🍔 FOODIE EXPRESS - Business Requirements Document (BRD)

Project: Food Delivery Platform (Microservices Architecture)

📋 TABLE OF CONTENTS

1. EXECUTIVE SUMMARY {#executive-summary}

1.1 Business Vision

FoodieExpress is an online food delivery platform connecting customers with local restaurants. The platform enables customers to browse menus, place orders, track deliveries in real-time, and make payments seamlessly.

1.2 Business Goals

Goal	Description	Success Metric
Customer Acquisition	Attract new customers to the platform	10,000 registered users in 6 months
Restaurant Partners	Onboard local restaurants	500 restaurant partners
Order Volume	Process orders efficiently	Handle 1,000 orders/day
Customer Satisfaction	Deliver great experience	4.5+ star rating
Delivery Speed	Fast delivery times	Average 30-minute delivery

1.3 Revenue Model

```
┌─────────────────────────────────────────────────────────────┐
│                    REVENUE STREAMS                            │
├─────────────────────────────────────────────────────────────┤
│                                                             │
│  1. COMMISSION (15-25% per order from restaurants)          │
│                                                             │
│  2. DELIVERY FEE (₹20-50 based on distance)                  │
│                                                              │
│  3. SURGE PRICING (Peak hours, bad weather)                  │
│                                                              │
│  4. SUBSCRIPTION (FoodieExpress Premium - free delivery)     │
│                                                              │
│  5. ADVERTISING (Featured restaurants, promoted listings)     │
│                                                              │
└──────────────────────────────────────────────────────────────┘
```

2. BUSINESS CONTEXT {#business-context}

2.1 Problem Statement

For Customers:

- Limited time to cook or go out
- Difficulty finding new restaurants
- No way to track order status
- Multiple apps for different restaurants

For Restaurants:

- Limited reach to customers
- High cost of delivery infrastructure
- No technology for order management
- Difficulty managing peak hours

For Delivery Partners:

- Inconsistent income
- No flexible work options
- Complex route planning

2.2 Solution Overview

```
┌──────────────────────────────────────────────────────────────────────────┐
│                        FOODIEEXPRESS ECOSYSTEM                           │
│                                                                          │
│  ┌──────────────┐    ┌──────────────┐    ┌──────────────┐               │
│  │   CUSTOMER   │    │  RESTAURANT  │    │   DELIVERY   │               │
│  │     APP      │    │    PORTAL    │    │ PARTNER APP  │               │
│  └──────┬───────┘    └──────┬───────┘    └──────┬───────┘               │
│         │                   │                   │                        │
│         └───────────────────┼───────────────────┘                        │
│                             │                                            │
│                     ┌───────▼────────┐                                   │
│                     │  PLATFORM API  │                                   │
│                    │  (Microservices)│                                   │
│                    └─────────────────┘                                   │
│                                                                          │
│  SERVICES:                                                               │
│  • User Management         • Restaurant/Menu Management                  │
│  • Order Processing        • Payment Processing                          │
│  • Delivery Tracking       • Notifications                               │
│  • Reviews & Ratings       • Analytics & Reporting                       │
│                                                                          │
└──────────────────────────────────────────────────────────────────────────┘
```

2.3 Target Users

User Type	Description	Key Needs
Customer	End user ordering food	Easy ordering, tracking, variety
Restaurant Owner	Business selling food	Order management, analytics
Restaurant Staff	Kitchen/counter staff	Order preparation workflow
Delivery Partner	Delivers orders	Route optimization, earnings
Admin	Platform operator	Monitoring, support, analytics

---

3. STAKEHOLDERS {#stakeholders}

3.1 Stakeholder Matrix

Stakeholder	Role	Interest	Influence
Customers	Primary users	High	High
Restaurants	Partners	High	High
Delivery Partners	Service providers	Medium	Medium
Platform Admin	Operations	High	High
Investors	Funding	High	Medium
Support Team	Customer service	Medium	Low

---

4. EPICS & USER STORIES {#epics-user-stories}

📦 EPIC 1: USER MANAGEMENT

Business Need

Customers need to create accounts, manage profiles, and authenticate securely to use the platform.

