**Технический проект**

**Функциональная структура продукта**

Для описания структуры используется объектно-ориентированный подход и диаграмма компонентов, представленная на рисунке 1.

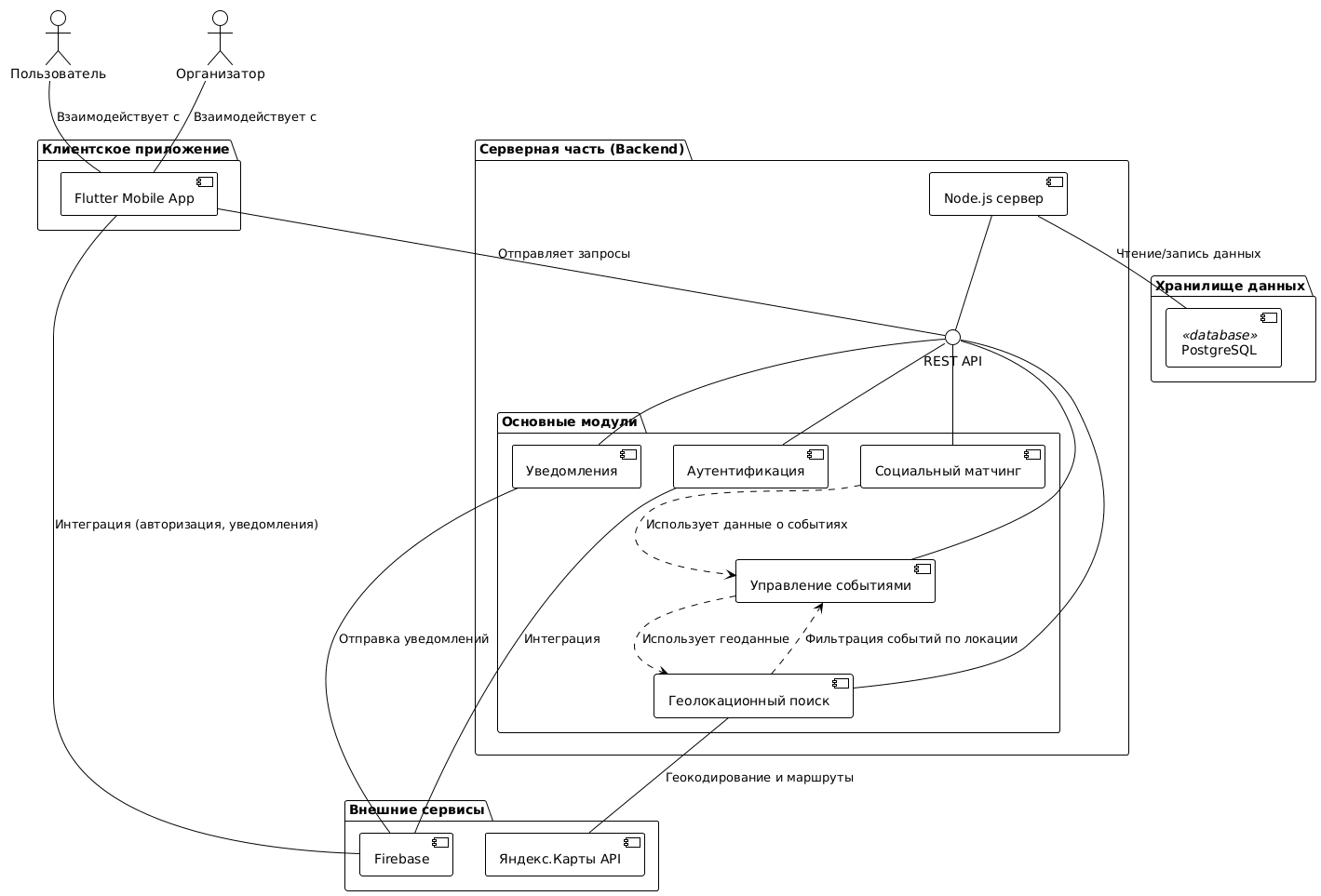


Рисунок – диаграмма компонентов

Описание Компонентов и Связей:

* Пользователь (Actor): Конечный пользователь системы — житель, турист или организатор мероприятий, взаимодействующий с приложением для поиска событий и социального взаимодействия;
* Мобильное приложение (Flutter): Клиентская часть на Flutter, обеспечивающая пользовательский интерфейс, работу с геолокацией, отображение событий на карте и отправку запросов к серверу через REST API;
* Backend сервер (Node.js): Серверная часть, реализующая бизнес-логику приложения — обработку запросов на поиск событий, управление профилями, реализацию матчинга, фильтрацию по геолокации и интеграцию с внешними сервисами;
* PostgreSQL: Основная база данных, хранящая информацию о пользователях, мероприятиях, участии, избранных событиях и матчах. Сервер взаимодействует с ней для чтения и записи данных;
* Firebase: Внешний сервис, используемый для аутентификации пользователей (Firebase Auth), отправки push-уведомлений (FCM) и хранения медиаконтента;
* Яндекс.Карты API: Внешний картографический сервис, обеспечивающий геокодирование адресов, отображение карты событий и построение маршрутов в мобильном приложении.

**Спецификация процессов**

Для спецификации процессов были составлены следующие диаграммы:

* Диаграмма последовательности (представлена на рисунке 2);
* Диаграммы состояний (представлены на рисунках 3, 4, 5, 6);
* Диаграммы деятельности (представлены на рисунках 7, 8, 9, 10.

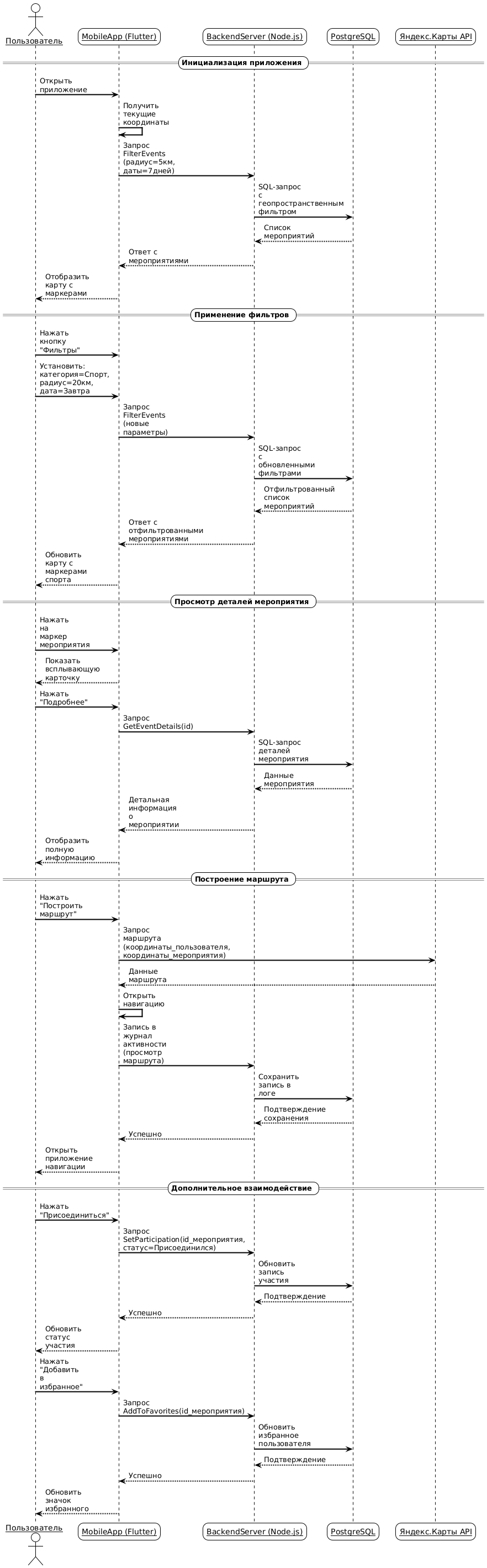


Рисунок – диаграмма последовательности

Диаграмма последовательности иллюстрирует основной сценарий взаимодействия пользователя с мобильным приложением геолокационной социальной сети.

