1h 28m left

ALL

1

2

1. Code Question 1

Amazon Shopping recently launched a new item whose daily customer ratings for *n* days is represented by the array *ratings*. They monitor these ratings to identify products that are not performing well. Find the number of groups that can be formed consisting of 1 or more *consecutive* days such that the rating continuously decreases over the days.

The rating is *consecutively decreasing* if it has the form: *r, r - 1, r - 2...* and so on, where *r* is the rating on the first day of the group being considered. Two groups are considered different if they contain the ratings of different consecutive days.

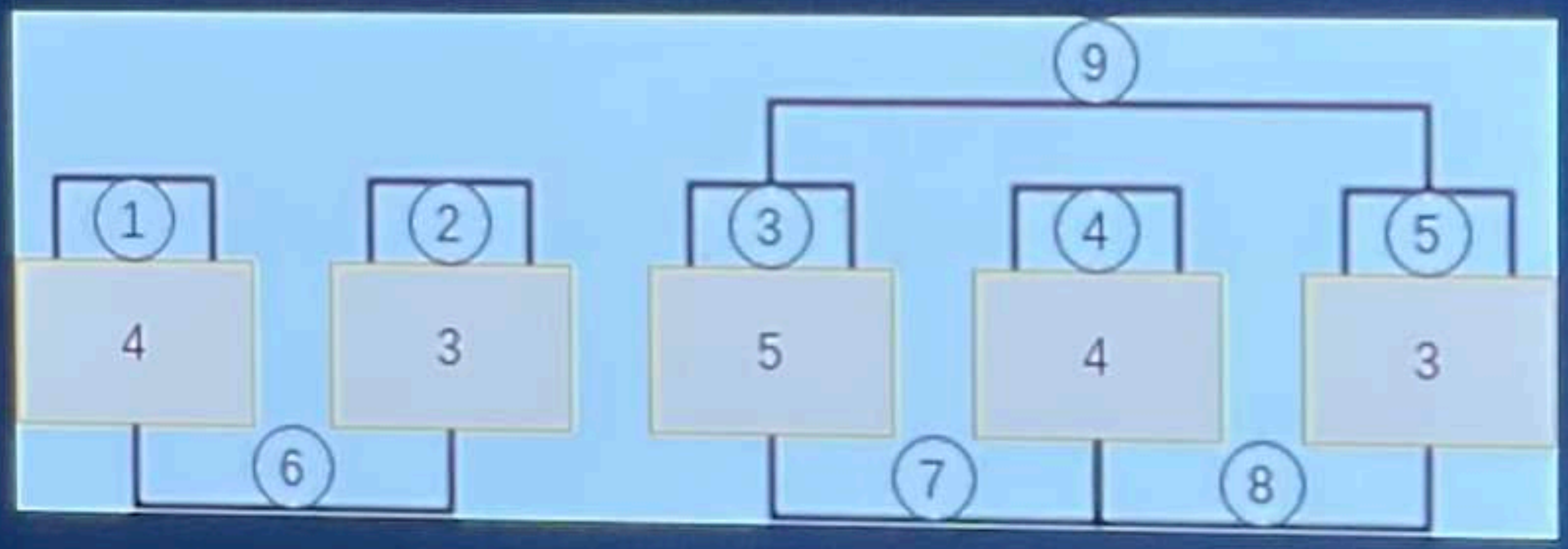
Example

ratings = [4, 3, 5, 4, 3]

There are 9 periods in which the rating *consecutively* decreases.

5 one day periods: [4], [3], [5], [4], [3]
3 two day periods: [4, 3], [5, 4], [4, 3]
1 three day period: [5, 4, 3]

These can be visualized in the figure shown below.



Return 9.

Function Description

Complete the function *countDecreasingRatings* in the editor.

countDecreasingRatings contains one parameter:
int ratings[n]: customer ratings for *n* days

Returns

long: the number of valid groups of ratings

Constraints

- $1 \le n \le 10^5$
- $0 \le ratings[i] \le 10^9$

1h 28m left

ALL

1

2

BETA Can't read the text? Switch theme

2. Code Question 2

Amazon Fresh is a new grocery *store* designed from the ground up to offer a seamless grocery *shopping* experience to consumers. As part of a stock clearance exercise at the store, given many piles of fresh products, follow the rules given below to stack the products in an orderly manner.

