**Order Management System High-Level Design**

This document contains a high-level description of the Order Management System which is to be implemented as microservices using Spring Boot. The following three microservices are to be implemented as part of Order Management.

1. User Service – Users can be buyer/seller/visitor. Buyer/Seller can login or register by providing their credentials. Buyer can opt for “privileged mode” using reward points. Buyer/seller should be able to inactivate their account.
2. Product Service – Product details need to be available to visitors, search products based on category and product name, Add/Removal/Move products to wish list, updation of product stock
3. Order Service – Add/Remove Products from cart, place an order, view orders placed on products

The project is to be developed with all functionalities and deployed in accordance with Agile process.

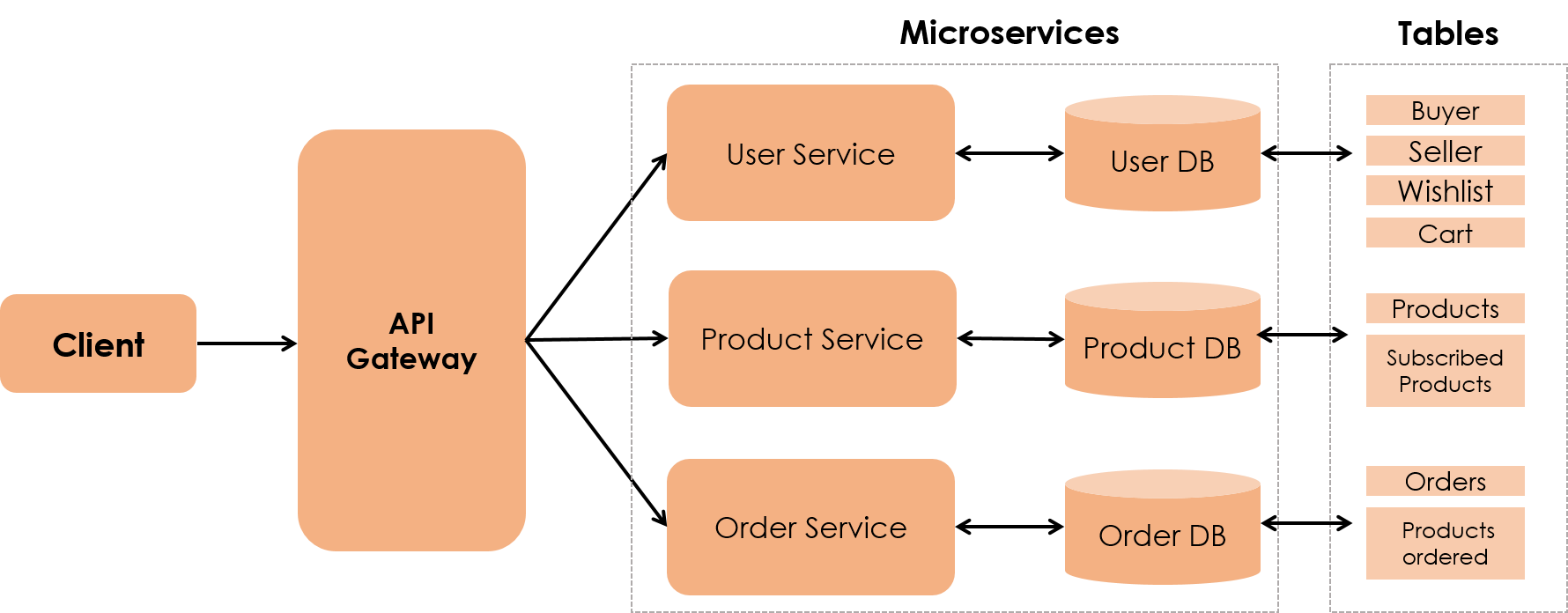
Sprint-1 would be development of each of the services based on user stories developed and need to perform vulnerability analysis using SonarCloud. The end points are to be tested using Postman.

