ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Разработка Telegram-бота учета автозапчастей с использованием искусственного интеллекта на основе ООО «ТОПЛАЙНЕР»

Киричков А. И. · 09.03.03 «Прикладная информатика»
Научный руководитель: к.т.н., доцент Д. Г. Алгазина
Барнаул, 2026

Актуальность разработки

Проблема решается не «еще одним ботом», а структурированием хаотичного обмена фото, OEM и ценами

В исходном процессе данные о деталях распадаются на переписки, фото и отдельные таблицы. В итоге каждый новый запрос снова превращается в ручной поиск.

- фото и цена часто передаются вручную
- OEM-номер может быть прочитан с ошибкой
- нет единой карточки товара и истории обновлений
- клиенту приходится ждать первичную проверку продавца

Поиск по фотографии и распознавание OEM

10–15
минут ручного поиска до внедрения

2–4
минуты после внедрения

ИИ
распознает OEM по фото

Функциональная структура ООО «ТОПЛАЙНЕР»

В проекте важны роли участников информационного обмена

Проектная организационно-функциональная схема для автоматизации каталога автозапчастей

Рисунок 2.1 — Обобщенная организационно-функциональная структура ООО «ТОПЛАЙНЕР»

- поставщик публикует карточки автозапчастей
- клиент ищет и открывает карточку товара
- ИТ-поддержка обеспечивает работу бота, БД, S3 и ИИ-модуля

Процесс «как есть»

Ручной обмен данными сохраняет гибкость общения, но ломается при росте числа товаров и запросов

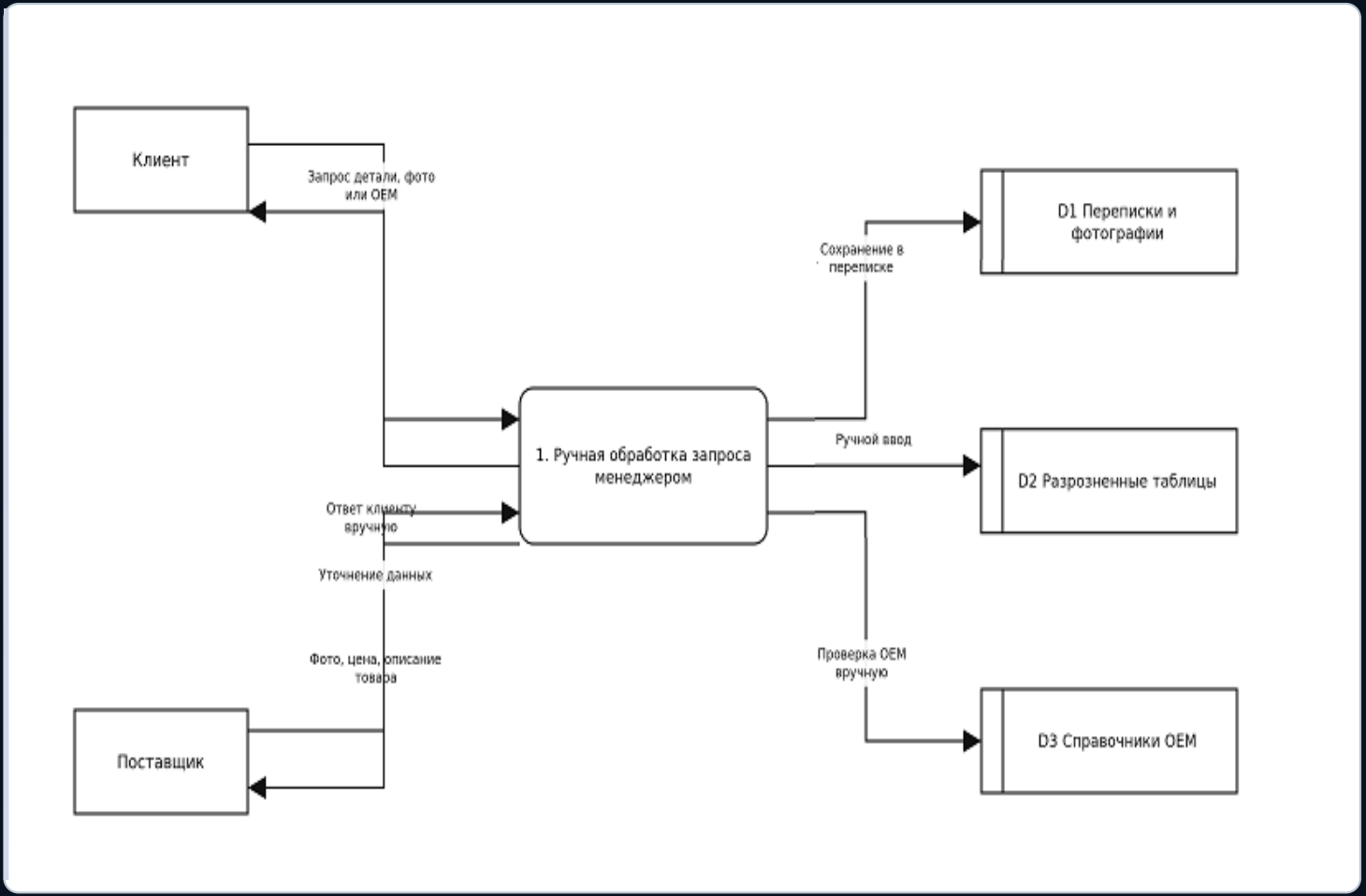


Рисунок 2.2 — DFD-модель процесса поиска и публикации «как есть»

КЛЮЧЕВЫЕ РИСКИ

- поиск по перепискам и фото
- ручная передача контактов
- дубли карточек и потеря актуальности
- ошибки при чтении OEM

Вывод: нужен единый цифровой контур с карточками, фото, поиском и ИИ-распознаванием.

