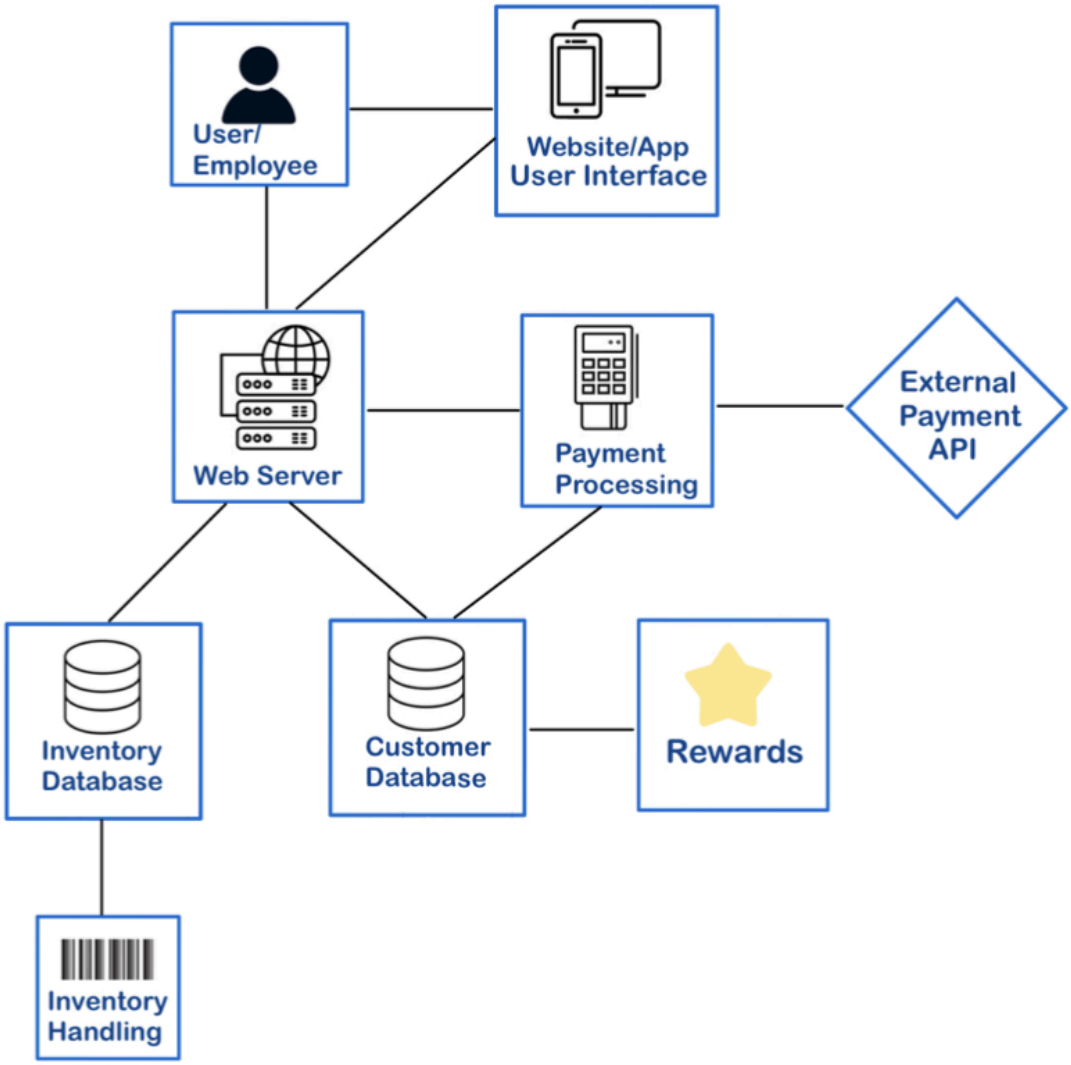
Phoebe Griffiths
Han Luu

1. **Software Architecture Diagram:** This is an architecture diagram that should reflect your most recent design, and reflect your data management strategy. This diagram can take any form you feel appropriate. You are also free to submit or modify an existing diagram you have from a previous assignment.
    a. Updated Software Architecture Diagram:



2. **Data Management Strategy:** All of your software systems revolve around persistent and potentially sensitive data. Here you should specify your data management strategy, SQL or non. (Hint: diagrams always help) You should also discuss your design decisions: how many databases you chose and why, how you split up the data logically, possible alternatives you could have used (both in technology and organization of data), and what the tradeoffs are between your choice and alternatives.

