

UX-сценарий и структура мок-дизайна Telegram Mini App «Юрист Сейчас»

UX-сценарий и структура мок-дизайна Telegram Mini App «Юрист Сейчас»

1. Пользовательский flow (MVP)

Шаг	Экран / действие	Суть UX-решения
1	/start → Welcome-WebApp	Мини-баннер «Найдите юриста за 1 минуту». Кнопка «Подобрать юриста». Чек-бокс «Мне 18+ и я резидент РФ».
2	Категории / быстрый фильтр	Горизонтальный карусель-чипов с 6 основными уголовными статьями (158, 228, 159, 264, экономика, 213). Кнопка «Все категории».
3	Настройка поиска (модалка)	Поля:• Город (по GeoIP, с ручным выбором).• Диапазон цены — два input или слайдер.• Чек-бокс «Показывать юристов с гонораром» (если включён — отображается второе поле «Мин. гонорар»).
4	Список юристов	Карточки в вертикальном списке:• Фото/инициалы, имя, стаж.• Специализация (иконка статьи).• Цена: «от 3 000 ₽/час» или «гонорар от 25 000 ₽».• Индикатор онлайн-статуса.• Кнопка «Чат».
5	Чат	Обычный чат Telegram. Первое авто-сообщение бота: «Опишите ситуацию». Кнопка «Закончить консультацию» по согласованию сторон.
6	Отзыв	После нажатия «Закончить...» бот присылает инлайн-кнопки 👍 / 👎 и поле «Оставить текстовый отзыв».

Почему так: мы убрали регистрацию — Telegram WebApp уже знает ID пользователя. Вся транзакция и договорённости происходят в чате, поэтому хранить персональные данные на сервере не требуется.

2. Карта экранов (Wireframes)

1. **Welcome / Start**
2. **Category Picker** (главный экран)
3. **Filters Modal**
4. **Lawyer List**
5. **Lawyer Profile Popup** (tap по карточке)
6. **Chat – User ↔ Lawyer**
7. **Feedback Sheet**

Для Figma достаточно low-fi wireframes на один «iPhone 14 Pro» фрейм — Telegram WebApp унаследует скролл и safe-area, что соответствует гайдлайнам Telegram («mobile-first», 60 fps, соблюдение safe area) (Telegram).

3. Навигация и паттерны Telegram Mini Apps

Паттерн	Реализация
Запуск	<code>web_app</code> inline-кнопка в ответ на <code>/start</code> .
Темы	Используем <code>Telegram.WebApp.themeParams</code> → динамически меняем primary/secondary цвета (поддержка Day/Night) (Telegram).
Header	<code>WebApp.setHeaderColor()</code> — чётко-синий при Day, тёмно-графитовый при Night.
Back	Внутренние модальки/листинги закрываются <code>WebApp.BackButton</code> .
Хаптик	<code>Telegram.WebApp.HapticFeedback.impact('light')</code> при выборе фильтра.

4. Дизайн-система (база)

Токен	Значение
Font	Inter 14 px base / 18 px sectionTitle / 24 px banner.
Primary	<code>themeParams.accent_text_color</code> * — Telegram автоматически даст фирменный.
BG / Card	Используем <code>background_color</code> и <code>secondary_bg_color</code> для карточек.
Elevation	Тень 0 1 6 rgba(0,0,0,.06) для карт, плавный hover (desktop).
Radius	12 px карточки, 20 px модалки.

- В пользовательских темах Telegram сам меняет accent, поэтому вручную перекрашивать не нужно – достаточно слушать события `themeChanged`.

5. Компоненты

- **ChipButton** (категория) — svg-иконка статьи + label, scroll-snap.
- **PricelInput** — два текстовых поля с плейсхолдерами «от...», «до...».
- **LawyerCard** — Lazy-loaded фото (blurhash), имя, стаж → rating bar (!) и СТА «Чат».
- **ModalSheet** — full-height, drag → dismiss.
- **FeedbackBar** — inline 👍/👎 + textarea (autosize).

6. Технический стек прототипа

Слой	Технология
Front-end	React + Vite + telegram-mini-apps UI (или TelegramUI kit) (GitHub)
State	Zustand / Redux Toolkit; хранение в browser memory (до 50 MB).
Design → Code	Figma Tokens → Style-dictionary → CSS Vars.
Hosting	Netlify / Vercel (https), домен вида <code>https://legalbot.app/...</code> .
Bot	Python 3.12 + aiogram 4 (WebApp handler).
Storage	Supabase (PostgreSQL) – профили юристов, отзывы.
Rating moderation	Простая админ-панель (Supabase Auth) для подтверждения отзывов.

7. План Sprint 0 (4 недели)

Неделя	Deliverable
1	User-flow → low-fi wireframes → согласование ТЗ.
2	High-fi mock (Figma components + прототип кликов).
3	Реализация React-каркаса + кастомные чипы/карты.
4	Интеграция с Telegram WebApp SDK, тест на TestFlight + Play Internal.

Точка «definition of ready»: кликабельный прототип + базовый фронт, который открывается через web_app в Telegram, фильтрует юристов из мок-JSON и позволяет открыть фейковый чат.

8. Риски и next steps

1. **Нагрузка модерации** — в уголовной практике важно верифицировать адвокатов; заложить time-buffer на процедуру «проверка ордера + диплома».
2. **Платежи** — если позже понадобится escrow/комиссия, придётся внедрять Telegram Stars или внешний acquiring – это отложено за рамки MVP.
3. **Безопасность данных** — не храним переписку, но логируем события (без текста) для KPI.

Что дальше?

- Подтвердите, подходит ли такая структура экранов и набор компонентов.
- Пришлите, если есть, фирменные цвета/шрифт — иначе возьмём Telegram accent.
- После подтверждения я соберу Figma-файл и экспортирую дизайн-токены в код.