User Stories

US-1.1: Customer Registration

```
AS A new customer
I WANT TO register on the platform
SO THAT I can order food from restaurants

ACCEPTANCE CRITERIA:
☑ User can register with email and password
☑ User can register using Google/Facebook OAuth
☑ User can register using phone number + OTP
☑ Email verification is sent after registration
☑ Password must be minimum 8 characters with 1 uppercase, 1 number
☑ Duplicate email/phone registration is prevented
☑ User receives welcome notification after registration

TECHNICAL TASKS:
□ Create User entity with validation
□ Implement registration API endpoint
□ Add email verification service
□ Implement OAuth2 integration
□ Add OTP service for phone verification
□ Create welcome notification event
```

US-1.2: Customer Login

```
AS A registered customer
I WANT TO login to my account
SO THAT I can access my orders and saved preferences

ACCEPTANCE CRITERIA:
☑ User can login with email/password
☑ User can login with phone + OTP
☑ User can login with Google/Facebook
☑ JWT token is issued on successful login
☑ Refresh token mechanism for session extension
☑ Account locks after 5 failed attempts (15 min cooldown)
☑ "Remember me" option persists session for 30 days
☑ Login from new device triggers email notification

TECHNICAL TASKS:
□ Implement JWT authentication
□ Create refresh token mechanism
□ Add account lockout logic
□ Implement OAuth2 login flow
□ Add new device detection
□ Create login notification event
```

US-1.3: Profile Management

```
AS A logged-in customer
I WANT TO manage my profile
SO THAT I can update my information and preferences

ACCEPTANCE CRITERIA:
☑ User can update name, email, phone
☑ User can add/edit multiple delivery addresses
☑ User can set default delivery address
☑ User can upload profile picture
☑ User can change password (requires current password)
☑ User can enable/disable notifications
☑ User can view order history

TECHNICAL TASKS:
□ Create profile update API
□ Implement address management (CRUD)
□ Add profile picture upload (S3/local storage)
□ Create password change API with validation
□ Implement notification preferences
```

US-1.4: Delivery Address Management

```
AS A customer
I WANT TO save multiple delivery addresses
SO THAT I can quickly select my location when ordering

ACCEPTANCE CRITERIA:
☑ User can add address with label (Home, Work, Other)
☑ User can add landmarks and delivery instructions
☑ User can set a default address
☑ User can edit and delete addresses
☑ Address includes: street, city, state, pincode, landmark
☑ GPS/Map integration for location selection
☑ Maximum 5 saved addresses per user

TECHNICAL TASKS:
□ Create Address entity
□ Implement address CRUD APIs
□ Add geocoding integration (optional)
□ Validate pincode against serviceable areas
```

US-1.5: Password Reset

```
AS A customer who forgot password
I WANT TO reset my password
SO THAT I can regain access to my account

ACCEPTANCE CRITERIA:
☑ User can request password reset via email
☑ Reset link expires in 15 minutes
☑ Link can only be used once
☑ User must set new password (different from last 3)
☑ All active sessions are invalidated on reset
☑ User receives confirmation email after reset

TECHNICAL TASKS:
□ Create password reset token generation
□ Implement reset email service
□ Add token validation and expiry logic
□ Create password history tracking
□ Implement session invalidation
```

📦 EPIC 2: RESTAURANT & MENU MANAGEMENT

Business Need

Restaurants need to manage their profile, menus, and availability to serve customers effectively.

User Stories

US-2.1: Restaurant Onboarding

```
AS A restaurant owner
I WANT TO register my restaurant on the platform
SO THAT I can receive orders from customers

ACCEPTANCE CRITERIA:
☑ Restaurant can submit registration request
☑ Required info: name, address, cuisine type, FSSAI license
☑ Restaurant can upload documents (license, menu photos)
☑ Admin reviews and approves/rejects registration
☑ Restaurant receives notification on approval
☑ Restaurant can set operating hours
☑ Restaurant can define delivery radius

TECHNICAL TASKS:
```

```
□ Create Restaurant entity
□ Implement registration API with document upload
□ Create admin approval workflow
□ Add operating hours configuration
□ Implement delivery area settings
```