Целевой процесс «как должно быть»

Бот превращает переписку в структурированный маршрут: запрос → БД → результат → контакт продавца

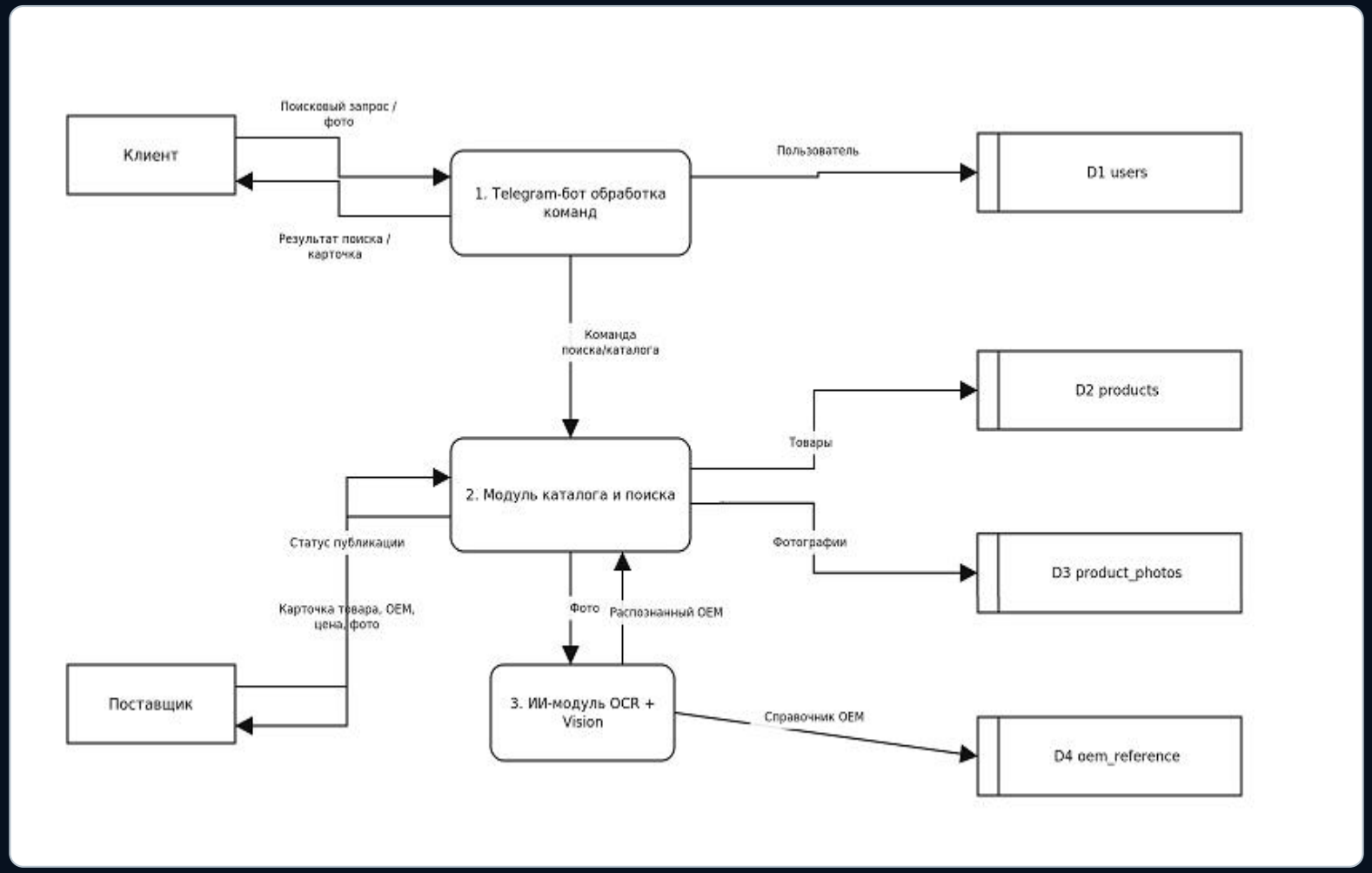


Рисунок 2.3 — DFD-модель целевого процесса

ЦЕЛЕВАЯ ЛОГИКА

- поставщик создает карточку товара
- бот сохраняет данные в PostgreSQL
- фото идут в Telegram file_id / S3
- ИИ выделяет OEM из фотографии
- клиент получает карточку и контакт

Функциональные требования

Два контура работы: поставщик наполняет каталог, клиент ищет и открывает карточку

ПОСТАВЩИК

- регистрация и контакт
- добавление товара
- OEM, тип, цена, описание
- загрузка нескольких фото
- раздел «Мои товары»
- редактирование и удаление своих карточек

