**Backend Design – FoodFast Drone Delivery**

**1. Subsystems**

Hệ thống FoodFast Drone Delivery gồm các subsystem chính:

* **User & Account Management**: quản lý tài khoản, thông tin khách hàng, merchant, admin.
* **Merchant & Store Management**: quản lý merchant, store, menu, categories, availability.
* **Order Management**: quản lý lifecycle đơn hàng từ đặt → xác nhận → thanh toán → giao → hoàn tất.
* **Payment Service**: tích hợp gateway thanh toán, sinh QR code, xử lý callback, cập nhật trạng thái.
* **Drone Service**: quản lý fleet drone, trạm base, gán drone cho order, tracking, unlock code.
* **Tracking & Telemetry**: tiếp nhận dữ liệu vị trí drone realtime qua MQTT/Redis, lưu checkpoint/route để báo cáo và hiển thị bản đồ.
* **Monitoring & Logging**: theo dõi sức khỏe drone, lịch sử flight, log event, audit unlock attempts.

**2. App**

* **Client**: mobile app (React Native) cho khách hàng, merchant; web portal cho admin.
* **Backend**: triển khai theo hướng microservice (NestJS / FastAPI).
* **Authentication**: JWT + Refresh token.
* **Realtime communication**:
  + Socket.IO / MQTT cho tracking drone và cập nhật trạng thái order theo thời gian thực.
  + Webhook từ Payment Provider.

**3. Data**

* **Database chính**: PostgreSQL + PostGIS extension để lưu toạ độ (geography(Point,4326)).
* **Realtime cache/stream**: Redis Streams hoặc Kafka để nhận dữ liệu drone 1s/lần.
* **Object storage**: AWS S3/Azure Blob để lưu hình ảnh menu, log file lớn.
* **Message broker**: RabbitMQ/Kafka để đồng bộ event giữa services (order\_created, payment\_success, drone\_assigned…).

**4. Route (Business Flows)**

**4.1 Order & Payment Flow**

1. **User đặt đơn** → tạo orders với status = pending, lưu coordinates vị trí giao hàng.
2. **Store xác nhận** → status = preparing. Khi món sẵn sàng → status = ready\_for\_payment.
3. **Payment QR sinh ra** → record trong payments với status = pending, gửi QR cho user.
4. **User thanh toán** → Payment Provider callback:
   * Thành công → payments.status = success, orders.status = confirmed.
   * Thất bại → orders.status = cancelled.
5. **Khi success** → backend sinh drone\_unlock\_code + drone\_unlock\_ttl, gửi cho user kèm ETA + tracking map.

**4.2 Drone Flow**

1. Dispatcher gán drone → orders.drone\_id + drone\_flights record.
2. Drone cất cánh → log takeoff.
3. **Tracking realtime**:
   * Drone gửi dữ liệu mỗi 1s → Redis/MQTT (không ghi thẳng DB).
   * Backend snapshot mỗi 1p hoặc mỗi 100m → ghi vào drone\_tracking (checkpoint).
   * Kết thúc chuyến bay → route tổng hợp lưu trong drone\_flights.flight\_path\_geo (GeoJSON/polyline).
4. Drone tới nơi → user quét QR drone\_unlock\_code.
   * Nếu đúng & trong TTL → mở hộp, status = completed.
   * Nếu sai/timeout → log unlock\_failed, fail-safe.
5. Drone hạ cánh hoặc quay về → log landing / return, drone về status = available.

**5. Monitoring**

* **Metrics (Prometheus + Grafana)**:
  + Drone uptime, battery %, flights/day, payment success rate.
  + Unlock fail attempts alert.
* **Logs (ELK/EFK)**:
  + Drone event logs (takeoff, landing, emergency).
  + Unlock attempts (thành công/thất bại).
  + Payment callback logs.
* **Alerting**:
  + Battery < 20%.
  + Unlock failed > 3 lần.
  + Payment failures tăng đột biến.