US-2.2: Menu Management

```
AS A restaurant owner
I WANT TO create and manage my menu
SO THAT customers can see what I offer

ACCEPTANCE CRITERIA:
✅ Restaurant can create menu categories (Starters, Main Course, etc.)
✅ Restaurant can add items with name, description, price, image
✅ Restaurant can mark items as vegetarian/non-vegetarian
✅ Restaurant can add item variants (size, toppings)
✅ Restaurant can set item availability (in stock/out of stock)
✅ Restaurant can add item preparation time
✅ Restaurant can create combo meals
✅ Restaurant can apply discounts to items

TECHNICAL TASKS:
□ Create Menu, Category, Product entities (exists partially)
□ Implement menu CRUD APIs
□ Add variant/customization support
□ Implement availability toggle
□ Add combo meal logic
□ Create discount/offer management
```

US-2.3: Restaurant Availability

```
AS A restaurant owner
I WANT TO control my restaurant's availability
SO THAT I don't receive orders when I can't fulfill them

ACCEPTANCE CRITERIA:
✅ Restaurant can set daily operating hours
✅ Restaurant can mark as temporarily closed
✅ Restaurant can set different hours for different days
✅ Restaurant can pause accepting orders (busy mode)
✅ System auto-hides restaurant outside operating hours
✅ Restaurant can set holiday closures in advance

TECHNICAL TASKS:
□ Create operating hours schedule
□ Implement availability toggle API
□ Add scheduled closure feature
□ Create busy mode with auto-resume
```

US-2.4: Restaurant Dashboard

```
AS A restaurant owner
I WANT TO see my business analytics
SO THAT I can make informed decisions

ACCEPTANCE CRITERIA:
☑ Dashboard shows today's orders count and revenue
☑ Dashboard shows popular items
☑ Dashboard shows average preparation time
☑ Dashboard shows customer ratings
☑ Dashboard shows peak order hours
☑ Export reports to CSV/PDF
☑ Compare with previous week/month

TECHNICAL TASKS:
□ Create analytics aggregation service
□ Implement dashboard APIs
□ Add report generation service
□ Create comparison logic
```

📦 EPIC 3: PRODUCT BROWSING & SEARCH

Business Need

Customers need to discover restaurants and menu items easily to make informed ordering decisions.

User Stories

US-3.1: Restaurant Discovery

```
AS A customer
I WANT TO browse available restaurants
SO THAT I can choose where to order from

ACCEPTANCE CRITERIA:
☑ Show restaurants based on user's delivery location
☑ Display restaurant name, image, cuisine, rating, delivery time
☑ Show distance from user's location
☑ Indicate if restaurant is open/closed
☑ Show minimum order amount and delivery fee
☑ Filter by: cuisine, rating, delivery time, price range
☑ Sort by: relevance, rating, distance, delivery time
☑ Pagination for large result sets

TECHNICAL TASKS:
□ Create restaurant listing API with filters
□ Implement location-based filtering
□ Add distance calculation
□ Create availability check
□ Implement sorting algorithms
```

US-3.2: Restaurant Search

```
AS A customer
I WANT TO search for restaurants and dishes
SO THAT I can find exactly what I'm craving

ACCEPTANCE CRITERIA:
☑ Search by restaurant name
☑ Search by dish name
☑ Search by cuisine type
☑ Show search suggestions while typing
☑ Show recent searches
☑ Highlight matching text in results
☑ Handle typos and synonyms

TECHNICAL TASKS:
□ Implement full-text search (Elasticsearch optional)
□ Create search suggestions API
□ Add recent searches storage
□ Implement fuzzy matching
```

US-3.3: Restaurant Details

```
AS A customer
I WANT TO view restaurant details and menu
SO THAT I can decide what to order

ACCEPTANCE CRITERIA:
☑ Show restaurant info: name, address, operating hours
☑ Show ratings and review count
☑ Display full menu organized by categories
☑ Show item details: name, description, price, image
☑ Indicate veg/non-veg items
☑ Show item ratings
☑ Display preparation time estimates
☑ Show customization options

TECHNICAL TASKS:
□ Create restaurant detail API
□ Implement menu retrieval with categories
□ Add item details with variants
□ Create rating aggregation
```

US-3.4: Favorites & Recently Viewed

```
AS A customer
I WANT TO save my favorite restaurants
SO THAT I can quickly reorder from them

ACCEPTANCE CRITERIA:
☑ User can add/remove restaurants from favorites
☑ User can view list of favorite restaurants
☑ Show recently viewed restaurants
☑ Quick reorder from favorites
☑ Get notified about offers from favorite restaurants

TECHNICAL TASKS:
□ Create favorites entity and APIs
□ Implement recently viewed tracking
□ Add reorder functionality
□ Create favorite restaurant notifications
```

📦 EPIC 4: CART & CHECKOUT

Business Need

Customers need a smooth cart and checkout experience to complete orders efficiently.

