Microservices Assignment

Use Case

Develop a backend for a web application of an ecommerce company using Microservices Architecture. The application should be able to perform following operations in first release (design must be extendable for further operations)

1. Admin can add/remove new products to inventory
2. Customers can search for the available products
3. Customer can add/remove products to his cart
4. Create an order for a user (Checkout cart)
5. Make payment for the order
6. Delivery team should be able to mark an order as delivered or undelivered due to some reason
7. View all orders along with status for a user

Cross-cutting concerns to be taken care of

1. Logging
2. Exception handling
3. Scalability
4. Security

Tools/Technolgies

1. Any language of choice for writing microservices
2. Service Discovery (Eureka for Java/DotNet, other languages can choose any)
3. API Gateway for implementing routing (Zuul for Java, Ocelot for DotNet, other languages can choose any)
4. Docker as deployment tool
5. A suitable kind of authentication mechanism to secure your APIs (Token based, JWT, username/password basic authentication etc. of your choice should be implemented)

Deliverables

1. A writeup of identified microservices for above use case along with reasonable explanation
2. URL definitions of the scenarios (Sample POSTMAN collection, or request/response JSONs)
3. Docker images to be built using Dockerfile and pushed to dockerhub (links to be submitted in document)
4. Source code of all microservices
5. **A batch file (.bat) that will start up the system using docker commands (or optionally docker-compose can be used). Intention is that your application should be able to get up and running on any other system properly.**

Guidelines:

1. This assignment only requires to identify all the microservices for above mentioned use case and expose the API endpoints.
2. No User interface is required.
3. Mention relevant assumptions taken while implementing application.
4. Your solution should be able to build/compile and run.
5. Standard coding guideline should be strictly followed.
6. Integration with database is not mandatory, in memory data structures (lists,maps) can be used to store data temporarily for the assignment.
7. Choose the proper communication and microservices design patterns as per the scenario and design.