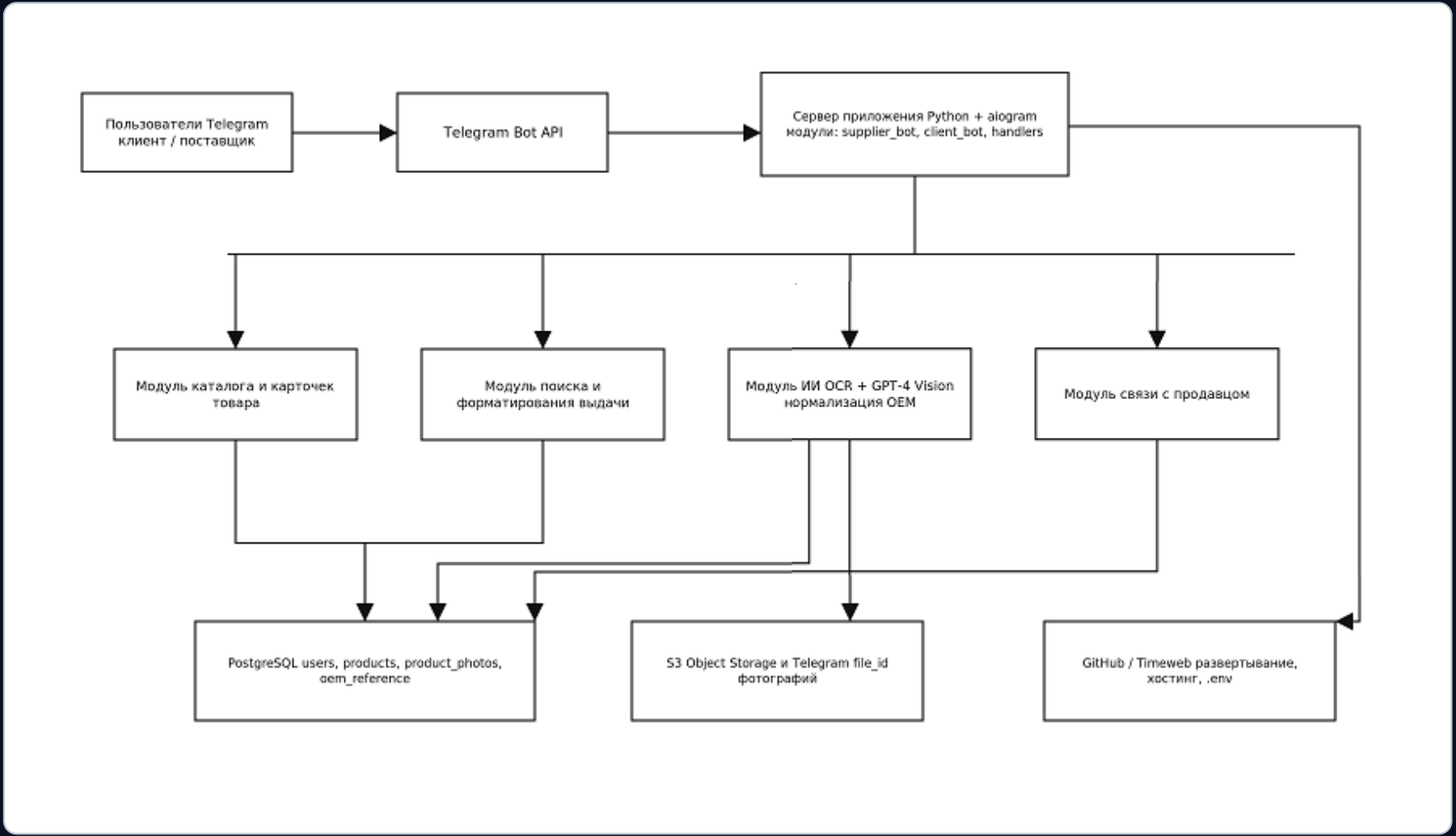
КЛИЕНТ

- просмотр каталога
- поиск по названию и OEM
- поиск по фотографии
- открытие полной карточки
- фото, цена, OEM и описание
- кнопка связи с продавцом

Фрагмент раздела «Мои товары» из ВКР

Проектная архитектура Telegram-бота

Единая база данных обслуживает клиентский и поставщицкий сценарии



- client_bot
- supplier_bot
- PostgreSQL
- S3 / file_id
- OCR + GPT Vision

Рисунок 3.1 — Проектная архитектура Telegram-бота учета автозапчастей

Технологический стек

Стек подобран под быстрый запуск MVP и дальнейшее масштабирование

Python

серверная логика и интеграции

aiogram

обработка Telegram Bot API

PostgreSQL

пользователи, товары, фото, OEM

S3 / file_id

хранение и повторная отправка фото

OCR + GPT-4 Vision

выделение OEM по фотографии

Timeweb / GitHub

размещение и контроль версий

Выбранный стек соответствует границам ВКР: каталог карточек, поиск, фото и ИИ-распознавание без перегруза складскими и платежными функциями.

