All Unique System Architecture Questions

Question 1: Enhanced Personalized Restaurant Recommendation Engine

Background:

Previously in your Assignment 1, you developed a Dining Concierge chatbot that provided restaurant suggestions based on user input for cuisines. However, this system did not offer personalized recommendations based on user feedback, nor did it dynamically adapt to external changes like restaurant availability or new openings.

Objective:

The goal is to develop a more intelligent and adaptive Dining Concierge system. This enhanced system should:

1. Offer personalized restaurant recommendations based on user feedback and interaction.
2. Offer recommendations based on geographic proximity.
3. Seamlessly integrate and respond to external data changes, such as updates in restaurant statuses or the introduction of new dining establishments. (assume any data source and mention it.)

When users request recommendations, they should receive 5 recommendations based on personal preferences, user's past searches, feedback etc and another 5 based on the trending restaurants around them.

You could assume that the application has an enhanced user screen that has a "like" button next to the name of a restaurant that is recommended to the user. You could

collect these "like" as input to your trending, recommendation engine.

Requirements:

1.**Data Stores**

- Clearly list and describe all data stores involved and why they are a good choice.
- Define the type of data stored (use schema/ERD etc), including any indexing mechanisms, and briefly explain why this is a good design for your data store.

2.**APIs**

- List all new APIs that will be integrated into the system.
- Describe the low-level design for your backend briefly, focusing on AWS services and infrastructure suitable for a high-traffic system.
- Ensure that the design is scalable, event-driven, and asynchronous, capable of handling the demands of a large user base.

3.**System Design Architecture (High-Level Backend Design)**

- Develop an architecture diagram showing the integration of the personalized recommendation engine and dynamic data system with the existing chatbot.
- This architecture should support extensive data from user interactions and be capable of adjusting recommendations dynamically.

4.**Feedback Loop, Real-Time Processing, and Adaptability Parts Working/ Explanation**

- Explain the feedback loop for capturing and integrating user preferences and feedback.
- Describe how the system will handle and adapt to real-time data, including user feedback and restaurant data changes.
- Identify AWS components and services used for real-time data management.

5.**Data Pipeline / Event Flow**

- Detail the data pipeline (or event flow) starting from user interaction to the delivery of personalized recommendations.

This system aims to transform the Dining Concierge chatbot into a more dynamic, and user-centric service. The upgraded system should demonstrate robustness, efficiency,

and the ability to adapt to the changes in the restaurant data out there and consumer preferences, leveraging AWS services effectively.

Feel free to make additional assumptions but remember to mention them.

Question 2: Restaurant Recommendation Engine with Current Wait-time and Special Offers

Background:

Previously in your Assignment 1, you developed a Dining Concierge chatbot that provided restaurant suggestions based on user input for cuisines. However, this system did not offer personalized recommendations based on user feedback, nor did it dynamically adapt to external changes like restaurant availability or new openings.

Objective:

The goal is to develop a more intelligent and adaptive Dining Concierge system. This enhanced system should:

1. Offer personalized restaurant recommendations based on user feedback and interaction with current wait-time in the respective restaurant.
2. Offer recommendations based on geographic proximity.
3. Seamlessly integrate and respond to external data changes, such as updates in restaurant statuses or the introduction of new dining establishments. (assume any data source and mention it.)

When users request recommendations, they should receive 5 recommendations based on personal preferences, user's past searches, feedback etc and another 5 based on the trending restaurants around them. These recommendations should include the current wait-time and if there is any special offer from the restaurant.

For wait-time, you assume that restaurants have an API that when you query, will provide you the current wait-time estimate. For the special offers, you assume that there is an API the restaurant provides what is the special offer for the next day if any. The offer expires the next day.

You could assume that the application has an enhanced user screen that has a "like" button next to the name of a restaurant that is recommended to the user. You could collect these "like" as input to your trending, recommendation engine.

Requirements:

1.**Data Stores**

- Clearly list and describe all data stores involved and why they are a good choice.
- Define the type of data stored (use schema/ERD etc), including any indexing mechanisms, and briefly explain why this is a good design for your data store.

2.**APIs**

- List all new APIs that will be integrated into the system.
- Describe the low-level design for your backend briefly, focusing on AWS services and infrastructure suitable for a high-traffic system.
- Ensure that the design is scalable, event-driven, and asynchronous, capable of handling the demands of a large user base.

3.**System Design Architecture (High-Level Backend Design)**

- Develop an architecture diagram showing the integration of the personalized recommendation engine and dynamic data system with the existing chatbot.
- This architecture should support extensive data from user interactions and be capable of adjusting recommendations dynamically with the wait-time and offers information.

4.**Feedback Loop, Real-Time Processing, and Adaptability Parts Working/ Explanation**

- Explain the feedback loop for capturing and integrating user preferences and feedback.
- Explain how you will handle the current wait-time and special offers APIs and how you would use these in a scalable way.
- Describe how the system will handle and adapt to real-time data, including user feedback and restaurant data changes.
- Identify AWS components and services used for real-time data management.

5.**Data Pipeline / Event Flow**

- Detail the data pipeline (or event flow) starting from user interaction to the delivery of personalized recommendations.

This system aims to transform the Dining Concierge chatbot into a more dynamic, and user-centric service. The upgraded system should demonstrate robustness, efficiency, and the ability to adapt to the changes in the restaurant data out there and consumer preferences, leveraging AWS services effectively.

Feel free to make additional assumptions but remember to mention them.

Question 8: Real-time Restaurant Recommendation System

Background:

Extend Assignment 1 to contain the features for:

- Calculate ratings for restaurants
- Trending restaurants in the user's area

Assume trending restaurants in the area can be obtained through an external API call that has already been provided to you. As user traffic increases, performance bottlenecks emerge. You must design a scalable, low-latency system that efficiently delivers recommendations while handling high traffic loads.

Requirements:

For both the above features, implement the following:

1.**APIs and Assumptions**

- State all APIs created and assumptions made.

2.**User Location Detection**

- Devise a way to figure out the user's location

3.**System Architecture**

- Design an architecture diagram for your backend system with all services.

- You need not draw the architecture provided in the assignment, just mention the connecting services properly.

4. **Database Schema Design**

- Design the schema for any Databases used for this.

5. **Scalability Strategy**

- How would you scale this system to handle millions of users while maintaining a low-latency experience?
