

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Институт компьютерных и инженерных наук
Кафедра информационных и управляющих систем
Направление подготовки 09.03.02 – Информационные системы и технологии
Направленность (профиль) образовательной программы Информационные системы и технологии

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой

_____ А.В. Бушманов

« _____ » _____ 2025 г.

БАКАЛАВРСКАЯ РАБОТА

на тему: Разработка веб-приложения «Спортивная карта города»

Исполнитель

студент группы 1104-об

_____ А.В. Аксёнов

(подпись, дата)

Руководитель

доцент, канд. техн. наук

_____ О.В. Жилиндина

(подпись, дата)

Консультант по безопасности и экологичности

доцент, канд. техн. наук

_____ А.Б. Булгаков

(подпись, дата)

Нормоконтроль

инженер кафедры

_____ В.Н. Адаменко

(подпись, дата)

Благовещенск 2025

Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
АМУРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
(ФГБОУ ВО «АмГУ»)

Факультет математики и информатики
Кафедра информационных и управляющих систем

УТВЕРЖДАЮ
Зав. кафедрой
_____ А.В. Бушманов
«_____» _____ 2025 г.

З А Д А Н И Е

К выпускной квалификационной работе студента Аксёнова А.В.

1. Тема выпускной квалификационной работы: Разработка веб-приложения
«Спортивная карта города»

(утверждена приказом от 14.04.2025 № 980-уч)

2. Срок сдачи студентом законченной работы: 10.06.2025г.

3. Исходные данные к выпускной квалификационной работе: отчет по пред-
дипломной практике.

4. Содержание выпускной квалификационной работы: анализ предметной об-
ласти, проектирование веб-приложения, разработка и реализация веб-прило-
жения, безопасность и экологичность.

5. Перечень материалов приложения: техническое задание, инфологическая
модель базы данных, логическая структура программы.

6. Консультанты по выпускной квалификационной работе: консультант по
безопасности и экологичности, А.Б. Булгаков, доцент, канд. техн. наук.

7. Дата выдачи задания: 02.10.2024г.

Руководитель выпускной квалификационной работы: Жилиндина Ольга Вик-
торовна, доцент, канд. техн. наук

Задание принял к исполнению: 02.10.2024 г.: _____

РЕФЕРАТ

Отчет содержит 74 с., 27 рисунков, 16 таблиц, 3 приложения, 35 источников.

БАЗА ДАННЫХ, SQL, СУБД, СУЩНОСТЬ, АТТРИБУТ, ОТНОШЕНИЕ, СВЯЗЬ, TELEGRAM MINI APPS, ЯНДЕКС.КАРТЫ API, ВЕБ-ПРИЛОЖЕНИЕ, PYTHON, СПОРТИВНАЯ ИНФРАСТРУКТУРА.

Цель работы – разработка веб-приложения «Спортивная карта города» для платформы Telegram Mini Apps, предоставляющего пользователям удобный инструмент для поиска спортивных объектов с учетом их специфических требований.

В ходе работы проведен анализ предметной области, современных картографических веб-сервисов и существующих решений, обоснована актуальность разработки. Исследованы особенности платформы Telegram Mini Apps и API Яндекс.Карт, на основе чего сформулированы функциональные и нефункциональные требования к приложению.

Разработана архитектура клиент-серверного приложения: серверная часть на Python (Flask) и клиентская часть (Telegram Mini App). Разработана структура базы данных с использованием СУБД SQLite3 для хранения информации об объектах, пользователях, отзывах и оборудовании. Реализован пользовательский интерфейс и API для взаимодействия между клиентом и сервером.

НОРМАТИВНЫЕ ССЫЛКИ

В настоящей выпускной квалификационной работе использованы ссылки на следующие стандарты:

Приказ Министерства труда и социальной защиты Российской Федерации от 29 октября 2021 г. N 774н «Об утверждении общих требований к организации безопасного рабочего места»

СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания»

СанПиН 2.1.3684-21 «Санитарно-эпидемиологические требования к содержанию территорий городских и сельских поселений, к водным объектам, питьевой воде и питьевому водоснабжению населения, атмосферному воздуху, почвам, жилым помещениям, эксплуатации производственных, общественных помещений, организации и проведению санитарно-противоэпидемических (профилактических) мероприятий»

СТО СМК 4.2.3.21-2018 Оформление выпускных квалификационных и курсовых работ (проектов)

ГОСТ Р 50948-2001 Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности

ГОСТ Р 52872-2019 Интернет-ресурсы и другая информация, представленная в электронно-цифровой форме. Приложение для стационарных и мобильных устройств, иные пользовательские интерфейсы

ГОСТ Р 58751-2019 Слаботочные системы. Кабельные системы. Телекоммуникационные пространства и помещения. Рабочее место

ГОСТ Р ИСО 9241-161-2016 Эргономика взаимодействия человек-система. Часть 161. Элементы графического пользовательского интерфейса

ОПРЕДЕЛЕНИЯ, ОБОЗНАЧЕНИЯ, СОКРАЩЕНИЯ

API – интерфейс программирования приложений

БД – база данных

ГУИ (GUI) – графический интерфейс пользователя

ИС – информационная система

ПО – программное обеспечение

ПК – персональный компьютер

СУБД – система управления базами данных

ТМА – мини-приложения Telegram (Telegram Mini Apps)

UI – пользовательский интерфейс (User Interface)

UX – опыт взаимодействия (User Experience)

СОДЕРЖАНИЕ

Введение	8
1 Анализ предметной области и постановка задачи	10
1.1 Обзор современных картографических веб-сервисов	10
1.2 Анализ существующих решений	13
1.3 Обоснование актуальности разработки	18
1.4 Особенности платформы Telegram Mini Apps	20
1.5 Обоснование выбора API Яндекс карт	23
1.6 Формулировка требований к приложению	24
2 Проектирование веб-приложения «Спортивная карта города»	28
2.1 Функциональная схема проекта	28
2.2 Выбор архитектуры приложения	29
2.3 Проектирование базы данных	31
2.3.1 Инфологическое проектирование	31
2.3.2 Логическое проектирование	36
2.3.3 Физическое проектирование	39
2.4 Проектирование пользовательского интерфейса	42
2.5 Проектирование API	46
3 Разработка и реализация веб-приложения	49
3.1 Выбор средств разработки	49
3.2 Реализация клиентской части	53
3.2.1 Интеграция с API Яндекс. Карт	53
3.2.2 Разработка пользовательского интерфейса	54
3.3 Реализация серверной части	54
3.4 Описание работы приложения	56
4 Безопасность и экологичность	61
4.1 Безопасность	61
4.2 Экологичность	64
4.3 Чрезвычайные ситуации	68

Заключение	71
Библиографические ссылки	72
Библиографический список	73
Приложение А	75
Приложение Б	82
Приложение В	83

ВВЕДЕНИЕ

В последние годы наблюдается значительный рост интереса к здоровому образу жизни, что выражается в увеличении числа людей, активно занимающихся спортом. Это мировая тенденция отражает персональные и общественные стремления к улучшению физического и психического здоровья. Урбанизация и плотность населения в городах превращают наличие доступной спортивной инфраструктуры в критический фактор, способствующий поддержанию здорового образа жизни.

Однако несмотря на значительное количество спортивных объектов, жители многих городов зачастую не обладают полной информацией о расположении, состоянии и доступности таких площадок. Отсутствие централизованной и актуальной информации приводит к недостаточному использованию этих объектов, а также к неудобствам при их поиске. Это, в свою очередь, препятствует формированию активных городских сообществ и снижает общую эффективность городской спортивной инфраструктуры.

Существующие популярные картографические сервисы, такие как Яндекс.Карты, Google Maps или 2ГИС, хотя и предоставляют информацию о местоположении некоторых спортивных объектов, часто лишены необходимой детализации. Пользователь может найти на карте отметку «спортивная площадка», но не получить сведений о наличии на ней конкретного оборудования, например, брусьев, турников, колец или специализированного покрытия для игровых видов спорта. Информация также может быть устаревшей, неполной или не отражать реального состояния объекта. Это создает неудобства для пользователей, которые ищут место для тренировок, соответствующее их специфическим потребностям и предпочтениям. Таким образом, возникает потребность в специализированном инструменте [7], который бы агрегировал актуальную и подробную информацию о спортивной инфраструктуре города.

Современные цифровые технологии открывают новые перспективы для решения этих проблем. Век цифровизации сопровождается развитием

приложений, которые делают информацию доступной и оперативной. В этом контексте важна роль мессенджеров, таких как Telegram, который благодаря своему функционалу становится не только инструментом для общения, но и платформой для таких сервисов как мини-приложения. Это особенно актуально в условиях, когда пользователи предпочитают иметь все необходимое в одном месте, без необходимости загружать и использовать множество отдельных приложений.

Цель работы: разработка веб приложения «Спортивная карта города»

Задачи:

- анализ предметной области;
- проектирование базы данных;
- реализация клиентской и серверной части;
- интеграция с API Яндекс.Карт и Telegram.

Объект исследования охватывает процессы, связанные с созданием веб-приложения, поддерживающего интеграцию современных картографических и геолокационных технологий. Применение технологического инструментария позволит точно отображать местоположение спортивных площадок на интерактивной карте, предоставляя при этом достоверную информацию о характеристиках объектов, их доступности и возможностях. Внедрение API Яндекс.Карт играет центральную роль в обеспечении актуальности данных и их динамического обновления в режиме реального времени. Это технологическое решение помогает пользователям постоянно иметь под рукой самую свежую и точную информацию.

1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ И ПОСТАНОВКА ЗАДАЧИ

1.1 Обзор современных картографических веб-сервисов

Картографические сервисы прочно вошли в повседневную жизнь современного человека, став незаменимым инструментом для навигации, поиска информации и исследования окружающего мира. За последние десятилетия они эволюционировали от статичных цифровых карт до сложных интерактивных платформ, интегрирующих множество функций и данных.

Основные тенденции развития:

– мобильность и повсеместный доступ. С распространением смартфонов и мобильного интернета картографические сервисы стали доступны практически в любом месте и в любое время. Мобильные приложения являются доминирующим способом взаимодействия с картами для большинства пользователей. Это требует адаптации интерфейсов под небольшие экраны, оптимизации производительности и учета особенностей мобильного использования;

– интерактивность и персонализация. Современные карты – это не просто статичные изображения. Пользователи могут масштабировать, панорамировать, вращать карту, переключаться между различными слоями (схема, спутник, панорамы улиц). Сервисы активно используют данные о пользователе (текущее местоположение, история поисков, сохраненные места) для персонализации отображаемой информации и предлагаемых маршрутов;

– интеграция с геолокационными технологиями. Точное определение местоположения пользователя с помощью GPS, Wi-Fi и сотовых сетей является основой для многих функций: навигация, поиск ближайших объектов, отображение собственного положения на карте;

– богатство контента и интеграция данных. Карты обогащаются огромным количеством дополнительной информации: данные об организациях, информация о дорожном движении в реальном времени, расписание общественного транспорта, данные о погоде, трехмерные модели зданий и рельефа

местности. Часто используются данные из внешних источников и пользовательский контент (отзывы, фотографии, исправления);

- социальные функции. Возможность делиться местоположением, маршрутами, найденными объектами с друзьями, оставлять отзывы и оценки, создавать списки избранных мест – все это превращает картографические сервисы в социальные платформы;

- использование векторных тайлов. Переход от растровых изображений к векторным тайлам позволяет добиться более плавной отрисовки, динамической стилизации карты на лету (например, смена темы оформления, скрывание/отображение определенных типов объектов), меньшего объема передаваемых данных и лучшего качества отображения на экранах с высокой плотностью пикселей.

Рассмотрим наиболее известные картографические сервисы:

- Google Maps: Глобальный лидер с огромным покрытием, подробными данными об организациях, панорамами улиц (Street View), мощным API. Основной фокус – глобальное покрытие и интеграция с экосистемой Google. Для России и некоторых других стран данные могут быть менее детализированы, чем у других платформ;

- Яндекс.Карты: Лидирующий сервис в России и странах СНГ. Отличается высокой детализацией карт российских городов, актуальной информацией о пробках, развитой системой отзывов и данных об организациях, качественными панорамами улиц. Предоставляет мощный и хорошо документированный JavaScript API, ориентированный на русскоязычных разработчиков;

- 2ГИС: Популярный сервис в России и некоторых других странах, известный своей высокой точностью и детализацией карт, включая внутренние планы зданий (этажи торговых центров). Содержит обширный справочник организаций с проверенными данными. Также предоставляет API для разработчиков;

- Apple Maps: Картографический сервис для устройств Apple. Активно развивается, улучшая покрытие и детализацию, добавляя новые функции

(например, Look Around - аналог Street View). Интегрирован в экосистему Apple, но его API менее доступен для кроссплатформенной веб-разработки по сравнению с Google или Яндексом;

– OpenStreetMap (OSM): Глобальный проект по созданию свободной, редактируемой карты мира силами сообщества добровольцев. Является источником данных для многих других сервисов и приложений. Преимущества: открытость данных, гибкость использования. Недостатки: неравномерность покрытия и детализации, которая зависит от активности сообщества в конкретном регионе, необходимость использования сторонних инструментов и сервисов для полноценной работы, таких как рендеринг тайлов, поиск и маршрутизация.