IDEF0 A0: декомпозиция функций

Основная функция разложена на регистрацию, добавление, распознавание, поиск и передачу контакта

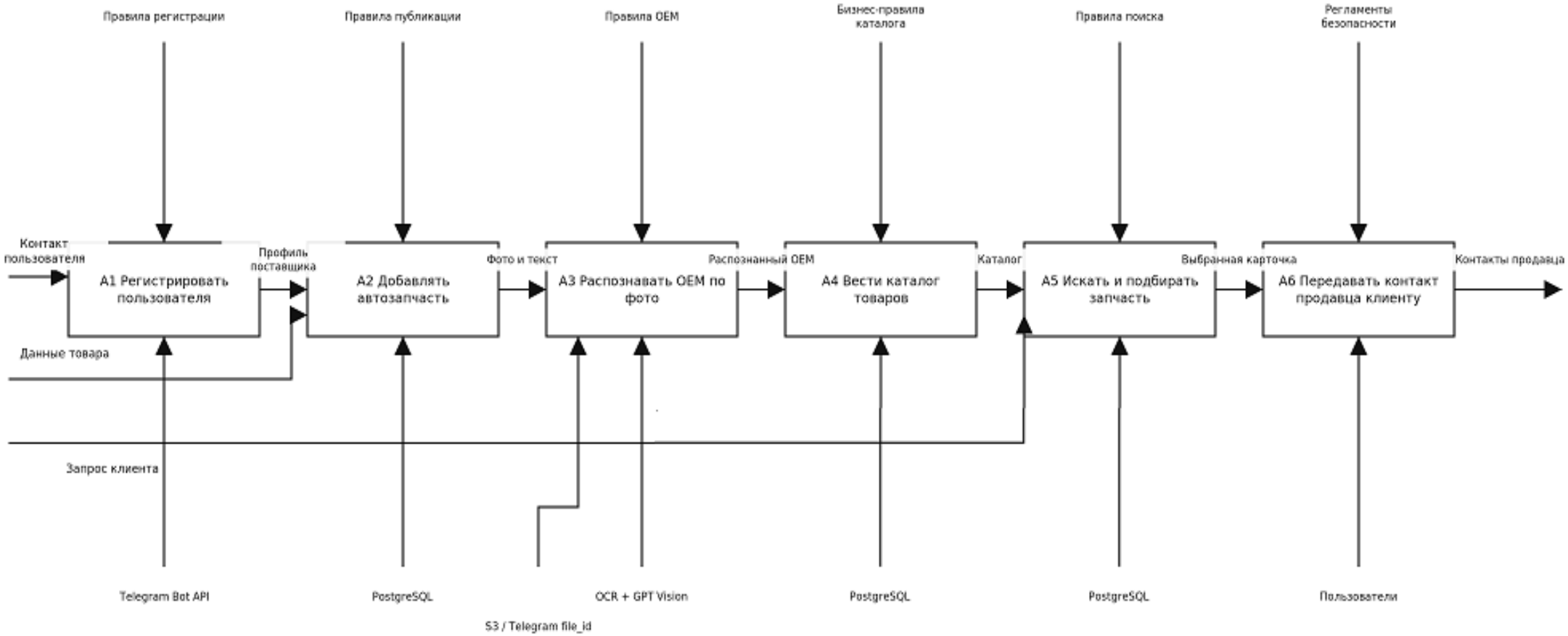


Рисунок 3.9 — Декомпозиция IDEF0 A0 функций Telegram-бота

ER-модель базы данных

Ключевая связь: products.seller_id → users.id, чтобы поставщик управлял только своими товарами