Пользователь запускает приложение, которое определяет его текущее местоположение. На основе координат клиент отправляет запрос на сервер для получения мероприятий в радиусе 5 км на ближайшие 7 дней. Сервер обрабатывает геопространственный запрос к базе данных PostgreSQL и возвращает отфильтрованный список событий, которые отображаются на карте.

При изменении фильтров (например, категория "Спорт", радиус 20 км) клиент повторно обращается к серверу, который снова запрашивает актуальные данные из базы. При просмотре деталей мероприятия сервер получает полную информацию из БД и передает ее клиенту.

Для построения маршрута приложение использует Яндекс.Карты API, а факт просмотра маршрута логируется: клиент уведомляет сервер, который сохраняет запись в базе данных.

Также показаны действия по участию в мероприятии ("Присоединиться") и добавлению в избранное — в обоих случаях клиент взаимодействует с сервером, а тот — с базой данных для обновления статуса.

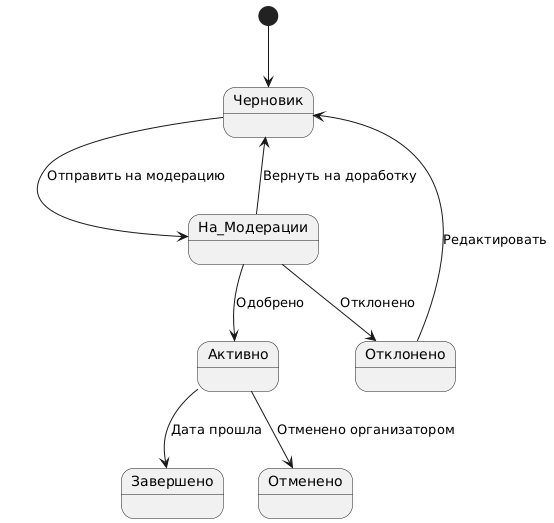


Рисунок – диаграмма состояний мероприятия

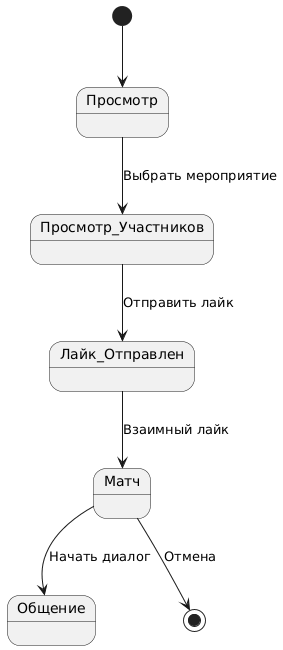


Рисунок – диаграмма состояний социального взаимодействия

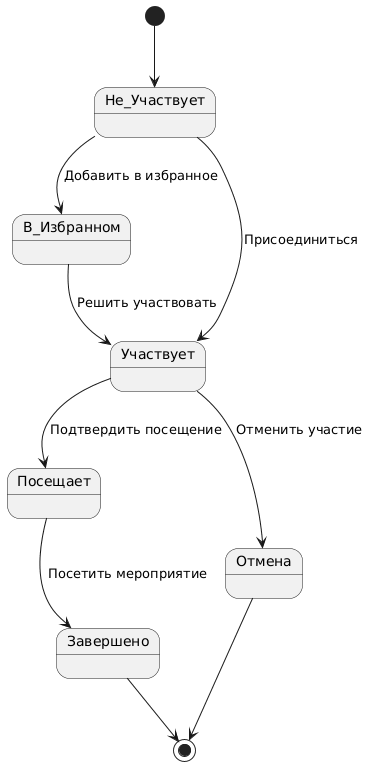


Рисунок – диаграмма состояний участия в мероприятии

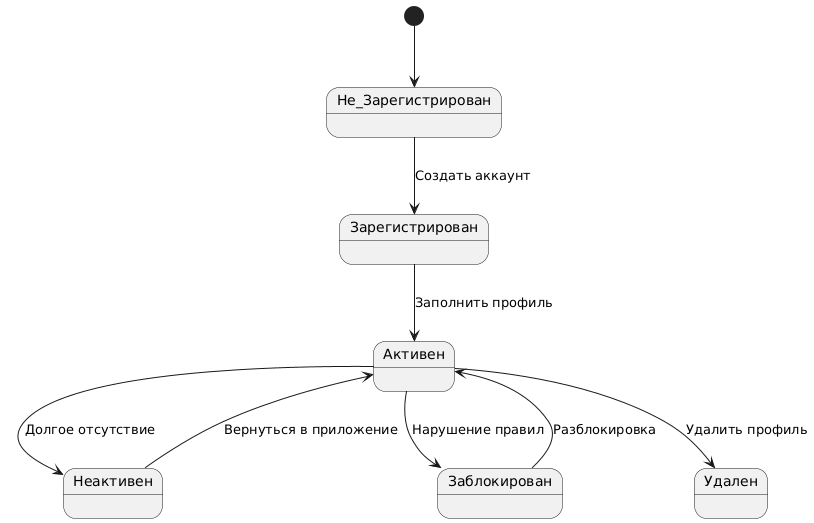


Рисунок – диаграмма состояний пользовательского профиля

Диаграммы состояний показывают жизненные циклы ключевых сущностей системы:

* мероприятие: черновик => на модерации => активно => завершено/отменено. Все события проходят обязательную модерацию перед публикацией.
* социальное взаимодействие: просмотр участников => отправка лайка => матч. матч возникает только при взаимных лайках пользователей, заинтересованных в одном мероприятии.
* участие в мероприятии: не участвует => в избранном/участвует => посещает => завершено. Пользователь может присоединиться к событию или добавить его в избранное для последующего просмотра.
* профиль пользователя: не зарегистрирован => зарегистрирован => активен. Профиль может быть временно заблокирован за нарушения или удален при длительной неактивности.

Все диаграммы отражают основные сценарии использования: поиск событий по геолокации, участие в них и социальное взаимодействие между участниками.