Основой современных веб-карт являются технологии веб-картографии. Данные карты обычно хранятся на сервере и передаются клиенту (браузеру или мобильному приложению) в виде небольших фрагментов – тайлов. Тайлы могут быть растровыми (готовые изображения в форматах PNG или JPEG) или векторными (данные о геометрии и атрибутах объектов, которые отрисовываются на стороне клиента с помощью библиотек типа Mapbox GL JS, Leaflet VectorGrid). Векторные тайлы обеспечивают большую гибкость и производительность.

Взаимодействие с картой (масштабирование, перемещение, клики), отображение объектов (меток, линий, полигонов), получение информации, построение маршрутов реализуются с помощью JavaScript API, предоставляемых картографическими платформами. Эти API инкапсулируют сложную логику работы с картой, предоставляя разработчику удобные методы и события.

Таким образом, современные картографические сервисы представляют собой сложные, многофункциональные платформы, построенные на передовых веб-технологиях и больших объемах геоданных. Они предоставляют пользователям мощные инструменты для взаимодействия с окружающим миром, а разработчикам – гибкие API для создания собственных геолокационных приложений.

1.2 Анализ существующих решений

Хотя универсальные картографические сервисы предоставляют базовые возможности для поиска различных мест, включая спортивные, их функционал не всегда отвечает специфическим запросам пользователей, ищущих конкретные условия для тренировок. Проведем анализ наиболее популярных в России сервисов – Яндекс.Карт, 2ГИС и Google Карты – с точки зрения их пригодности для поиска спортивных объектов.

Все рассматриваемые платформы предлагают схожий набор базовых функций для поиска объектов на карте:

- поиск по ключевым словам: пользователь может ввести запрос типа «стадион», «фитнес-клуб», «бассейн», «спортивная площадка». Сервис отобразит на карте найденные объекты и предоставит список с краткой информацией;

- поиск по категориям: сервисы имеют встроенные каталоги организаций и объектов, сгруппированных по категориям. Часто существуют категории вроде «Спорт», «Фитнес», «Активный отдых», которые включают стадионы, спорткомплексы, фитнес-центры, бассейны, катки и т.д. Однако детализация этих категорий может быть недостаточной. Например, общедоступные уличные площадки могут попадать в общую категорию «Спортивная площадка» или даже отсутствовать.

- отображение на карте: найденные объекты помечаются иконками на карте. При клике на метку или элемент списка отображается карточка объекта.

- карточка объекта: содержит основную информацию: название, адрес, тип объекта, рейтинг (если есть), фотографии (часто добавленные пользователями или владельцами), отзывы пользователей, иногда часы работы (для коммерческих объектов).

- отзывы и рейтинги: пользователи могут оставлять комментарии и ставить оценки объектам (обычно по 5-звездочной шкале). Это важный источник информации, но он субъективен и не структурирован для поиска по конкретным параметрам оборудования.

– фотографии: визуальная информация помогает оценить состояние объекта, но редко дает полное представление о всем имеющемся оборудовании. Пользовательские фото могут быть неполными или устаревшими.

– построение маршрута: все сервисы позволяют построить пешеходный, автомобильный или транспортный маршрут до выбранного объекта.

Специфика платформ:

Яндекс.Карты: обладают обширной базой данных по России. Поиск по запросу «спортивная площадка» обычно выдает некоторое количество результатов, включая дворовые объекты. Информация в карточке может содержать пользовательские фото и отзывы. Главный экран приложения представлен на рисунке 1.

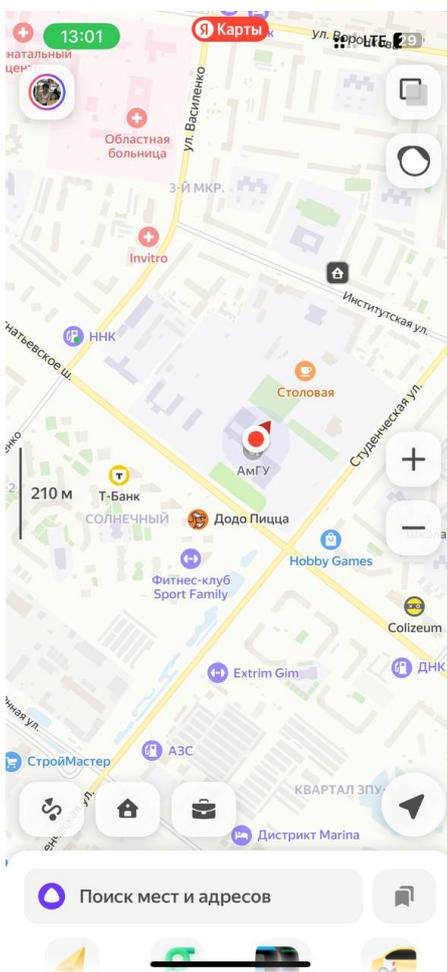


Рисунок 1 – Главный экран Яндекс карты

Основная сложность заключается в том, что Яндекс Карты не всегда содержат полные данные о спортивных объектах, такие как наличие

оборудования, доступные виды спорта и текущая загрузка площадки. Это делает сервис недостаточно удобным для целенаправленного поиска.

2ГИС – это еще одна популярная платформа, сочетающая в себе возможности навигации и предоставления информации о предприятиях и объектах городской инфраструктуры. Славится точностью данных об организациях и детализацией карт. Хорошо ищет коммерческие спортивные объекты (фитнес-клубы, секции). Поиск общедоступных уличных площадок может быть менее эффективным, чем в Яндексe. Специализированных спортивных фильтров также нет. Главный экран приложения представлен на рисунке 2.

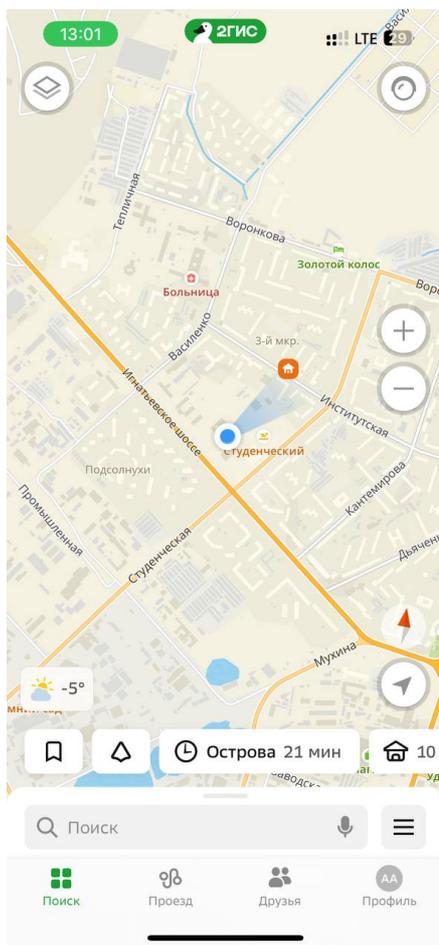


Рисунок 2 – Главный экран 2ГИС

Google Карты – имеет хорошее глобальное покрытие, но детализация данных по общедоступным спортивным площадкам в России часто уступает Яндексe и 2ГИС. Поиск по категориям и ключевым словам работает, но результаты могут быть менее релевантными для специфических запросов вроде

«площадка с брусьями». Фильтрация по оборудованию отсутствует. Главный экран приложения представлен на рисунке 3.

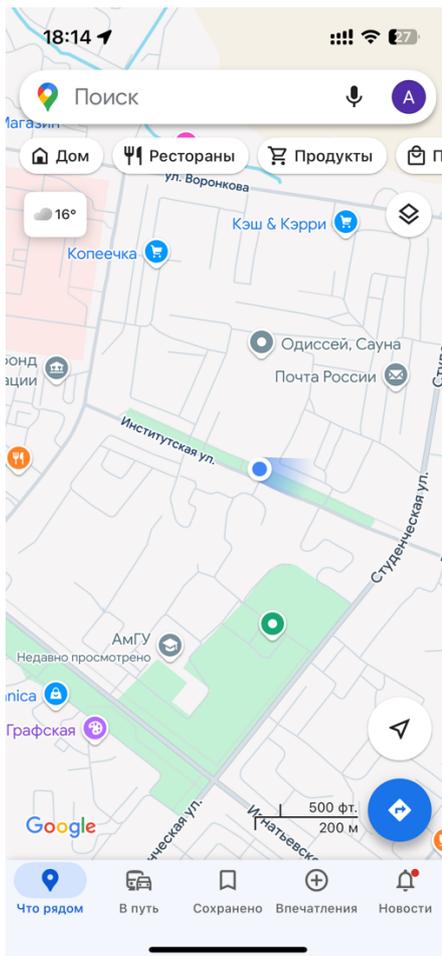


Рисунок 3 – Главный экран Google Карты

Достоинства рассмотренных картографических сервисов:

- широкое покрытие и известность: Эти сервисы охватывают практически все населенные пункты и знакомы большинству пользователей, что снижает порог входа;
- большая база данных: содержат информацию о миллионах объектов, включая многие крупные и коммерческие спортивные сооружения;
- пользовательский контент: отзывы и фотографии от других пользователей могут дать представление о популярности и общем состоянии объекта;
- интеграция и удобство: возможность построить маршрут, посмотреть панорамы, сохранить место – все в одном приложении;

– регулярные обновления: карты и данные постоянно обновляются (хотя актуальность информации о небольших объектах может страдать).

Недостатки:

– отсутствие специализированной фильтрации: ключевой недостаток. Невозможно отфильтровать площадки по наличию конкретного оборудования (турники, брусья, скамьи для пресса, тип покрытия беговой дорожки и т.д.). Пользователь вынужден просматривать фотографии и читать отзывы по каждому объекту, что крайне неудобно и долго;

– слабая категоризация спортивных объектов: общедоступные уличные площадки, стадионы школ, частные фитнес-клубы часто попадают в одну общую категорию или ищутся по общим ключевым словам, что затрудняет целевой поиск;

– неполнота и неактуальность данных: информация о небольших, бесплатных уличных площадках часто отсутствует или устаревает. Пользовательский контент не всегда систематизирован и может быть неактуальным.

– ориентация на коммерческие объекты: основной фокус сервисов часто смещен в сторону коммерческих организаций (фитнес-центры, платные секции), информация о которых более полная и проверенная. Бесплатные общественные пространства освещаются хуже;

– отсутствие специфической информации: нет данных о качестве покрытия, высоте турников, освещенности в темное время суток, наличии раздевалок или питьевой воды поблизости (что может быть важно для спортсменов).

Как показал анализ, ни один из ведущих картографических сервисов (Яндекс.Карты, 2ГИС, Google Карты) не предоставляет пользователям возможности целенаправленного поиска спортивных объектов по наличию конкретного инвентаря или оборудования. Если пользователю нужны, например, площадки исключительно с брусьями и турником, или беговая дорожка с резиновым покрытием, он не сможет задать такие критерии поиска. Ему придется вручную перебирать все найденные «спортивные площадки», изучая их

фотографии и отзывы в надежде найти нужную информацию. Это делает процесс поиска неэффективным и зачастую безрезультатным.

Кроме того, в существующих аналогах отсутствуют другие потенциально полезные для спортсменов функции:

- структурированное описание оборудования. Вместо простого перечисления в тексте отзыва, необходима система тегов или чекбоксов, четко указывающая на наличие каждого типа оборудования;

- оценка состояния оборудования. Помимо общего рейтинга места, была бы полезна возможность оценить состояние конкретных тренажеров;

- возможность добавления нового объекта. Такой функционал позволит сообществу активных пользователей самостоятельно добавлять в приложение новые спортивные площадки, которые еще не отмечены на карте.

1.3 Обоснование актуальности разработки

Актуальность и необходимость создания специализированного приложения «Спортивная карта города» обусловлены несколькими факторами, связанными как с общими социальными тенденциями, так и с ограничениями существующих инструментов.

Рост популярности здорового образа жизни и уличного спорта. В последние годы наблюдается устойчивый тренд на повышение внимания к собственному здоровью и физической активности. Все больше людей стремятся регулярно заниматься спортом, причем значительная часть из них предпочитает тренировки на свежем воздухе, используя доступную городскую инфраструктуру. Такие направления, как воркаут (уличная гимнастика), бег, занятия на уличных тренажерах, становятся массовыми. Это формирует четкий запрос на удобные инструменты для поиска подходящих мест для тренировок.

Специфические потребности целевой аудитории. Разрабатываемое приложение ориентировано на конкретные группы пользователей, чьи потребности не удовлетворяются универсальными картографическими сервисами:

- воркаут-атлеты: для них критически важно наличие определенного набора снарядов (турники разной высоты, брусья, шведские стенки, рукоходы)

и их состояние. Поиск площадок отнимает время, которое можно было бы потратить на тренировку;

– бегуны: ищут беговые дорожки с определенным покрытием (грунт, асфальт, резина), информацией о длине круга, рельефе, освещенности;

– люди, занимающиеся общей физической подготовкой: нуждаются в площадках с разнообразными тренажерами для комплексной тренировки, например, уличные аналоги тренажеров из зала или скамьи для прессы;

– родители с детьми: ищут безопасные площадки с детским спортивным оборудованием рядом с домом.

Для всех этих групп возможность быстро найти объект с нужным набором оборудования является ключевым фактором для регулярных и эффективных занятий.

Популярные картографические сервисы не предоставляют необходимой детализации и функционала фильтрации по спортивному инвентарю. Это создает информационный вакуум и неудобства для целевой аудитории. Пользователи вынуждены полагаться на самостоятельное исследование местности.