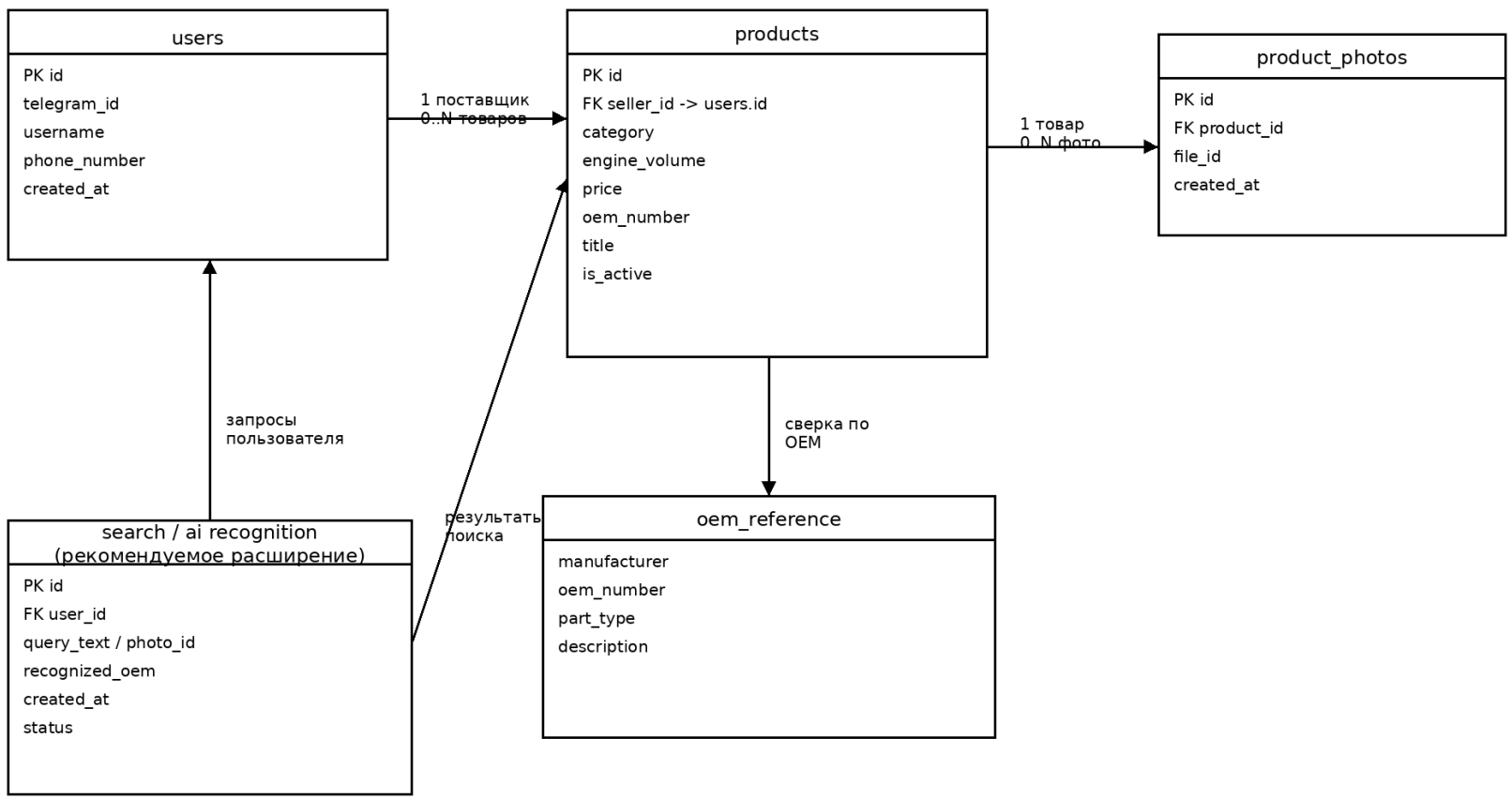


Рисунок 3.7 — ER-модель базы данных Telegram-бота

ТАБЛИЦЫ

- users
- products
- product_photos
- oem_reference
- search_ai_log — при развитии

BPMN: добавление автозапчасти поставщиком

Поставщик последовательно вводит данные, загружает фото и публикует карточку

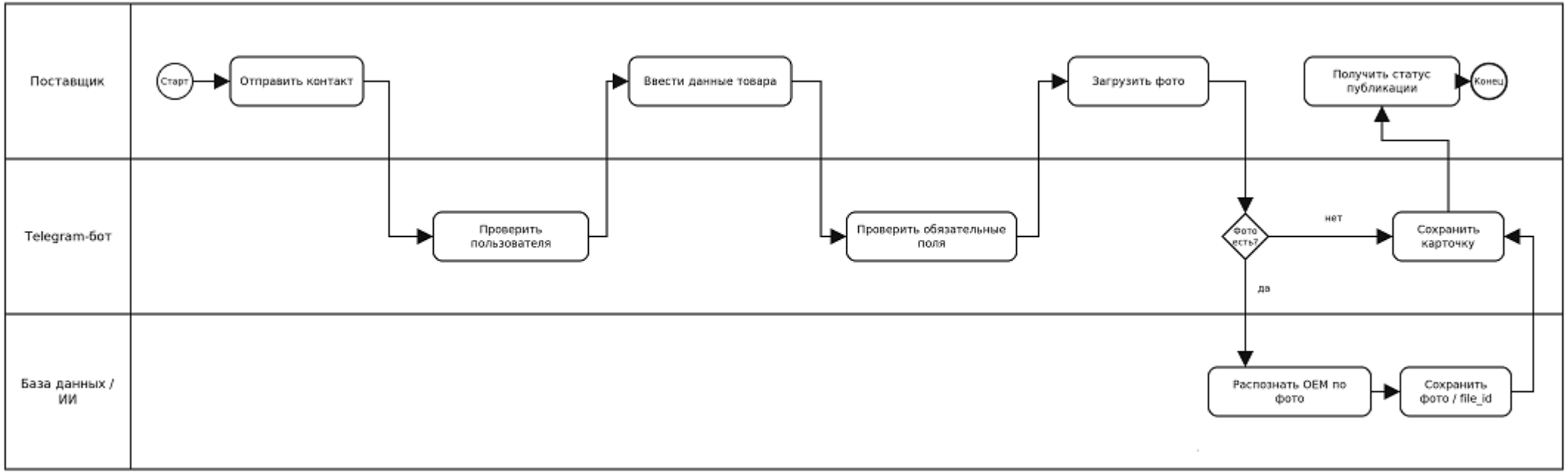


Рисунок 3.2 — BPMN-процесс добавления автозапчасти поставщиком

Формат Telegram-диалога удобен тем, что сложная форма разбивается на короткие шаги: название → OEM → тип → цена → фото → сохранение.