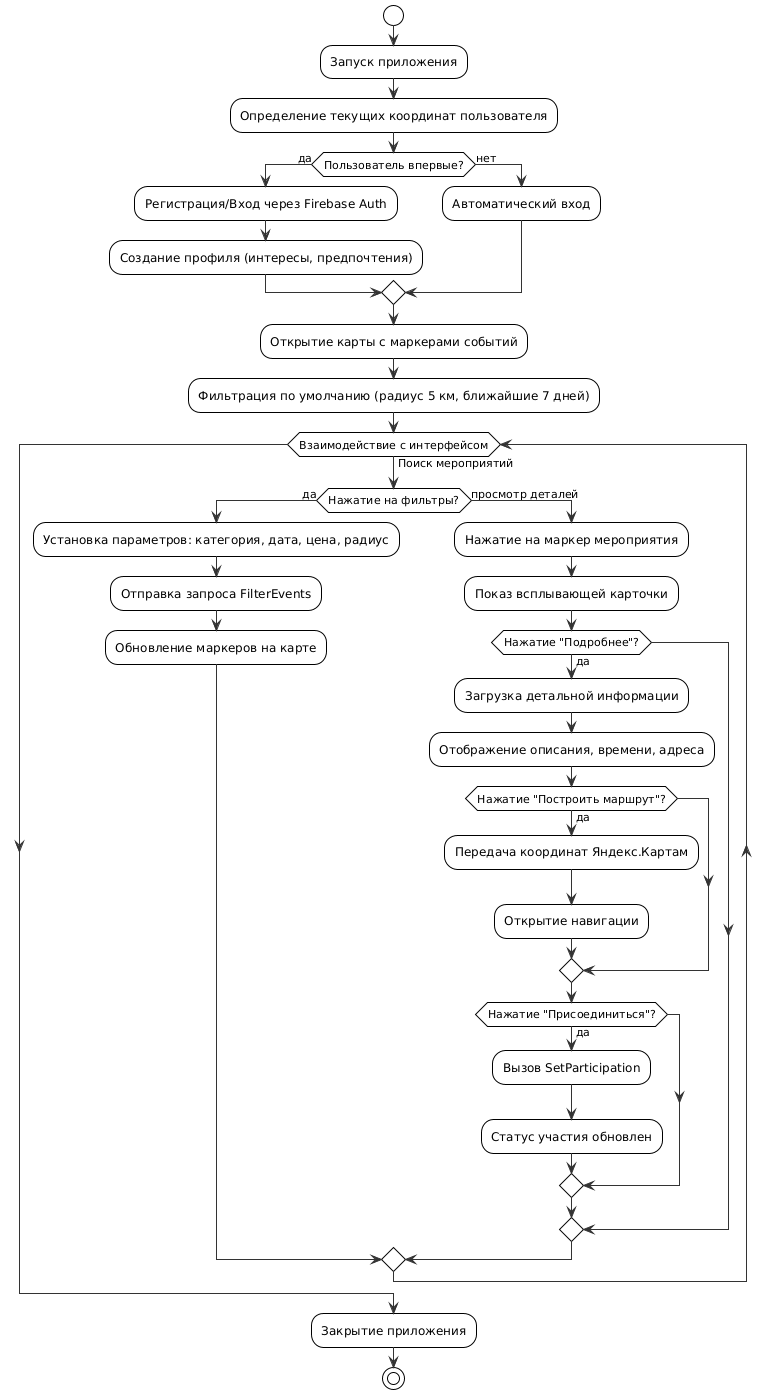


Рисунок – диаграмма деятельности поиска и просмотра мероприятий

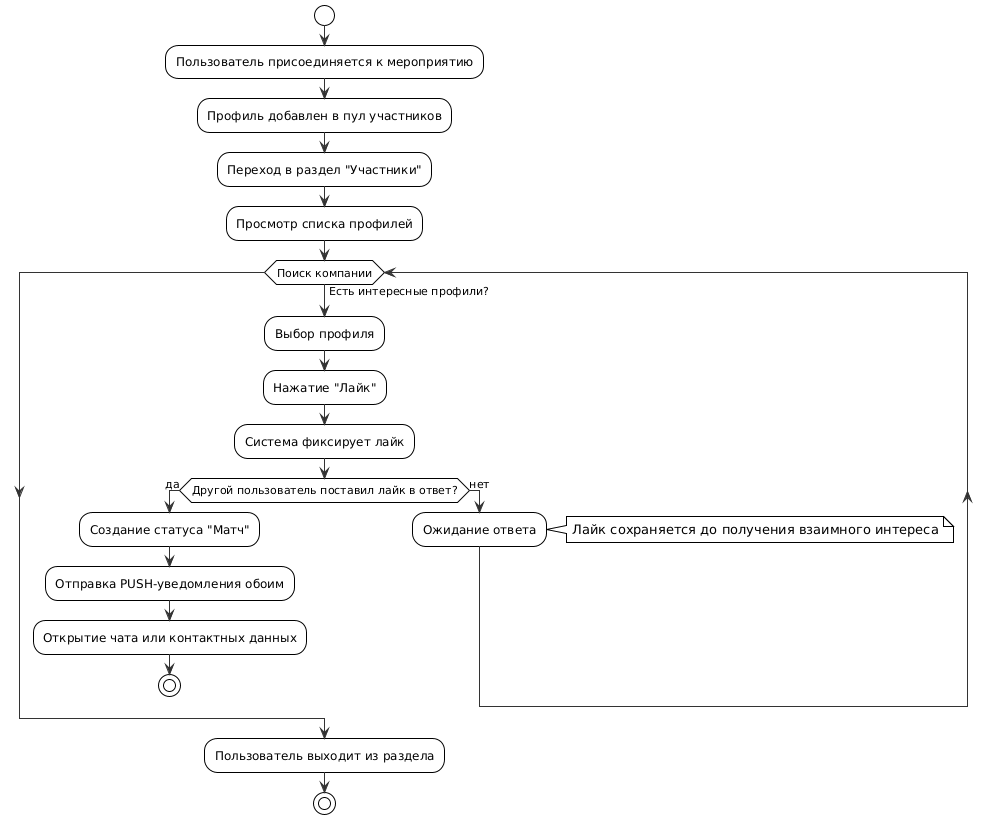


Рисунок – диаграмма деятельности процесса матчинга