Потребность в актуальной и проверенной информации: Состояние уличных спортивных площадок может быстро меняться (установка нового оборудования, поломки, демонтаж). Централизованная платформа с возможностью добавления и обновления информации самими пользователями, с последующей модерацией, позволит поддерживать базу данных в актуальном состоянии. Система рейтингов и отзывов, сфокусированная на спортивных аспектах (качество оборудования, покрытие, удобство), будет более информативна, чем общие отзывы на универсальных картах.

Создание локального, но детализированного ресурса может быть особенно полезным для города. Оно способствует развитию местного спортивного сообщества, помогает жителям лучше ориентироваться в доступной инфраструктуре и стимулирует городские власти к улучшению существующих и созданию новых спортивных зон, видя реальный спрос и получая обратную связь.

Таким образом, разработка «Спортивной карты города» является актуальной задачей, решающей реальную проблему целевой аудитории и предлагающей уникальный функционал, отсутствующий в существующих аналогах. Приложение способно стать полезным инструментом для спортсменов-любителей, способствовать развитию спортивной инфраструктуры города и популяризации здорового образа жизни.

1.4 Особенности платформы Telegram Mini Apps

Выбор платформы для реализации приложения является важным стратегическим решением. Telegram Mini Apps представляет собой веб-приложения, запускаемые непосредственно внутри мессенджера Telegram. Рассмотрим преимущества, технические аспекты и ограничения этой платформы применительно к задаче разработки «Спортивной карты города».

Преимущества:

- нативная интеграция и доступность: Пользователям не нужно искать, скачивать и устанавливать отдельное мобильное. Доступ к Mini App осуществляется через чат-бот в Telegram или по прямой ссылке, что значительно снижает барьер для начала использования. Приложение ощущается как часть привычного интерфейса мессенджера;

- огромная аудитория: Telegram является одним из самых популярных мессенджеров в России и мире. Размещение приложения на этой платформе открывает доступ к многомиллионной аудитории потенциальных пользователей без дополнительных затрат на маркетинг и продвижение, присущих традиционным мобильным приложениям;

- простота авторизации и получения данных пользователя: ТМА имеют встроенную интеграцию с Telegram API. Приложение может легко и безопасно получить идентификатор пользователя, его имя (username), имя и фамилию без необходимости реализации сложных процессов регистрации и аутентификации. Это упрощает персонализацию, например, для системы отзывов;

- использование стандартных веб-технологий: Разработка ТМА основана на хорошо известных и широко распространенных веб-технологиях:

HTML для структуры, CSS для стилизации и JavaScript для логики и интерактивности. Это позволяет использовать существующие библиотеки и инструменты веб-разработки, что ускоряет и упрощает процесс создания приложения. Код приложения является стандартным веб-контентом;

– легкость обновления: В отличие от нативных мобильных приложений, требующих сборки, публикации и обновления через магазины приложений, ТМА обновляются так же легко, как обычные веб-сайты. Достаточно обновить файлы на сервере, и пользователи при следующем открытии увидят последнюю версию;

– возможности взаимодействия с интерфейсом Telegram: Платформа предоставляет JavaScript библиотеку `telegram-web-app.js`, которая позволяет веб-приложению взаимодействовать с нативным интерфейсом Telegram: показывать всплывающие уведомления, запрашивать доступ к геолокации, управлять цветовой схемой приложения в соответствии с темой Telegram пользователя, закрывать Mini App и т.д;

– потенциал для распространения и социальных функций: Пользователи могут легко делиться ссылкой на Mini App или конкретные объекты внутри него в чатах Telegram, что способствует естественному распространению приложения. Интеграция с мессенджером открывает возможности для реализации социальных функций, таких как приглашение друзей на совместную тренировку, создание групповых рейтингов и т.д.

Ограничения:

– веб-приложение в IFrame: технически, Mini App представляет собой веб-страницу, загружаемую в изолированном окружении внутри клиента Telegram. Это накладывает определенные ограничения на доступ к API устройства и браузера из соображений безопасности;

– зависимость от `telegram-web-app.js`: Взаимодействие с функциями Telegram (кнопки, уведомления, данные пользователя) осуществляется исключительно через библиотеку `telegram-web-app.js`, предоставляемую платформой [5]. Необходимо корректно инициализировать ее и использовать ее методы.

– ограниченный доступ к API устройства: Прямой доступ к некоторым аппаратным возможностям устройства, например, Bluetooth, NFC, данные о контактах может быть ограничен или отсутствовать, в отличие от нативных приложений. Однако доступ к геолокации, камере предоставляется через стандартные веб-API или запросы через Telegram API;

– требования к UI/UX: хотя разработчик имеет свободу в дизайне, рекомендуется придерживаться гайдлайнов Telegram по внешнему виду и поведению элементов интерфейса, чтобы приложение органично вписывалось в экосистему мессенджера. Платформа предоставляет информацию о текущей цветовой теме пользователя и переменные CSS, которые используются в style.css для адаптации внешнего вида;

– производительность: сложные веб-приложения, особенно работающие с интерактивными картами и большим количеством данных, могут сталкиваться с ограничениями производительности на менее мощных мобильных устройствах. Требуется тщательная оптимизация кода JavaScript, CSS и работы с данными. Использование тяжелых фреймворков может увеличить время первоначальной загрузки;

– необходимость хостинга: Mini App должно быть размещено на доступном через HTTPS веб-сервере. Разработчик сам отвечает за хостинг и обеспечение доступности своего приложения. В данном проекте используется Flask для предоставления статических файлов и API, что решает вопрос хостинга;

– безопасность: данные, передаваемые между Mini App и сервером, должны быть защищены с использованием HTTPS. При обработке данных пользователя, полученных от Telegram (initData), необходимо проверять их подлинность на стороне сервера.

В целом, для веб-приложения «Спортивная карта города» платформа Telegram Mini Apps представляет собой удачный выбор, сочетающий удобство для пользователя, простоту разработки и распространения с достаточной функциональностью для реализации всех поставленных задач, включая интеграцию со сторонними API.

1.5 Обоснование выбора API Яндекс.Карт

Выбор картографического провайдера и его API является критически важным для приложения, основной функцией которого является отображение интерактивной карты и работа с геоданными. Выбор API Яндекс.Карт обусловлен следующими причинами:

- высокое качество и детализация карт России: Яндекс традиционно уделяет большое внимание качеству картографического покрытия на территории России и стран СНГ. Карты Яндекса часто содержат более актуальную и подробную информацию о дорожной сети, зданиях, организациях и других объектах в российских городах, включая Благовещенск, по сравнению с глобальными конкурентами вроде Google Maps или Apple Maps. Это особенно важно для поиска не только крупных, но и небольших локальных спортивных площадок;

- мощный и функциональный JavaScript API: Яндекс.Карты предоставляют хорошо документированный и гибкий JavaScript API, который позволяет реализовать весь необходимый функционал ;

- качественная документация и поддержка на русском языке: Наличие подробной документации, примеров кода, технической поддержки на русском языке значительно упрощает процесс разработки и решения возникающих проблем для русскоязычных разработчиков [4];

- условия использования и бесплатный лимит: API Яндекс.Карт предлагает бесплатный тариф с достаточно щедрыми лимитами на количество запросов к API и загрузок карты, что подходит для запуска и начального этапа развития некоммерческого или небольшого коммерческого проекта;

- релевантность для целевой аудитории: пользователи в России привыкли к интерфейсу и логике работы Яндекс.Карт, что может сделать взаимодействие с приложением более интуитивным.

Сравнение с альтернативами:

Google Maps API: также мощный API с глобальным покрытием. Однако его использование стало преимущественно платным даже для небольших

объемов запросов, что увеличивает порог входа. Качество карт и данных по России может уступать Яндекс.Картам в некоторых аспектах.

OpenStreetMap (OSM): предоставляет свободные геоданные, но не является готовым картографическим сервисом «из коробки». Для его использования требуется настройка собственного сервера тайлов или использование сторонних сервисов (часто платных), а также интеграция с отдельными библиотеками для поиска, маршрутизации и отображения. Это требует больших затрат на разработку и поддержку инфраструктуры.

2ГИС API: предлагает качественные данные и API, но может быть менее гибким в плане кастомизации и иметь более строгие условия использования по сравнению с Яндекс.Картами для некоторых сценариев.

Учитывая высокое качество карт России, богатый функционал API, наличие русскоязычной документации и приемлемые условия использования, API Яндекс.Карт представляется оптимальным выбором для реализации проекта «Спортивная карта города Благовещенска». Интеграция с данным API позволяет обеспечить пользователей качественной и интерактивной картографической основой для поиска спортивных объектов.

1.6 Формулировка требований к приложению

На основе проведенного анализа предметной области, выявленных потребностей пользователей и возможностей выбранных технологий были сформулированы основные требования к разрабатываемому веб-приложению «Спортивная карта города». Требования разделены на функциональные, определяющие основные возможности системы, и нефункциональные, описывающие качественные характеристики приложения.

Функциональные требования:

Приложение должно предоставлять пользователю комплексный инструмент для взаимодействия с информацией о спортивных объектах. Ключевой функцией является отображение интерактивной карты города Благовещенска с использованием API Яндекс.Карт с возможностью масштабирования, панорамирования и отображения текущего местоположения

пользователя. На карте должны отображаться спортивные объекты в виде меток, при взаимодействии с которыми пользователь получает доступ к краткой, а затем и к детальной информации об объекте. Эта информация включает название, адрес, описание, фотографию, средний рейтинг, список оборудования, а также отзывы других пользователей и форму для добавления собственного отзыва.

Параллельно с картой должен отображаться список объектов с основной информацией и возможностью постраничной загрузки «Загрузить больше». Как клик по метке на карте, так и выбор элемента из списка должны приводить к экрану с детальной информацией.

Важной частью функционала является система фильтрации объектов по наличию конкретного спортивного оборудования. Пользователь должен иметь возможность выбрать один или несколько типов оборудования, после чего на карте и в списке будут отображены только соответствующие объекты. Предусмотрена возможность сброса фильтров и визуализация активных фильтров.

Для персонализации опыта использования внедряется система «Избранное», позволяющая авторизованным пользователям сохранять понравившиеся объекты в личный список, просматривать их и удалять.

Система отзывов и рейтингов дает возможность авторизованным пользователям делиться своим мнением об объектах, оставлять текстовые комментарии и выставлять оценки по 5-звездочной шкале. На основе этих оценок должен рассчитываться и отображаться средний рейтинг каждого объекта.

Пользователи также должны иметь возможность добавлять новые спортивные объекты. Этот процесс включает указание местоположения на карте, заполнение формы с названием, адресом (с возможностью автоматической подстановки), описанием, загрузкой фотографии и выбором имеющегося оборудования. Данные отправляются на сервер для сохранения.

К нефункциональным требованиям относятся следующие аспекты:

Производительность:

- время первоначальной загрузки приложения не должно превышать 5 секунд на среднем мобильном устройстве при стабильном интернет-соединении (3G/4G);

- время отклика интерфейса на действия пользователя (клик по кнопке, выбор фильтра, открытие деталей объекта) не должно превышать 1 секунду;

- отображение и панорамирование карты с метками должно быть плавным.

Удобство использования (Usability):

- интерфейс должен быть интуитивно понятным для пользователей, знакомых с Telegram и стандартными картографическими приложениями;

- навигация по приложению должна быть простой и логичной;

- дизайн приложения должен соответствовать цветовой схеме Telegram пользователя для лучшей интеграции;

- приложение должно корректно отображаться на экранах различных мобильных устройств.

Надежность:

- приложение должно стабильно работать без сбоев и зависаний при выполнении стандартных пользовательских сценариев;

- ошибки должны корректно обрабатываться и сопровождаться информативными сообщениями для пользователя;

- доступность серверной части (API) должна быть не ниже 99 %.

Безопасность:

- взаимодействие между клиентской частью (ТМА) и серверной частью (Flask API) должно осуществляться по защищенному протоколу HTTPS;

- серверная часть должна быть защищена от распространенных веб-уязвимостей;

- проверка аутентичности данных пользователя, получаемых от Telegram, должна выполняться на сервере;

- загружаемые пользователями файлы должны проходить проверку на допустимые типы и размеры.

Масштабируемость:

- архитектура приложения должна позволять относительно легко добавлять новые типы оборудования для фильтрации;
- система должна справляться с ростом количества спортивных объектов и пользователей.

Поддерживаемость:

- код приложения должен быть хорошо структурирован, прокомментирован и следовать общепринятым стандартам оформления для облегчения дальнейшей поддержки и модификации.

Сформулированные требования служат основой для проектирования, разработки и последующего тестирования веб-приложения «Спортивная карта города».

2 ПРОЕКТИРОВАНИЕ ВЕБ-ПРИЛОЖЕНИЯ «СПОРТИВНАЯ КАРТА ГОРОДА»

2.1 Функциональная схема проекта

В ходе разработки веб-приложения были выделены основные категории функций, которые представлены на рисунке на рисунке 4.

