**网络程序的服务器部署**

1. 购买云服务器

利用课程提供的华为云代金券选择适合本项目的服务器，并且进行初始配置。配置选择所考虑的因素：

1. 性能需求：根据访问量等业务需求选择合适的CPU、内存、存储空间等配置。
2. 安全等级：根据华为云所提供的安全措施，如防DDoS攻击、数据加密、访问控制等进行考量。
3. 操作系统选择：windows、linux及其拓展版。
4. 稳定性：选择性能稳定的云服务器以减少维护费用和保证业务连续性。
5. 可拓展性：确保云服务器具备良好的可扩展性以适应业务发展。
6. 成本控制：在满足性能和稳定性要求的前提下，考虑价格因素。

根据本项目需求，侧重网络程序的搭建以及部署实践，暂未考虑备案发布并且进行商业化，所以网站的访问量并不大，对性能需求不高；安全等级也并不在考量范围之内；由于linux系统的灵活性以及在服务器端的先天优势；结合成本控制。我们最终选择了单核2G，40G存储的CPU进行项目部署，并且选择linux稳定发行版Ubuntu20.04 server版本的操作系统进行配置。相比于宝塔面板，ubuntu原生系统具有很多优势，例如性能强悍、可拓展性强、内存占用低等。

1. 配置安全组

安全组是一个逻辑上的分组，为具有相同安全保护需求并相互信任的云服务器提供访 问策略，类似于Windows的防火墙。安全组创建后，用户可以在安全组中定义各种访问规则，当云服务器加入该安全组后，即受到这些访问规则的保护。

系统会为每个用户默认创建一个默认安全组，默认安全组的规则是在出方向上的数据 报文全部放行，入方向访问受限，安全组内的云服务器无需添加规则即可互相访问。默认安全组可以直接使用，也可以根据需要创建自定义的安全组。

  对于刚购买的服务器公网ip需要添加以下安全组规则，以实现公网访问服务器：

1. 开放TCP-22号端口作为远程登录端口。
2. 开放TCP-5000端口作为网站服务器端口。
3. 开放TCP-3306端口作为远程访问控制Mysql服务器端口。



