Backlog

1. **Flights Service**
   - Implement Create Flight feature using Command Pattern
   - Implement Update Flight feature using Command Pattern
   - Implement Delete Flight feature using Command Pattern
   - Implement Undo/Redo functionality for Create/Update/Delete flights
   - Implement simple queries (list all flights) using Query Object Pattern
   - Implement detailed flight queries using Query Object Pattern
   - Sync flight data between PostgreSQL and Elasticsearch with Signal (Observer Pattern)
   - Implement indexing of flight data in Elasticsearch
2. **Flights Filter Service**
   - Implement advanced search filters in Elasticsearch for flights
3. **Train Service**
   - Implement Create Train feature using Command Pattern
   - Implement Update Train feature using Command Pattern
   - Implement Delete Train feature using Command Pattern
   - Implement Undo/Redo functionality for Create/Update/Delete trains
   - Implement simple queries (list all trains) using Query Object Pattern
   - Implement detailed train queries using Query Object Pattern
   - Sync train data between PostgreSQL and Elasticsearch with Signal (Observer Pattern)
   - Implement indexing of train data in Elasticsearch
4. **Trains Filter Service**
   - Implement advanced search filters in Elasticsearch for Trains
5. **Profiles Mirco Service**
   - Implement Create Profile feature using Service Layer Pattern
   - Implement Update Profile feature using Service Layer Pattern
   - Implement Delete Profile feature using Service Layer Pattern
   - Implement simple queries (list all profiles)
   - Implement detailed profile queries
6. **Authentication & Authorization Micro Service**
   - Implement user signup
   - Implement user login
   - Implement password change feature
   - Implement Role-Based Access Control (RBAC)
   - Implement dynamic permission management
7. **Wallet Service**
   - Implement Create Wallet feature for users by User National Code
   - Implement Update Wallet feature
   - Save wallet transaction history
   - Wallet Credit by gRPC and Observer Pattern communicate with Payment Gateway Service
   - Wallet Debit by gRPC and Observer Pattern communicate with Refund Service
8. **Payment Gateway Service**
   - Implement Create PaymentGateway
   - Implement Update PaymentGateway
   - Implement Simple Queries on PaymentGateway Model
   - Implement Communicate with Wallet Service by gRPC with Observer Pattern
   - Implement Communicate with Reservation Service by gRPC with Observer Pattern

9. Refund Service
    - Implement Create Refund request feature
    - Implement Update Refund request feature
10. Reservation Service
    - Implement hotel reservation in two steps (operator-based)
    - Implement reservation of flights, tours, and trains with Kafka event producer
    - Implement conditions for triggering reservations
11. Notification Service
    - Implement event-driven notification system
    - Create notification queues and listeners
12. Hotel Service
    - Implement Create Hotel feature
    - Implement Update Hotel feature
    - Implement Delete Hotel feature
    - Implement simple queries (list all hotels)
    - Implement detailed hotel queries
    - Sync hotel data between PostgreSQL and Elasticsearch
    - Implement indexing of hotel data in Elasticsearch
13. Hotel Filter Service
    - Implement advanced search filters in Elasticsearch for hotels
14. Hotel Reviews Service
    - Implement Create Hotel Review feature
    - Implement Update Hotel Review feature
    - Implement Delete Hotel Review feature
    - Implement advanced queries for hotel reviews
15. Tour Service
    - Implement Create Tour feature
    - Implement Update Tour feature
    - Implement Delete Tour feature
    - Implement simple queries (list all tours)
    - Implement detailed tour queries
    - Sync tour data between PostgreSQL and Elasticsearch
    - Implement indexing of tour data in Elasticsearch
16. Tour Filter Service
    - Implement advanced search filters in Elasticsearch for tours