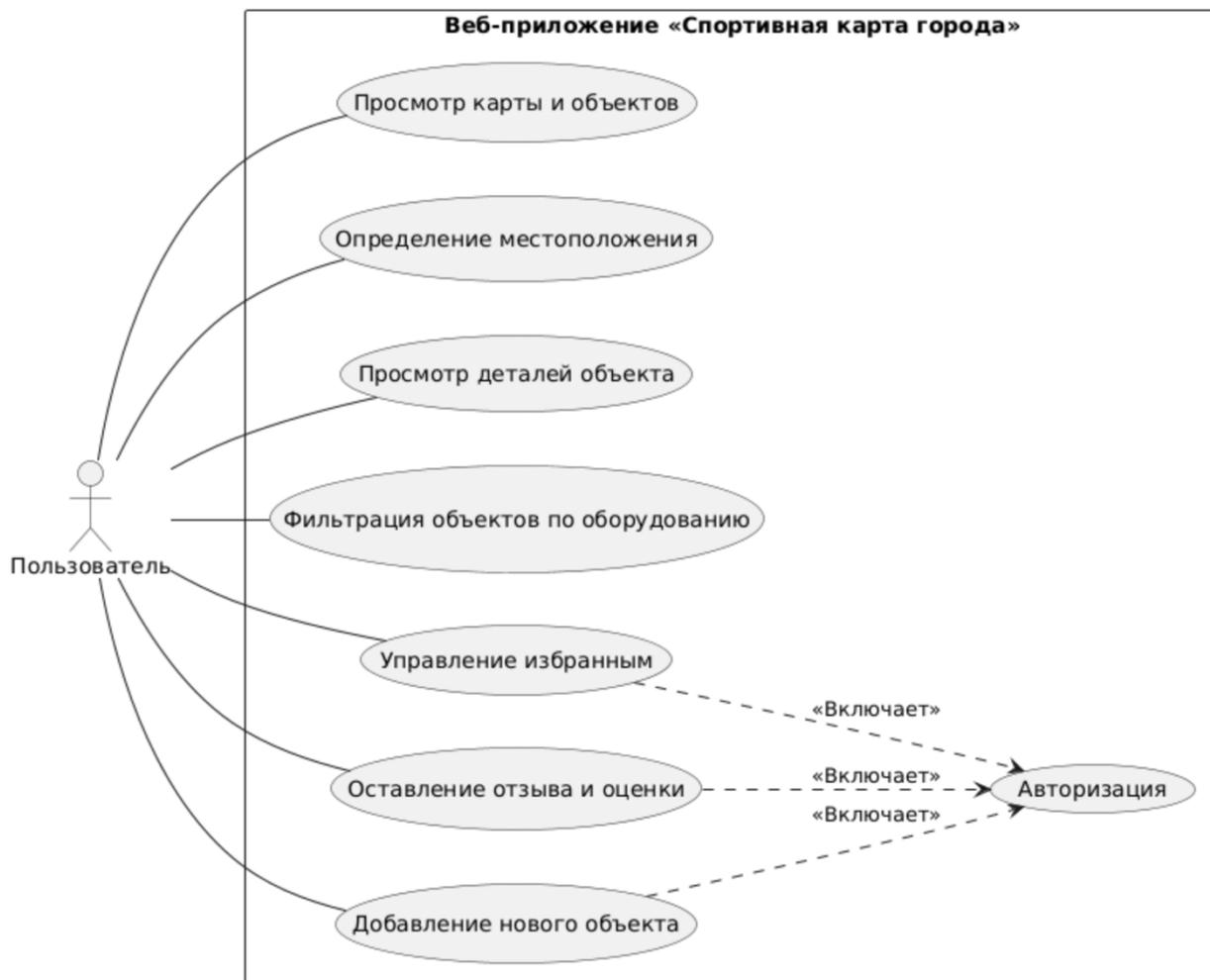


Рисунок 4 – Диаграмма прецедентов

Основными функциями приложения являются:

- отображение и навигация по карте;
- просмотр информации об объектах;
- возможность фильтрации;
- система «Избранное»;
- рейтинг и отзывы;
- добавление новых объектов пользователями.

Эти функции в совокупности обеспечивают выполнение основных задач, поставленных перед приложением: предоставление актуальной информации о спортивной инфраструктуре города, облегчение поиска подходящих площадок и вовлечение пользователей в процесс наполнения и оценки контента.

2.2 Выбор архитектуры приложения

Архитектура приложения определяет его общую структуру, взаимодействие компонентов и способ распределения ответственности между ними. Выбор правильной архитектуры критически важен для обеспечения масштабируемости, поддерживаемости, производительности и надежности системы. Для приложения «Спортивная карта города» была выбрана классическая клиент-серверная архитектура, представленная на рисунке 5.

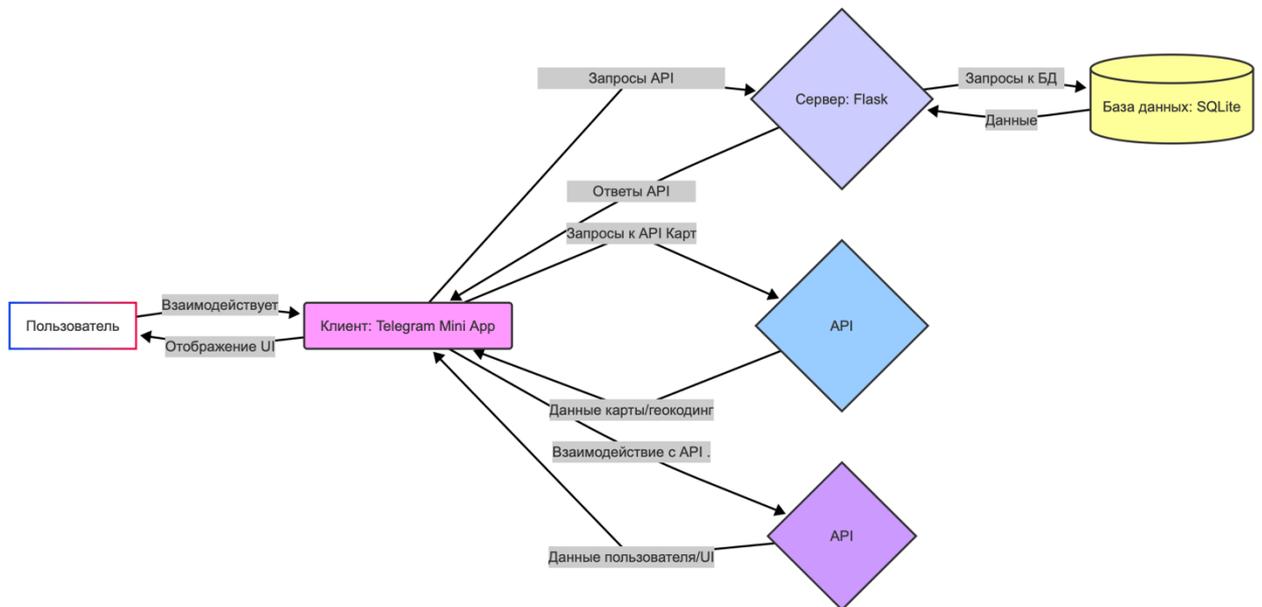


Рисунок 5 – Архитектура веб-приложения

Клиент-серверная модель предполагает разделение приложения на две основные части:

– клиентская часть (Client): это фронтенд-приложение, с которым непосредственно взаимодействует пользователь. В данном случае это Telegram Mini App, реализованное с помощью веб-технологий (HTML, CSS, JavaScript) и работающее внутри мессенджера Telegram. Клиент отвечает за отображение

пользовательского интерфейса (карта, списки, формы), обработку пользовательского ввода, взаимодействие с API Яндекс.Карт для отображения карты и меток, а также за отправку запросов на сервер для получения и изменения данных (объекты, отзывы, избранное).

– серверная часть (Server): это бэкенд-приложение, которое работает на удаленном сервере. Оно отвечает за бизнес-логику, обработку запросов от клиента, взаимодействие с базой данных (хранение и извлечение информации о спортивных объектах, пользователях, отзывах, избранном, оборудовании) и предоставление данных клиенту через API (Application Programming Interface). Сервер также отвечает за обработку загружаемых файлов.

Обоснование выбора клиент-серверной архитектуры:

– централизованное хранение данных: Данные о спортивных объектах, пользователях, отзывах и т.д. должны быть доступны всем пользователям приложения и сохраняться постоянно. Хранение этих данных на сервере в централизованной базе данных является наиболее логичным решением. Попытка хранить все данные только на клиенте (например, в LocalStorage или Telegram CloudStorage) привела бы к проблемам синхронизации, ограничениям по объему и сложности совместного доступа.

– бизнес-логика на сервере: Вынесение основной логики (фильтрация данных по сложным критериям, расчет рейтинга, обработка добавления новых объектов, управление пользователями) на сервер позволяет сделать клиентскую часть более легкой и обеспечивает консистентность данных.

– безопасность: Сервер позволяет реализовать контроль доступа, валидацию данных и защиту от несанкционированных действий, что сложнее обеспечить при хранении всей логики на клиенте. Например, проверка прав пользователя на удаление из избранного или добавление отзыва происходит на сервере.

– масштабируемость: хотя текущая реализация использует SQLite, серверная архитектура позволяет в будущем относительно легко перейти на более

мощную СУБД и масштабировать серверную часть при росте нагрузки, не затрагивая клиентское приложение.

– управление пользовательским контентом: Обработка загружаемых фотографий и модерация добавляемых объектов – задачи, которые естественным образом решаются на серверной стороне.

Клиентская часть представляет собой одностраничное веб-приложение (Single Page Application - SPA), работающее как Telegram Mini App.

Такая архитектура обеспечивает четкое разделение ответственности между клиентом и сервером, хорошую организацию кода и гибкость для дальнейшего развития.

2.3 Проектирование базы данных

2.3.1 Инфологическое проектирование

На основании проведенных исследований предметной области, были выделены следующие сущности, необходимые для построения информационной базы [1]. Все сущности приведены в таблице 1.

Таблица 1 – Формирование сущностей

Название сущности	Описание сущности
Пользователь	Содержит информацию о пользователях
СпортивныйОбъект	Содержит информацию о спортивных объектах
Отзыв	Содержит информацию об отзывах
Оборудование	Содержит информацию о спортивном оборудовании
Избранное	Содержит информацию об избранных объектах пользователей

После определения ключевых сущностей, следующим шагом инфологического проектирования является детализация их атрибутивного состава. Для каждой выделенной сущности необходимо определить атрибуты, которые будут наиболее полно описывать ее свойства и обеспечивать необходимые связи

с другими сущностями системы. Этот процесс позволяет уточнить структуру данных и подготовить основу для последующего перехода к логическому и физическому проектированию базы данных.

Спецификация атрибутов сущностей представлена в таблицах 2-7:

Таблица 2 – Спецификация атрибутов сущности «Пользователь»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код_пользователя	Уникальный идентификатор пользователя	Числовой	> 0	5
Имя_пользователя	Имя или никнейм пользователя	Текстовый	–	Иван Петров
Код_телеграм	Идентификатор в Telegram	Числовой	>0	12345678
Дата_регистрации	Дата регистрации пользователя	Дата	>01.01.2025	01.11.2025

Таблица 3 – Спецификация атрибутов сущности «СпортивныйОбъект»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код_объекта	Уникальный идентификатор объекта	Числовой	>0	12
Название	Наименование спортивного объекта	Текстовый	–	Парк Победы
Адрес	Физический адрес объекта	Текстовый	–	ул. Ленина, 10
Описание	Описание объекта	Текстовый	–	Сквер с тренажерами
Широта	Географическая широта	Числовой	от -90.0 до +90.0	55.7558
Долгота	Географическая долгота	Числовой	от -180.0 до +180.0	37.6173
Фото	Ссылка на фото объекта	Текстовый	–	photo1.jpg

Таблица 4 – Спецификация атрибутов сущности «Оборудование»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код_оборудования	Уникальный идентификатор оборудования	Числовой	>0	3
Наименование	Название оборудования	Текстовый	Произвольная строка	Турник

Таблица 5 – Спецификация атрибутов сущности «Отзыв»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код_отзыва	Уникальный идентификатор отзыва	Числовой	>0	8
Код_пользователя	Ссылка на автора отзыва	Числовой	>0	5
Код_объекта	Ссылка на объект отзыва	Числовой	>0	12
Текст	Содержание отзыва	Текстовый	–	Отличное место!
Оценка	Оценка объекта (целое число)	Числовой	>=1 и <=5	5
Дата	Дата создания отзыва	Дата	>01.01.2025	02.01.2025

Таблица 6 – Спецификация атрибутов сущности «Избранное»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код_избранного	Уникальный идентификатор избранного	Числовой	>0	1
Код_пользователя	Ссылка на пользователя	Числовой	>0	5
Код_объекта	Ссылка на спортивный объект	Числовой	>0	12
Дата_добавления	Дата добавления в избранное	Дата	>01.01.2025	02.01.2025

Таблица 7 – Спецификация атрибутов сущности «СпортивныйОбъект Оборудование»

Название атрибута	Описание атрибута	Тип данных	Диапазон значений	Пример
Код_объекта	Уникальный идентификатор объекта	Числовой	>0	1
Код_оборудования	Уникальный идентификатор оборудования	Числовой	>0	5

После определения сущностей и их атрибутов необходимо спроектировать связи между сущностями для их функционирования.

В результате были спроектированы следующие связи:

Связь: «Пользователь – Избранное» рисунок 6.



Рисунок 6 – Связь «Пользователи – Избранное»

Образуется связь «один ко многим». Один пользователь может добавить множество объектов в избранное, в то время как одна запись в избранном может принадлежать только одному пользователю.

Связь «СпортивныйОбъект – Избранное» рисунок 7.



Рисунок 7 – Связь «СпортивныйОбъект – Избранное»

Образуется связь «один ко многим». Один спортивный объект может быть добавлен в избранное у многих пользователей, в то время как одна запись в избранном относится только к одному спортивному объекту.

Связь «Пользователь – Отзыв» рисунок 8.



