Assessment Questions:

1. What are the questions you would ask the Product Manager on this project to better understand how to architect the system?

Ans: As the requirement clearly specifies the end goal of the product, I will be asking the below questions to get more clarity before starting system design.

1. What type of notifications we are targeting like SMS, email & push notifications?
2. Are we targeting for real time system like we need to deliver on every past immediately or there can be any delay in delivery like 1hr or daily basis?
3. AS different devices need to handle differently like security layers. What are the devices we are targeting?
4. Are we providing opt-out like unsubscribe for tags & sub-tags?
5. Lot of notifications a day will annoy the user & may think of un-subscription, is there any number of counts can be delivered per day?
6. AS there are lot of articles delivery for any tag, sub tags & even addition of a new sub tag, are we coming up with the approach upscaling & down scaling our backend services?
7. Is there chance of any article can have 2 tags at a time?
8. Are we allowing users to select notification type he can receive?

Rest based on the above questions answers, will arise & will get clarity.

2. What are some of the things you will consider when designing this system from a technical perspective?

Ans: As this is subscription-based notification system, I will consider following things:

Frontend:

UI need to be simpler with attractive layout & faster rendering using SSR libraries like NextJS, with Redux & etc.

Backend:

I am assuming above asked questions are handling in our product, with this I am considering the below technical perspectives.

As news articles can be huge, so we need to think of managing the notifications in queuing mechanism like handling in batches.

Will store user preferences of notification type & other details in non-SQL DB like Cassandra or Mongo DB, because here we don’t have complex relations to use RDBMS, this gives faster while data querying from DB & other No-SQL advantages. Will build micro services accordingly.

As notifications may increase or decrease any time, so we need to do auto upscale or downscale like AWS or firebase of our services & this will be applicable for all critical services like publishers, service worker, etc.

As we are dealing with same set of articles with different users. So, we can use caching mechanisms like Redis or elastic search, which can save lot of bandwidth of querying with main DB & improves performance.

As we need to send same kind of template for all notifications, we can store/cache the template.

As users may feel bad if we send hundreds of notifications through push notifications, so we need to have rate limiter mechanism, which will check previous notifications count & user metrics/analytics based on this can send more or less for a user & even we can consider event driven/firing approach.

Sometimes there may be changes for failure of delivering notifications due to multiple factors, so there must me mechanism to retry the failed notifications.

We need to send notifications with all security practices, as different devices/ type of notifications has different security principles.

We need to monitor user’s actions on received notifications like he is reading the notifications or not by analytics, based on this we can build more user preferred actions or articles.

We need to store better logging metrics for failed mechanisms to better understand & for auditing purpose.

With above things in mind, I had prepared low level system design diagram & attached separately.