### РЕКОМЕНДАЦИИ ОТ ДИПСИК. Архитектурные рекомендации для NestJS

**1. Структура модулей (по доменам):**

text

modules/

├── catalog/ # Категории, бренды, продукты

├── spine/ # Группировки товаров

├── product-unit/ # Учетные единицы + все статусы

├── supply/ # Поставщики

├── customer/ # Клиенты

├── cash/ # Касса и денежные операции

└── reporting/ # Отчеты и логи

**2. Ключевые сервисы и их ответственность:**

**ProductUnitService** - главный orchestrator:

* Смена статусов (CLEAR → CANDIDATE → IN\_REQUEST → IN\_DELIVERY → IN\_STORE → SOLD/CREDIT)
* Создание дочерних юнитов при множественной заявке
* Валидация переходов между статусами

**SpineAggregationService** - для статистики по карточкам:

* Агрегация counts по статусам (CLEAR, CANDIDATE, IN\_REQUEST, etc.)
* Группировка по брендам внутри spine

**ProductUnitLoggerService** - для логирования:

* Автоматическое логирование ВСЕХ изменений статусов
* Запись в ProductUnitLog с типом и метаданными

**3. Критические моменты для реализации:**

**Транзакции!** Многие операции должны быть в транзакциях:

* Создание дочерних юнитов + изменение родителя
* Изменение статуса + логирование
* Продажа + создание CashEvent

**Статус-машина** - реализуйте строгую проверку переходов:

typescript

const allowedTransitions = {

CLEAR: ['CANDIDATE', 'SPROUTED'],

CANDIDATE: ['IN\_REQUEST', 'CLEAR'],

SPROUTED: ['IN\_REQUEST'],

*// ...*

}

**DTO для операций:**

* CreateProductUnitDto (из Product)
* ChangeStatusDto (статус + метаданные)
* CreateMultipleRequestDto (родитель + количество)

**Что нужно пересмотреть в схеме:**

1. **ProductUnitPhysicalStatus** - добавьте RESERVED или IN\_DELIVERY
2. **CashEventType** - добавьте CREDIT\_PAYMENT для оплаты кредитов
3. **Индексы** на часто фильтруемые поля: statusCard, statusProduct, spineId, createdAt

**Next steps:**

1. Начните с модуля product-unit и реализуйте статус-машину
2. Добавьте автоматическое логирование через хуки Prisma
3. Создайте SpineAggregationService для отображения карточек

Ваша бизнес-логика продумана очень хорошо. Теперь важно реализовать ее в коде без "размазывания" логики по разным местам.

Отлично! Ставлю на паузу ⏸️

### Карта разработки (сохранена):

**✅ Завершено:**

* Фаза 1: Настройка проекта и БД
* Фаза 2: Модуль брендов (Brand)
* Фаза 3: Модуль категорий (Category) - Materialized Path
* Фаза 4: Модуль поставщиков (Supplier)
* Фаза 5: Модуль клиентов (Customer)
* Фаза 6: Модуль Spine (группировки)
* Верстка создания категорий и Spine

**➡️ Следующие этапы:**  
7. Модуль продуктов (Product) - CRUD + Images  
8. Модуль ProductUnit (ядро системы)  
9. Сервис логирования ProductUnitLog  
10. Сервис агрегации для Spine  
11. Workflow: Кандидаты и заявки  
12. Workflow: Поставки  
13. Workflow: Продажи  
14. Модуль кассы (CashDay/CashEvent)

Отличная работа сегодня! Вечером продолжим с этапа 7 - модуль продуктов.

Жду вашего возвращения! 👋