Рисунок 8 – Связь «Пользователь – Отзыв»

Образуется связь «один ко многим». Один пользователь может оставить множество отзывов, в то время как один отзыв может быть написан только одним пользователем.

Связь: «СпортивныйОбъект – Отзыв» рисунок 9.

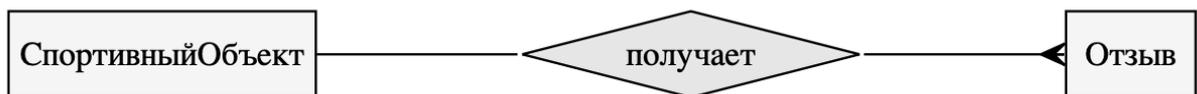


Рисунок 9 – Связь «СпортивныйОбъект – Отзыв»

Образуется связь «один ко многим». Один спортивный объект может получить множество отзывов, в то время как один отзыв может относиться только к одному спортивному объекту.

Связь: «СпортивныйОбъект – Оборудование» рисунок 10.



Рисунок 10 – Связь «СпортивныйОбъект – Оборудование»

Образуется связь «многие ко многим». Один спортивный объект может содержать множество видов оборудования, в то время как один вид оборудования может находиться на множестве спортивных объектов. Связь осуществляется через ассоциативную таблицу «СпортивныйОбъект_Оборудование».

Цель инфологического проектирования заключается в представлении семантики предметной области. Для описания предметной области наиболее часто используется ER–модель (Сущность – Связь).

Инфологическая модель базы данных в нотации Чена представлена рисунке Б.1 в приложении Б.

2.3.2 Логическое проектирование базы данных

Для установления связи между объектами предметной области и создания набора нормализованных отношений, этот процесс включает два шага:

– преобразование концептуально-инфологической модели в реляционную модель: на данном этапе ключевые элементы взаимосвязанных записей представляются в виде отношений. Это подразумевает, что каждая сущность или объект становятся отдельными таблицами, а связи между ними реализуются посредством ключей или столбцов, соединяющих таблицы;

– анализ соответствия полученных отношений трём нормальным формам: после перевода в реляционную модель, отношения проверяются на соответствие трём нормальным формам. Эти формы (первая, вторая и третья нормальные формы) задают правила для организации данных в таблицах, устраняя избыточность и обеспечивая эффективное хранение и обработку информации.

Процесс нормализации данных включает трансформацию модели в реляционную форму и последующий анализ полученных отношений на соответствие трём нормальным формам.

Сущности:

«Избранное» – Таблица «Избранное» с первичным ключом «Код_избранного».

«Оборудование» – Таблица «Оборудование» с первичным ключом «Код_оборудования».

«Отзыв» – Таблица «Отзыв» с первичным ключом «Код_отзыва».

«Пользователь» – Таблица «Пользователь» с первичным ключом «Код_пользователя».

«СпортивныйОбъект» – Таблица «СпортивныйОбъект» с первичным ключом «Код_объекта».

«СпортивныйОбъект_Оборудование» – Связующая таблица с составным первичным ключом («Код_объекта», «Код_оборудования»).

Связи:

Связь «Пользователь – Избранное» реализована через внешний ключ «Код_пользователя» в таблице «Избранное».

Связь «СпортивныйОбъект – Избранное» реализована через внешний ключ «Код_объекта» в таблице «Избранное».

Связь «Пользователь – Отзыв» реализована через внешний ключ «Код_пользователя» в таблице «Отзыв».

Связь «СпортивныйОбъект – Отзыв» реализована через внешний ключ «Код_объекта» в таблице «Отзыв».

Связь «СпортивныйОбъект – Оборудование» реализована через связующую таблицу «СпортивныйОбъект_Оборудование».

Все представленные сущности соответствуют первой нормальной форме, так как значения всех атрибутов являются неделимыми или атомарными. Кроме того, сущности также соответствуют второй нормальной форме, так как все не ключевые атрибуты функционально полно зависят от первичного ключа. Кроме того, все отношения находятся в третьей нормальной форме, поскольку не ключевые атрибуты не взаимозависимы. Функциональные зависимости между атрибутами отображены на рисунках 11-15.



Рисунок 11 – Функциональная зависимость сущности «Пользователь»

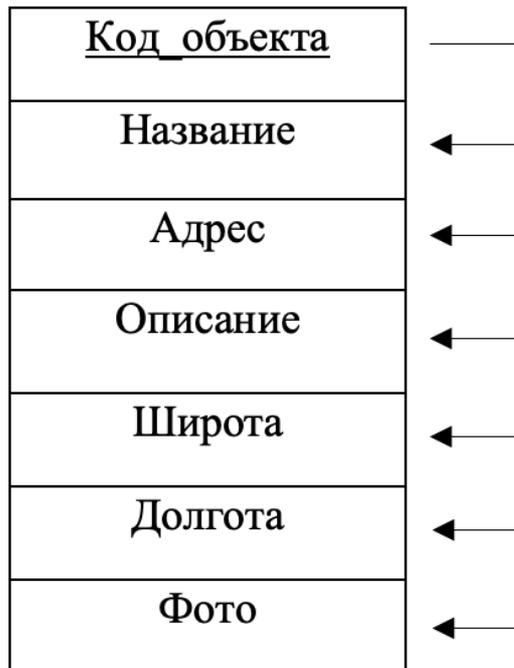


Рисунок 12 – Функциональная зависимость сущности «СпортивныйОбъект»

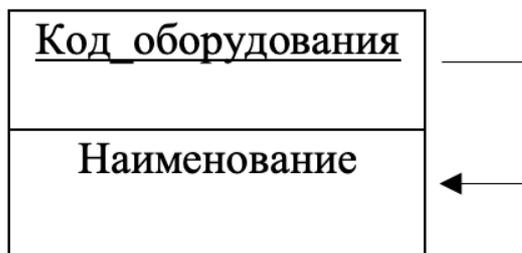


Рисунок 13 – Функциональная зависимость сущности «Оборудование»



Рисунок 14 – Функциональная зависимость сущности «Избранное»



Рисунок 15 – Функциональная зависимость сущности «Отзывы»

Логическая модель базы данных в виде диаграммы IDEF1X представлена на рисунке 16.

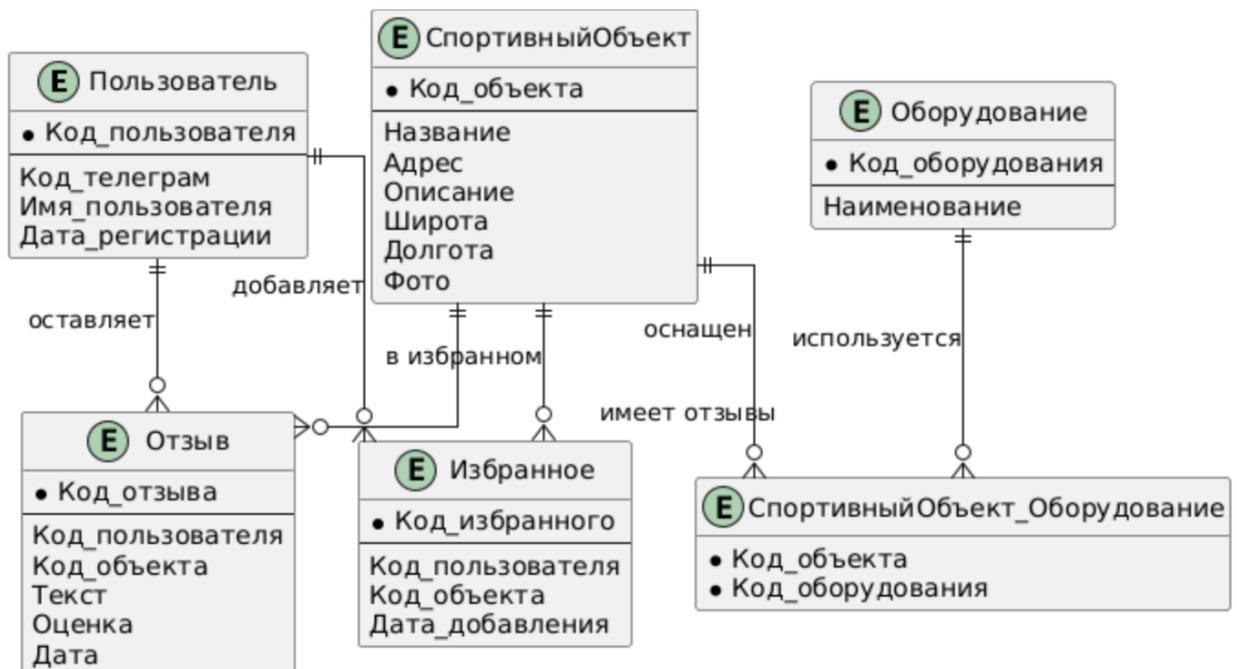


Рисунок 16 – Логическая модель БД

2.3.3 Физическое проектирование базы данных

На данном этапе представлены проекты таблиц, которые будут реализованы в СУБД. Физическое определение атрибутов сущностей приведено в таблицах 8–13.

Таблица 8 – Физическая структура данных отношения «Избранное»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Код_избранного	INTEGER	PRIMARY KEY, AUTOINCREMENT	Целое число	PRIMARY KEY
Код_пользователя	INTEGER	NOT NULL, FOREIGN KEY	Целое число	FOREIGN KEY
Код_объекта	INTEGER	NOT NULL, FOREIGN KEY	Целое число	FOREIGN KEY
Дата_добавления	TEXT	NOT NULL	Текст (дата)	–

Таблица 9 – Физическая структура данных отношения «Оборудование»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Код_оборудования	INTEGER	PRIMARY KEY, AUTOINCREMENT	Целое число	PRIMARY KEY
Наименование	TEXT	–	Текст	–

Таблица 10 – Физическая структура данных отношения «Отзыв»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Код_отзыва	INTEGER	PRIMARY KEY, AUTOINCREMENT	Целое число	PRIMARY KEY
Код_пользователя	INTEGER	NOT NULL, FOREIGN KEY	Целое число	FOREIGN KEY
Код_объекта	INTEGER	NOT NULL, FOREIGN KEY	Целое число	FOREIGN KEY
Текст	TEXT	NOT NULL	Текст	–
Оценка	INTEGER	–	Целое число	–
Дата	TEXT	NOT NULL	Текст (дата)	–
Код_пользователя	INTEGER	NOT NULL, FOREIGN KEY	Целое число	FOREIGN KEY

Таблица 11 – Физическая структура данных отношения «Пользователь»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Код_пользователя	INTEGER	PRIMARY KEY, AUTOINCREMENT	Целое число	PRIMARY KEY
Имя_пользователя	TEXT	-	Текст	–
Код_телеграм	INTEGER	UNIQUE	Целое число	UNIQUE
Дата_регистрации	TEXT	–	Текст (дата)	–

Таблица 12 – Физическая структура данных отношения «Спортивный Объект»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Код_объекта	INTEGER	PRIMARY KEY, AUTOINCREMENT	Целое число	PRIMARY KEY
Название	TEXT	NOT NULL	Текст	–
Адрес	TEXT	NOT NULL	Текст	–
Описание	TEXT	NOT NULL	Текст	–
Широта	TEXT	NOT NULL	Текст	–
Долгота	TEXT	NOT NULL	Текст	–
Фото	TEXT	NOT NULL	Текст (путь)	–

Таблица 13 – Физическая структура данных отношения «СпортивныйОбъект_Оборудование»

Название атрибута	Тип данных	Условия	Формат данных	Индексация
Код_объекта	INTEGER	NOT NULL, FOREIGN KEY, PRIMARY KEY	Целое число	PRIMARY KEY, FOREIGN KEY
Код_оборудования	INTEGER	NOT NULL, FOREIGN KEY, PRIMARY KEY	Целое число	PRIMARY KEY, FOREIGN KEY

Физическое проектирование также предусматривает построение структуры физической модели данных, которая представлена на рисунке 17.



Рисунок 17 – Физическая модель базы данных

2.4 Проектирование пользовательского интерфейса

Проектирование пользовательского интерфейса (UI) для веб-приложения «Спортивная карта города» является критически важным этапом, направленным на создание удобного, интуитивно понятного и функционального инструмента для целевой аудитории.

Основной задачей было разработать интерфейс, который бы органично вписывался в экосистему Telegram Mini Apps, обеспечивая при этом быстрый доступ ко всем ключевым функциям приложения [9].

При разработке концепции интерфейса особое внимание уделялось простоте навигации. Пользователь должен легко ориентироваться в приложении, будь то поиск объекта на карте, фильтрация по специфическому оборудованию или добавление нового места. Для этого была выбрана одностраничная архитектура (SPA), где основные секции приложения динамически сменяют друг друга, создавая ощущение целостного и отзывчивого пространства.

Структура интерфейса и основные экраны:

Центральным элементом главного экрана приложения является интерактивная карта Яндекс.Карт, на которой отображаются метки спортивных объектов. Под картой или рядом с ней (в зависимости от ориентации устройства) располагается список этих объектов в виде информативных карточек. В верхней части карты сгруппированы основные элементы взаимодействия: кнопка вызова фильтров, кнопка для отображения избранных объектов, кнопка определения геолокации пользователя и кнопка для перехода в режим добавления

нового спортивного объекта. Ниже карты также предусмотрена область для отображения тегов активных фильтров, что позволяет пользователю всегда видеть текущие критерии поиска. Эскиз главного экрана представлен на рисунке 18.

