

---

## *decision\_report*

---

### **Feature 1: Customer Account Management**

- **Pattern Chosen:** Request/Response
  - **Reasoning:**
    - Users need immediate confirmation when logging in or updating profile.
    - Secure and reliable handling of payment info is required.
  - **Why Only This:** Request/Response is simple, reliable, and perfect for CRUD operations.
- 

### **Feature 2: Order Tracking for Customers**

- **Pattern Chosen:** Long Polling
  - **Reasoning:**
    - Updates are frequent but don't need instant delivery.
    - Helps conserve mobile battery compared to continuous WebSockets.
  - **Why Only This:** Long Polling balances “near real-time” feel with server efficiency.
- 

### **Feature 3: Driver Location Updates**

- **Pattern Chosen:** WebSockets
  - **Reasoning:**
    - Drivers' location updates every 10–15 seconds.
    - Only the specific customer sees the driver's location.
    - Smooth real-time map tracking is required.
  - **Why Only This:** WebSockets provide interactive, bi-directional communication efficiently.
-

## Feature 4: Restaurant Order Notifications

- **Pattern Chosen:** WebSockets + Pub/Sub (Redis)
  - **Reasoning:**
    - Orders must appear instantly on multiple staff dashboards.
    - High reliability is needed to avoid missed orders.
  - **Why Only This:** Pub/Sub ensures decoupled, reliable delivery; WebSockets push updates instantly.
- 

## Feature 5: Customer Support Chat

- **Pattern Chosen:** WebSockets
  - **Reasoning:**
    - Messages must appear instantly for both customer and agent.
    - Supports multiple simultaneous conversations with typing indicators.
  - **Why Only This:** WebSockets handle real-time bi-directional chat best.
- 

## Feature 6: System-Wide Announcements

- **Pattern Chosen:** WebSockets (Namespace)
  - **Reasoning:**
    - Broadcast messages to thousands of users simultaneously.
    - Minor delay acceptable; not critical.
  - **Why Only This:** Efficient broadcasting without overwhelming server, works well with Socket.io namespaces.
- 

## Feature 7: Image Upload for Menu Items

- **Pattern Chosen:** Request/Response + Async Processing (Cloudinary)
- **Reasoning:**
  - Users upload large files (2–10MB) with progress indication.
  - Processing takes time (resizing/compression).
- **Why Only This:** Request/Response handles upload; async processing updates status when complete.