User Stories

US-4.1: Add to Cart

```
AS A customer
I WANT TO add items to my cart
SO THAT I can build my order before checkout

ACCEPTANCE CRITERIA:
☑ Add item with quantity to cart
☑ Add item with customizations (size, toppings)
☑ Show cart item count in header
☑ Cart persists across sessions (logged-in users)
☑ Guest cart stored in local/session storage
☑ Cannot add items from multiple restaurants (show warning)
☑ Show clear cart option when switching restaurants

TECHNICAL TASKS:
□ Create Cart entity and CartItem entity
□ Implement add to cart API
□ Add customization handling
□ Create cart persistence logic
□ Implement restaurant conflict handling
```

US-4.2: Cart Management

```
AS A customer
I WANT TO modify my cart
SO THAT I can adjust my order before checkout

ACCEPTANCE CRITERIA:
☑ View all cart items with details
☑ Increase/decrease item quantity
☑ Remove item from cart
☑ Clear entire cart
☑ Show item-wise subtotal
☑ Show cart total with taxes
☑ Apply promo code / coupon
☑ Show savings from offers

TECHNICAL TASKS:
□ Implement cart update APIs
□ Create price calculation logic
□ Add tax calculation
□ Implement coupon validation and application
□ Create savings calculation
```

US-4.3: Checkout Process

```
AS A customer
I WANT TO complete checkout
SO THAT I can place my order

ACCEPTANCE CRITERIA:
☑ Select/confirm delivery address
☑ Add delivery instructions
☑ Select delivery type (standard/express)
☑ Choose payment method
```

```
✅ View order summary before placing
✅ Apply/change coupon at checkout
✅ Validate restaurant is still open
✅ Validate items are still available
✅ Calculate estimated delivery time

TECHNICAL TASKS:
□ Create checkout API
□ Implement delivery time estimation
□ Add availability validation
□ Create order summary generation
□ Implement address selection
```

US-4.4: Coupon & Offers

```
AS A customer
I WANT TO apply discount coupons
SO THAT I can save money on my order

ACCEPTANCE CRITERIA:
✅ Enter coupon code manually
✅ Show available coupons for the order
✅ Auto-apply best available offer
✅ Show discount breakdown
✅ Coupon validation (min order, expiry, usage limit)
✅ Show why coupon is not applicable
✅ Bank/card specific offers
✅ First-time user offers

TECHNICAL TASKS:
□ Create Coupon entity with rules
□ Implement coupon validation logic
□ Create coupon application API
□ Add offer discovery API
□ Implement usage tracking
```

📦 EPIC 5: ORDER MANAGEMENT

Business Need

Efficient order processing is critical for customer satisfaction and restaurant operations.

User Stories

US-5.1: Place Order

```
AS A customer
I WANT TO place my order
SO THAT I can receive my food

ACCEPTANCE CRITERIA:
☑ Order is created with all cart items
☑ Payment is processed before order confirmation
☑ Order confirmation displayed with order ID
☑ Confirmation email/SMS sent
☑ Order appears in restaurant portal immediately
☑ Delivery partner assigned automatically
☑ Estimated delivery time shown
☑ Stock/availability updated

TECHNICAL TASKS:
□ Create Order entity with OrderItems
□ Implement order placement API
□ Integrate payment gateway
□ Create order confirmation notification
□ Implement delivery partner assignment
□ Add inventory update logic
```

