Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	21 October 2023
Team ID	PNT2022TMID590947
Project Name	Project – Express Eats
Maximum Marks	4 Marks

Technical Architecture:

The **food delivery app cost** is also affected by the technology stack that is used. Before developing the final app working on a prototype would be a great idea.

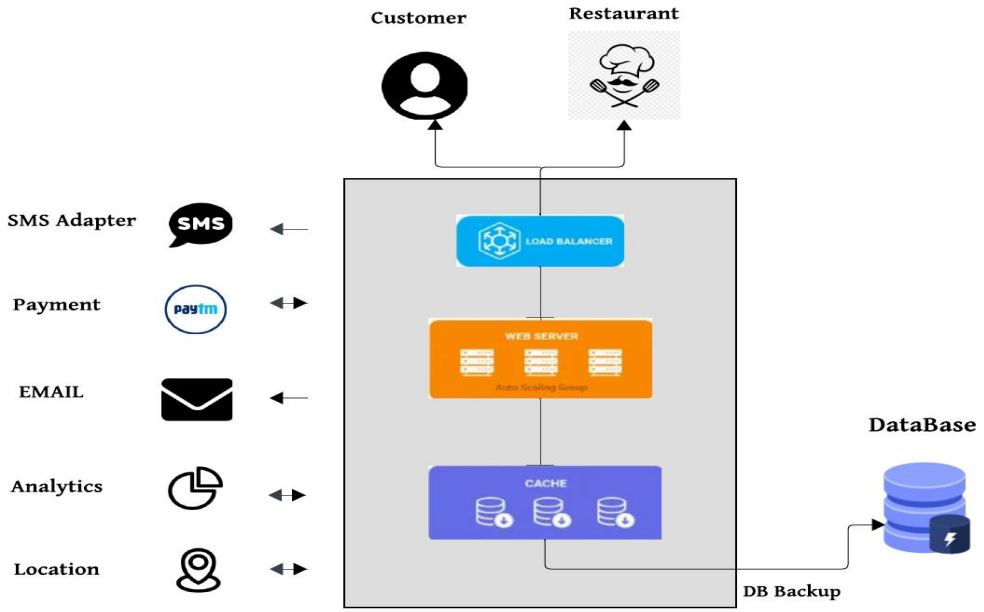

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	To create a Web UI for Android, you can use WebView, which is a component that allows you to embed web content in your Android application.	Android Studio -Kotlin
2.	Mobile App	You can create mobile app UIs using XML layouts and Kotlin for logic.	Kotlin
3.	Database	The app, developed in Kotlin for Android, utilizes a local database for efficient data storage and retrieval, including user profiles, orders, and menus. Additionally, the database serves as a backup mechanism, ensuring data integrity and availability, even in offline scenarios.	Local Database
4.	Cloud	The app, built in Kotlin for Android, leverages auto-scaling web servers in the cloud to efficiently handle varying traffic loads. This ensures seamless performance, scalability, and responsiveness for users, whether during peak dining hours or quieter periods.	Auto-scaling web servers for storage.
5.	File Storage	A food delivery app in Kotlin for Android Studio requires file storage for local databases & images (restaurant menus, user profile pictures) and caching.	Local Filesystem
6.	Location	For GPS tracking, the business entity can use Google Location API for Android.	Google Location API
7.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Set up the Android Studio development environment on your local machine. You can use the Android emulator or physical Android devices for testing. Ensure you have the necessary dependencies, such as a local database (e.g., SQLite), and configure the development environment for testing and debugging. Cloud Server Configuration: configure the server environment with the necessary runtime	Android Studio, cloud-auto scaling and load balancing

		components, such as the Android emulator, and ensure that the app can connect to the cloud database securely. Implement auto-scaling and load balancing to handle varying user loads.	

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Scalable Architecture	Involves load balancing to distribute traffic, stateless application servers for horizontal scalability, a cloud-based, scalable database for data storage, and caching mechanisms to reduce server load.	Kotlin ,Load balancer ,auto scaling
2.	Availability	Load balancers and distributed servers ensure the food delivery app's availability by evenly distributing traffic and allowing for horizontal scaling to meet fluctuating user demand. High availability measures, such as failover mechanisms and redundancy, guarantee uninterrupted service, even in the face of server or database failures. These technologies are essential to provide a reliable, responsive user experience and maintain continuous app functionality, particularly during peak hours or unexpected disruptions.	Kotlin, firebase ,analytics tool

S.No	Characteristics	Description	Technology
4.	Performance	Implement a robust caching strategy to reduce the load on your servers. Utilize Content Delivery Networks (CDNs) for efficient delivery of static assets like images, ensuring faster load times for users across various locations. Design the application to handle a high number of requests per second, allowing for horizontal scaling and auto-scaling to accommodate peak usage times, guaranteeing responsive service even during rush hours or promotional events. Google Analytics for analytics and performance comparison.	CDN, auto scaling ,load balancer, analytics