- There are a total of *n* piles of products.
- The number of products in each pile is represented by the array *numProducts.*
- Select any subarray from the array *numProducts* and pick up products from that subarray such that the number of products you pick from the i^{th} pile is strictly less than the number of products you pick from the $(i+1)^{th}$ pile for all indices *i* of the subarray.

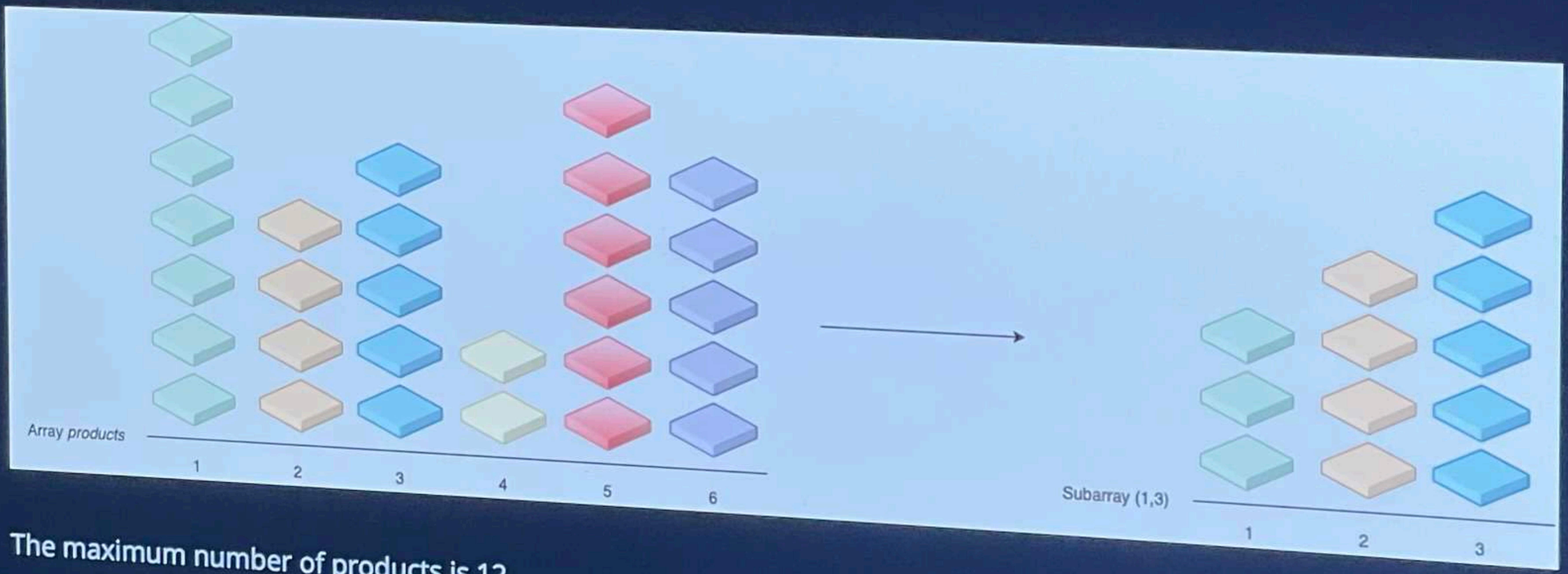
Find the maximum number of products that can be picked.

Example

The numbers of products in each pile are *numProducts = [7, 4, 5, 2, 6, 5].*

These are some ways strictly increasing subarrays can be chosen (1-based index):

- Choose subarray from indices (1, 3) and pick products [3, 4, 5] respectively from each index, which is 12 products. Note that we are forced to pick only 3 products from index 1 as the maximum number of products we can pick from index 2 is 4 and we need to make sure it is greater than the number of products picked from index 1.
- Choose subarray from indices (3, 6) and pick products [1, 2, 4, 5] respectively from each index, which is 12 products. Similar to the above case, we are forced to pick only 1 product from index 3 as the number of products at index 4 is only 2.
- Choose subarray from indices (3, 5) and pick products [1, 2, 6] respectively from each index, which is 9 products.
- Choose subarray from indices (1, 1) and pick all the 7 products.