US-5.2: Order Status Tracking

```
AS A customer
I WANT TO track my order status
SO THAT I know when my food will arrive

ACCEPTANCE CRITERIA:
☑ Real-time order status updates
☑ Status stages: Placed → Confirmed → Preparing → Ready → Picked Up → Delivered
☑ Push notifications on status change
☑ Show delivery partner details when assigned
☑ Live tracking on map when out for delivery
☑ Estimated time updates dynamically
☑ Option to contact delivery partner

TECHNICAL TASKS:
□ Create order status enum and tracking
□ Implement WebSocket for real-time updates
□ Create notification triggers for status changes
□ Add delivery partner info API
□ Implement map tracking integration
```

US-5.3: Order History

```
AS A customer
I WANT TO view my past orders
SO THAT I can reorder or track my spending

ACCEPTANCE CRITERIA:
☑ List all past orders with date, restaurant, amount
☑ View order details (items, address, payment)
☑ Download invoice/receipt
☑ Reorder entire previous order
☑ Filter by date range
☑ Search orders

TECHNICAL TASKS:
□ Create order history API with pagination
□ Implement order detail retrieval
□ Add invoice generation
□ Create reorder functionality
□ Add search and filters
```

US-5.4: Order Cancellation

```
AS A customer
I WANT TO cancel my order
SO THAT I'm not charged for food I don't want

ACCEPTANCE CRITERIA:
☑ Cancel order before restaurant confirms
☑ Partial cancellation not allowed after confirmation
☑ Full refund if cancelled before preparation
☑ Partial refund based on order stage
☑ Cancel reason is required
☑ Cancellation confirmation sent
☑ Restaurant notified of cancellation

TECHNICAL TASKS:
□ Implement cancellation API with rules
□ Create refund calculation logic
□ Add cancellation notification
□ Implement refund processing
```

US-5.5: Restaurant Order Management

```
AS A restaurant staff
I WANT TO manage incoming orders
SO THAT I can prepare and fulfill them efficiently
```

```
ACCEPTANCE CRITERIA:
✅ See new orders with notification/sound
✅ View order details (items, customer notes)
✅ Accept or reject order
✅ Mark order as "Preparing"
✅ Mark order as "Ready for Pickup"
✅ Adjust preparation time if needed
✅ View order history
✅ Handle multiple orders simultaneously

TECHNICAL TASKS:
□ Create restaurant order dashboard API
□ Implement order status update APIs
□ Add real-time order notifications (WebSocket)
□ Create preparation time adjustment
```

📦 EPIC 6: PAYMENT PROCESSING

Business Need

Secure and flexible payment options are essential for completing transactions.

User Stories

US-6.1: Multiple Payment Methods

```
AS A customer
I WANT TO choose my preferred payment method
SO THAT I can pay conveniently

ACCEPTANCE CRITERIA:
✅ Credit/Debit card payment
✅ UPI payment (PhonePe, GPay, Paytm)
✅ Net banking
✅ Wallet balance (FoodieExpress wallet)
✅ Cash on Delivery
✅ Save cards for future use
✅ Set default payment method

TECHNICAL TASKS:
□ Create Payment entity
□ Integrate payment gateway (Razorpay/Stripe)
□ Implement UPI integration
□ Create wallet system
□ Add saved cards management
```

US-6.2: Payment Processing

```
AS A customer
I WANT TO complete payment securely
SO THAT my money and data are safe

ACCEPTANCE CRITERIA:
☑ Secure payment page (PCI compliant)
☑ OTP verification for cards
☑ Payment timeout handling (5 minutes)
☑ Retry failed payments
☑ Show payment status (success/failure)
☑ Generate transaction ID
☑ Handle double payment prevention

TECHNICAL TASKS:
□ Implement payment flow with gateway
□ Add payment status tracking
□ Create payment retry mechanism
□ Implement idempotency for payments
□ Add payment confirmation handling
```

US-6.3: Refunds

```
AS A customer
I WANT TO receive refunds for cancelled orders
SO THAT I get my money back

ACCEPTANCE CRITERIA:
☑ Automatic refund for eligible cancellations
☑ Refund to original payment method
☑ Refund to wallet option (faster)
☑ Refund status tracking
☑ Refund timeline visibility (5-7 days)
☑ Partial refund for partial issues

TECHNICAL TASKS:
□ Create refund processing service
□ Implement refund to source
□ Add wallet credit option
□ Create refund tracking
```

