

Схема сервис-ориентированной архитектуры (SOA)

1 Схема SOA в монолитном подходе

1.1 Описание подхода

Поскольку мы используем монолитную архитектуру, все функции реализованы внутри одного Android-приложения. Однако код организован по принципам SOA, разделён на логические модули (пакеты в Java), которые взаимодействуют через вызовы методов. Такой подход упрощает разработку и поддержку, сохраняя модульность.

1.2 Ключевые компоненты системы

- **Клиентское приложение (Android):** Монолитное приложение, выполняющее все функции. Включает пользовательский интерфейс (UI), бизнес-логику и доступ к данным.
- **Логические модули (сервисы внутри приложения):** Функциональность разделена на модули для лучшей организации кода.
- **База данных (SQLite):** Локальная база данных на устройстве для хранения данных.
- **Нейросеть:** Собственная нейросеть, реализованная на Java с использованием библиотеки DeepLearning4J.
- **Внешние сервисы:** Yandex MapKit для отображения карты.

1.3 Логические модули (сервисы внутри приложения)

- **User Module (Модуль пользователей):**
 - *Функции:* Хранение данных пользователя (`username`, `email`, `current_emotion`); обновление эмоции после анализа нейросетью.
 - *Таблицы:* `users`.
 - *Класс:* `UserManager`.
- **Map Module (Модуль карты):**
 - *Функции:* Получение списка точек (`nodes`) с координатами; построение маршрута между двумя точками (алгоритм Дейкстры).
 - *Таблицы:* `nodes`, `edges`.

- *Класс*: MapManager.
- **Place Module (Модуль мест):**
 - *Функции*: Получение списка категорий; фильтрация точек по категориям (например, кафе, достопримечательности).
 - *Таблицы*: place_category, node_category, nodes.
 - *Класс*: PlaceManager.
- **Landmark Module (Модуль достопримечательностей):**
 - *Функции*: Получение информации о достопримечательности; выбор текста в зависимости от эмоции пользователя; предоставление аудиофайлов.
 - *Таблицы*: landmarks, nodes.
 - *Класс*: LandmarkManager.
- **Excursion Module (Модуль экскурсий):**
 - *Функции*: Получение списка экскурсий; получение точек экскурсионного маршрута.
 - *Таблицы*: excursion_routes, excursion_nodes, nodes.
 - *Класс*: ExcursionManager.
- **Emotion Module (Модуль эмоций):**
 - *Функции*: Анализ эмоций пользователя (например, через фото); обновление current_emotion в User Module.
 - *Использует*: Собственную нейросеть на Java (DeepLearning4J).
 - *Класс*: EmotionAnalyzer.

1.4 Взаимодействие компонентов

- **Клиент (UI) → Логические модули**: UI (Activity/Fragment) вызывает методы модулей.
- **Логические модули → База данных (SQLite)**: Каждый модуль использует SQLiteOpenHelper для доступа к базе данных.
- **Логические модули → Нейросеть**: Emotion Module использует собственную нейросеть (DeepLearning4J) для анализа эмоций.
- **Логические модули → Внешние сервисы**: Map Module использует Yandex MapKit для отображения карты.
- **Логические модули → Логические модули**: Landmark Module вызывает UserManager для получения эмоции.

1.5 Схематическое представление SOA

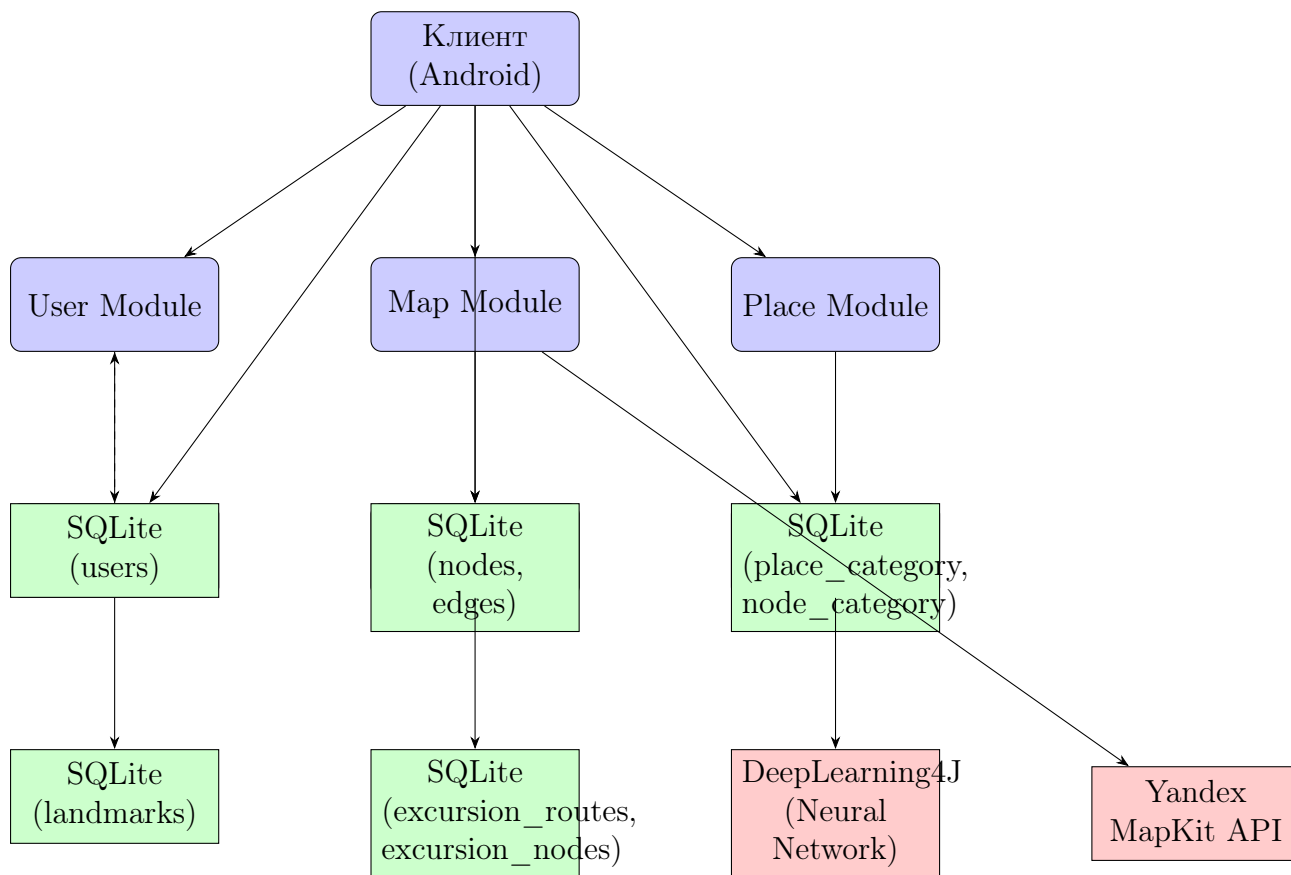


Рис. 1: Схема SOA