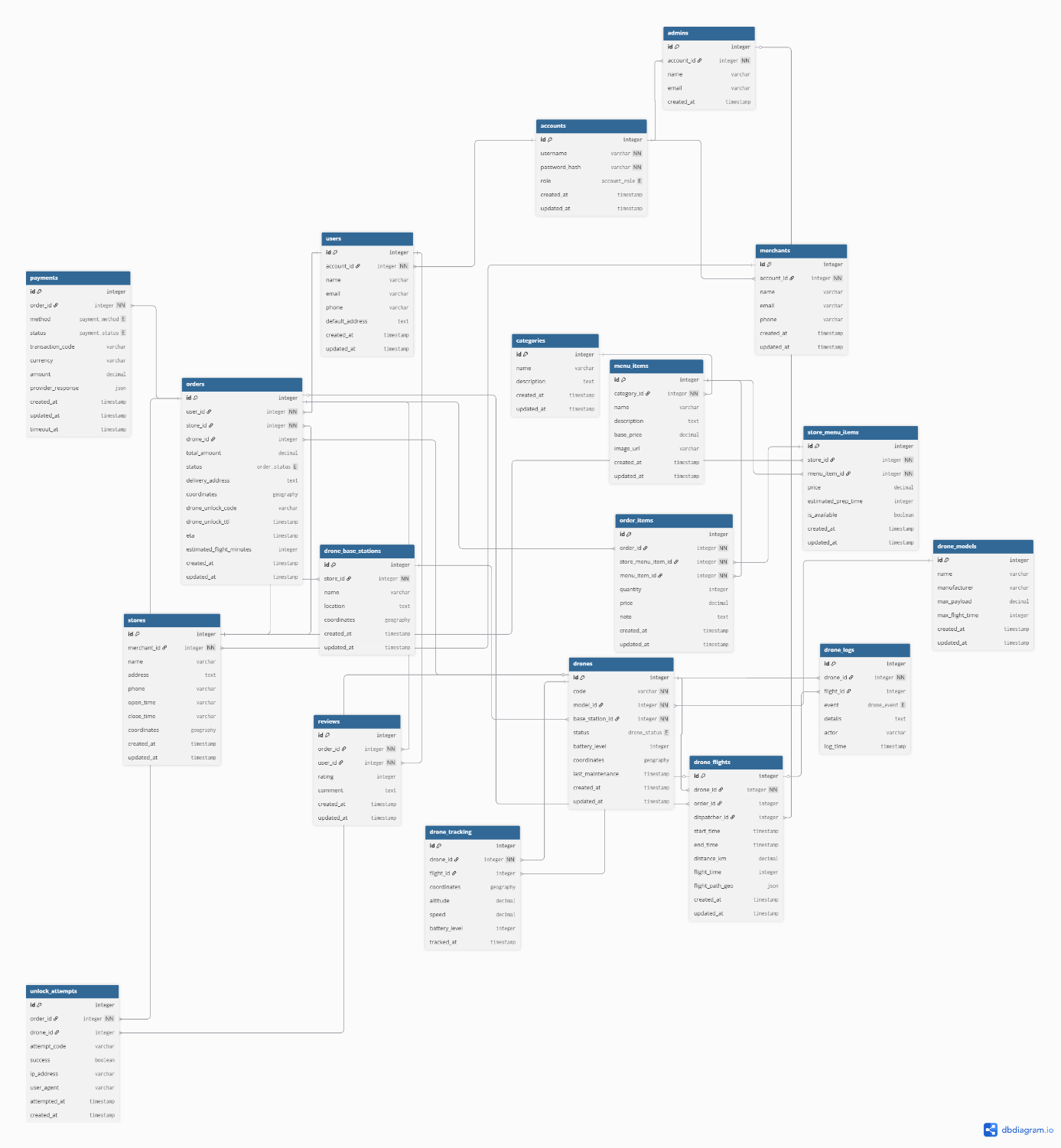
**6. Database – Key Entities**

**Toạ độ**

* Chuẩn hóa: tất cả entity có vị trí (stores, orders, drone\_base\_stations, drones, drone\_tracking) đều lưu dưới dạng **geography(Point,4326)**.

