Я хочу, чтобы ты выступил в роли опытного технического и продуктового аналитика. Вот описание моего проекта:

**ЖильеGO** — это чат‑бот для посуточной аренды квартир, который объединяет основные сценарии поиска, бронирования, управления и аналитики в одном удобном интерфейсе.

## **1. Цель проекта**

Создать MVP‑версию Telegram‑бота, позволяющего пользователям быстро и просто найти свободную квартиру на сутки, забронировать её с оплатой через Kaspi, отслеживать свои брони и оставлять отзывы. При этом владельцы и риелторы получают инструмент для оперативного ввода данных и мониторинга ключевых бизнес‑метрик.

## **2. Пользовательские сценарии**

1. **Приветствие и главное меню**
   * Бот приветствует: «Привет! Я ЖильеGO — помогу быстро найти и забронировать квартиру на сутки.»
   * В меню — четыре кнопки:  
     1. Поиск квартир
     2. Мои бронирования
     3. Статус текущей брони
     4. Помощь
2. **Поиск квартир**
   * Пошаговый фильтр:  
     1. Выбор города (Астана, Алматы, Шымкент)
     2. Выбор района (динамически из БД)
     3. Класс жилья (Комфорт / Бизнес / Премиум)
     4. Количество комнат (1 / 2 / 3 / 4+)
   * Результаты выдаются карточками с шестью фотографиями, кратким описанием, ценой за сутки и кнопками «Забронировать» / «Следующая».
3. **Процесс бронирования**
   * Запрос дат и времени заезда/выезда (по умолчанию «сегодня → завтра»)
   * Расчёт стоимости: цена за сутки × число дней
   * Оплата через Kaspi — при успешной оплате бот выдаёт все данные (адрес, подъезд, код домофона/сейфа, телефон риелтора) и помечает бронь как «Забронировано».
4. **Личный кабинет**
   * «Мои бронирования» — история завершённых сделок
   * «Статус текущей брони» — список активных, оплаченных броней с кнопкой «Детали»
   * Автоматический перенос в историю после выезда
5. **Отзывы**
   * На карточке жилья отображается средний рейтинг и счётчик
   * После выезда бот напоминает оставить отзыв (оценка + текст, а также возможность прикрепить фото)
   * Просмотр комментариев по 10 штук с пагинацией

## **3. Административный контур**

* **Ввод данных**
  + Риелторы через бота, суперпользователи — через Django‑админку
* **Управление каталогом квартир**
  + Поля: 6 фото, описание, цена, класс, город, район, адрес, статус
  + Жизненный цикл статусов: «Свободна» → «Забронирована» → «Занята» → «Свободна»
* **Безопасность и логирование**
  + Шифрование кодов домофонов/сейфов и аудита доступа
  + Логирование платежей Kaspi (ID, сумма)

## **4. Статистика и отчётность**

1. **Роли и права доступа**
   * **Админ‑риелтор** — видит данные по своим квартирам
   * **Суперадмин** — сводные отчёты по всем регионам и администраторам
2. **Ключевые метрики**
   * Доход, количество бронирований и отмен
   * Коэффициент загрузки (отношение занятых ночей к доступным)
   * ТОП‑5 квартир и гостей по разным критериям
3. **Инструменты отчётности**
   * В боте: интерактивные кнопки для выборов периодов, выгрузка CSV
   * В Django‑панели: полноразмерные графики, фильтры, экспорт, heat‑map занятости
4. **Дополнительные возможности**
   * Автоматические уведомления при загрузке < 40 %
   * Оценка надёжности гостей (KO‑фактор) и предоплата для рисковых
   * Ежемесячные PDF‑отчёты суперадмину с диаграммами и анализом «план‑факт»

**Итог:**   
 ЖильеGO сочетает в себе простой пользовательский путь от поиска до оплаты и мощный административный бэк‑офис для управления квартирами и аналитики. Такой подход позволит быстро запустить MVP и постепенно расширять функционал в соответствии с реалиями рынка посуточной аренды.

\*\*Твоя задача:\*\*

1. \*\*Идейный план\*\*

- Оценить видение и общую концепцию проекта

- Проанализировать целевую аудиторию и ключевые пользовательские гипотезы

- Выявить недостающие элементы:

- описание MVP и приоритезация фич

- сценарии использования (user stories) и эпики

- конкурентный анализ и уникальное торговое предложение

- бизнес-модель и дорожная карта развития

2. \*\*Технический план\*\*

- Оценить существующую архитектуру (инфраструктуру, схемы данных, API)

- Проанализировать подбор технологического стека (Frontend, Backend, СУБД, DevOps)

- Указать пробелы по:

- масштабируемости и отказоустойчивости

- безопасности и соответствию стандартам

- интеграциям с внешними сервисами

- процессам CI/CD, покрытию тестами и мониторингу

- документации кода и инфраструктуры

3. \*\*Результат\*\*

- Сформировать структурированный отчёт в формате Markdown с разделами «Идейный план» и «Технический план»

- В каждом разделе перечислить конкретные недостающие элементы

- Для каждого пункта предложить приоритетные задачи и рекомендации по следующему шагу