BPMN: поиск автозапчасти клиентом

Клиент может идти по обычному поиску или отправить фотографию маркировки

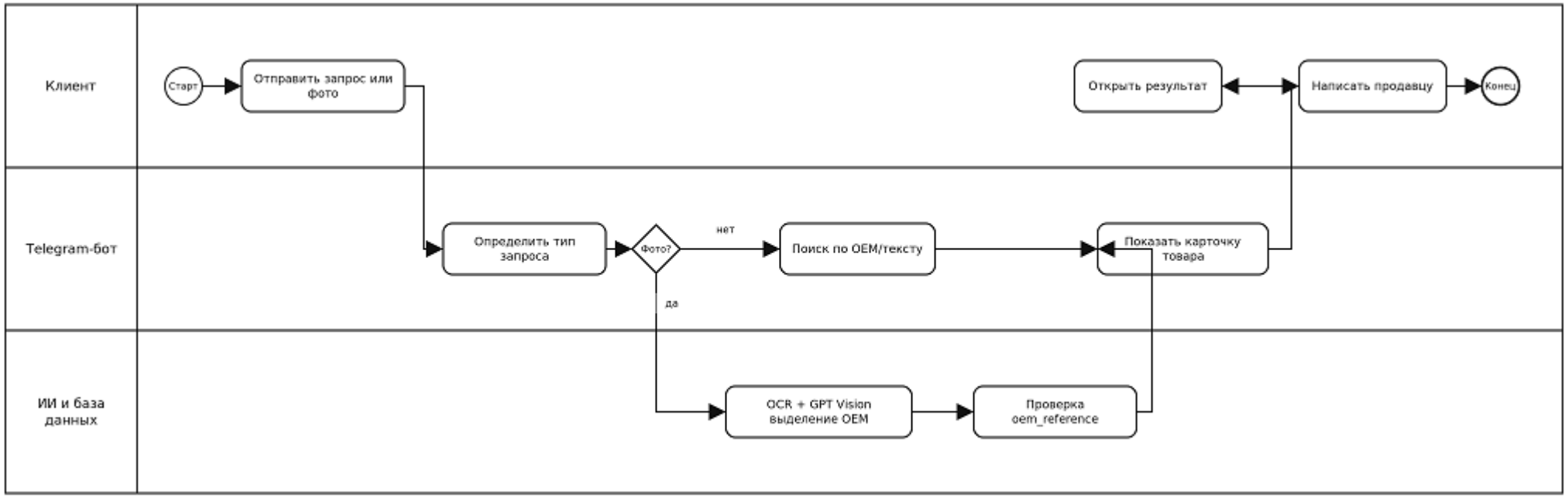


Рисунок 3.3 — BPMN-процесс поиска с ИИ-распознаванием OEM

- если запрос текстовый — поиск сразу идет по БД
- если запрос с фото — сначала OCR/GPT выделяет OEM
- после открытия карточки клиент получает фото, цену и контакт продавца

