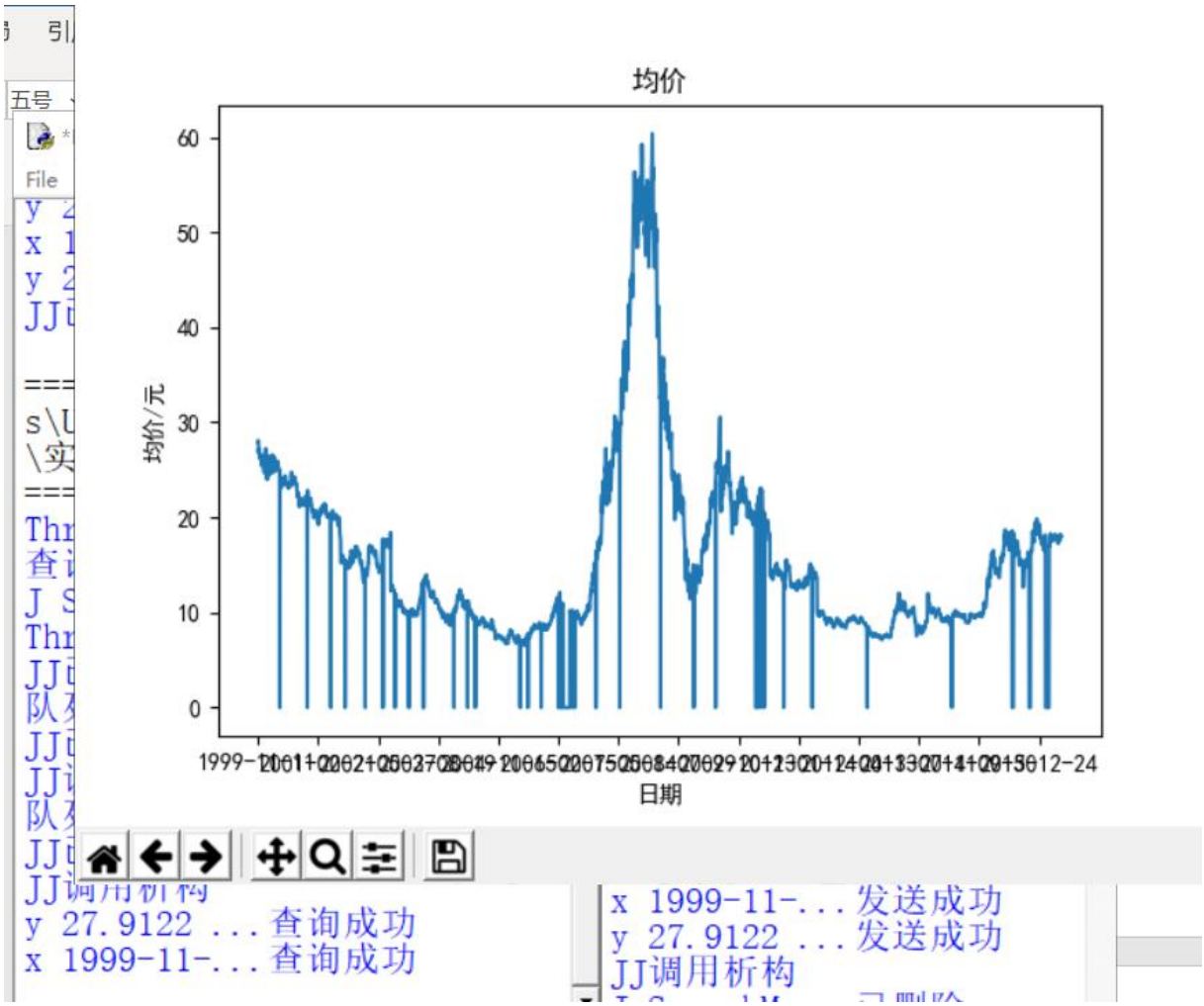


图 6 本地部署三层架构完成

⑥ 通过远程桌面链接到 EC2，使用百度网盘上传自己的代码以及环境配置的安装包，配置完成后在 EC2 上下载安装，然后运行服务器，最终将数据传递回本机，结果如下：

