| **Name:** | Lin Mei An |
| --- | --- |
| **Email:** | lin\_mei\_an@mpa.gov.sg |

**Question 1**

1. Name a (or more) cryptographic algorithm you would use to perform the following
   1. encryption
      1. AES
      2. RSA
   2. non repudiation
      1. MD5
      2. SHA256
      3. SHA1

(hash and then sign with digital signature)

* 1. no tempering

same as non-repudiation

1. Name the following JWT registered claim names (see <https://tools.ietf.org/html/rfc7519#page-9>)
   1. unique JWT identifier

"jti" (JWT ID) Claim

* 1. cannot be used before a certain date

"nbf" (Not Before) Claim

* 1. issue date

"iat" (Issued At) Claim

* 1. token recipient

"aud" (Audience) Claim

* 1. mobile number

-proprietary information hence it goes into the payload

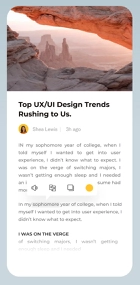
**Question 2**

You are developing a hotel reservation application. After your user have successfully booked a hotel, the application can (opt in) update a user’s Google calendar with the stay’s detail and alerts. The application needs to create, update and delete calendar entries.

What are the required steps to allows the reservation system to update the a user’s calendar?

* enable the calendar api from the google api library
* create credentials using OAUTH or using API key generated by google cloud

**Question 3**

You and a few friends have co-founded a hot social media startup. Like any good social media startups, you will need a new feed. A news feed is a list of post that is constantly updated with stories, activities, polls, etc from your friends. A post content includes the poster, text, images, videos, simple questionnaires, links, locations, etc. 

The post will also include likes, the number of people reacted to it.

Two REST endpoints have been designated for users to publish and retrieve their feeds.

Publish a post

POST /api/v1/feed/me

Retrieving a feed

GET /api/v1/feed/me

The endpoints are secured with JWT.

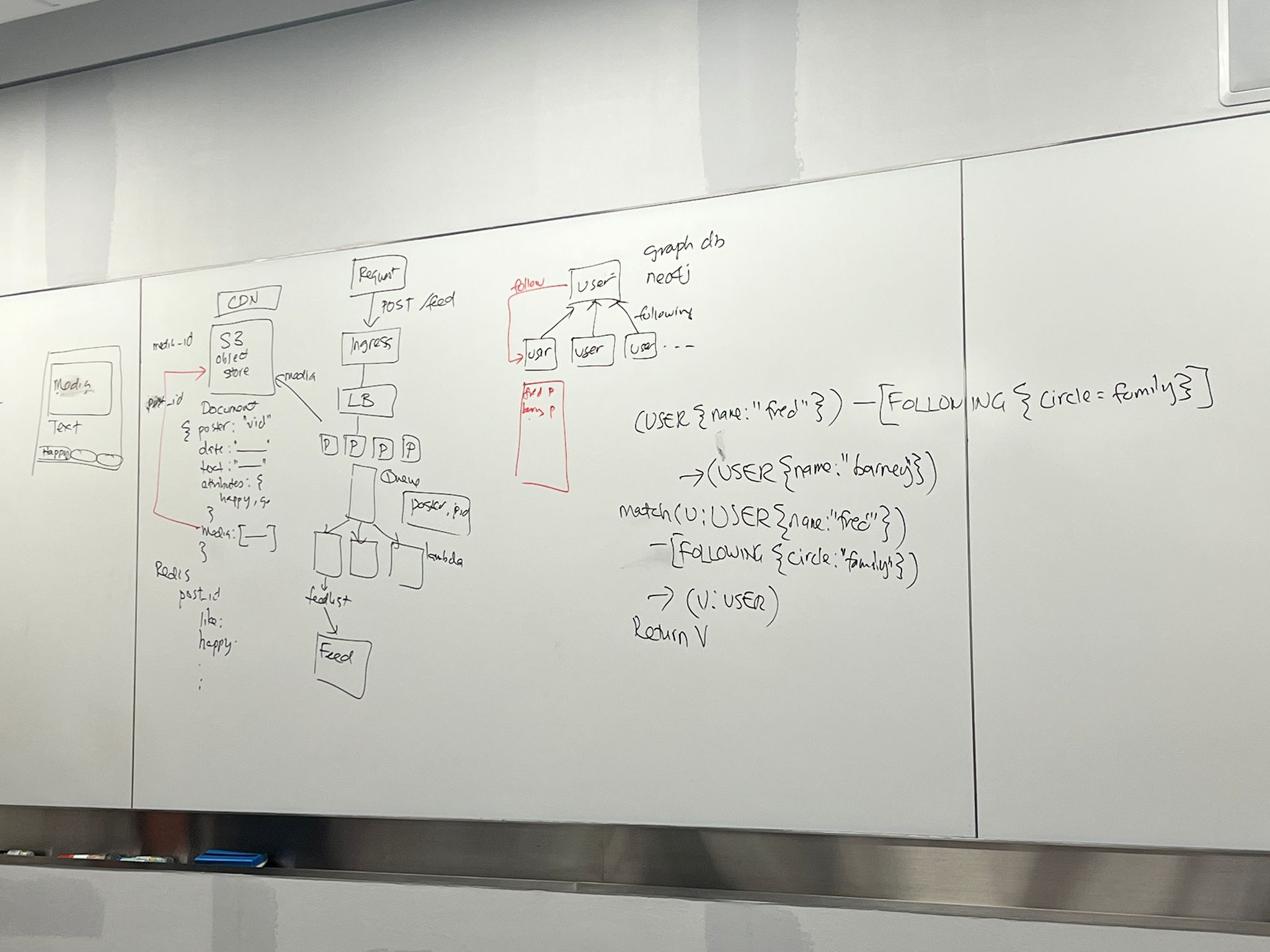
Each user of your social site can have up to a total of 1000 friends/followers. You anticipate 5 million daily active users with about 70% of them posting at least 1 post.

Design a system that will support your REST endpoints along with the given requirements. Be as detailed as possible with your design.

GET /api/v1/feed/me

POST /api/v1/feed/me

-due to heavy traffic, easier to split into 2 different applications

****

POST:

request -> ingress -> load balancer -> cluster of microservices dedicated to posting

* the post is broken up into the constituents e.g. media, text, others and stored on different sites tagged with uid and a reference to the media (and maybe other attributes e.g. date and location)
* likes/comments/etc that are more dynamic as compared to the actual post are stored separately from the content of the post
  + common methods to store: key value pair e.g. Redis
* post id needs to be pushed to friends’ feeds for them to receive it on their feed
  + however, pushing actively may waste resources and take alot of time if there are too many followers
* hence, the pull method can be used instead
  + may be slower since the user needs to actively look up all the following and see if they have any new posts to retrieve the new posts
* (user, post) is queued

too many relationships cannot use relational database because too many joins, better to use graphdb e.g. neo4j

**Submission**

Copy this Word document to your repository and commit it.

git add .

git commit -m ‘worksheet03’

git push origin master