

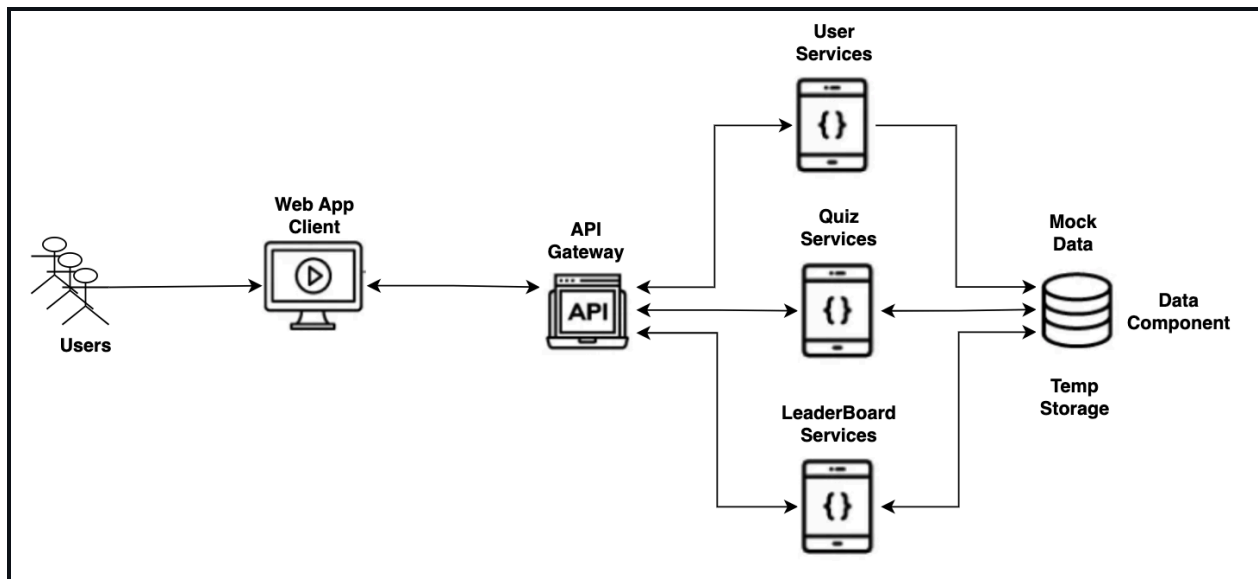
Real-Time Vocabulary Quiz Coding Challenge

Challenge Requirements

Part 1: System Design

1. System Design Document:

Architecture Diagram



Component Description, Data Flow and Technologies and Tools

Users:

- Users do Quiz by the web browser to access the address of Quiz.

1. Web Application Client:

- Description: This web application will present the UI/UX interacting with users through features provided. This practice includes User, Quiz and LeaderBoard Services.
- Tech and Tools: Propose ReactJS/NodeJS to develop the Quiz's web.
- Dataflow: The web application shows data that were obtained from services or inputted from the user's actions. The data is sent to and got back from the service apis through the API Gateway accordingly.

2. API Gateway

- Description: The API Gateway plays an role of proxy doing the both routing and controlling the flow of requests to the Service Endpoints.
- Tech and Tools: Propose Nginx to serve this function.
- Dataflow: Get requests from web application routing to services and on the other hand, get responses to send back to the request owner.

3. User Services

- Description: User Service supports the features of User such as Authen, Sign up, Sign in and Sign out.
- Tech and Tools: Propose Go/NodeJS-TypeScript Language to develop the business services.
- Dataflow:
 - Receive request from web application, processing on context and send response result.
 - Based on the demand of request, these services can connect to the other services such as the 3rd parties or Database.

4. Quiz Services

- Description: Quiz Service supports the features of Quiz such as loading questions, answer keys, checking answers and saving User's result of Quiz.
- Tech and Tools: Propose Go/NodeJS-TypeScript Language to develop the business services.
- Dataflow:
 - Receive request from web application, processing on context and send response result.
 - Based on the demand of request, these services can connect to the other services such as the 3rd parties or Database.