A sample endpoint is as below:

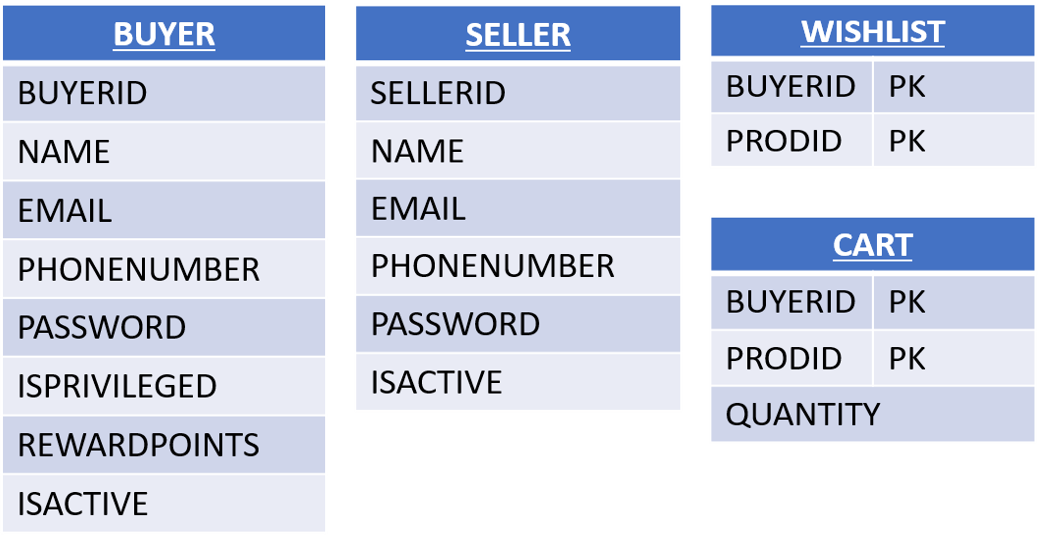
|  |  |
| --- | --- |
| Method | Endpoints |
| POST | api/seller/register |
| POST | api/seller/login |
| DELETE | api/seller/deactivate |
| GET | api/seller |

Sprint II would be setting up a Continuous delivery pipeline for container-based microservices using AWS Code Pipeline and Amazon ECS.

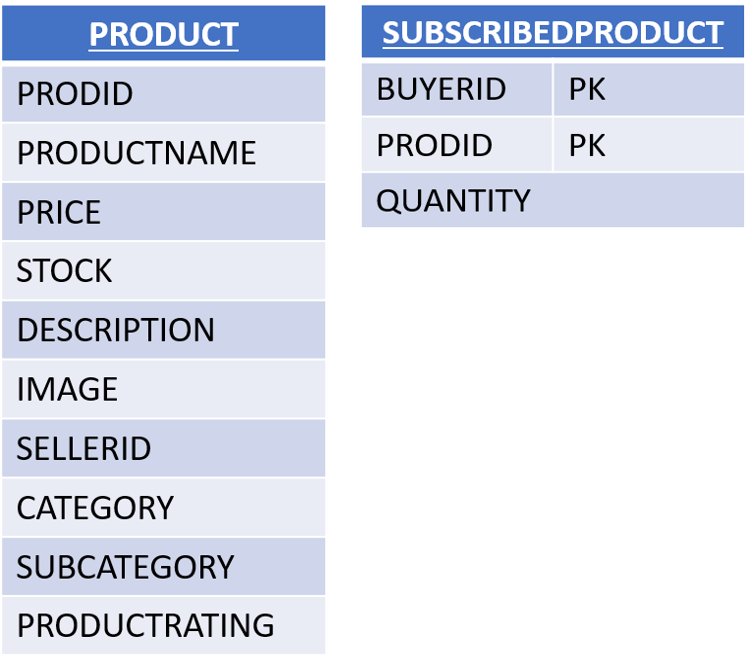
The following diagram shows the microservices architecture and tables used:



**USER DB**



**PRODUCT DB**



**ORDER DB**