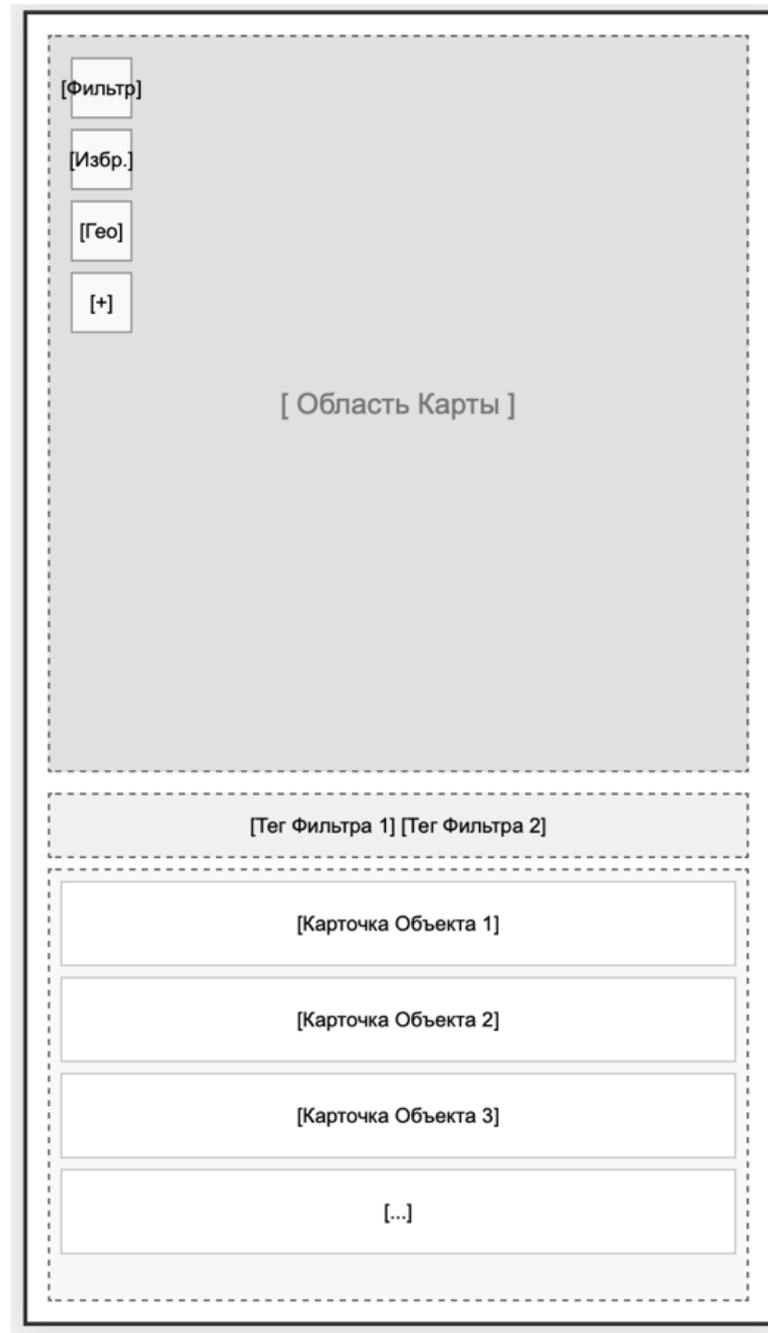


Рисунок 18 – Эскиз главного экрана приложения

При выборе конкретного объекта из списка или по клику на метку на карте, пользователь переходит на экран деталей объекта (рисунок 19).

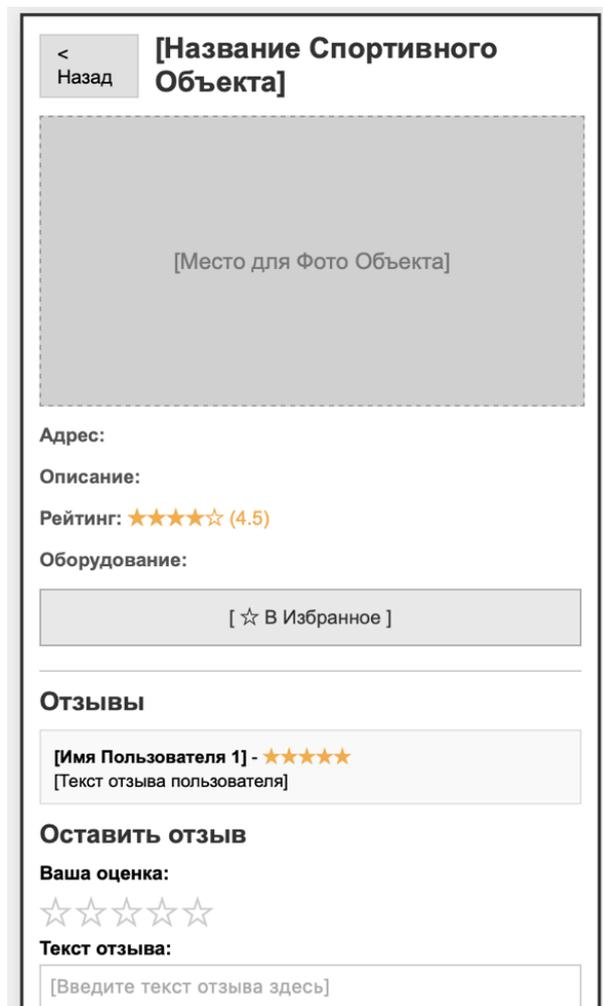


Рисунок 19 – Эскиз экрана деталей объекта

Этот экран спроектирован так, чтобы предоставить исчерпывающую информацию: крупное фото объекта, его название, точный адрес, подробное описание, текущий рейтинг и перечень доступного оборудования. Ниже располагается секция с отзывами других пользователей и форма для добавления собственного отзыва и оценки. Важными элементами этого экрана являются кнопка «Назад» для возврата к основному виду, и кнопка «Добавить в избранное/Удалить из избранного».

Для функции фильтрации спроектировано выпадающее меню или модальное окно, где пользователь может выбрать интересующие типы спортивного оборудования из предложенного списка чекбоксов. После применения или сброса фильтров интерфейс немедленно обновляет как карту, так и список объектов.

Процесс добавления нового объекта начинается с перехода в специальный режим, где пользователю предлагается указать местоположение на карте. После этого появляется модальное окно (рисунок 20) с формой, включающей поля для названия, адреса, описания, кнопку для загрузки фотографии и секцию для выбора имеющегося на площадке оборудования.

Схема модального окна «Добавление нового объекта»:

- Заголовок: **[Добавление нового объекта]**
- Поле для названия: [Название объекта *] / [Поле для ввода названия]
- Поле для адреса: [Адрес * (предзаполнен/редактируем)] / [Поле для адреса]
- Поле для описания: [Описание *] / [Область для текста описания]
- Поле для фото: [Фото *] / [Кнопка "Выбрать фото"] / [Пreview фото]
- Секция оборудования: [Оборудование] с вариантами: [Турник], [Брусья], [...]
- Кнопки: [Отмена] и [Сохранить объект (MainButton)]

Рисунок 20 – Схема модального окна добавления объекта

Проектирование интерфейса велось с учетом ключевых пользовательских сценариев. Например, сценарий «Поиск площадки с турниками и брусьями» предполагает, что пользователь легко найдет кнопку фильтров, выберет нужные опции и увидит результат на карте и в списке. Сценарий «Добавление новой спортивной площадки» продуман так, чтобы минимизировать количество шагов: от указания точки на карте до заполнения простой формы.

Обратная связь с пользователем реализуется через визуальные изменения (например, подсветка активных кнопок фильтров, изменение текста на кнопке «В избранном»), а также через системные уведомления, предоставляемые API Telegram (например, при успешной отправке отзыва или добавлении объекта). Все элементы управления спроектированы таким образом, чтобы быть достаточно крупными для удобного использования на сенсорных экранах мобильных устройств. Адаптивность интерфейса под различные размеры экранов также является важным аспектом проектирования.

Стилистическое оформление интерфейса ориентировано на использование переменных темы Telegram, что позволяет приложению выглядеть нативно и согласованно с общей стилистикой мессенджера, независимо от выбранной пользователем цветовой схемы. Это достигается за счет использования CSS Custom Properties, предоставляемых платформой Telegram Mini Apps.

Таким образом, спроектированный пользовательский интерфейс нацелен на обеспечение эффективного и приятного взаимодействия пользователя с приложением «Спортивная карта города», позволяя ему быстро находить нужную информацию и легко участвовать в наполнении базы данных новыми объектами.

2.5 Проектирование API

Проектирование программного интерфейса приложения (API) является фундаментальным шагом в разработке клиент-серверной архитектуры веб-приложения «Спортивная карта города». API определяет контракт взаимодействия между клиентской частью (Telegram Mini App) и серверной частью, описывая, какие запросы клиент может отправлять, какие данные он должен передавать, и какие ответы он будет получать от сервера [8]. Для данного проекта выбран подход, близкий к стилю RESTful, с использованием стандартных методов HTTP и формата JSON для обмена данными.

Основные HTTP-методы и их применение в API:

В спроектированном API преимущественно используются два основных HTTP-метода: GET и POST.

GET запросы предназначены для получения (извлечения) данных с сервера. Они являются идемпотентными, что означает, что многократное выполнение одного и того же GET-запроса должно приводить к одному и тому же результату и не должно изменять состояние сервера. Параметры для GET-запросов обычно передаются в строке URL (query parameters). Например, для получения списка отзывов по конкретному объекту, его идентификатор будет передан как параметр URL.

POST запросы используются для отправки данных на сервер с целью создания нового ресурса или выполнения операции, которая может изменить состояние сервера. Данные для POST-запросов обычно передаются в теле запроса, чаще всего в формате JSON или как multipart/form-data (если требуется передача файлов, как в случае загрузки фотографий). В отличие от GET, POST-запросы не всегда идемпотентны; например, повторная отправка запроса на создание отзыва приведет к созданию нескольких одинаковых отзывов.

Метод DELETE также будет использоваться для удаления ресурсов, например, при удалении объекта из списка избранного пользователя.

Для всех запросов и ответов в качестве основного формата обмена данными выбран JSON (JavaScript Object Notation) из-за его легковесности, простоты парсинга на стороне клиента (JavaScript) и хорошей читаемости.

Серверная часть, в случае возникновения ошибок, будет возвращать клиенту соответствующие HTTP-статус коды (например, 400 Bad Request, 404 Not Found, 500 Internal Server Error) вместе с JSON-объектом, содержащим описание ошибки. Это позволит клиентской части корректно обрабатывать ошибочные ситуации и информировать пользователя.

На основе анализа функциональных требований к приложению были спроектированы следующие ключевые API-эндпоинты, представленные в таблице 14.

Таблица 14 – Сводка спроектированных API-эндпоинтов

Назначение	URL	Метод	Параметры / Тело запроса	Ожидаемый ответ (успех)
Получить все объекты	/api/objects	GET	–	Массив объектов
Получить список оборудования	/api/objects/equipment	GET	–	Массив оборудования
Получить отзывы для объекта	/api/reviews	GET	object_id	Массив отзывов
Добавить отзыв	/api/review	POST	JSON: {telegram_user_id, telegram_username, object_id, text}	{status: "success"}
Получить избранное пользователя	/api/favorites	GET	telegram_user_id	Массив ID объектов
Добавить в избранное	/api/favorite	POST	JSON: {telegram_user_id, object_id}	{status: "success"} или {status: "already_exists"}
Удалить из избранного	/api/favorite	DELETE	JSON: {telegram_user_id, object_id}	{status: "success"} или {status: "not_found"}
Добавить новый спорт. объект	/api/object/add	POST	FormData: (name, address, photo)	{status: "success", object_id}

Данное проектирование API обеспечивает необходимый набор функций для взаимодействия клиентской и серверной частей, позволяет четко разделить ответственность и закладывает основу для дальнейшей разработки и возможного расширения функционала приложения «Спортивная карта города».

3 РАЗРАБОТКА И РЕАЛИЗАЦИЯ ВЕБ-ПРИЛОЖЕНИЯ

3.1 Выбор средств разработки

Определение подходящего набора инструментальных средств является ключевым этапом проектирования любого программного продукта, включая веб-приложение «Спортивная карта города». Выбранные технологии должны не только соответствовать текущим задачам проекта, но и учитывать перспективы его развития.

Центральным компонентом для хранения информации о спортивных объектах, пользователях, отзывах, оборудовании и избранном является СУБД. Для данного проекта, который на начальном этапе предполагает умеренное количество пользователей и данных, а также требует простоты развертывания и интеграции, SQLite3 представляется наиболее целесообразным выбором. Эта встраиваемая, файловая СУБД не требует настройки отдельного сервера и хранит все данные в одном файле, что значительно упрощает как процесс разработки, так и потенциальное развертывание приложения [10].

С точки зрения финансовых затрат, SQLite3 является полностью бесплатной СУБД, не требующей лицензионных отчислений. PostgreSQL и MariaDB (форк MySQL) также являются СУБД с открытым исходным кодом и бесплатны для использования, но могут потребовать затрат на более производительный хостинг при росте нагрузки.

Рассматривались и более мощные клиент-серверные СУБД, такие как PostgreSQL или MySQL/MariaDB. Они обеспечивают высокую производительность при больших нагрузках, масштабируемость и расширенные функции для сложных запросов и администрирования, что делает их стандартом для крупных веб-приложений. Однако для «Спортивной карты города», особенно на стадии первоначального запуска, их внедрение сопряжено с дополнительными сложностями в настройке и обслуживании, а также с потенциально большими требованиями к хостингу, что не является оправданным на данном этапе.