US-6.4: Wallet System

```
AS A customer
I WANT TO use FoodieExpress wallet
SO THAT I can pay faster and earn cashback

ACCEPTANCE CRITERIA:
✅ Add money to wallet
✅ View wallet balance
✅ View transaction history
✅ Earn cashback to wallet
✅ Use wallet + other payment (split)
✅ Wallet limits (max ₹10,000)
✅ Refunds credited to wallet

TECHNICAL TASKS:
□ Create Wallet entity
□ Implement wallet top-up
□ Create transaction history
□ Add cashback logic
□ Implement split payment
```

📦 EPIC 7: DELIVERY MANAGEMENT

Business Need

Efficient delivery operations ensure timely food delivery and partner satisfaction.

User Stories

US-7.1: Delivery Partner Onboarding

```
AS A person wanting to deliver
I WANT TO register as a delivery partner
SO THAT I can earn money delivering food

ACCEPTANCE CRITERIA:
✅ Submit application with personal details
✅ Upload documents (ID, license, vehicle RC)
✅ Background verification process
✅ Training completion requirement
✅ Approval notification
✅ Access to delivery partner app

TECHNICAL TASKS:
□ Create DeliveryPartner entity
□ Implement registration API
□ Add document upload and verification
□ Create approval workflow
```

US-7.2: Order Assignment

```
AS a delivery partner
I WANT TO receive order assignments
SO THAT I can pick up and deliver food

ACCEPTANCE CRITERIA:
☑ Receive new order notification
☑ See pickup location and drop location
☑ See estimated earnings
☑ Accept or reject within 30 seconds
☑ Auto-reassign if not accepted
☑ Batch multiple orders if same direction
☑ Consider partner location for assignment

TECHNICAL TASKS:
□ Create order assignment algorithm
□ Implement proximity-based assignment
□ Add acceptance timeout handling
□ Create batch order logic
□ Implement reassignment flow
```

US-7.3: Delivery Execution

```
AS A delivery partner
I WANT TO manage my delivery
SO THAT I can complete it efficiently

ACCEPTANCE CRITERIA:
☑ Navigate to restaurant (map integration)
☑ Mark "Arrived at Restaurant"
☑ Mark "Order Picked Up"
☑ Navigate to customer location
☑ Mark "Delivered"
☑ Collect payment for COD orders
☑ Handle delivery issues (customer not available)

TECHNICAL TASKS:
□ Create delivery status workflow
□ Implement location tracking
□ Add map navigation integration
□ Create COD collection handling
□ Implement issue reporting
```

US-7.4: Delivery Partner Earnings

```
AS A delivery partner
I WANT TO track my earnings
SO THAT I know how much I've made

ACCEPTANCE CRITERIA:
☑ View daily/weekly/monthly earnings
☑ See per-delivery earnings breakdown
☑ Track tips received
☑ View incentive bonuses
☑ Weekly payout schedule
☑ Download earnings report

TECHNICAL TASKS:
□ Create earnings calculation service
□ Implement earnings dashboard API
□ Add incentive/bonus logic
□ Create payout processing
```

📦 EPIC 8: RATINGS & REVIEWS

Business Need

Reviews help customers make decisions and restaurants improve their service.

User Stories

US-8.1: Rate Order

```
AS A customer
I WANT TO rate my order
SO THAT I can share my experience

ACCEPTANCE CRITERIA:
☑ Rate after order is delivered
☑ Rate restaurant (1-5 stars)
☑ Rate food quality (1-5 stars)
☑ Rate delivery partner (1-5 stars)
☑ Add text review (optional)
☑ Upload food photos (optional)
☑ Edit rating within 24 hours
☑ Prompt for rating via notification

TECHNICAL TASKS:
□ Create Review entity
□ Implement rating submission API
□ Add photo upload for reviews
□ Create rating prompt notification
□ Implement edit functionality
```

US-8.2: View Reviews

```
AS A customer
I WANT TO see restaurant reviews
SO THAT I can decide where to order

ACCEPTANCE CRITERIA:
☑ Show overall rating with count
☑ List recent reviews
☑ Filter by rating (5 star, 4 star, etc.)
☑ Sort by recent, helpful
☑ Show review photos
☑ Show restaurant response to reviews
☑ Mark review as helpful

TECHNICAL TASKS:
□ Create review listing API
□ Implement filtering and sorting
□ Add helpful vote functionality
□ Create restaurant response feature
```

📦 EPIC 9: NOTIFICATIONS

Business Need

Timely notifications keep users informed and engaged with the platform.