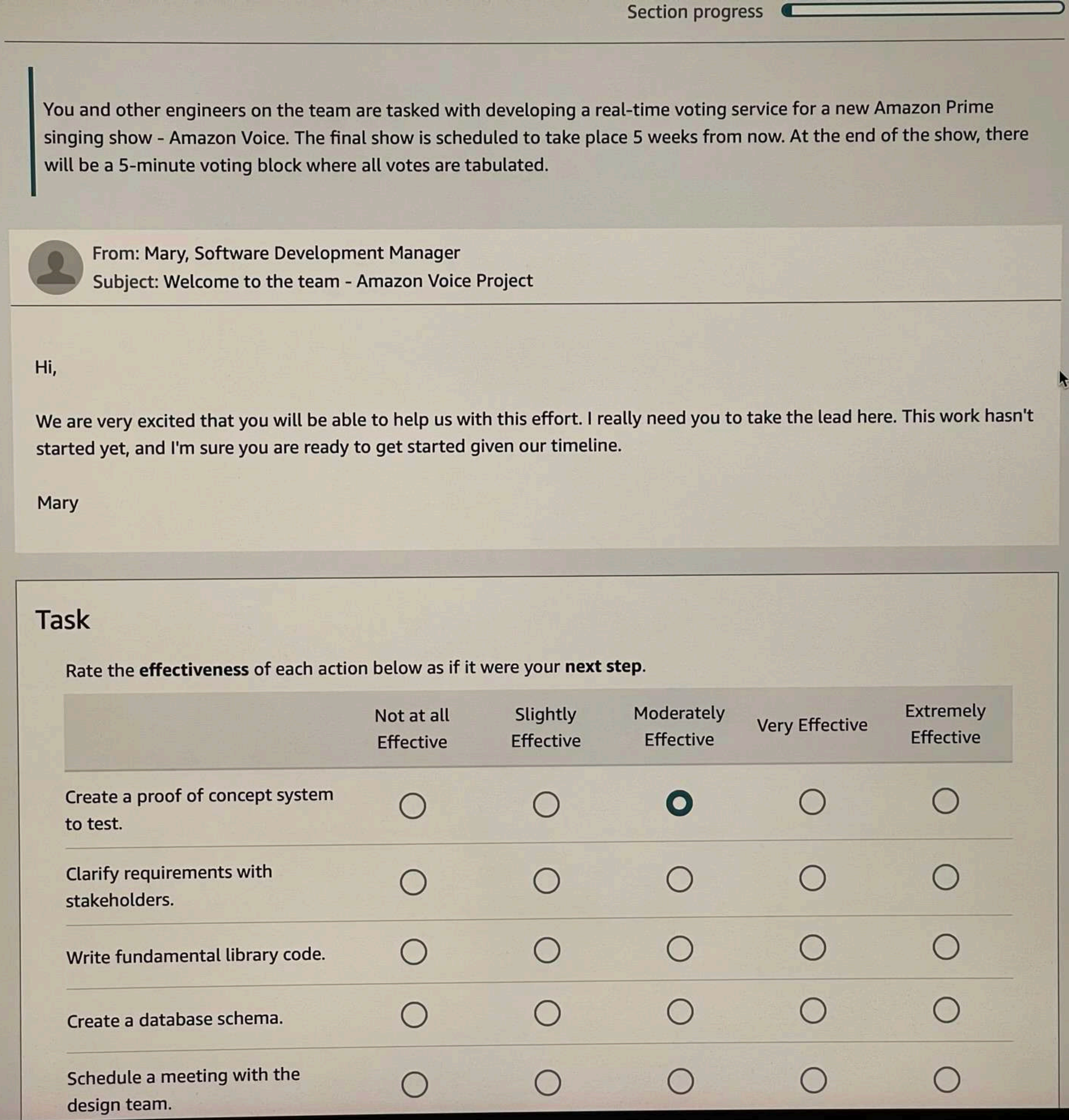


The maximum number of products is 12.

Function Description

Complete the function *findMaxProducts* in the editor below.

findMaxProducts has the following



Section progress

> You and other engineers on the team are tasked with developing a real-time voting service for a new Amazon Prime singing show - Amazon Voice. The final show is scheduled to take place 5 weeks from now. At the end of the show, there will be a 5-minute voting block where all votes are tabulated.