5. LeaderBoard Services

- Description: LeaderBoard Service supports the real-time statistics such as leader board of quiz, user's results.
- Tech and Tools: Propose Go/NodeJS-TypeScript Language to develop the business services for back-end layers.
- Dataflow:
 - Receive request from web application, processing on context and send response result.
 - Based on the demand of request, this module was connected to the Quiz and User services to update the reporting data.

6. Database:

- Description: where stores data for applications such as User, Question, Answer Keys, User's Quiz, ...
- Tech and Tools: Propose Postgres DB to store data, going with caching DB such as Redis DB to get higher performance.
- Dataflow.
 - Receive CRUD request, process and send responses back to services.
 - Store data and sync data to the other system on demand.
 - The real-time statistical data will be updated from User and Quiz Services accordingly based on events of the user's actions.

Part 2: Implementation

Pick a Component:

Requirements for the Implemented Component:

In this challenge, I will use the Go/NodeJS-TypeScript Language play a role of full stack to develop the application from Front-End, Back-End to Data Component (Postgres)

Following the application architecture of microservices, I will develop two independent applications, one for Web Client to present UI and Data, the other is the business services including User, Quiz and Leaderboard (reporting) services.

Steps to operate Real-time Vocabulary Challenge:

1. Browse to <http://localhost:3001/> to run App

The challenging Quizlet

Home Quizlet Settings						
Ranking						
#	FirstName	LastName	Email	Score	Status	Date
1	Lam Ngan	Tan	komodo1080@gmail.com	2	Completed	2024-07-13 20:54:38
2	2	2	2@gmail.com	0	Uncompleted	2024-07-13 20:57:32
3	1	1	111@gmail.com	0	Uncompleted	2024-07-13 20:57:29

2. Create a Quiz
 - a. Navigate to Tab “Setting” to create a quiz (exp: Realtime Vocabulary Challenging)

The challenging Quizlet

Home	Quizlet	Settings
------	---------	----------

Create Quiz

Title

Score

- b. Create question for a quiz

Quiz type

Realtime Vocabulary Challenge

Question

Lighthouse

Score

2

Options

Answer

Bánh ứt

☐ Is correct?

Add answer

#	Answer	Is correct	Action
1	Hải đăng	Yes	
2	Bánh bèo	No	

Submit

- c. Enroll and try a Quiz
 - i. Navigate to Tab “Quizlet”
 - 1. Register to start a quiz

The challenging Quizlet

Home Quizlet Settings

Input your information to start the quiz

Quiz type

Realtime Vocabulary Challenge

First Name

Lâm Ngâ

Last Name

Tân

Email

komodo1080@gmail.com

Start quiz

- 2. Do Quiz

The challenging Quizlet

[Home](#) [Quizlet](#) [Settings](#)

