SOA projek - food delivery app

UI:
-Frontend api - TJAN

Mikrostoritve:

1. Restaurant service - TJAN
- CRUD za restavracije
- CRUD izdelki na menjiu za določeno restavracijo
- status restavracij (open/close)

endpoints:

GET /restaurants
GET /restaurants/{id}
POST /restaurants/{id}
PUT /restaurants/{id}
DELETE /restaurants/{id}
POST /restaurants/{id}/open
POST /restaurants/{id}/menu
GET /restaurants/{id}/menu
GET /restaurants/{id}/menu/{item_id}
PUT /restaurants/{id}/menu/{item_id}
DELETE /restaurants/{id}/menu/{item_id}

DB - restaurant_db

2. User service - SIMON
- CRUD za uporabnike (register/login, delete user itd,)
- authentication
- različni uporabniki, guest, logged in premium user (optional)

endpoints:

POST /users/register
GET users/{id}
PUT users/{id}
DELETE users/{id}
GET /users

DB - user_db

3. Order service - JURE
- iz košarice naredi naročilo - order
- različni statusi naročila (pending, sprejeto, na_poti, dostavljeno idt.)
- komunicira z:

- Restaurant service (preverja obstoj izdelka)
- Delivery service (določi dostavljalca)
- Payment service (potrditev plačila)

endpoints:

POST /orders
GET /orders
GET /orders/{id}
GET (orders/user/{user_id}
GET /orders/{id}/status

DB - order_db

4. Delivery service - SIMON
- sprejem naročila iz Order service
- pripiše dostavljalca
- spremlja status naročila (pending, sprejeto, na poti, dostavljeno)

endpoints:

GET /deliveries
POST /deliveries
GET /deliveries/{id}
PUT /deliveries/{id}/status
PUT /deliveries/{id}/assign-driver
GET /deliveries/driver/{driver_id}

DB - delivery_db

5. Payment service - JURE
- process payments (Stripe ?)
- hrani obvestila o plačilih
- obvesti order service o uspešnosti plačila (success/failure)

endpoints:

GET /payments
POST /payments
GET /payments/{id}
GET payments/user/{user_id}
POST /payments/{id}/confirm

DB - payment_db

6. API gateway (opcijsko)

- način, da frontent komunicira z eno storitivjo, ne 5timi

-(nima svoj db)