Алгоритм ИИ-распознавания OEM

ИИ помогает превратить фотографию маркировки в поисковый запрос

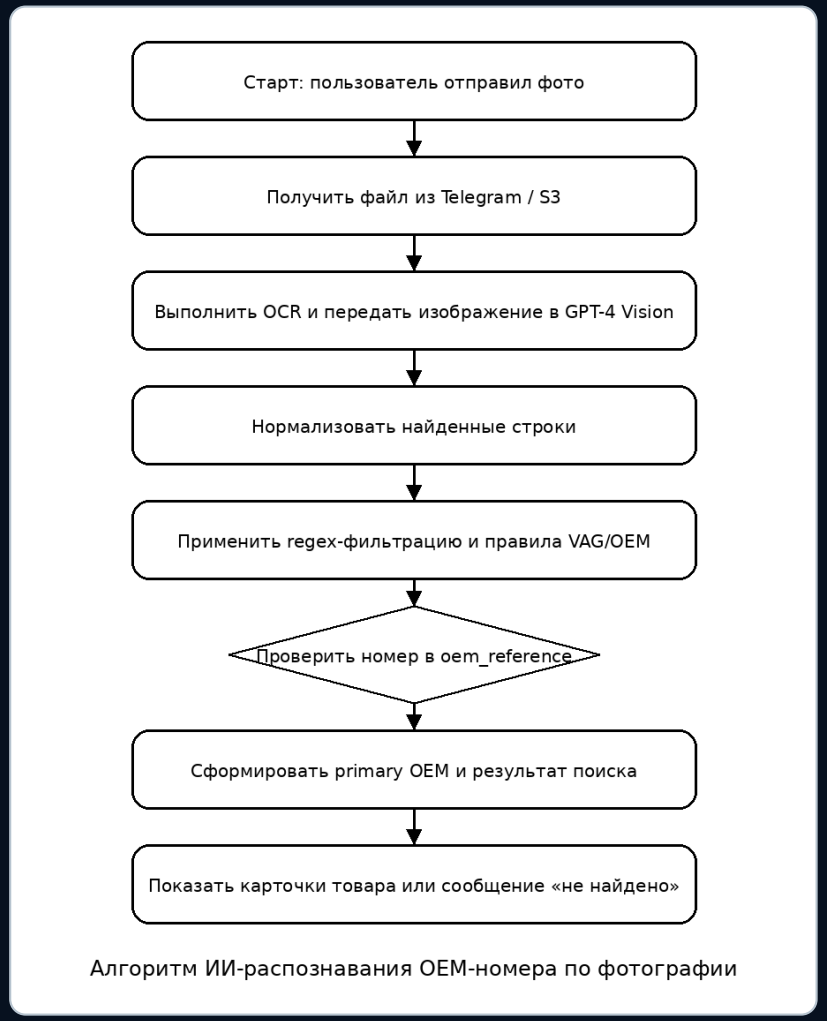


Рисунок 1.1 / 3.10 — Алгоритм ИИ-распознавания OEM-номера по фотографии

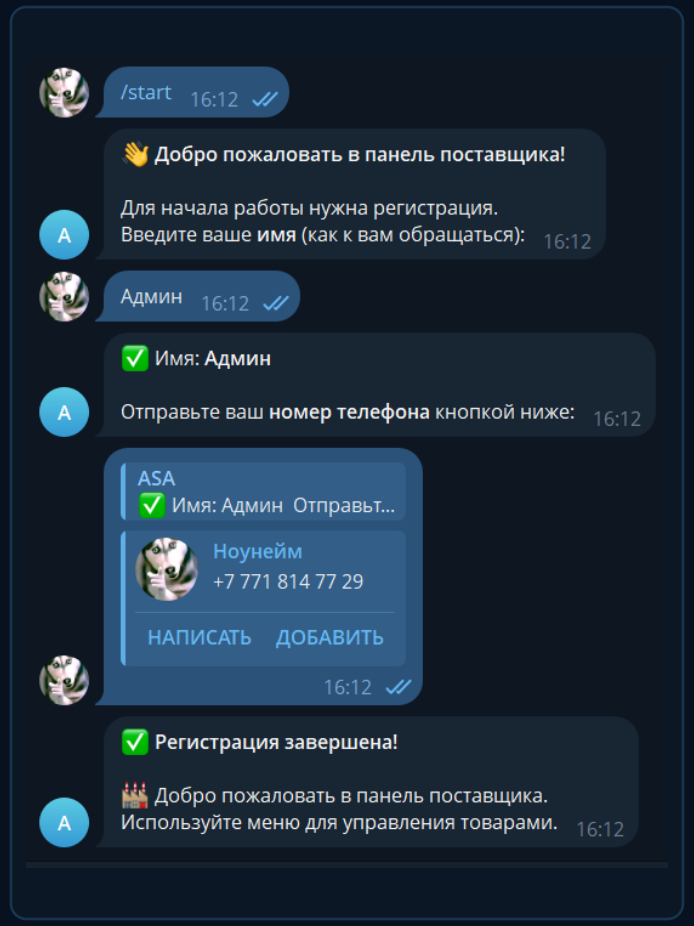
ЛОГИКА ОБРАБОТКИ

- бот получает фото от клиента или поставщика
- изображение передается в OCR / GPT Vision
- модель возвращает primary_oem и варианты
- номер нормализуется и ищется в PostgreSQL
- если уверенность низкая — бот просит ручной ввод

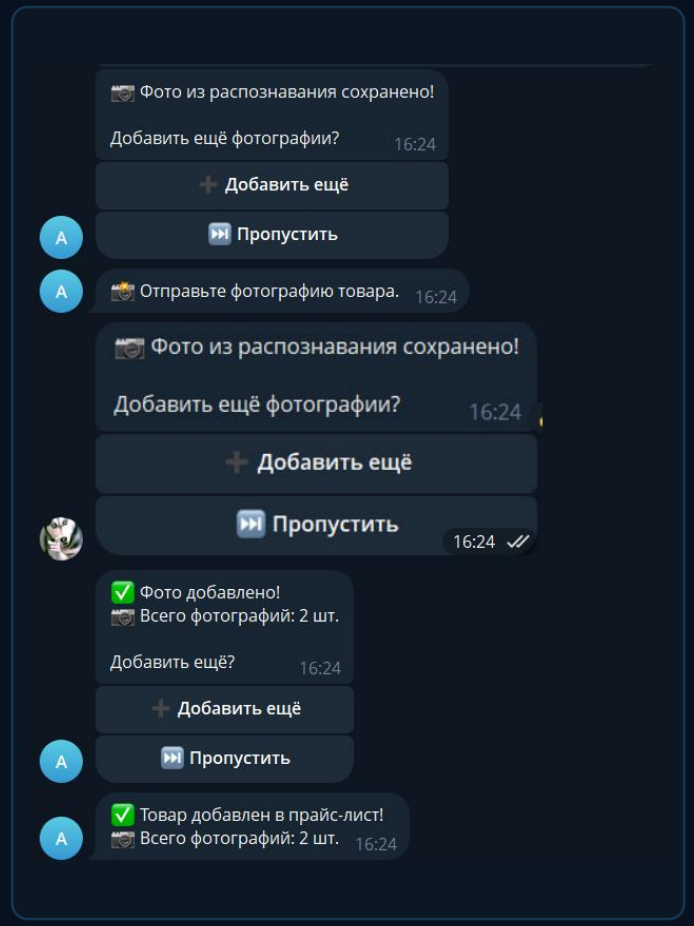
ИИ не заменяет эксперта по совместимости. Он ускоряет первичное извлечение OEM.

Реальные экраны: сценарий поставщика

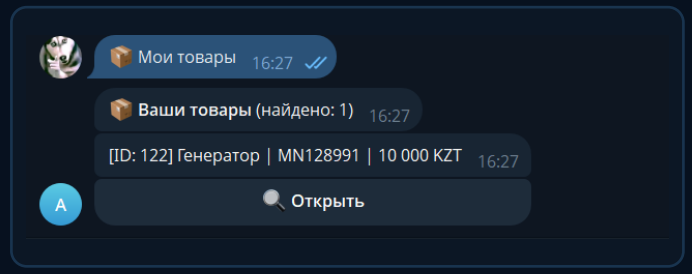
Скриншоты из ВКР показывают регистрацию, добавление фото и раздел собственных товаров



