

Project Design Phase-I

Solution Architecture

Date	23 October 2023
Team ID	Team-591034
Project Name	SNACK SQUAD: A CUSTOMIZABLE SNACK
Maximum Marks	4 Marks

Solution Architecture:

Proposed Solution

Proposed below is a high-level reference architecture for Online Snack Delivery systems. This proposed architecture is generic and it can be deployed to any of cloud provider like AWS/GCP/Azure and can be build using Jetpack Compose

Assumptions

Delivery partners are provided with devices that have inbuilt GPS. This will help to locate their current location accurately.

Integration with map provider is there and we get details about routes, traffic and commute time

Solution Details

Core Features of Customer App

- Searching menu:** Allow your users to search for different restaurants, cafes by location, and cuisines. Using the search filter, users can easily find their favorite eating places, list menu, offers, etc.
- Order placement:** The user can place an order of selected dishes and food. They just need to cross-verify their preferred dish, delivery time, and proceed check-out.
- Tracking Delivery Partners:** With real-time tracking features, it becomes easy for users to track delivery drivers and know their real-time location information. Users can check the time taken by the food delivery executive to deliver their parcel.
- Payment gateway integration:** You provide the users with multiple payment options like credit/debit cards, different wallets like Google Pay, Paytm, Phonepe, UPI, etc

Core Features of Delivery Partner

- Delivery Partner's profile:** Through this feature, a driver can keep his profile update. It contains his full name, address, email, contact number, photo, and other personal information.
- Notification for orders:** Through push notifications, drivers can get constant updates & alerts for new orders. It will help in the accurate delivery service of your restaurant.
- Map for the delivery route:** Integrate Google Map or other providers and allow drivers to choose the shortest and fastest routes to reach the location.

Core Features of Food Partners/ Restaurants

- Restaurant Profile/Menu:** Through this feature, a restaurant owner can add their restaurant details, menu and its availability, price, preparation times, etc

- Notification for orders:** Through push notifications, Restaurants can get constant updates & alerts for new orders. It will help in the accurate delivery service of your restaurant.
- Notifications for Pickup Partners:** They will get alerts about delivery partners, their location when they will pick up, etc.
- Payment Details:** Information about the payment received from the food delivery system for their orders

Core Features for Food Ordering System Admin

- Restaurant management:** Being on the admin panel, one can directly manage all the restaurants by adding, updating, and removing any eating joint from the list. He can also check active restaurant status.
- Analytics & report generation:** Using analysis and report features, you can get real-time insights of reports and other accounting information which helps you to identify the growth and opportunities to expand reach.
- Monitoring every action:** Monitor all the food orders, delivery partners, deliveries, reviews & ratings of delivery partners, canceled orders, and other important data related to the driver's performance.

Application Flow

We are considering here microservices-based architecture. Different services are listed in the architecture diagram

- 1.All requests made from a mobile app or UI will go to different services via the API gateway. API gateway will take care of load balancing and routing requests to services. This will authenticate and authorize the user and send back the token ID. This token is used for further communication.
- 2.Different services like, user registration and management service, order service, payment service will use transactional databases. We will use the Amazon Aurora relational database. This is a highly scalable database service to manage users and concurrent orders etc.
- 3.Information about different restaurants, their menu, price, offers, etc will be stored in JSON document storage in Elasticsearch. We can use a multi-node cluster here. Whenever a customer searches for a menu/cuisines it will be fetched from elastic search. Elastic search provides fast scalable search options.
- 4.Once the user selects the dishes and quantity from the restaurant. He will go to the checkout option and then do payment. Different payment gateways and payment options are integration with the system and upon successful payments, the order is successfully placed.
- 5.Once the order is placed all the information is sent to the central message Queue like Kafka. The order processing unit reads the order info and then notifies the selected restaurant about the order. At the same time, it searches for available delivery partners to nearby locations to pick up the order. It also gets the information like preparation time from the restaurant and estimated pickup time from the delivery partner based on his location and other details. it will select the best available delivery partner and he is notified about order and restaurant details.

6. The user gets push notification about the order. The order processing and tracking service will work together and the user can track their order status, live location of the delivery person, etc
7. Delivery person pickup order and deliver to customers. Customer is real-time notified with ETA for the order

Easy onboarding and Searchability of Restaurant

From the Admin panel, the admin can add the restaurant. Admin adds details like restaurant name, city, address, postal code, cuisine type, operational hours, owner details, payment shares, etc. All this information is stored in a relational database. We use Amazon Aurora here.

Once the restaurant is added we will generate a Unique ID for the restaurant. This unique ID will be used in Elastic search to store information like different menus, their price, preparation time, etc.

Restaurants have access to add /update/delete menu, price, preparation time, etc

So when a customer searches food options by dish name, restaurant name, location then ElasticSearch is queried. Elasticsearch is a highly available, scalable open-source full-text search and analytics engine. With elastic search, you can store, analyze, search large volumes of data quickly and in near realtime.

When a customer opens the app, the first call is made to the Inventory/Menu system to figure out:

1. Nearby serviceable restaurants to customer locations. This can be done by customer location and restaurant location.
2. The restaurants that are actually serviceable to the user/customer, those restaurants which can deliver food within some stipulated time (say 45 minutes)
3. The expected delivery time for your food order from a potential restaurant.

Solution Architecture Diagram