User Stories

US-9.1: Order Notifications

```
AS A customer
I WANT TO receive order updates
SO THAT I know my order status

ACCEPTANCE CRITERIA:
☑ Push notification on order confirmation
☑ Notification when order is being prepared
☑ Notification when order is out for delivery
☑ Notification with delivery partner details
☑ Notification on delivery
☑ SMS backup for critical updates
☑ Email receipt after delivery
```

```
TECHNICAL TASKS:
□ Create notification service
□ Implement push notifications (Firebase)
□ Add SMS integration (Twilio)
□ Create email notification templates
□ Implement notification preferences
```

US-9.2: Promotional Notifications

```
AS A customer
I WANT TO receive relevant offers
SO THAT I can save money

ACCEPTANCE CRITERIA:
☑ Notify about new offers/coupons
☑ Notify about offers from favorite restaurants
☑ Personalized recommendations
☑ Abandoned cart reminders
☑ Respect notification preferences
☑ Time-based notifications (lunch time offers)

TECHNICAL TASKS:
□ Create promotional notification system
□ Implement user segmentation
□ Add preference-based filtering
□ Create abandoned cart tracking
```

📦 EPIC 10: ADMIN & SUPPORT

Business Need

Platform operations require administrative tools and customer support capabilities.

User Stories

US-10.1: Admin Dashboard

```
AS A platform admin
I WANT TO monitor platform operations
SO THAT I can ensure smooth functioning

ACCEPTANCE CRITERIA:
☑ Real-time order volume
☑ Active users count
☑ Restaurant performance metrics
```

```
  ☑ Delivery partner metrics
  ☑ Revenue analytics
  ☑ Issue/complaint tracking
  ☑ System health monitoring

TECHNICAL TASKS:
□ Create admin analytics APIs
□ Implement real-time metrics
□ Add alert system
□ Create admin authentication
```

US-10.2: Customer Support

```
AS A customer
I WANT TO get help with issues
SO THAT my problems are resolved

ACCEPTANCE CRITERIA:
☑ In-app chat support
☑ Report order issues
☑ Request refund
☑ Escalation mechanism
☑ FAQ section
☑ Ticket tracking
☑ Resolution feedback

TECHNICAL TASKS:
□ Create support ticket system
□ Implement chat integration
□ Add issue categorization
□ Create escalation workflow
```

---

5. NON-FUNCTIONAL REQUIREMENTS {#non-functional-requirements}

5.1 Performance Requirements

Metric	Requirement
API Response Time	< 200ms for 95th percentile
Page Load Time	< 3 seconds
Order Processing	< 500ms end-to-end
Concurrent Users	Support 10,000 simultaneous users

Metric	Requirement
Database Queries	< 50ms average

5.2 Scalability Requirements

Metric	Requirement
Horizontal Scaling	Auto-scale based on load
Peak Load Handling	3x normal traffic during peak hours
Geographic Distribution	Multi-region deployment

5.3 Availability Requirements

Metric	Requirement
Uptime	99.9% availability
Disaster Recovery	RPO: 1 hour, RTO: 4 hours
Failover	Automatic failover to backup

5.4 Security Requirements

Requirement	Description
Authentication	JWT with refresh tokens
Authorization	Role-based access control
Data Encryption	TLS 1.3 in transit, AES-256 at rest
PCI Compliance	For payment processing
GDPR Compliance	User data privacy

---

6. IMPLEMENTATION ROADMAP {#roadmap}

Phase 1: MVP (Minimum Viable Product)

Duration: 8-10 weeks

```
SPRINT 1-2: Foundation
├── US-1.1: Customer Registration ⬅ START HERE
├── US-1.2: Customer Login
├── US-1.3: Profile Management
└── US-1.4: Delivery Address Management

SPRINT 3-4: Restaurant & Menu
```

```
  ├── US-2.1: Restaurant Onboarding (basic)
  ├── US-2.2: Menu Management
  ├── US-3.1: Restaurant Discovery
  └── US-3.3: Restaurant Details

  SPRINT 5-6: Ordering
  ├── US-4.1: Add to Cart
  ├── US-4.2: Cart Management
  ├── US-5.1: Place Order
  └── US-5.2: Order Status Tracking

  SPRINT 7-8: Payment & Delivery
  ├── US-6.1: Multiple Payment Methods
  ├── US-6.2: Payment Processing
  ├── US-7.2: Order Assignment
  └── US-7.3: Delivery Execution
```