From: Mary, Software Development Manager
Subject: Welcome to the team - Amazon Voice Project

Hi,

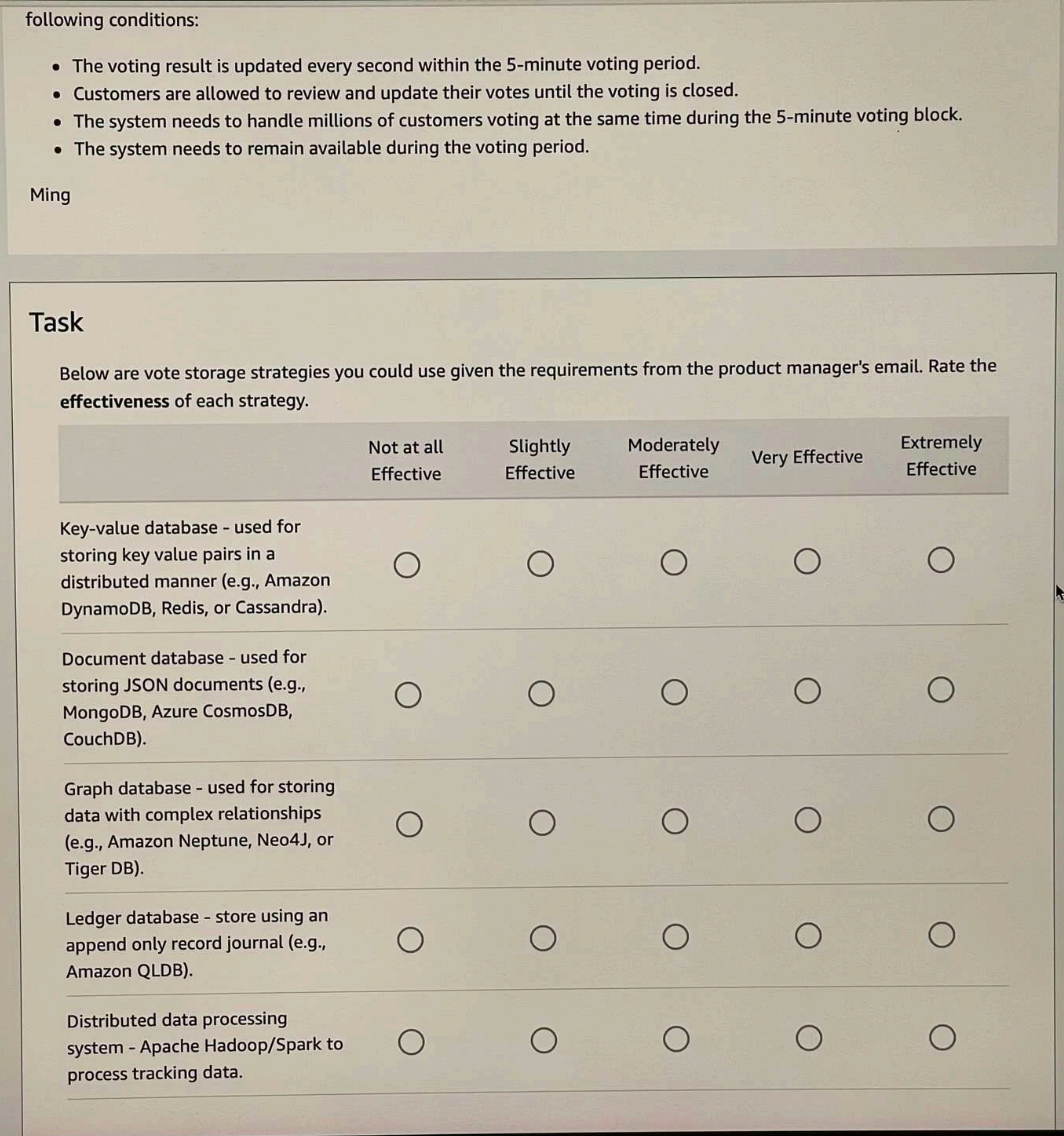
We are very excited that you will be able to help us with this effort. I really need you to take the lead here. This work hasn't started yet, and I'm sure you are ready to get started given our timeline.

Mary

Task

Rate the **effectiveness** of each action below as if it were your **next step**.

	Not at all Effective	Slightly Effective	Moderately Effective	Very Effective	Extremely Effective
Create a proof of concept system to test.	○	○	●	○	○
Clarify requirements with stakeholders.	○	○	○	○	○
Write fundamental library code.	○	○	○	○	○
Create a database schema.	○	○	○	○	○
Schedule a meeting with the design team.	○	○	○	○	○



following conditions:

- The voting result is updated every second within the 5-minute voting period.
- Customers are allowed to review and update their votes until the voting is closed.
- The system needs to handle millions of customers voting at the same time during the 5-minute voting block.
- The system needs to remain available during the voting period.