Challenging yourself

Lemon

- ☒ Cam
- ☐ Quyt
- ☐ Chanh

Chicken

- ☐ Cho
- ☐ Meo
- ☐ Vit
- ☒ Ga

Bottle

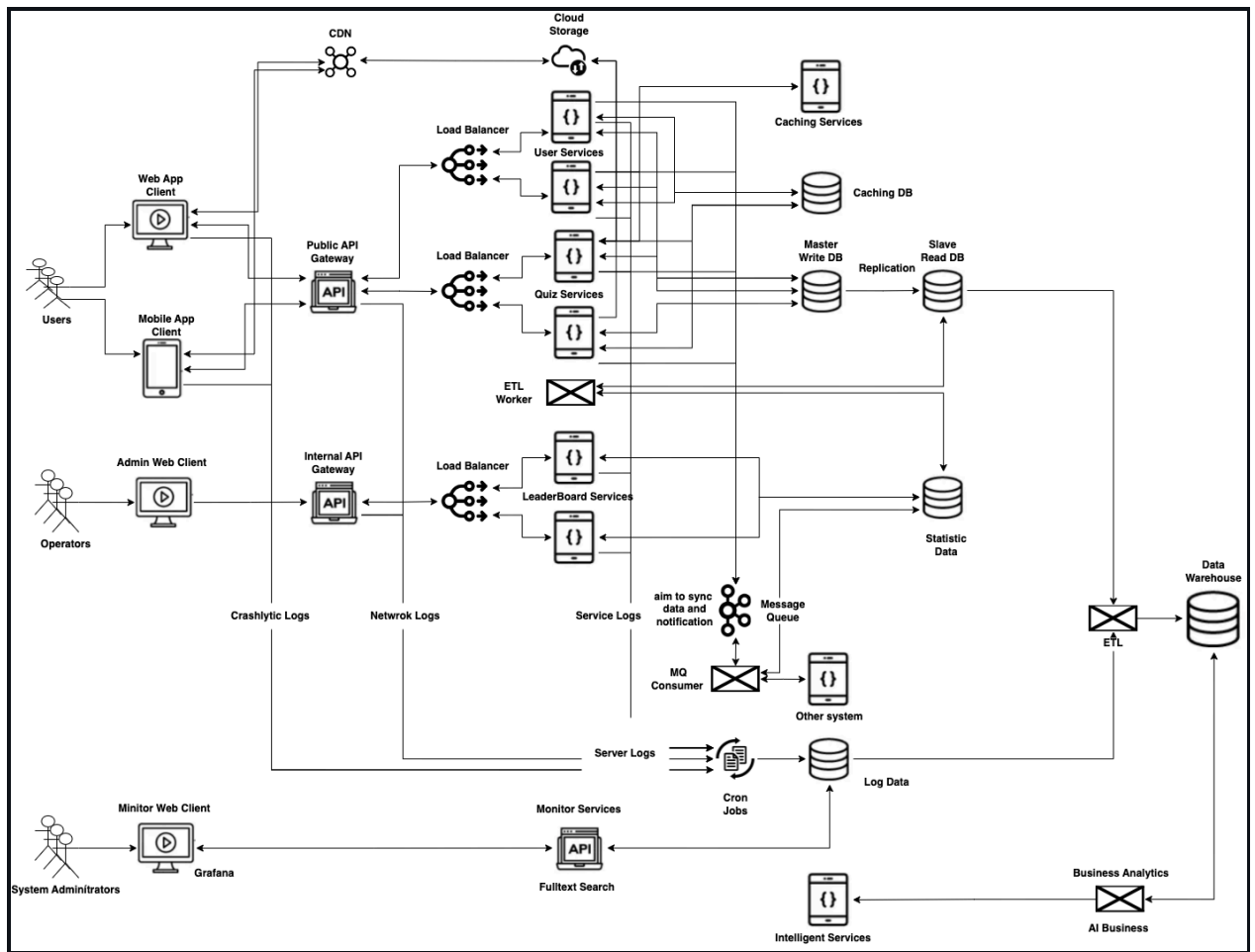
- ☐ Ghe
- ☐ Ban
- ☒ Tu
- ☐ Chai

3. Candidates and Results will be updated real-time at leaderboards.

Build For the Future:

System Design Document:

Architecture Diagram for a large traffic adapting to expansion, scale system up and down rapidly that not impact the whole system.



Component Description, Technologies, Tools and Data Flow.

Layer 1: Client Application

1. Quiz Web Application

Description: The program that users use to experience the Quiz Services through the web browsers.

Technologies and Tools: NodeJS/ReactJS/Typescript and Flutter/React Native for mobile platforms, embedded Firebase SDK for Notification, Crashlytics, ... to measure, monitor and maintain applications.

DataFlow:

- Data Sources:
 - UI: from CDN or hosting services.
 - Data: from Services or User's actions
- Connected to the services through public API Gateway.
- Present the UI/UX interacting with users through features provided

2. Admin Web Application

Description: The program that users operate the Quiz Services through the web browsers. Publish in an internal (office) network or VPN (if from the internet) to secure it.

Technologies and Tools: NodeJS/ReactJS/Typescript, ... embedded Firebase SDK for Notification, Crashlytics, ... to measure, monitor and maintain applications.