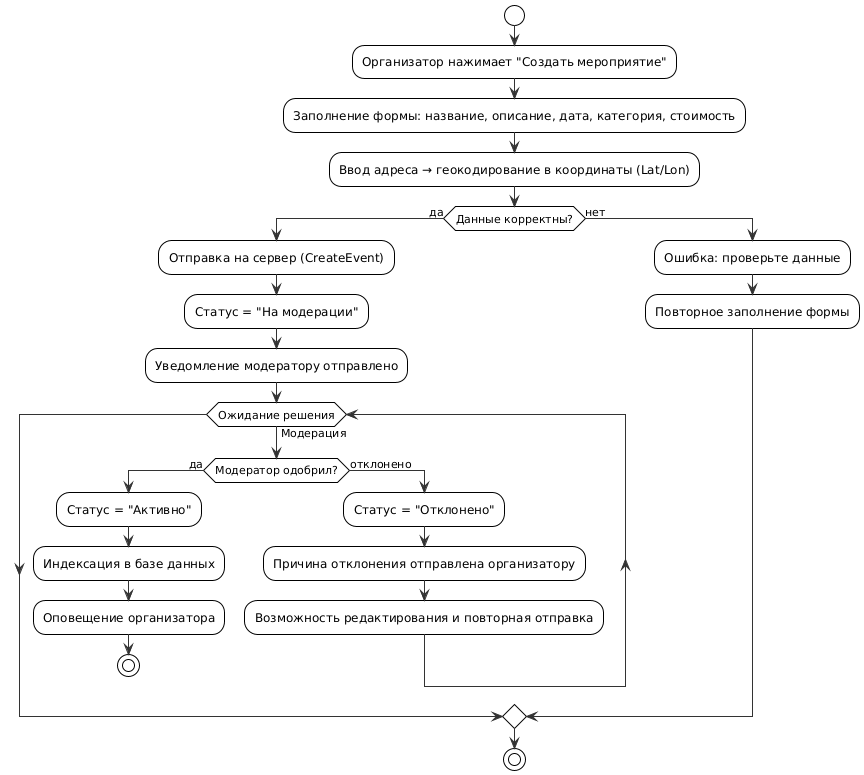


Рисунок – диаграмма деятельности создания и публикации мероприятия

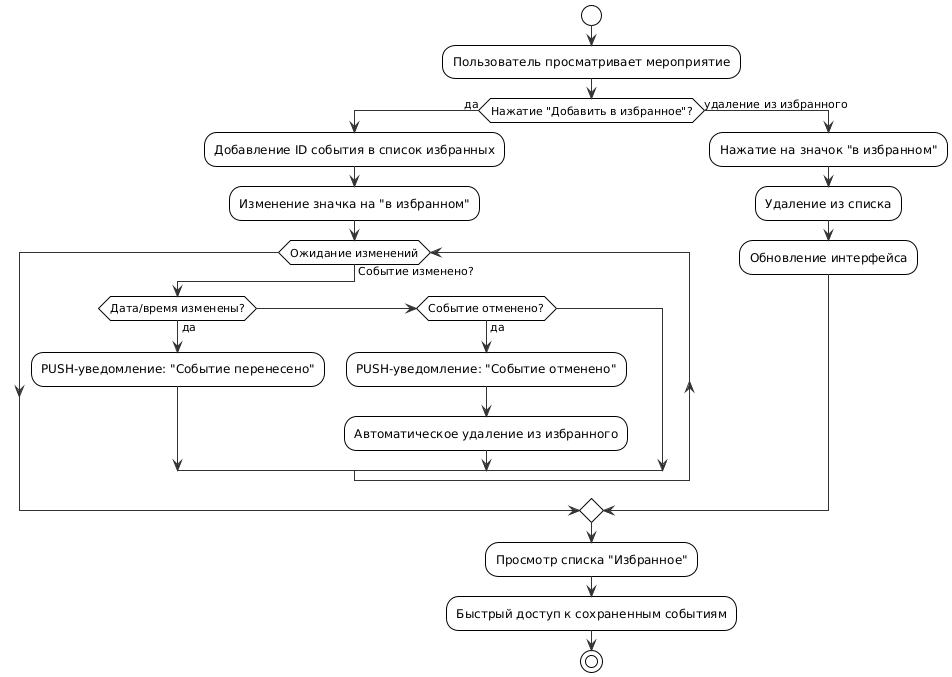


Рисунок - диаграмма деятельности управления избранным

Диаграммы деятельности описывают ключевые бизнес-процессы системы, показывая последовательность действий пользователей и системы.