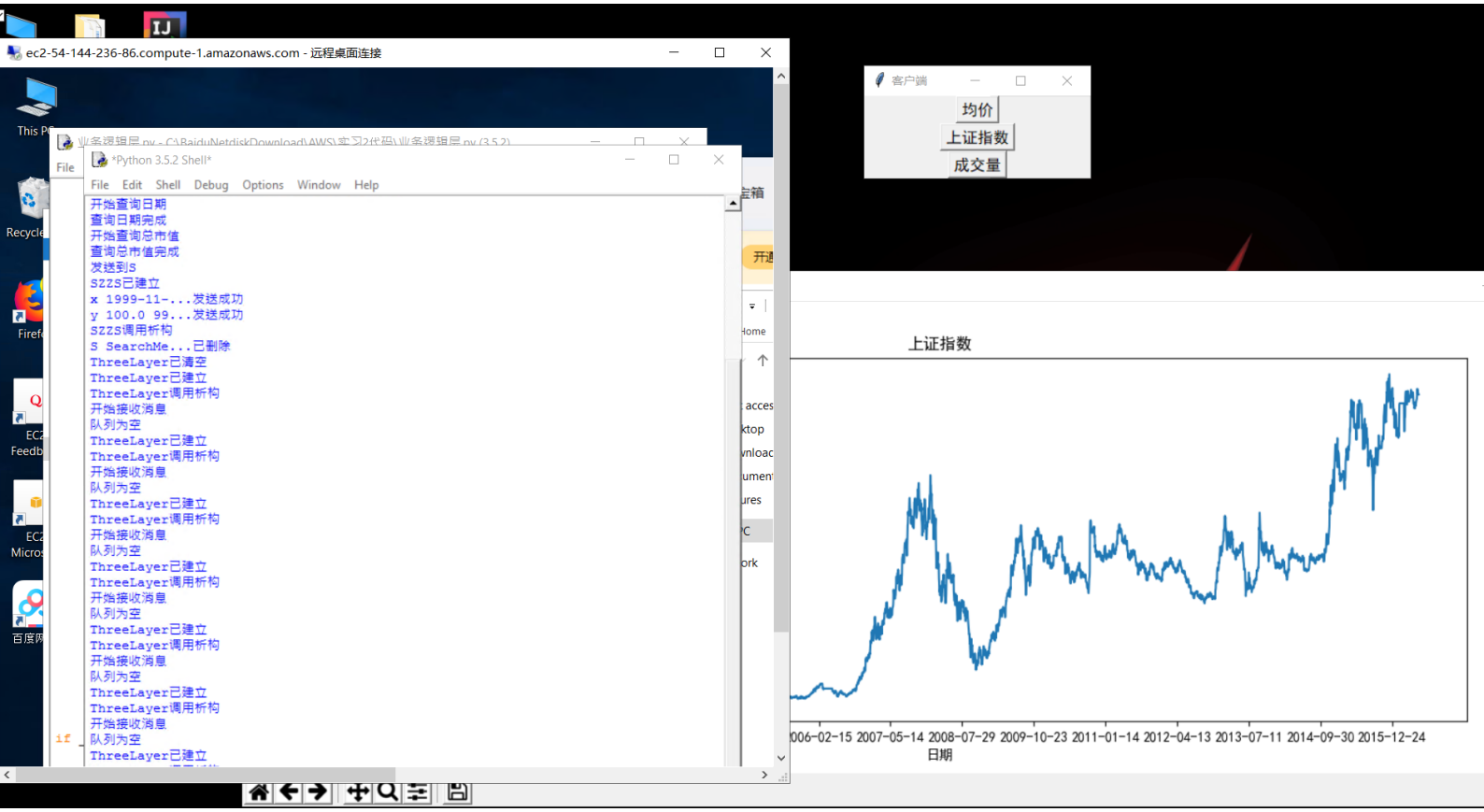


图 7 EC2 部署三层架构

四、实验评价

4.1 实验结果

实验结果在上面进行了给出，我认为较为完整地实现了三层架构的功能。在设计构件的时候也将每一层都设计成只有信息传递的接口的形式，实现了构件的独立性，减少了构件之间的耦合程度，通过数据的传输串联起整个系统。

4.2 结果分析

本次实习我认为实现了三层架构的功能，从结果来看，客户端，业务处理层，和数据访问，分别都处于不同的位置，但最终依靠数据传输链接到了一起。实际中的很多系统都像这种部署结构一样，用户下载客户端，无需管理其数据发送是如何做的，以及后台的消息处理是如何运行的，只需要掌握如何操作客户端就可以了。

实验的局限性在于依然是一个小的课设项目，和大型的工程相比，我认为并不能以小见大，因为大型的工程无论是数据处理还是网络链接等处理方式都有更严格的要求和更成熟的解决方案，以及 AWS 对实验的阻碍太大了= =连不上的时候真的是 nanding，想必在现实的开发中肯定是不允许这种事情出现的，不过对结果的影响不是很大，影响的是编程时的心情和快乐程度。

五、总结

(1) 本次实习完成了三层架构的处理目标，实现了不同应用之间的数据传输，学习到了很多知识。和上一次的实习相比，我在本次实习中更重视了代码的结构，以及重用复用的功能。相比上次实习时每次都要单独调用发送和接收 SQS 的语句，我在本次实习中将他们进行封装，在使用的时候更加方便并且直观。同样的我对访问数据库的操作也做了这样的封装，在开发的时候相关的代码就变得更加精悍，只需要调用面向对象的一些方法就可以了。在数据处理层由于对于数据要进行相似的处理，我将不同的处理方式都汇总在了一个函数中，只通过传入数据的标识来决定该执行哪个。这样提高了构件的复用性和可扩展性，当需要添加处理的数据的时候，就可以通过在该函数中添加判断条件和处理的代码来添加功能。本次实习不足的地方在于界面没有好好设计，以及对数据的处理并不是很复杂，功能不是很丰富，我认为这是我没有很好安排时间的结果，因此下次实习的时候我也会注意对时间的安排，加强对时间的利用，尽快的实现更多的功能。

(2) 开发中遇到的问题主要有两个，一个是链接数据库，一个是在 EC2 上配置文件。链接数据库的时候由于尝试了一下午+一晚上无法用 python 连接到 RDS 数据库，因此我最终选择了使用 java 链接数据库，然后使用 python 调用 java 代码，完成调用数据库的功能。在这过程中我掌握了 jar 包压缩和配置，以及对外部依赖项的链接，收获还是很大的。配置 EC2 的时候我是使用百度网盘将所有文件传输到了 EC2，除了有延时以外，在这上面配置环境没有想象中的难。总之还算是顺利了解决了问题。

(3) 本实习的应用场景很广，各种系统都有可能用到三层架构的模型，不过从我这里写的时候已经发现了三层架构的一些不好的地方，比如数据传输的过程过于复杂，针对不同的类型还要添加不同的发送格式。不过如果用套接字的话也许会更好一些。

以上就是本次实习的总结，其中缺点和不足都会在下次实习中改进。

参考文献：

[1]jar 包相关：https://blog.csdn.net/weixin_39407066/article/details/83625316

[2]python 调用 jar: https://www.cnblogs.com/ai594ai/p/8615818.html

附件：

代码已经上传到 GitHub：https://github.com/UncoDong/SAInCug