Регистрация поставщика



Добавление фотографий

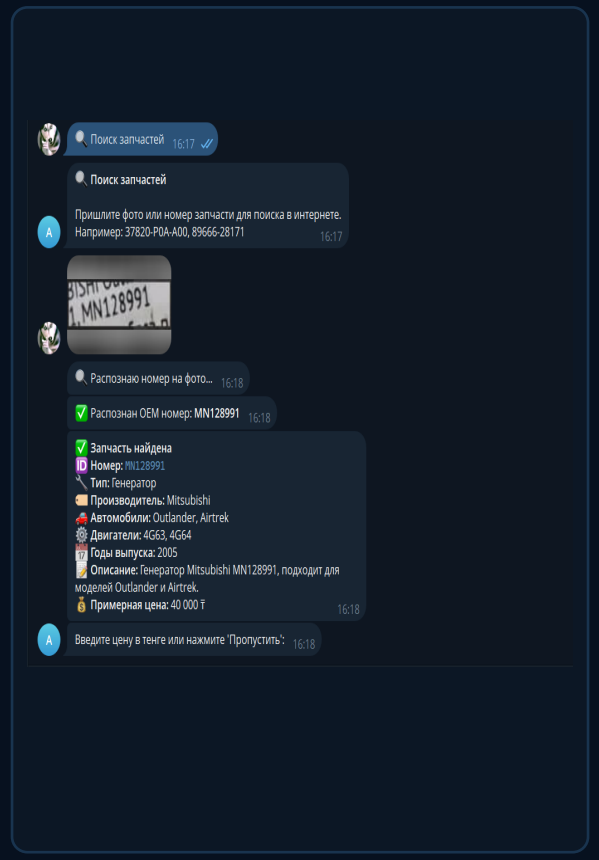


Раздел «Мои товары»

Поставщик работает не с таблицей, а с коротким пошаговым диалогом. Это снижает барьер входа и дисциплинирует структуру карточки.

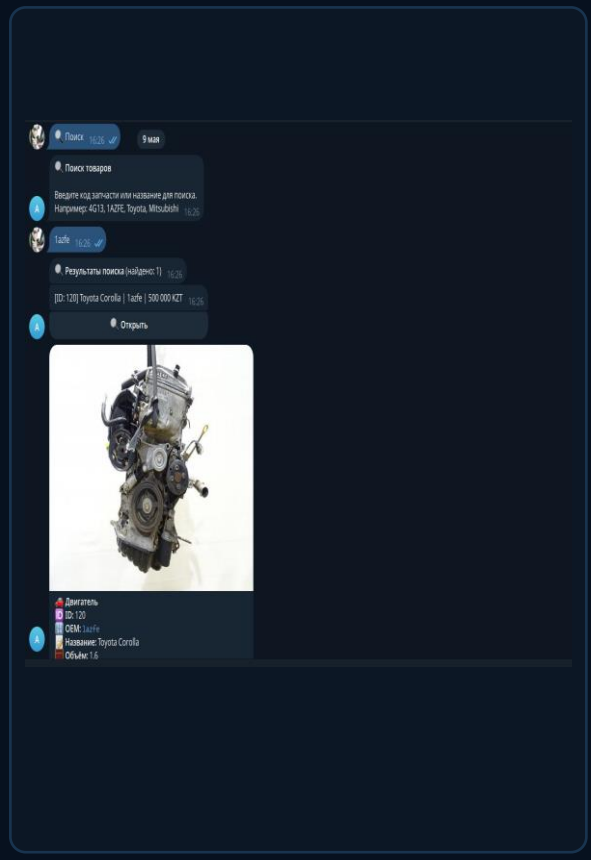
Реальные экраны: сценарий клиента

Поиск, распознавание, открытие карточки и переход к продавцу внутри Telegram

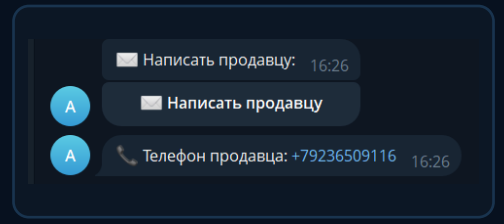


1 · Фото → OEM

2 · Результат поиска



3 · Карточка товара



4 · Контакт продавца

Клиент получает не просто ответ в чате, а структурированную карточку с фото, OEM, ценой и прямым действием — написать продавцу.

Тестирование и контрольный пример

Проверены основные сценарии регистрации, добавления товара, поиска, карточки и ограничения доступа

Рисунок 3.15 — контрольный пример поиска по фотографии и карточке товара

ПРОВЕРЕННЫЕ СЦЕНАРИИ

- запуск /start и сохранение контакта
- добавление товара и нескольких фото
- поиск по OEM и названию
- ИИ-распознавание фото с маркировкой
- открытие карточки и связь с продавцом
- запрет редактирования чужого товара

Эффективность и итог работы

ВКР завершена проектным решением и прототипной логикой Telegram-бота

125 000 ₽
расчетная себестоимость разработки прототипа

15 000 ₽
ожидаемый месячный эффект от экономии ручного труда

8–9 мес.
ориентировочный срок окупаемости

30 ч/мес.
условная экономия ручной обработки запросов

Главный результат

Спроектирован Telegram-бот учета автозапчастей с клиентским и поставщицким сценариями, общей базой PostgreSQL, хранением фотографий и модулем ИИ-распознавания OEM по фотографии.

Перспективы развития: фильтры по марке, модели и году выпуска, рейтинг продавцов, аналитика запросов, уведомления и интеграция с внешними каталогами автозапчастей.

Спасибо за внимание