**Câu Trả Lời:**

**Câu 1**: Dưới đây là một số cách để đảm bảo mỗi thẻ cào chỉ được sử dụng cho một người duy nhất nếu không yêu cầu xác thực(AuthN) hoặc kiểm soát quyền truy cập(AuthZ) :

* Sử dụng Token duy nhất cho mỗi người dung :

+ Khi tạo thẻ cào, hãy liên kết mỗi thẻ với một token duy nhất cho người dùng đó.

+ Mỗi lần người dùng thực hiện một yêu cầu, họ sẽ phải cung cấp token của mình để xác định.

* Giới hạn phạm vi của thẻ cào :

+ Đảm bảo rằng mỗi thẻ cào chỉ có thể được sử dụng để một số lượng cố định của các yêu cầu hoặc trong một khoảng thời gian cụ thể.

+ Sau khi số lượng yêu cầu được thực hiện hoặc sau thời gian cụ thể, thẻ cào sẽ trở nên vô hiệu

* Sử dụng hàng đợi(Queue) :

+ Sử dụng một hàng đợi để xử lý các yêu cầu từ thẻ cào. Mỗi thẻ cào được xử lý lần lượt theo thứ tự đến từ hàng đợi

+ Điều này giúp đảm bảo rằng mỗi thẻ cào chỉ có thể được xử lý bởi một yêu cầu tại một thời điểm, giảm khả năng trao thưởng cho nhiều người

* Ghi nhận và Theo dõi :

+ Ghi nhận mọi yêu cầu và kiểm tra xem có bất kỳ sự lạc quan nào xuất hiện.

+ Theo dõi các thẻ cào đã được sử dụng và thông tin người dùng liên quan.

* Giám sát và Cảnh báo :

+ Thiết lập hệ thống giám sát để theo dõi các hoạt động đáng ngờ hoặc vi phạm an toàn.

+ Cảnh báo khi phát hiện bất thường và thực hiện biện pháp phòng ngừa.

***Câu 3***: Một số đề xuất kiến trúc/Giải pháp để scale hệ thống :

* Sử dụng microservice
* Sử dung Kubernetes để quản lý và triển khai các container.
* Sử dụng các hệ thống caching như Redis, Memcached, Nginx, Apache traffic Server, … để giảm áp lực lên cơ sở dữ liệu.
* Sử dụng các giải pháp như sharding, replication, hoặc caching để tối ưu hiệu suất và đồng bộ dữ liệu.
* Sử dụng các biện pháp bảo mật như OAuth, HTTPS, và cấu hình chính xác các quyền truy cập
* Sử dụng các công cụ như Prometheus, Grafana, và ELK stack để theo dõi và ghi lại thông tin về hiệu suất, lỗi, và log của hệ thống.
* Sử dụng load balancer để phân phối tải đều giữa các instances của ứng dụng. Ngoài ra Spring Cloud có thể tích hợp với Ribbon để cung cấp cân bằng tải dựa trên các chiến lược khác nhau.