DataFlow:

- Data Sources:
 - UI: from CDN or hosting services.
 - Data: from Services or User's actions
- Connected to the services through the internal API Gateway.
- Present the UI/UX interacting with users through features provided.

3. Monitor Web Application

Description: The program for system administrators to monitor and maintain the whole Quiz system through the web browsers. Publish in an internal (office) network or VPN (if from the internet) to secure it.

Technologies and Tools: Use Grafana, Splunk to visualize the status of system services.

DataFlow:

- Data Sources:
 - Logged from Services, servers and 3rd parties.
- Connected to the services through the private/internal services.

Layer 2: API Gateway

1. Public API Gateway

- Description: Used to publish services for application in the Internet environment.
- Technologies and Tools: Use NginX for routing, load balance of services.
- DataFlow: In from Public application and out to internal Services or vice versa

2. Internal / Private API Gateway

- Description: Used to publish services for application in the Office (internal) environment.
- Technologies and Tools: Use NginX for routing, load balance of services.
- DataFlow: In from admin application and out to internal Services or vice versa

Layer 3: Load Balancer and Business Services

1. Load Balancer

- Description: In case of a large system with high traffic, we would have much more than 1 node to adapt their performance, and Load Balancer is a tool that helps to check health, workload, ... to coordinate requests between nodes.
- Technologies and Tools: NginX

2. Business Service (User / Quiz / Leaderboard)

- Description: provide their service with functions defined in their scope User, Quiz, .. accordingly.
- Technologies and Tools:
 - Kubernetes (hosting service),
 - Git and Jenkins (CI/CD) for deployment
 - Go/Node/Python/Flutter for development
 - S3 for CDN
- Dataflow:
 - Receive request from web application, processing on context and send response result.
 - Based on the demand of request, these services can connect to the other services such as the 3rd parties: CDN Services, Caching Services, MQ services, ... and Databases

3. ETL Workers

- Description: helps to
 - Centralize and transform the original data of services / logs to certain format data to serve for other business goals such as Analytics, AI Services Training, Segmentation or Personalization.
- Technologies and Tools: Framework with Apache Spark, Python, Go, Node or Java.
- Dataflow: come from:
 - Application/Services Data
 - Logs: Client Application, Services, Database, ...
 - Store them into Data warehouse, Statistic DB, ... on demand.

4. MQ Services

- Description: Plays a role of Bridge that helps to sync data for the whole system or do jobs that do not wait for response to the request's owners such as sending notification. It's useful in a large system with millions of requests.
- Technologies and Tools: Kafka / RabbitMQ / AWS SQS.
- Dataflow: with methods of publisher/consumers allowing one service to share their information to others by events.

5. Other Services

- CDN Service: with a particular mechanism it supports high performance in storing and reading to the static data such as media files, ... on nodes of locations that we have set up before.
 - Technology and Tool: AWS S3.
- Caching Service: Supports to cache data, static pages without reading Database, reduce workload of Back-End (business services), Databases.
 - Technology and Tool: Memcached, AWS CloudFront.

Layer 4: Database System

1. Application DB (Master)

- Description: Serve to store, read master, application data from services.

- Technologies and Tools: Postgres
- DataFlow:
 - Data in/out from business services.
 - Replicate data to slave DB

2. Read DB (Slave DB and Statistic DB)

- Description: Serve to read only data.
- Technologies and Tools: Postgres and Replication setup
- DataFlow:
 - Connect to Master DB to Sync Data.
 - Support for ETL Workers to transform data.

3. Logging DB

- Description: Where store logs from network, client application/services and servers to serve for monitoring and maintenance systems or tracing issues.
- Technologies and Tools: Nginx, Grafana, Splunk, Kibana, Elastic Search.
- DataFlow: Logs from many sources will be centralized in ES DB

4. Data Warehouse

- Description: store the whole valuable data from service, application and user's behaviors. It serves many goals in business needs.
- Technologies and Tools: MongoDB, Postgres, depend on demand.
- DataFlow: come from DBs or ETL services.