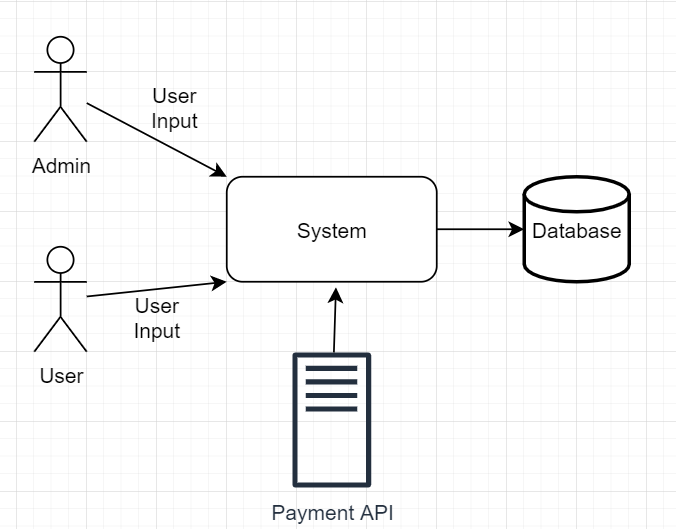
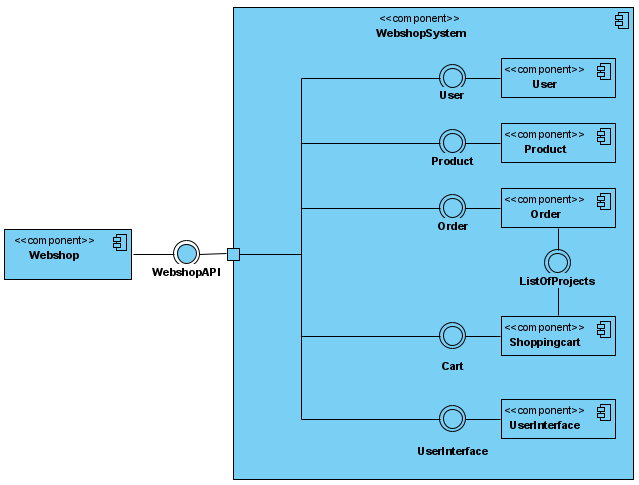
Distributed Software Architecture

# Context Diagram



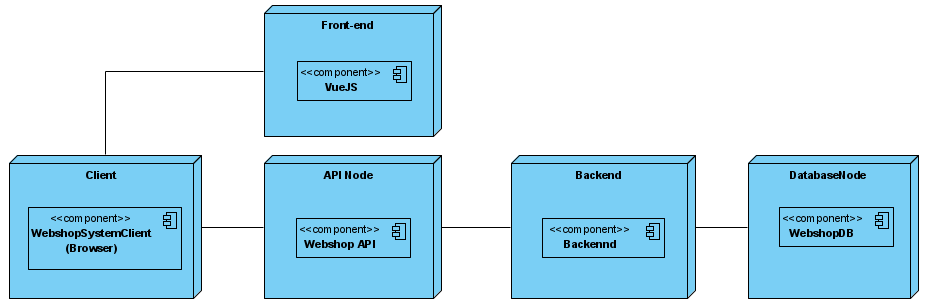
In my system there will be two usertypes both will have a other system with different functionalities. I will be using one database to store all the data. Maybe I’m going to use a Payment API but I have to do some more research to make a decision.

# Components Diagram



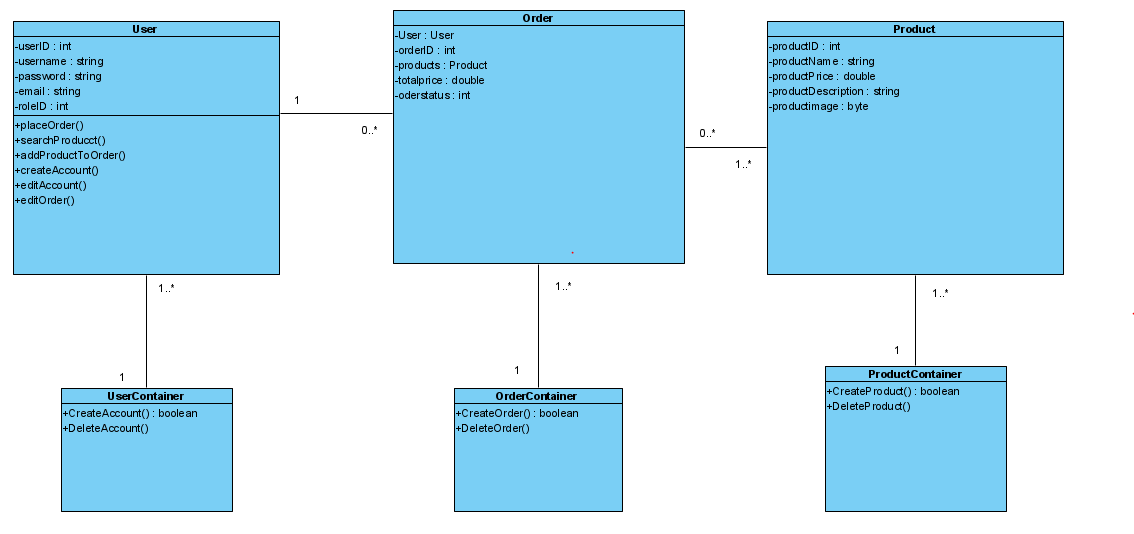
In my component diagram each component has its individual interface, each of the component are linked to the gateway that goes to the clients webbrowser.

# Deployment Diagram



My application will be deployed in different pieces, so i will deploy my backend and frontend separated from each other. I think I am going to divide my backend into two pieces (one for the user and one for the admin). This is so the user can’t get to the admin funtionalities. My database will also run on a separate server.

# Class Diagram



For my application I think that i’m going to use three classes, each one has a container to create the classes. This diagram is still a Work in Progress, so I will go ask some feedback and then change the diagram a bit.

# Which questions should I be able to answer after this module?

1. What is a distributed software system?  
   *A distributed software system is a system with multiple software components that are on multiple computers, but run as a single system.*
2. What is the difference between a monolith and a distributed software system?  
   *A monolith software system is a single tiered software application. The user interface and data access code are combined into a single program, that runs from a single platform.  
   So the difference between a monolith and a distributed software system is that a monolith system is a single program on a single platform. And a distributed system has multiple components, on multiple computers but runs as a single system.*
3. Which advantages does a distributed software system offer?

*A distributed software system is very maintainable. Also when one of the services crashes, it will effect your other services. With a distributed software system it’s very easy to add a service, what makes a distributed software system very scalable.*

1. Explain why your architecture is distributed  
   *My architecture is distributed because, I am going to use multiple platforms to run my application. So if one part crashes, the application won’t crash completely but only one part of the application won’t work.*
2. What are the important architectural principles and techniques when developing a distributed software system?  
   *One of the vital skills of an architect is tob e able to view the arhitecture from many different standpoint: each one of them individually might not be fully relevent, but combining them together gives a helicopter view of the product.*