Поиск и Просмотр Мероприятий

Пользователь запускает приложение, определяется его местоположение, и отображаются мероприятия в радиусе 5 км на ближайшие 7 дней. Пользователь может применять фильтры (категория, дата, цена, радиус), просматривать детали событий, строить маршрут через Яндекс.Карты и присоединяться к мероприятию. Процесс центрирован вокруг геолокационного поиска и персонализации.

Процесс Матчинга

После присоединения к мероприятию профиль пользователя попадает в пул участников. При просмотре списка он может поставить лайк другому участнику. Если лайк взаимный — система фиксирует «Матч» и отправляет обоим push-уведомление, открывая возможность для общения. Это основа социального взаимодействия в приложении.

Создание и Публикация Мероприятия

Организатор создает событие, заполняя форму с названием, датой, адресом и категорией. Адрес геокодируется в координаты. После отправки мероприятие переходит в статус «На модерации». Модератор проверяет контент: при одобрении событие становится активным, при отклонении — отправляется на доработку.

Управление Избранным

Пользователь добавляет интересующие события в избранное. Система автоматически отслеживает изменения (перенос, отмена) и уведомляет пользователя. При открытии списка избранного обеспечивается быстрый доступ ко всем сохраненным мероприятиям.

Все диаграммы подчеркивают интеграцию геолокационного поиска с социальной функцией, формируя единый цифровой опыт для поиска событий и компании.

**Архитектурно-структурное решение**

Для реализации проекта выбрана монолитная архитектура с интеграцией с внешними сервисами.

Система состоит из трех основных слоев:

* клиентский слой: мобильное приложение на Flutter, обеспечивающее кроссплатформенную поддержку iOS и Android. Приложение реализует пользовательский интерфейс для поиска мероприятий, геолокационного отображения, социального взаимодействия и управления профилем. Интеграция с Firebase обеспечивает аутентификацию, push-уведомления и хранение медиаконтента;
* серверный слой: бэкенд на Node.js с четким разделением на функциональные модули (аутентификация, управление событиями, геолокационный поиск, социальный матчинг, уведомления). Архитектура серверной части построена вокруг REST API;
* слой данных: PostgreSQL как основное хранилище данных с геопространственными расширениями (PostGIS) для эффективной обработки геолокационных запросов. Для интеграции с Firebase реализована отдельная прослойка, отвечающая за аутентификацию, уведомления и хранение медиафайлов.

Дополнительно в архитектуре выделены ключевые компоненты:

* интеграция с Яндекс.Карты API для геокодирования, отображения карт и построения маршрутов;
* система модерации контента, обеспечивающая контроль качества публикуемых мероприятий;
* механизм социального взаимодействия (матчинг), основанный на взаимных лайках пользователей;

Для визуализации архитектуры разработана диаграмма компонентов, представленная на рисунке 11.