Ming

Task

Below are vote storage strategies you could use given the requirements from the product manager's email. Rate the **effectiveness** of each strategy.

	Not at all Effective	Slightly Effective	Moderately Effective	Very Effective	Extremely Effective
Key-value database - used for storing key value pairs in a distributed manner (e.g., Amazon DynamoDB, Redis, or Cassandra).	○	○	○	○	○
Document database - used for storing JSON documents (e.g., MongoDB, Azure CosmosDB, CouchDB).	○	○	○	○	○
Graph database - used for storing data with complex relationships (e.g., Amazon Neptune, Neo4J, or Tiger DB).	○	○	○	○	○
Ledger database - store using an append only record journal (e.g., Amazon QLDB).	○	○	○	○	○
Distributed data processing system - Apache Hadoop/Spark to process tracking data.	○	○	○	○	○

> Your team is building a new inventory management system that is going to be offered as a SaaS product (Software-as-a-Service).

From: Stansilov, Tech Lead
Subject: System Architecture

Hi,

The product owner has finalized requirements for the inventory management system. We need to architect a system to support these requirements. I'm hoping you can help with this effort.

Stansilov

Task

Rate the **effectiveness** of each action below as if it were your **next step**.

	Not at all Effective	Slightly Effective	Moderately Effective	Very Effective	Extremely Effective
Set up time to discuss the requirements with the product owner.	○	○	○	○	○
Sketch together a high level diagram.	○	○	○	○	○
Set up an operational dashboard of metrics, alarms, and documentation.	○	○	○	○	○
Develop a proof of concept with your team and senior engineers to test the key system constructs.	○	○	○	○	○
Prepare and host a system-level architecture review.	○	○	○	○	○

Kate, you

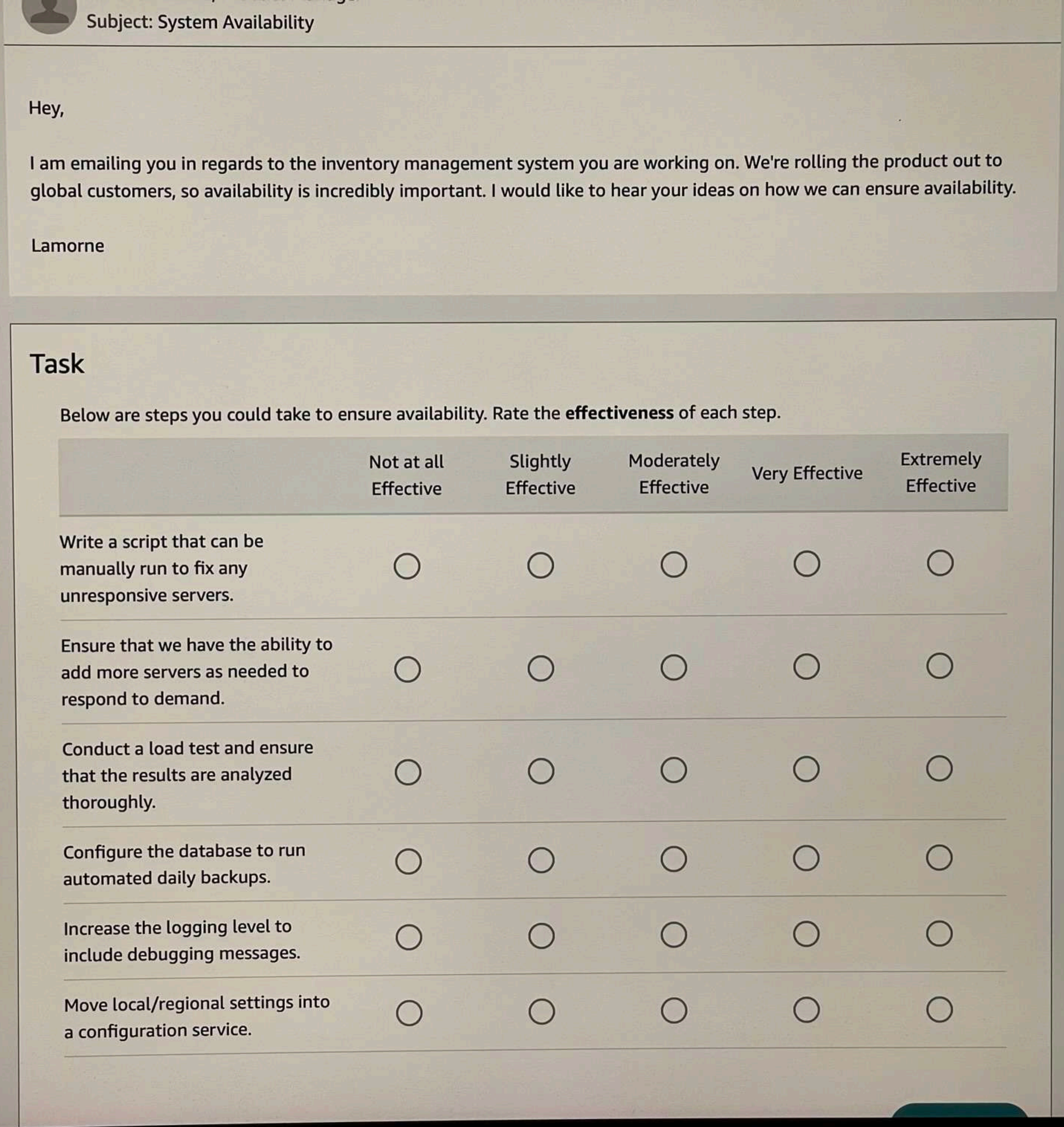
Kate (Software Development Engineer)