Сравнительный анализ СУБД представлен в таблице 15.

Таблица 15 – Сравнительный анализ СУБД

Название СУБД	Основные преимущества	Основное назначение	Применимость к проекту
PostgreSQL	Высокая надежность, расширяемость (JSONB, PostGIS), соответствие стандартам SQL.	Крупные веб-приложения, геоинформационные системы, сложные аналитические задачи.	Избыточен на данном этапе.
MySQL/MariaDB	Популярность, большое сообщество, хорошая производительность для типичных веб-задач.	Веб-сайты и приложения различного масштаба, особенно в связке с PHP.	Избыточен на данном этапе.
SQLite3	Простота использования, нулевая конфигурация, встраиваемость, кроссплатформенность.	Локальное хранение данных, мобильные приложения, небольшие веб-проекты, прототипы.	Оптимальный выбор.

Коммерческие СУБД, такие как MS SQL Server или Oracle Database, для данного проекта не рассматривались из-за их высокой стоимости и избыточной функциональности.

Для хранения структурированной информации о спортивных объектах и пользователях, SQLite3 обеспечивает достаточную функциональность при минимальных затратах и сложности.

Выбор языка программирования для серверной части:

Серверная часть приложения будет отвечать за обработку запросов от клиента (ТМА), взаимодействие с базой данных, управление загрузкой файлов и реализацию бизнес-логики. Для этих задач выбран язык программирования Python в связке с микрофреймворком Flask [2].

Python привлекает своим лаконичным синтаксисом, большим количеством доступных библиотек и активным сообществом, что способствует быстрой и эффективной разработке. Flask, будучи микрофреймворком, предоставляет гибкость и минималистичный старт, позволяя подключать только необходимые компоненты. Это идеально подходит для создания API, которое будет обслуживать Mini App.

В качестве альтернатив могли бы рассматриваться:

- Node.js с Express.js: позволил бы использовать JavaScript на всех уровнях приложения, но для задач, связанных с обработкой данных и типичной логикой веб-сервисов, Python часто оказывается более продуктивным;

- PHP с Laravel/Symfony: Традиционный стек для веб-разработки, но Python с Flask предлагает более современный и, для многих разработчиков, более удобный подход к созданию API;

- Python с Django: более полнофункциональный фреймворк, включающий ORM и админ-панель. Для данного проекта его возможности могут быть избыточны, усложняя структуру без существенной необходимости.

Выбор Python и Flask обеспечивает баланс между скоростью разработки, гибкостью и производительностью, достаточной для нужд проекта.

Выбор технологий для клиентской части:

Клиентская часть приложения будет реализована как Telegram Mini App, что означает разработку веб-приложения, работающего внутри мессенджера Telegram. Для этого будут использованы стандартные веб-технологии [3]:

- HTML (HyperText Markup Language): для определения структуры веб-страниц;

- CSS (Cascading Style Sheets): для описания внешнего вида и стилизации элементов;

- JavaScript: для реализации интерактивности, взаимодействия с API Яндекс.Карт, отправки запросов на бэкенд и управления логикой на стороне клиента, включая взаимодействие с API.

Выбор среды разработки (IDE):

Для написания кода на Python, JavaScript, HTML и CSS будет использоваться Visual Studio Code (VS Code). Это легковесный, но мощный и бесплатный редактор кода с открытым исходным кодом, который обладает широкими возможностями кастомизации благодаря огромному количеству плагинов. VS Code отлично подходит для веб-разработки, поддерживает отладку, интеграцию с системами контроля версий (например, Git) и предоставляет удобную среду для работы над проектом.

Таблица 16 – Сравнительный анализ сред разработки

Средство разработки	Преимущества	Недостатки	Применимость к проекту
PyCharm	Мощная IDE, специализированная для Python, глубокая интеграция с фреймворками, продвинутые инструменты отладки.	Может быть ресурсоемкой, платная для полной версии (Community Edition бесплатна).	Отличный выбор, но VS Code более универсален.
WebStorm	Мощная IDE для JavaScript, HTML, CSS, фреймворков.	Платная. Избыточна, если основной фокус на Python-бэкенде.	Избыточна.
VS Code	Легковесность, скорость, бесплатность, огромное количество расширений, кроссплатформенность, отличная поддержка Python и веб-технологий.	Меньше «коробочных» функций для специфичных задач по сравнению с полнофункциональными IDE.	Оптимальный выбор.
Sublime Text/Atom	Легковесные текстовые редакторы, расширяемые плагинами.	Меньше встроенных инструментов для отладки и управления проектом, чем в VS Code.	Хорошие варианты, но VS Code предлагает больше удобств.

Учитывая универсальность, бесплатность и широкие возможности VS Code для работы как с Python, так и с фронтенд-технологиями, он является оптимальным выбором для данного проекта.

Таким образом, для разработки веб-приложения «Спортивная карта города» будет использован следующий стек: Python и Flask для бэкенда, СУБД SQLite3, HTML, CSS и JavaScript для фронтенда (Telegram Mini App), API Яндекс.Карт для картографических функций, и среда разработки Visual Studio Code. Этот набор инструментов обеспечит эффективную реализацию проекта в установленные сроки.

3.2 Реализация клиентской части

3.2.1 Интеграция с API Яндекс.Карт

Одной из центральных задач клиентской части является интеграция с API Яндекс.Карт для отображения интерактивной карты, меток спортивных объектов, а также для реализации функций, связанных с геолокацией и выбором местоположения.

Подключение API Яндекс.Карт осуществляется путем добавления соответствующего скрипта в HTML-документ `index.html`)

В URL подключения указывается API-ключ и необходимые модули. Инициализация карты происходит в JavaScript-коде после полной загрузки DOM и API Яндекс.Карт. В процессе инициализации задаются начальные параметры карты: центр (например, координаты Благовещенска), уровень масштабирования и набор элементов управления.

Для отображения спортивных объектов на карте используются метки (`ymaps.Placemark`). Каждая метка создается с указанием координат объекта, а также может содержать информацию для балуна (всплывающего окна при клике на метку) и хинта (подсказки при наведении). Метки группируются в коллекцию (`ymaps.GeoObjectCollection`), что упрощает управление ими (например, одновременное добавление или удаление).

Интеграция также включает использование модуля `geocode` для определения адреса по координатам, что применяется, например, при добавлении

нового объекта пользователем после указания точки на карте. Функция `umaps.geocode(coords).then(...)` позволяет асинхронно получить адрес.

Для определения местоположения пользователя используется `umaps.geolocation.get()`, что позволяет центрировать карту на текущей позиции пользователя и отобразить соответствующую метку.

3.2.2. Разработка пользовательского интерфейса

Пользовательский интерфейс (UI) разработан с использованием HTML для структурирования элементов, CSS для их стилизации и JavaScript для управления их поведением и динамическим обновлением. Основной целью было создание интуитивно понятного, адаптивного и визуально согласованного с экосистемой Telegram интерфейса.

Структура интерфейса (`index.html`) включает следующие основные блоки:

- контейнер для карты (`<div id="map-container">`);
- контейнер для элементов управления на карте (кнопки фильтров, геолокации, добавления объекта);
- контейнер для списка объектов (`<div id="objects-list-container">`), который динамически заполняется карточками объектов;
- блок для отображения детальной информации об объекте (`<div id="object-details">`), который показывается при выборе объекта;
- Модальное окно для добавления нового объекта (`<div id="add-object-modal">`).

Стилизация элементов (`style.css`) выполнена с учетом переменных темы Telegram (`--tg-theme-bg-color`, `--tg-theme-text-color` и т.д.), что обеспечивает адаптацию внешнего вида приложения под текущую тему пользователя в мессенджере. Используются Flexbox и Grid для построения адаптивных макетов.

3.3 Реализация серверной части

Серверная часть веб-приложения «Спортивная карта города» разработана на языке программирования Python с использованием микрофреймворка Flask. Она выполняет роль API-сервера, обрабатывая запросы от клиентского Telegram Mini App, взаимодействуя с базой данных SQLite3 для хранения и

извлечения данных, а также управляя загрузкой пользовательских файлов (фотографий объектов).

Рассмотрим инициализация приложения и конфигурация:

Основой серверной части является экземпляр приложения Flask. При запуске настраивается подключение к базе данных `DATABASE = './db/db.sqlite3'` и определяется папка для загружаемых файлов `UPLOAD_FOLDER_PATH`. Эта папка создается автоматически, если не существует. Также определен список разрешенных расширений для загружаемых изображений (`ALLOWED_EXTENSIONS`).

Для работы с базой данных SQLite3 реализована функция `get_db_connection()`, которая устанавливает соединение и настраивает `conn.row_factory = sqlite3.Row` для удобного доступа к данным по именам колонок. Все операции с базой данных выполняются в рамках контекстного менеджера `with get_db_connection() as conn:`, что обеспечивает автоматическое закрытие соединения и фиксацию (`commit`) транзакций при успешном выполнении, либо откат (`rollback`) в случае ошибок.

Обработка данных пользователей:

Ключевой для многих операций является функция `get_or_create_user(telegram_id, username)`. Она проверяет наличие пользователя в таблице Пользователь по его Код_телеграм. Если пользователь не найден, он создается с текущей датой регистрации. Эта функция гарантирует, что для каждого взаимодействия с системой (добавление отзыва, избранного, объекта) существует соответствующая запись о пользователе в БД.

Серверная часть предоставляет API-эндпоинты для взаимодействия с клиентским приложением.

Формирование ответов и обработка ошибок:

Все ответы API формируются в формате JSON с использованием функции `jsonify()`. При возникновении ошибок (например, отсутствия обязательных полей, неверного формата данных, серверных сбоев) клиенту возвращаются

соответствующие HTTP-коды состояния (400, 404, 500) и JSON-объект с описанием ошибки. Используется `app.logger` для логирования ошибок на сервере.

Общая логика взаимодействия реализованных клиентской и серверной частей, а также их обращений к базе данных, наглядно показана на рисунке В.1 в Приложении В, представляющем логическую структуру программы.

3.4 Описание работы приложения

При запуске приложения открывается карта города Благовещенска с несколькими метками спортивных объектов. Под картой отображается начальный список объектов в виде карточек (название, адрес, фото, рейтинг). В левом верхнем углу карты видны кнопки управления (Фильтры, Избранное, Геолокация, Добавить). Главный экран приложения представлен на рисунке 21.

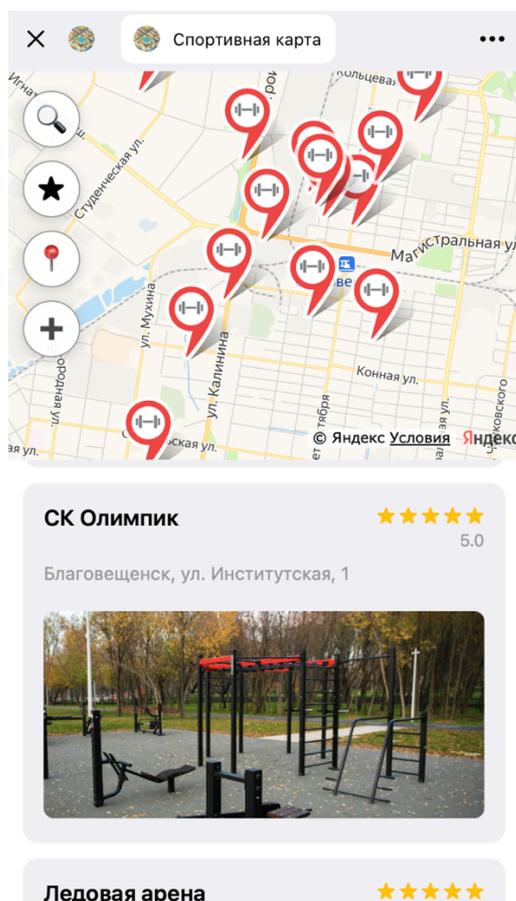


Рисунок 21 – Главный экран приложения

При нажатии на кнопку поиск открывается меню фильтров (рисунок 22). В меню видны чекбоксы с названиями оборудования («Турник», «Брусья», «Беговая дорожка» и т.д.). Пользователь выбрал «Брусья» и «Турник».

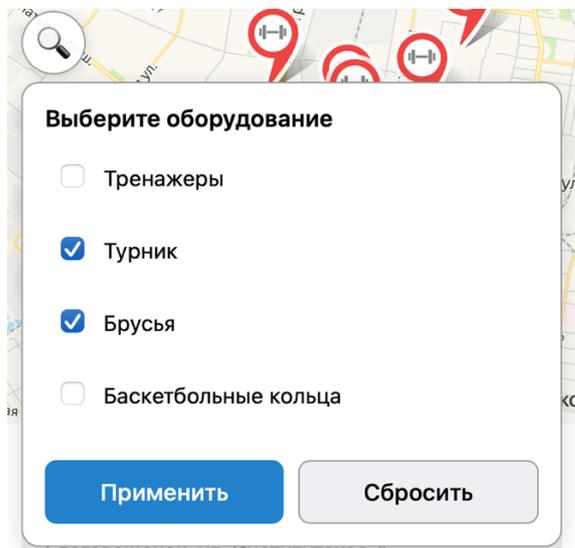


Рисунок 22 – Процесс фильтрации по оборудованию

После применения фильтров «Брусья» и «Турник». Меню фильтров скрывается. На карте и в списке под ней отображаются только те объекты, которые соответствуют выбранным критериям. Внизу карты (или над списком) видны теги активных фильтров (рисунок 23).

