**Подробный документ технического дизайна для eSIM Mini App (фронтенд)**

**1. Общий обзор проекта**

**Технологический стек:**

* **Фронтенд:** React.js (функциональные компоненты, хуки) и CSS для стилизации.
* **Интеграция:** Telegram WebApp API для аутентификации и взаимодействия с пользователями; данные (страны, пакеты) загружаются из JSON-файлов.
* **Основной функционал:**
  + Покупка и управление eSIM-пакетами.
  + Отображение FAQ, гайдов и информации об устройствах.
  + Интерактивное руководство по настройке eSIM.

**Управление состоянием и навигация:**

* Глобальное состояние (activePage, activeTab, информация о пользователе) хранится в корневом компоненте.
* Навигация осуществляется с помощью вкладок и условного рендеринга контента.

**2. Детальное описание файлов**

**2.1 App.js (Основной контейнер приложения)**

**Обязанности:**

* Является корневым компонентом, управляющим глобальным состоянием:
  + **activePage:** Определяет, отображается ли главная страница или один из табов.
  + **activeTab:** Отслеживает активную вкладку (например, «buy», «myesims», «guides», «faq», «devices»).
  + **user:** Содержит данные аутентифицированного пользователя (получаются через Telegram WebApp API и authService).
  + **isMobile:** Флаг, определяющий, используется ли мобильное устройство (на основе ширины окна).
  + **Состояния для меню и блока помощи:** Контролируют видимость мобильного меню и выпадающего списка справки.
* Сохраняет значения activePage и activeTab в localStorage для сохранения состояния при обновлении страницы.
* Интегрируется с Telegram WebApp API:
  + Проверяет наличие объекта Telegram в браузере.
  + Выполняет автоаутентификацию через authService.
* Рендерит следующие разделы:
  + **Шапка (Header):** Отображает логотип, навигационные кнопки, информацию о пользователе и элемент выбора языка (планируется добавить).
  + **Вкладки (Tabs):** Рендерит кнопки для навигации между Home, Buy eSIM, My eSIMs и другими разделами.
  + **Основной контент (Main Page):** Содержит заголовок с выделением «eSIM Unlimited», подзаголовок, основную кнопку CTA, секцию «Наши преимущества» и интерактивное руководство по настройке eSIM.
  + **Дополнительные виды:** В зависимости от activeTab отображаются компоненты BuyESIM, FAQTab, Guides, EsimSupportedDevices и MyESIMs.

**Ключевые функции:**

* **Управление состоянием и сохранение:**  
  Используются хуки useState и useEffect для локального состояния и синхронизации с localStorage.
* **Обработчики событий:**
  + toggleHelpCenter и toggleMenu – управление видимостью выпадающих списков.
  + handleTabChange(tab) – изменение активной вкладки и страницы.
  + handleManualLogin() и handleLogout() – управление процессом аутентификации.
* **Функции рендера:**  
  Разделены на renderHeader, renderTabs, renderMainPage и renderContent для лучшей модульности.
* **Адаптивность:**  
  Обработчик событий окна обновляет флаг isMobile при изменении размеров экрана.

**Интеграция:**

* Импортируются дочерние компоненты (FAQTab, Guides, EsimSupportedDevices, BuyESIM, MyESIMs).
* Используется authService для аутентификации через Telegram.
* Внутри компонента определён интерактивный компонент для руководства по настройке eSIM (можно позже вынести в отдельный файл).

**2.2 BuyESIM.js**

**Обязанности:**

* Отображает доступные eSIM-пакеты для покупки.
* Фильтрует пакеты по странам (Local), регионам (Regional) и глобальным параметрам (Global).
* Управляет состоянием для:
  + Активной вкладки (например, local, regional, global).
  + Выбранной страны или региона.
  + Поискового запроса и количества отображаемых стран.
  + Фильтрации и сортировки пакетов (на основе цены, объёма данных и т.д.).

**Ключевые особенности:**

* **Загрузка данных:**  
  Используются хуки useEffect для загрузки данных из JSON-файлов (countries.json, countryPackages.json, regionalPackages.json, globalPackages.json).
* **Фильтрация и сортировка:**  
  Логика фильтрации на основе выбранных параметров, сортировка пакетов по цене.
* **Интерактивность:**  
  Кнопки для выбора страны, региона и обновления фильтров.

**Интеграция:**

* Данные о странах и пакетах получаются динамически.
* Компонент взаимодействует с родительским через обновление состояния (например, очистка фильтров при смене вкладки).