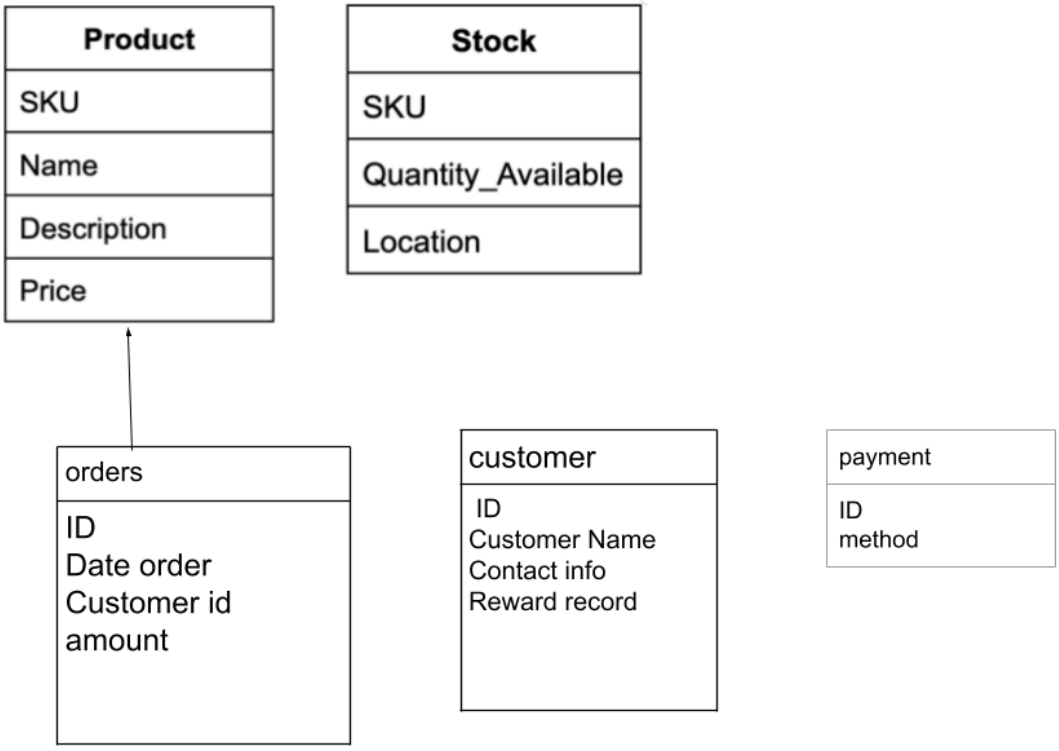
In this manicured design, ShopEase implements only two functional databases: one for customer information and one for inventory management. This setup upholds a standard of balancing performance while still completing core features for the application. We think using two databases within our system makes the most sense as it optimizes performance, security, and simplicity. Each database is easily accessible for the types for queries made to the system so operations like payment processing and inventory updating happen seamlessly. The separation of customer and product data allows for more privacy by reducing the risk of information leaks. Separate databases allow for separate policies to be enforced depending on the sensitivity of data being stored. Lastly, a standard of simplicity is upheld with having just two databases. Less administrative overhead is needed which is beneficial if a smaller team is incorporating this software.

SQL databases are the preferred choice mainly for their relational nature. With an online shopping software system, complex relationships between products, customers, inventory, sales, rewards, etc. exist, so maintaining data accuracy and consistency is critical. SQL databases support relationships between different types of data through keys and other constraints that ultimately limit errors in the information being stored.

Inventory Database: This database is for all information related to products including descriptions, prices, stock updates, and SKU.
Customer Database: This database stores all customer-related data including personal information, purchase history, and rewards points.

An alternative plan was to have only one database, but ultimately this didn't align with the standards outlined in the software's mission. Although one database could simplify data management and reduce complexity when multiple databases interact with one another, the increase in security was not a valid tradeoff. One database could also complicate scalability.

Inventory Database SQL Management Strategy: two tables above
Customer Database SQL Management Strategy: three tables below
We decided to link two databases together. Linking the customer database with the inventory database using a key allows for efficient access to product information when customers place orders. By establishing this link, we enable the order processing system to check the availability of items in the inventory before completing the order.


Product
SKU
Name
Description
Price
Stock
SKU
Quantity_Available
Location
orders
ID
Date order
Customer id
amount
customer
ID
Customer Name
Contact info
Reward record
payment
ID
method