图1 协议端口配置

1. 远程登录与环境搭建

本项目通过使用远程登录连接华为云服务器。

使用SSH客户端（如PuTTY或Terminal）通过开放的TCP-22端口远程登录到服务器。通过输入服务器的公网IP地址，以及购买时设置的用户名和密码。

连接好服务器之后我们需要进行环境设置，使得与本地的环境相匹配。并且通过FTP

件传输协议，我们把本地完成的项目上传到云端，同时修改相应的路由设置进行服务器匹配，与数据库连接工作。

**电商网站的支付宝接入模拟**

1. 原理

我们设计的支付原理如下图所示：

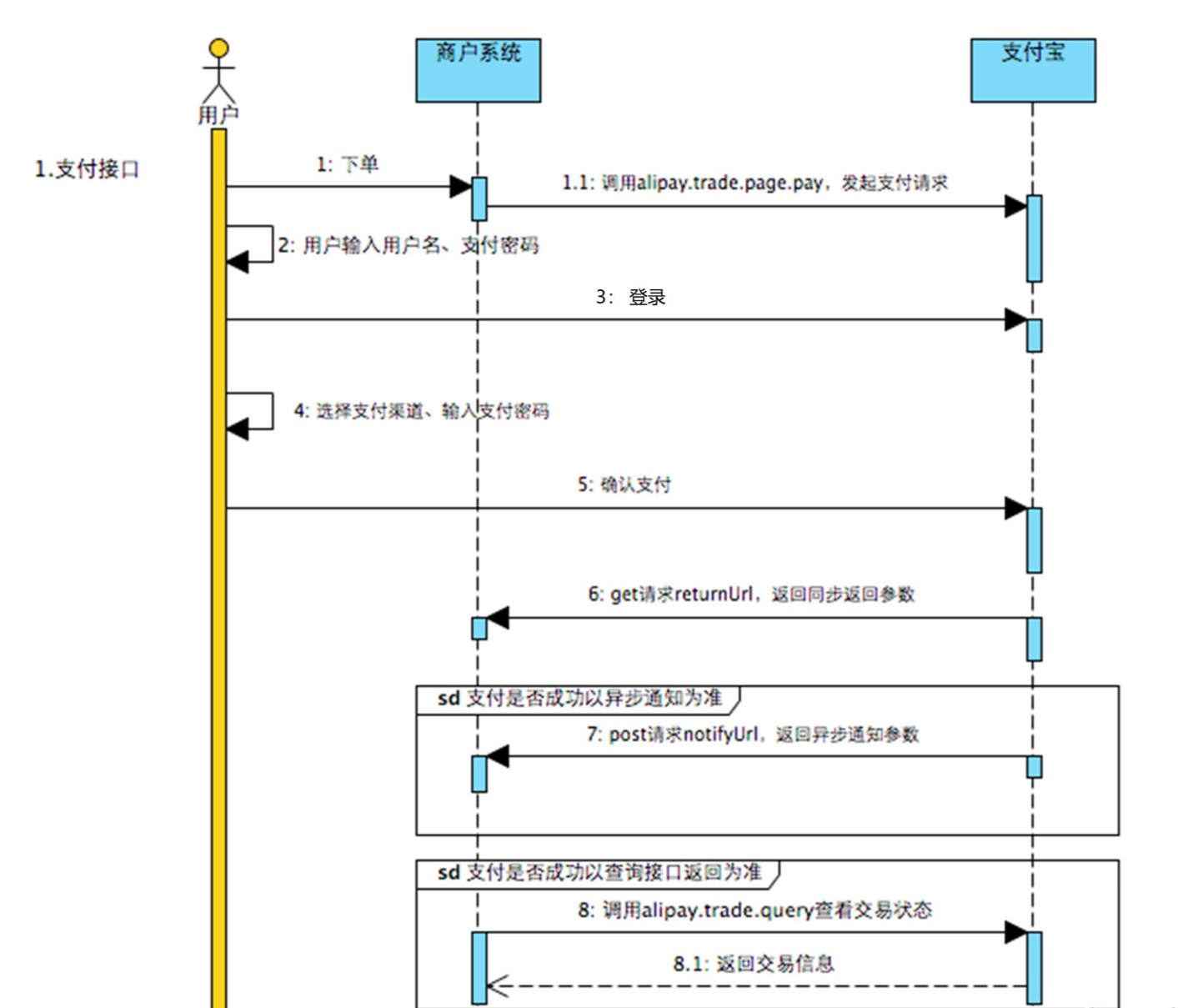


图2 支付宝支付拉取原理图

1. 触发场景：
2. 用户确认了订单并且选择支付方式
3. 点击立即支付，在点击立即支付后发送一个拉起支付请求到我们的服务器，参数就是订单的基本信息。
4. 服务端响应：
5. 服务器端处理用户的下单请求。
6. 根据客户端传来的订单信息，计算出每一个商品的单价、订单的总金额等（金额是不应该由客户端直接传的，应该由服务器端计算，更安全），再生成一个订单号。
7. 调用支付宝支付接口，订单信息进行上传，成功后支付宝会返回一段html文档，就是平时在支付时经常看到的带二维码的PC端支付页面。
8. 当买家扫码支付完成以后，支付宝会调用同步回调API接口跳转到一个自定义的成功页面（这个页面只是告诉用户支付成功了，只做展示用途，并非真正的支付回调）
9. 支付完成以后，支付宝会调用异步回调API接口，将一些参数传给服务端，如交易流水号、订单号，服务端可以根据这些信息查询交易是否真的成功了(第8步)，从而执行后续的业务。
10. 支付宝沙盒

支付宝有一个供开发者测试使用的沙箱环境，会提供一个沙箱版的支付宝app、一个商家账户、一个买家账户。有了这个，可以让我们跳过商家入驻、企业资质审核等过程，开箱即用。从而完成支付模拟的功能。

1. 私钥与公钥

对于数据的安全考虑，工具的使用需要生成一份公钥和私钥，公钥提供给支付宝，支付宝对数据进行加密；私钥用于解析支付宝传来的加密数据，由我们自行保管。支付宝开放平台为我们提供了密钥生成工具（开发助手）来对应用的客户端服务端之间的交互进行加密保护。工具主要功能有生成密钥、签名、验签、格式转换、密钥匹配、智能反馈、开放社区。



1. Alipay模块的类

本项目所使用到的Alipay需要自己编写格式转换、字符串排序、参数传递以及服务器请求等功能，支付宝只提供数据的加密与页面返回。这里我们通过设置类Alipay来进行上述操作的打包。相应内容如下（为了完整显示出结构，函数的参数以及具体细节并未展示）：

class AliPay(object):

    def \_\_init\_\_(): #构造函数，进行对象的初始化

    def direct\_pay():#支付界面参数生成，通过给这个函数传参返回对应的url参数

    def direct\_query():

    def build\_body(): #数据结构定义，字典格式

    def sign\_data(self, data): #签名函数

    def ordered\_data(self, data):

    def sign(self, unsigned\_string):

    def \_verify(self, raw\_content, signature):

    def verify(self, data, signature):

**管理员后台界面设计思想**

为了不重复用户界面的操作，尽量做到更多工具的学习与使用，加深对项目的印象，本项目管理员后台界面使用Axure RP软件设计。

1. 平台介绍

Axure RP是一款专业的快速原型设计工具。Axure代表美国Axure公司；RP则是Rapid Prototyping（快速原型）的缩写。

Axure RP的使用者主要包括商业分析师、信息架构师、产品经理、IT咨询师、用户体验设计师、交互设计师、UI设计师等，另外，架构师、程序员也在使用Axure。

Axure RP Pro能快速帮助设计者设计出快捷而简便的创建基于网站构架图的带注释页面示意图、操作流程图、以及交互设计，并可自动生成用于演示的网页文件和规格文件，以提供演示与开发。

1. 平台优势：

Axure这个软件功能强大，可以做出很多交互效果，直接通过拖拽画出原型、生成网页；Axure这个软件效果高，只有遵循一些可用性原创，很快模拟出高保真页面原型。避免掉了直接写html效率低的问题。

1. 页面介绍



图3 基于Axure开发的管理员后台首页

目前本页面主要开发了四个功能：

1. 添加商品页面跳转
2. 删除商品页面跳转
3. 修改商品页面跳转
4. 管理员注册页面跳转

通过添加js页面跳转函数实现此功能。

后续关于订单的管理也可添加至本页面，作为此课程项目的拓展。通过Axure的发布功能可以将此页面生成为HTML，以及其附带的静态原件，只需要放到本项目flask框架下的对应位置，并且改变HTML文件中的相对寻址路径即可完成迁移（Axure项目与Flask项目的base\_dir不相同）。