**. Scopul documentului:**

1. **Elaborarea unei diagrame logice pentru a defini arhitectura aplicației.**
   * Specifică procesul de planificare inițială pentru dezvoltarea aplicației.
   * Servicii incluse: Diagramă logică și diagramă ER.
2. **Crearea unei diagrame ER pentru a structura baza de date.**
   * Detalierează necesitatea de a crea o structură clară a bazei de date.
   * Servicii incluse: Design ER.
3. **Realizarea unui design inițial în Figma pentru a elimina orice ambiguitate legată de interfața grafică.**
   * Accentuează importanța unui design grafic inițial pentru claritate și aprobare.
   * Servicii incluse: Design în Figma.

**II. Propunere structură (Exemplu: 335 Ore - 2 Programatori):**

**Dezvoltarea de bază:**

* **Aplicație Client (iOS și Android):**
  + Căutare Restaurante
  + Plasare Comenzi
  + Urmărire Livrare
  + Recenzii și Evaluări
  + Integrare API Stripe pentru Plata cu Cardul
  + Opțiune de Plată Ramburs
* **Aplicație Rider (iOS și Android):**
  + Acceptare Comenzi
  + Gestionare Livrări
  + Raportare Financiară
  + Interfață de Navigare
* **Aplicație Restaurant (iOS și Android):**
  + Gestionare Meniu
  + Confirmare Livrări
  + Raportare Financiară
* **Aplicație Admin (Doar Browser):**
  + Gestionare Utilizatori
  + Gestionare Restaurante
  + Gestionare Rider
  + Monitorizare Comenzi
  + Rapoarte și Analize

**III. Sugestii suplimentare (Exemplu: 24 ore de muncă \* 2 programatori):**

* Generarea automată a facturilor prin integrare cu API SmartBill.

**IV. Preț și timp de implementare:**

* **Dezvoltare de bază cu 2 programatori:** X Euro + TVA
* **Dezvoltare de bază și sugestii suplimentare cu 2 programatori:** X Euro + TVA

**Termeni de achiziție:**

* **Confidențialitate:** Menționarea confidențialității ofertei.
* **Prețurile:** Specificarea că prețurile sunt fără TVA.
* **Servicii incluse:** Detalierea echipelor și suportului oferit pentru fiecare componentă a proiectului.

**Definiții tehnice:**

* **React, Ionic, NestJS, Firebase, MongoDB:** Explicații succinte pentru tehnologiile utilizate.
* **API (Interfață de Programare a Aplicațiilor):** Descriere a conceptului de API și utilizarea acestuia în proiect.

**Concluzie:**

Oferta este detaliată și organizată pentru a clarifica fiecare aspect al proiectului propus, incluzând specificații tehnice, servicii incluse, costuri estimate și termeni de colaborare. Aceasta este concepută pentru a oferi o viziune completă și transparentă asupra proiectului propus și a modului în care vor fi gestionate etapele inițiale de dezvoltare.

**Funcționalități cerute pentru fiecare categorie de utilizatori:**

**1. Șofer (Rider):**

* **Hartă cu traseul:** Trebuie să fie capabil să vizualizeze traseul optim către punctele de ridicare și livrare.
* **Timp estimat:** Aplicația trebuie să ofere estimări precise pentru timpul de la punctul de ridicare până la livrare.
* **Preluare comenzi:** Opțiunea de a accepta sau refuza comenzi direct din aplicație.
* **Raport zilnic:** Generarea automată a unui raport zilnic cu detaliile livrărilor efectuate.
* **Actualizare în timp real:** Monitorizarea în timp real a parcursului către client, inclusiv notificări pentru fiecare stadiu al livrării.

**2. Restaurant:**

* **Preluare comenzi:** Capacitatea de a primi și confirma comenzile de la clienți.
* **Status comenzi în timp real:** Notificarea clienților cu privire la statusul comenzii și timpul estimat de livrare.
* **Opțiune limitare comenzi:** În orele de vârf, restricționarea posibilității de a plasa noi comenzi dacă timpul de livrare depășește estimarea aplicată.
* **Vizualizare parcurs șofer:** Posibilitatea de a monitoriza traseul și locația șoferului pe hartă în timp real.

**3. Client:**

* **Informații detaliate despre restaurant:** Timpul estimat de livrare, timpul de preparare, recenzii, meniul complet.
* **Plasare comenzi:** Adăugarea produselor în coș și opțiuni de plată (card sau numerar).
* **Notificări pe parcursul comenzii:** Confirmare comandă, ridicare comandă, livrare comandă.
* **Detalii despre șofer:** Nume, număr de contact și parcurs pe hartă după confirmarea comenzii.

**Funcționalități suplimentare:**

* **Admin:**
  + Monitorizarea activității tuturor ramurilor (șoferi, restaurante, clienți).
  + Generarea de rapoarte detaliate pe fiecare categorie.
* **Livrări de colete mici sau plicuri:**
  + Adăugarea opțiunii pentru clienți de a solicita livrări de colete mici, inclusiv detalii despre tipul (plic sau cutie) și dimensiuni/greutate.

**Tehnologii necesare:**

Pentru a satisface aceste cerințe, sunt recomandate următoarele tehnologii:

* **Frontend:** Utilizarea unui framework modern pentru dezvoltarea aplicațiilor mobile, cum ar fi **React** sau **Ionic** pentru o interfață de utilizator reactivă și performantă.
* **Backend:** Implementarea unui server-side framework robust, cum ar fi **NestJS** cu **Node.js**, pentru a gestiona logica de business și integrările cu bazele de date.
* **Bază de date:** Utilizarea unei soluții NoSQL flexibile și scalabile, cum ar fi **MongoDB**, pentru a stoca și gestiona datele din aplicație.
* **Servicii Cloud:** Integrarea cu servicii precum **Firebase** pentru autentificare, stocare și notificări în timp real.
* **API-uri:** Dezvoltarea și integrarea API-urilor pentru comunicarea între diferitele module ale aplicației și pentru integrările cu serviciile externe (de exemplu, API pentru plăți online).

**Concluzie:**

Analiza cerințelor evidențiază nevoia unei soluții complexe și integrate, care să ofere o experiență fluidă și eficientă atât pentru utilizatorii finali (șoferi și clienți), cât și pentru partenerii de afaceri (restaurante). Alegerea tehnologiilor potrivite va asigura realizarea unei aplicații scalabile, securizate și ușor de întreținut, capabilă să gestioneze cerințele variate și volumul de date generat de operarea zilnică.