Hey - I've been asked to develop a solution to provide image thumbnails of inventory items. I've identified a few potential storage options, but I was hoping to hear your thoughts on each.

Task

Below are image storage options you could use. Rate the **effectiveness** of each option.

	Not at all Effective	Slightly Effective	Moderately Effective	Very Effective	Extremely Effective
Flat/file store.	○	○	○	○	○
Relational database.	○	○	○	○	○
Key-value data store.	○	○	○	○	○
Elastic search.	○	○	○	○	○
Cloud file store.	○	○	○	○	○



Subject: System Availability

Hey,

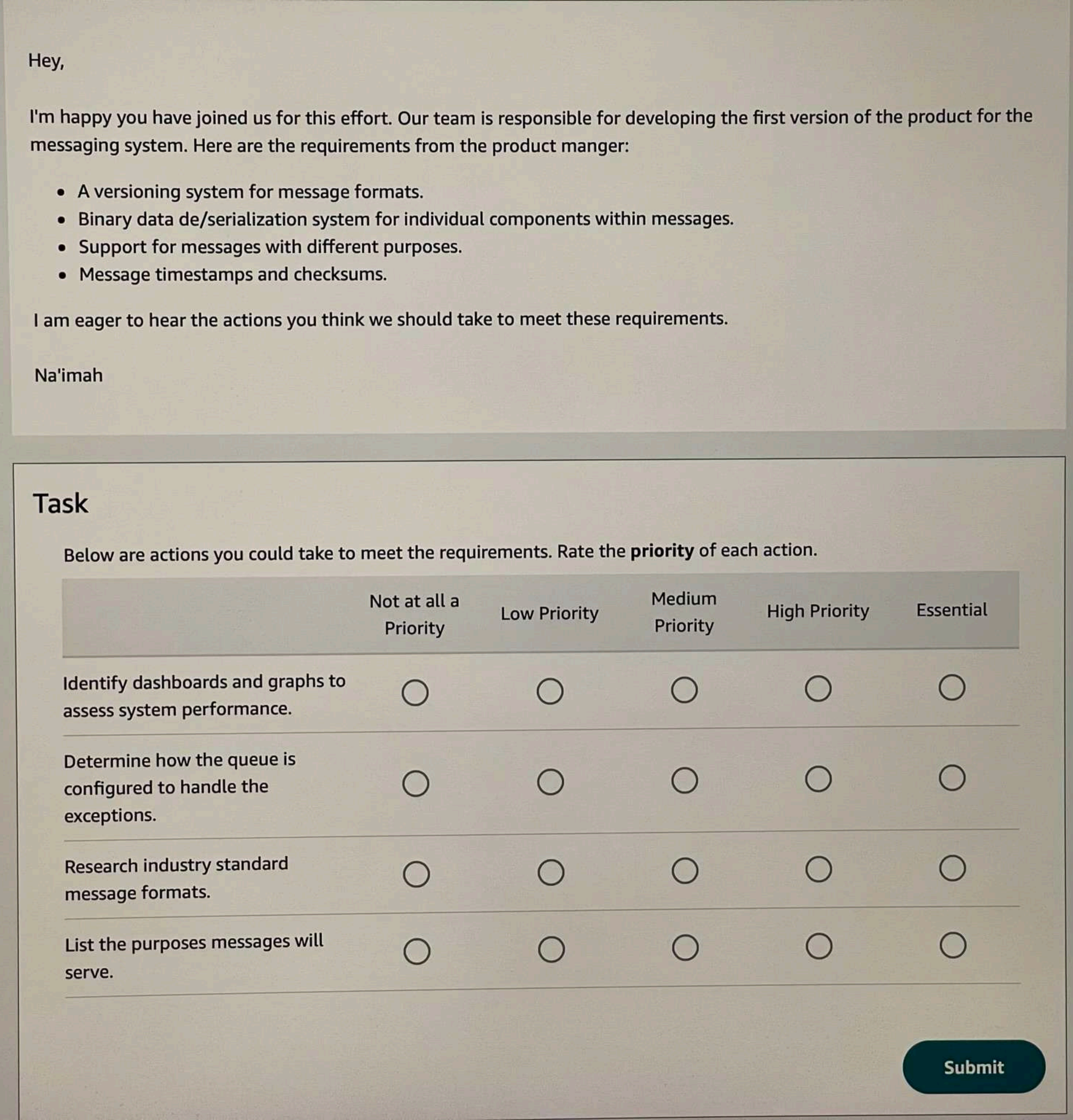
I am emailing you in regards to the inventory management system you are working on. We're rolling the product out to global customers, so availability is incredibly important. I would like to hear your ideas on how we can ensure availability.