**Drone Tracking Strategy**

* **Realtime (Redis/MQTT)**: 1s/lần → phục vụ map UI.
* **Checkpoint (Postgres)**: mỗi 30s → drone\_tracking.
* **Flight summary**: route cuối cùng lưu dạng GeoJSON trong drone\_flights.flight\_path\_geo.



|  |
| --- |
| // ===================== ENUMs =====================  Enum account\_role {  customer  admin  merchant  }  Enum order\_status {  pending // user vừa tạo đơn  preparing // store xác nhận & đang chuẩn bị  ready\_for\_payment// store đã chuẩn bị xong, chờ thanh toán  confirmed // payment success, sẵn sàng cho dispatch  delivering // drone đang giao  completed // user đã nhận  cancelled // huỷ  failed\_delivery // giao không thành, drone quay về/timeout  }  Enum payment\_method {  e\_wallet  card  cod  }  Enum payment\_status {  processing  pending  success  failed  timeout  }  Enum drone\_status {  available  delivering  maintenance  offline  }  Enum drone\_event {  takeoff  landing  return  emergency  unlock\_attempt  unlock\_success  unlock\_failed  }  // ===================== TABLES =====================  Table accounts {  id integer [primary key]  username varchar [unique, not null]  password\_hash varchar [not null, note: 'hashed password']  role account\_role  created\_at timestamp  updated\_at timestamp  }  Table users {  id integer [primary key]  account\_id integer [not null, unique]  name varchar  email varchar  phone varchar  default\_address text  created\_at timestamp  updated\_at timestamp  }  Table admins {  id integer [primary key]  account\_id integer [not null, unique]  name varchar  email varchar  created\_at timestamp  }  // Merchants & Stores  Table merchants {  id integer [primary key]  account\_id integer [not null, unique]  name varchar  email varchar  phone varchar  created\_at timestamp  updated\_at timestamp  }  Table stores {  id integer [primary key]  merchant\_id integer [not null]  name varchar  address text  phone varchar  open\_time varchar [note: 'HH:MM']  close\_time varchar [note: 'HH:MM']  coordinates geography [note: 'Vị trí cửa hàng (POINT, SRID=4326)']  created\_at timestamp  updated\_at timestamp  }  // Menu categories & items  Table categories {  id integer [primary key]  name varchar  description text  created\_at timestamp  updated\_at timestamp  }  Table menu\_items {  id integer [primary key]  category\_id integer [not null]  name varchar  description text  base\_price decimal  image\_url varchar  created\_at timestamp  updated\_at timestamp  }  Table store\_menu\_items {  id integer [primary key]  store\_id integer [not null]  menu\_item\_id integer [not null]  price decimal  estimated\_prep\_time integer [note: 'minutes']  is\_available boolean [default: true]  created\_at timestamp  updated\_at timestamp  }  // Orders & Payments  Table orders {  id integer [primary key]  user\_id integer [not null]  store\_id integer [not null]  drone\_id integer  total\_amount decimal  status order\_status  delivery\_address text  coordinates geography [note: 'Vị trí khách lúc đặt (POINT, SRID=4326)']  drone\_unlock\_code varchar [note: 'QR/OTP mở hộp drone, tạo khi payment success']  drone\_unlock\_ttl timestamp [note: 'thời hạn hiệu lực mã mở']  eta timestamp  estimated\_flight\_minutes integer  created\_at timestamp  updated\_at timestamp  }  Table order\_items {  id integer [primary key]  order\_id integer [not null]  store\_menu\_item\_id integer [not null]  menu\_item\_id integer [not null]  quantity integer  price decimal  note text  created\_at timestamp  updated\_at timestamp  }  Table payments {  id integer [primary key]  order\_id integer [not null]  method payment\_method  status payment\_status  transaction\_code varchar [note: 'QR code / mã giao dịch thanh toán