| **Name:** | Wong Ru Peng |
| --- | --- |
| **Email:** | wong\_ru\_peng@mpa.gov.sg |

**Question 1**

1. Name a (or more) cryptographic algorithm you would use to perform the following
   1. encryption

**AES, RSA**

* 1. non repudiation

**Hashing - SHA1, SHA512, MD5**

* 1. no tempering

**MD5**

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

**jti**

* 1. cannot be used before a certain date

**nbf**

* 1. issue date

**iat**

* 1. token recipient

**aud**

* 1. mobile number

**Private Claim Names**

**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?

**Create project in GCP -> Select API and services -> Enable API Services -> look for the Google Calendar API and enable it -> Create credentials -> Create API Key**

**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 details a possible with your design.

**1) Have 1 system to handle all POST request and another to handle all GET request. Each system have a LB to distribute the load.**

**2) Store the medias in a S3 store that is behind a CDN**

**3) Store the document in a separate server which has a media ID reference the media in the S3 store**

**4) Store the “likes” and comments in redis**

**Submission**

Copy this Word document to your repository and commit it.

git add .

git commit -m ‘worksheet03’

git push origin master