Lamorne

Task

Below are steps you could take to ensure availability. Rate the **effectiveness** of each step.

	Not at all Effective	Slightly Effective	Moderately Effective	Very Effective	Extremely Effective
Write a script that can be manually run to fix any unresponsive servers.	○	○	○	○	○
Ensure that we have the ability to add more servers as needed to respond to demand.	○	○	○	○	○
Conduct a load test and ensure that the results are analyzed thoroughly.	○	○	○	○	○
Configure the database to run automated daily backups.	○	○	○	○	○
Increase the logging level to include debugging messages.	○	○	○	○	○
Move local/regional settings into a configuration service.	○	○	○	○	○



Hey,

I'm happy you have joined us for this effort. Our team is responsible for developing the first version of the product for the messaging system. Here are the requirements from the product manger:

- A versioning system for message formats.
- Binary data de/serialization system for individual components within messages.
- Support for messages with different purposes.
- Message timestamps and checksums.

I am eager to hear the actions you think we should take to meet these requirements.

Na'imah

Task

Below are actions you could take to meet the requirements. Rate the **priority** of each action.

	Not at all a Priority	Low Priority	Medium Priority	High Priority	Essential
Identify dashboards and graphs to assess system performance.	○	○	○	○	○
Determine how the queue is configured to handle the exceptions.	○	○	○	○	○
Research industry standard message formats.	○	○	○	○	○
List the purposes messages will serve.	○	○	○	○	○

queuing is planned, so it's necessary to design and support an appropriate message format.