từ payment gateway']  currency varchar [note: 'e.g., VND, USD']  amount decimal  provider\_response json [note: 'raw callback payload if needed']  created\_at timestamp  updated\_at timestamp  timeout\_at timestamp  }  // Drone management  Table drone\_models {  id integer [primary key]  name varchar  manufacturer varchar  max\_payload decimal  max\_flight\_time integer [note: 'minutes']  created\_at timestamp  updated\_at timestamp  }  Table drone\_base\_stations {  id integer [primary key]  store\_id integer [not null]  name varchar  location text  coordinates geography [note: 'Vị trí trạm drone (POINT, SRID=4326)']  created\_at timestamp  updated\_at timestamp  }  Table drones {  id integer [primary key]  code varchar [unique, not null]  model\_id integer [not null]  base\_station\_id integer [not null]  status drone\_status  battery\_level integer [note: '0-100 percent']  coordinates geography [note: 'Vị trí hiện tại của drone (POINT, SRID=4326)']  last\_maintenance timestamp  created\_at timestamp  updated\_at timestamp  }  Table drone\_flights {  id integer [primary key]  drone\_id integer [not null]  order\_id integer  dispatcher\_id integer [note: 'admin/staff who authorized']  start\_time timestamp  end\_time timestamp  distance\_km decimal  flight\_time integer [note: 'minutes']  flight\_path\_geo json [note: 'optional polyline/geojson snapshot']  created\_at timestamp  updated\_at timestamp  }  Table drone\_logs {  id integer [primary key]  drone\_id integer [not null]  flight\_id integer  event drone\_event  details text  actor varchar [note: 'who triggered event (system/dispatcher/user)']  log\_time timestamp  }  Table drone\_tracking {  id integer [primary key]  drone\_id integer [not null]  flight\_id integer  coordinates geography [note: 'Vị trí realtime (POINT, SRID=4326)']  altitude decimal [note: 'meters']  speed decimal [note: 'm/s']  battery\_level integer [note: 'snapshot']  tracked\_at timestamp  }  // Security / Audit  Table unlock\_attempts {  id integer [primary key]  order\_id integer [not null]  drone\_id integer  attempt\_code varchar  success boolean  ip\_address varchar  user\_agent varchar  attempted\_at timestamp  created\_at timestamp  }  // Reviews  Table reviews {  id integer [primary key]  order\_id integer [not null]  user\_id integer [not null]  rating integer [note: '1-5']  comment text  created\_at timestamp  updated\_at timestamp  }  // ===================== RELATIONSHIPS =====================  Ref: users.account\_id > accounts.id  Ref: admins.account\_id > accounts.id  Ref: merchants.account\_id > accounts.id  Ref: stores.merchant\_id > merchants.id  Ref: menu\_items.category\_id > categories.id  Ref: store\_menu\_items.menu\_item\_id > menu\_items.id  Ref: store\_menu\_items.store\_id > stores.id  Ref: orders.user\_id > users.id  Ref: orders.store\_id > stores.id  Ref: orders.drone\_id > drones.id  Ref: order\_items.order\_id > orders.id  Ref: order\_items.store\_menu\_item\_id > store\_menu\_items.id  Ref: order\_items.menu\_item\_id > menu\_items.id  Ref: payments.order\_id > orders.id  Ref: drones.model\_id > drone\_models.id  Ref: drones.base\_station\_id > drone\_base\_stations.id  Ref: drone\_base\_stations.store\_id > stores.id  Ref: drone\_flights.drone\_id > drones.id  Ref: drone\_flights.order\_id > orders.id  Ref: drone\_flights.dispatcher\_id > admins.id  Ref: drone\_logs.drone\_id > drones.id  Ref: drone\_logs.flight\_id > drone\_flights.id  Ref: drone\_tracking.drone\_id > drones.id  Ref: drone\_tracking.flight\_id > drone\_flights.id  Ref: unlock\_attempts.order\_id > orders.id  Ref: unlock\_attempts.drone\_id > drones.id  Ref: reviews.order\_id > orders.id  Ref: reviews.user\_id > users.id |