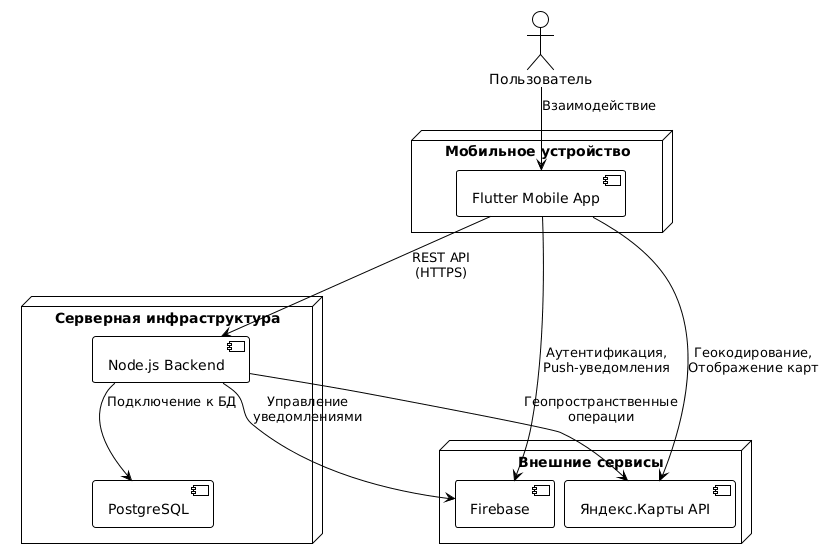


Рисунок – диаграмма размещения

**Прототипы основных экранных форм**

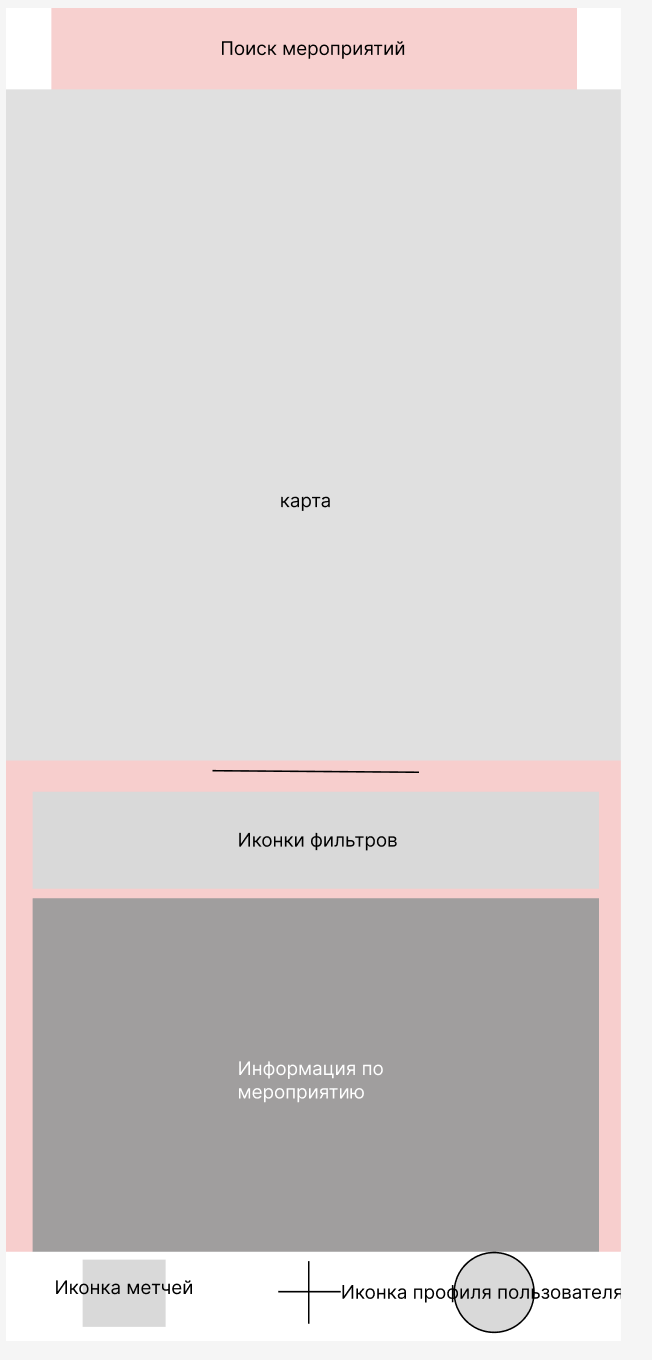
****

Рисунок – Основной экран

**A screenshot of a phone

AI-generated content may be incorrect.**

Рисунок – Экран регистрации

**A grey rectangular object with black text

AI-generated content may be incorrect.**

Рисунок – Профиль пользователя

**A screenshot of a cell phone

AI-generated content may be incorrect.**

Рисунок – Экран мэтчей

**A screenshot of a cell phone

AI-generated content may be incorrect.**

Рисунок – Экран создания мероприятия