**2.3 FAQ.js, Guides.js, EsimSupportedDevices.js, MyESIMs.js**

**Обязанности:**

* Рендерят соответствующий контент:
  + **FAQ.js:** Список часто задаваемых вопросов и ответов.
  + **Guides.js:** Инструкции и гайды по использованию сервиса.
  + **EsimSupportedDevices.js:** Список устройств, поддерживающих eSIM.
  + **MyESIMs.js:** Личный кабинет пользователя с информацией о активированных eSIM.

**Ключевые особенности:**

* **Отображение контента:**  
  Контент может быть статичным или динамически загружаться из JSON/базы данных.
* **Интерактивность:**  
  Возможна реализация выпадающих списков, фильтрации, поиска.

**Интеграция:**

* Эти компоненты рендерятся в App.js в зависимости от выбранной вкладки.
* Каждый компонент имеет собственный CSS для стилизации (FAQ.css, Guides.css и т.д.).

**2.4 auth.js (Сервис аутентификации)**

**Обязанности:**

* Управляет аутентификацией через Telegram WebApp API.
* Предоставляет методы:
  + **Аутентификации:** Проверка initData от Telegram и получение данных пользователя.
  + **Ручного логина и логаута:** Возможность входа и выхода пользователя.

**Ключевые особенности:**

* **HTTP-запросы:**  
  Используются запросы к бэкенду для валидации Telegram данных.
* **Обработка ошибок:**  
  Функции возвращают данные пользователя или ошибки для дальнейшей обработки в App.js.

**Интеграция:**

* Вызывается из App.js для автоматической и ручной аутентификации.
* Полученные данные используются для персонализации шапки и управления сессией.

**2.5 index.js**

**Обязанности:**

* Является точкой входа в React-приложение.
* Рендерит корневой компонент (MiniApp) в DOM.
* Может использовать глобальные провайдеры (например, Context API) при необходимости.

**Ключевые особенности:**

* **Инициализация:**  
  Импортируются глобальные CSS и настраивается рендеринг через ReactDOM (или createRoot для React 18+).

**Интеграция:**

* Связывает React-приложение с HTML-файлом (public/index.html).

**3. Глобальная конфигурация и переменные окружения**

Все конфигурационные параметры (например, URL для веб-приложения, Telegram токен, API базы данных) должны быть вынесены в глобальные переменные или загружены из файла .env.  
Пример файла .env:

ini

Копировать

REACT\_APP\_TELEGRAM\_BOT\_TOKEN=YOUR\_TELEGRAM\_BOT\_TOKEN

REACT\_APP\_WEBAPP\_URL=https://your-mini-app-url.com

REACT\_APP\_API\_BASE\_URL=https://your-api-endpoint.com

В коде доступ к этим переменным осуществляется через process.env.REACT\_APP\_<VARIABLE\_NAME>.

**4. Стилизация и адаптивность**

* **По новому макету**

**5. Поток данных и обработка событий**

**Поток данных:**

* **Инициализация:**  
  При загрузке index.js рендерится компонент MiniApp, который инициализирует глобальное состояние (activePage, activeTab, user) из localStorage.
* **Загрузка данных:**  
  Компоненты, такие как BuyESIM.js, используют хуки useEffect для загрузки данных из JSON-файлов и сохранения их в локальном состоянии.
* **Аутентификация:**  
  App.js вызывает методы из auth.js для автоматической аутентификации через Telegram WebApp API.
* **Сохранение состояния:**  
  Значения activePage и activeTab сохраняются в localStorage, чтобы сохранять пользовательские настройки при обновлении страницы.

**Обработка событий:**

* **Глобальные события:**  
  Обработчик resize обновляет состояние isMobile.
* **События клика:**  
  Обработчики в шапке и навигационных элементах обновляют состояние (например, переключение вкладок, открытие/закрытие меню).
* **События внутри компонентов:**  
  В BuyESIM.js – клики по странам, регионам и пакетам обновляют соответствующие состояния.
* **Аутентификация:**  
  Кнопки Login/Logout вызывают функции handleManualLogin() и handleLogout().

**6. Рекомендации по рефакторингу**

* **Модульность:**  
  Рекомендуется вынести компоненты, такие как интерактивное руководство по настройке eSIM (InteractiveSetupGuide), в отдельные файлы для улучшения читаемости и повторного использования.
* **Управление конфигурацией:**  
  Все константы (например, URL, токены) должны быть вынесены в отдельный конфигурационный файл (config.js) или загружены из .env.
* **Оптимизация состояния:**  
  При необходимости, рассмотрите использование Context API или Redux для управления глобальным состоянием, если проект станет более сложным.
* **CSS-модули/Styled Components:**  
  Рассмотрите возможность использования CSS-модулей или styled components для локализации стилей и избежания конфликтов.