Anne, you

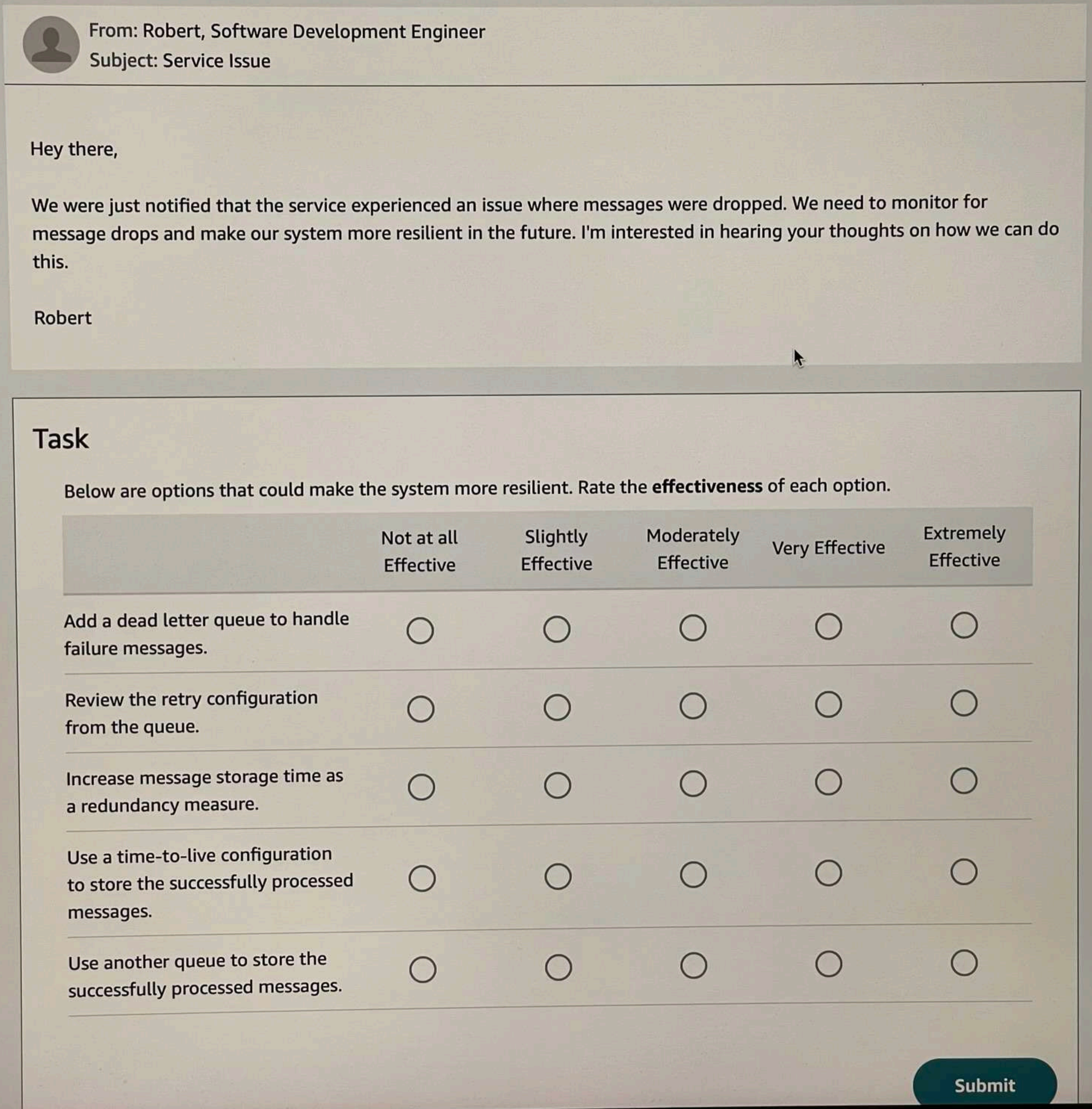
Anne (Principal Software Development Engineer)

Hey - some of the data these traffic cameras produce can be quite large, and the customer wants to transmit these messages to the service. I have a few ideas on how we could do this, but I'm interested in hearing your thoughts.

Task

Below are approaches you could take to transmit large messages to the service. Rate the **effectiveness** of each approach.

	Not at all Effective	Slightly Effective	Moderately Effective	Very Effective	Extremely Effective
Use different message types to transmit the request metadata and large data.	○	○	○	○	○
Fragment the large data into multiple messages.	○	○	○	○	○
Develop a streaming protocol for large messages.	○	○	○	○	○
Develop a cadence to send large messages during low volume time.	○	○	○	○	○
Physically download large messages from each location.	○	○	○	○	○



From: Robert, Software Development Engineer
Subject: Service Issue

Hey there,

We were just notified that the service experienced an issue where messages were dropped. We need to monitor for message drops and make our system more resilient in the future. I'm interested in hearing your thoughts on how we can do this.

Robert

Task

Below are options that could make the system more resilient. Rate the **effectiveness** of each option.

	Not at all Effective	Slightly Effective	Moderately Effective	Very Effective	Extremely Effective
Add a dead letter queue to handle failure messages.	○	○	○	○	○
Review the retry configuration from the queue.	○	○	○	○	○
Increase message storage time as a redundancy measure.	○	○	○	○	○
Use a time-to-live configuration to store the successfully processed messages.	○	○	○	○	○
Use another queue to store the successfully processed messages.	○	○	○	○	○