Phase 2: Enhanced Features

Duration: 6-8 weeks

```
  ├── US-3.2: Restaurant Search
  ├── US-4.4: Coupon & Offers
  ├── US-5.3: Order History
  ├── US-5.4: Order Cancellation
  ├── US-6.3: Refunds
  ├── US-8.1: Rate Order
  └── US-9.1: Order Notifications
```

Phase 3: Advanced Features

Duration: 6-8 weeks

```
  ├── US-2.4: Restaurant Dashboard
  ├── US-6.4: Wallet System
  ├── US-7.4: Delivery Partner Earnings
  ├── US-9.2: Promotional Notifications
  ├── US-10.1: Admin Dashboard
  └── US-10.2: Customer Support
```

📊 MICROSERVICES MAPPING

Based on these requirements, here's how your services should be organized:

```
┌──────────────────────────────────────────────────────────────────────────────┐
│                        MICROSERVICES ARCHITECTURE                              │
├──────────────────────────────────────────────────────────────────────────────┤
│                                                                              │
│  ┌───────────────────┐  ┌───────────────────┐  ┌───────────────────┐       │
│  │   USER SERVICE    │  │RESTAURANT SERVICE│  │   ORDER SERVICE   │       │
│  │                   │  │                   │  │                   │       │
│  │ • Registration    │  │ • Restaurant CRUD│  │ • Cart Management│       │
│  │ • Authentication  │  │ • Menu Management│  │ • Order Placement│       │
│  │ • Profile Mgmt    │  │ • Availability    │  │ • Order Tracking  │       │
│  │ • Addresses       │  │ • Search          │  │ • Order History   │       │
│  │ • Favorites       │  │ • Categories      │  │                   │       │
│  └───────────────────┘  └───────────────────┘  └───────────────────┘       │
│                                                                              │
│  ┌───────────────────┐  ┌───────────────────┐  ┌───────────────────┐       │
│  │ PAYMENT SERVICE   │  │ DELIVERY SERVICE  │  │NOTIFICATION SVC   │       │
│  │                   │  │                   │  │                   │       │
│  │ • Payment Process│  │ • Partner Mgmt    │  │ • Push Notif      │       │
│  │ • Refunds         │  │ • Order Assign    │  │ • SMS             │       │
│  │ • Wallet          │  │ • Tracking        │  │ • Email           │       │
│  │ • Transactions    │  │ • Earnings        │  │ • In-App          │       │
│  └───────────────────┘  └───────────────────┘  └───────────────────┘       │
│                                                                              │
│  ┌───────────────────┐  ┌───────────────────┐                              │
│  │  REVIEW SERVICE   │  │  COUPON SERVICE   │                              │
│  │                   │  │                   │                              │
│  │ • Ratings         │  │ • Coupon CRUD     │                              │
│  │ • Reviews         │  │ • Validation      │                              │
│  │ • Photos          │  │ • Usage Tracking  │                              │
│  └───────────────────┘  └───────────────────┘                              │
│                                                                              │
└──────────────────────────────────────────────────────────────────────────────┘
```

🚀 START HERE

Your First 3 User Stories to Implement

Based on your current project state, start with these:

1. **US-1.4: Delivery Address Management** (Expand User Service)

- Add Address entity linked to User
- Create address CRUD APIs
- Set default address functionality

2. **US-4.1: Add to Cart** (Expand Order Service)

- Create Cart and CartItem entities
- Implement cart APIs

- Handle restaurant conflict logic

3. **US-5.2: Order Status Tracking** (Expand Order Service)

   - Add order status workflow
   - Create status update APIs
   - Implement order timeline

This document should be your product backlog. Pick user stories, implement them, and check off the acceptance criteria. Each completed story teaches you microservices concepts while building a real product!