# Case Study- Redeem Mall

RedeemMall is an ecommerce application where users can redeem their loyalty points earned through credit card. Users cannot order items exceeding the loyalty points earned by them. As this ecommerce application has become popular and its users are growing exponentially as well as changes in application also seem to be frequent, RedeemMall owner wants this application to be designed using Microservice style of architecture.

You need to design micro services for this ecommerce application. A customer can order certain number of products available in this ecommerce application and the value of this order is deducted from the loyalty points accumulated by the customer. An order placed by the customer must pass the following validations:

* Maximum number of items in an order should not be greater than 5 (may change in future)
* Maximum quantity of items should not exceed 10 (may change in future)
* The overall cost of the order must not exceed loyalty points earned by the customer.
* In case overall customer’s order exceeds loyalty points, then application should suggest customer to reduce the quantity of the product with the least value possible.
* Quantity of each item must not exceed the inventory available with the mall.

You can use any RDBMS already installed on your laptop or you can use H2 database. You can also use collections like Arraylist, Hashmap, etc. and in such case; you do not need to depict relationship between entities like Customer, Order, etc.

You should follow RESTful principles and coding best practices while implementing this solution. You must implement proper exception handling and logging.

You may design three services (or more based on your understanding) for this application:

1. Order Service
2. Customer Service
3. Inventory Service

Order Service may have APIs like

* Accepting Order

Customer Service may have API for

* Checking whether customer has required loyalty points

Inventory Service may have API for

* Checking whether quantity of a product ordered by customer is available or not

You should not take care of any concurrency issues and transaction handling.