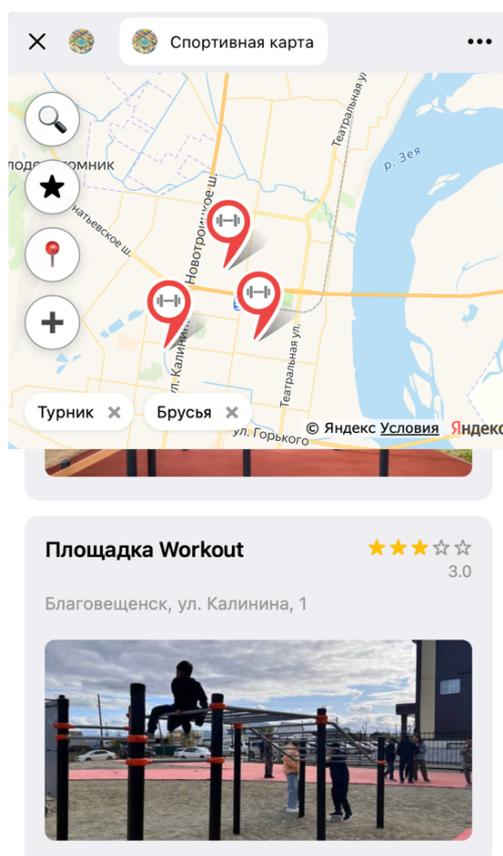


Рисунок 23 – Отображение активных фильтров

После выбора спортивного объекта открывается экран с подробной информацией. Видны: кнопка «Назад», название объекта, крупное фото, адрес, описание, средний рейтинг (звезды и цифра) и кнопка добавления в избранное (рисунок 24).

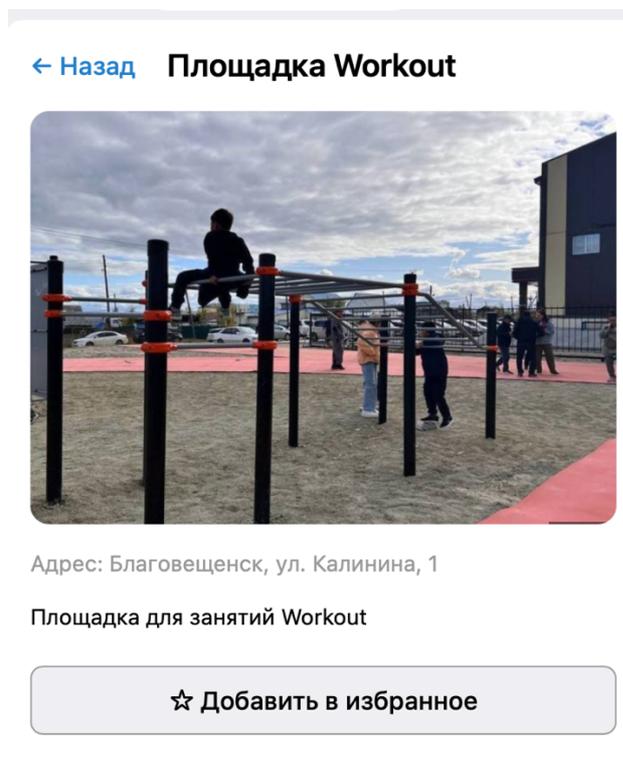


Рисунок 24 – Экран детальной информации об объекте

Продолжение экрана деталей объекта (прокрутка вниз). Показан заголовок «Отзывы», список существующих отзывов (дата, оценка, текст). Ниже расположена форма для добавления нового отзыва с интерактивными звездами для оценки, текстовым полем и кнопкой «Отправить отзыв» (рисунок 25).

Ваш отзыв

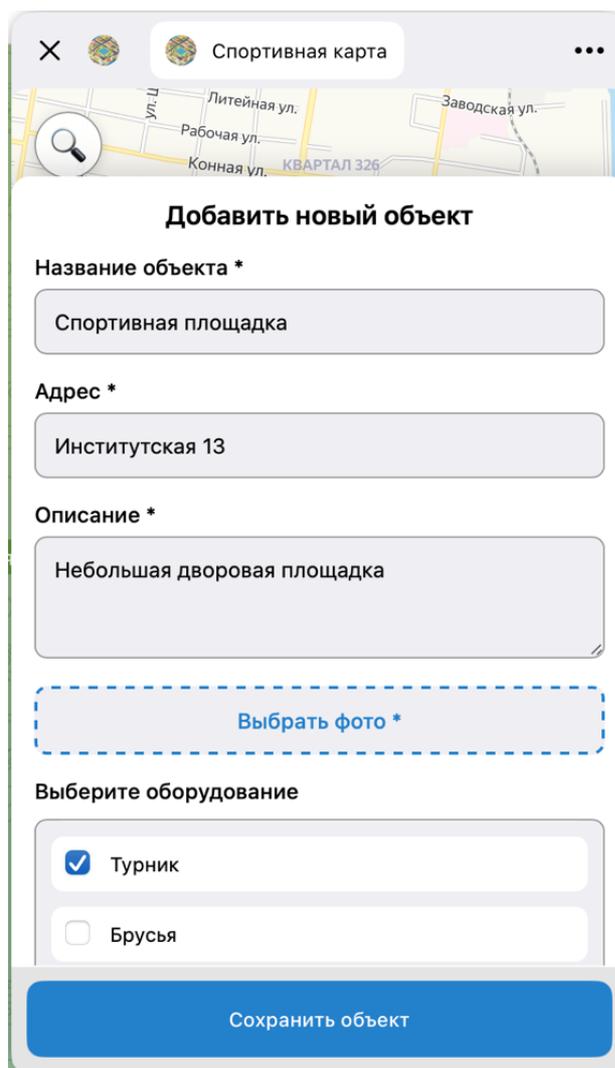
Оценка: ☆☆☆☆

Напишите, что думаете об этом месте...

Отправить отзыв

Рисунок 25 – Секция отзывов и форма добавления отзыва

При нажатии на кнопку «Добавить объект» появляется системное сообщение или подсказка о дальнейших действиях. Пользователю предлагается установить метку нового объекта на карте, затем открывается модальное окно с формой для ввода данных нового объекта (рисунок 26). Видны поля: «Название», «Адрес» (предзаполнен по координатам метки), «Описание», кнопка выбора фото, секция с чекбоксами оборудования. Внизу кнопка «Отмена». MainButton Telegram внизу экрана имеет текст «Сохранить объект»



Скриншот интерфейса приложения «Спортивная карта» в режиме добавления нового объекта. В верхней части экрана отображена карта с меткой. Модальное окно «Добавить новый объект» содержит следующие элементы:

- Название объекта *: Спортивная площадка
- Адрес *: Институтская 13
- Описание *: Небольшая дворовая площадка
- Кнопка «Выбрать фото *» (обозначена пунктирной линией)
- Секция «Выберите оборудование»:
 - Турник
 - Брусья
- Кнопка «Сохранить объект» (синяя)

Рисунок 26 – Режим добавления нового объекта

После нажатия кнопки «Избранное» на карте и в списке отображаются только те объекты, которые пользователь ранее добавил в избранное, так же виден тег активного фильтра (рисунок 27).

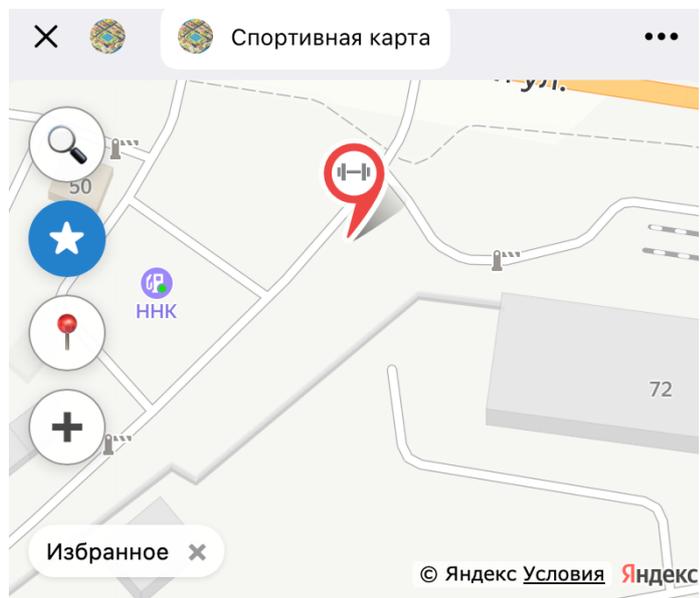


Рисунок 27 – Отображение избранных объектов

Все функции веб-приложения работоспособны. Реализованный интерфейс соответствует спроектированным макетам и сценариям использования.

4 БЕЗОПАСНОСТЬ И ЭКОЛОГИЧНОСТЬ

4.1 Безопасность

Для специалиста, занимающегося созданием веб-приложения «Спортивная карта города», основной объем работы связан с использованием персонального компьютера (ПК) и сопутствующего офисного оборудования. Длительная работа в таких условиях может приводить к возникновению профессиональных рисков для здоровья, связанных с малоподвижным образом жизни, напряжением зрения и опорно-двигательного аппарата. Поэтому крайне важно создать условия труда, минимизирующие эти риски и обеспечивающие высокую работоспособность [11].

Комфортные и безопасные условия труда на рабочем месте разработчика формируются совокупностью факторов производственной среды. Ключевые гигиенические нормативы и требования к этим факторам установлены в СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

– микроклимат: в помещениях, где осуществляется работа с ПК, должны поддерживаться оптимальные параметры температуры, относительной влажности и скорости движения воздуха. Для теплого периода года рекомендуемая температура воздуха составляет 23-25 °С, для холодного – 22-24 °С. Относительная влажность воздуха должна находиться в пределах 40-60 %. Скорость движения воздуха не должна превышать 0,1 м/с. Регулярное проветривание и, при необходимости, использование систем кондиционирования и увлажнения воздуха способствуют поддержанию комфортного микроклимата;

– освещение: рабочее место должно иметь как естественное, так и искусственное освещение. Естественное освещение обеспечивается через оконные проемы, которые следует располагать так, чтобы прямые солнечные лучи не создавали бликов на экране монитора и не слепили пользователя. Искусственное освещение должно быть равномерным, достаточным и не создавать резких

теней. Рекомендуется использовать комбинированную систему освещения (общее и местное). Светильники общего освещения должны располагаться таким образом, чтобы исключить отраженную блесккость на экране;

– шум и вибрация: уровни шума на рабочем месте не должны превышать допустимых значений. Источниками шума могут быть офисная техника, разговоры коллег, системы вентиляции. Для снижения уровня шума могут применяться звукопоглощающие материалы в отделке помещения, а также рациональное размещение оборудования. Вибрация на рабочих местах с ПК, как правило, отсутствует или находится в пределах допустимых норм;

– электромагнитные поля (ЭМП): современные персональные компьютеры и мониторы, соответствующие ГОСТ Р 50948-2001 «Средства отображения информации индивидуального пользования. Общие эргономические требования и требования безопасности», генерируют ЭМП на уровнях, не превышающих установленные гигиенические нормативы. Тем не менее, рекомендуется соблюдать безопасные расстояния от источников ЭМП и использовать сертифицированное оборудование.

Правильная организация рабочего места разработчика имеет решающее значение для предотвращения утомляемости и развития профессиональных заболеваний, связанных с длительной работой в сидячем положении и напряжением зрительного аппарата. Общие требования к организации безопасного рабочего места утверждены Приказом Министерства труда и социальной защиты РФ от 29 октября 2021 г. N 774н.

Эргономика:

– рабочий стол: высота стола должна обеспечивать удобное положение рук и ног. Рекомендуется глубина стола, достаточная для размещения монитора на оптимальном расстоянии, и ширина, позволяющая свободно разместить клавиатуру, мышь и необходимые рабочие материалы [6];

– рабочее кресло: кресло должно быть подъемно-поворотным, регулируемым по высоте сиденья и спинки, а также по углу наклона спинки. Оно

должно обеспечивать поддержку поясничного отдела позвоночника и иметь подлокотники [2];

– размещение монитора: экран монитора должен находиться на расстоянии 60-70 см от глаз пользователя, а его верхний край – на уровне глаз или немного ниже. Это минимизирует напряжение мышц шеи и глаз. Необходимо избегать бликов на экране от источников света;

– клавиатура и мышь: клавиатура должна располагаться так, чтобы при работе предплечья были параллельны полу, а кисти рук находились в нейтральном положении. Мышь должна находиться на одном уровне с клавиатурой и в непосредственной близости от нее;

– пространство и планировка: рабочее место должно обеспечивать достаточное пространство для движений и смены рабочей позы. Согласно ГОСТ Р 58751-2019, телекоммуникационные пространства и помещения, включая рабочие места, должны соответствовать определенным требованиям по площади и объему на одного работающего;

– режим труда и отдыха: длительная непрерывная работа за компьютером приводит к утомлению. Необходимо соблюдать регламентированные перерывы. Во время перерывов рекомендуется выполнять упражнения для глаз и легкую физическую разминку для снятия мышечного напряжения [12].

Организация графического интерфейса пользователя (GUI)

Качество графического интерфейса программных средств, с которыми работает разработчик (операционная система, среда разработки, текстовые редакторы и т.д.), также влияет на его безопасность и комфорт. Эргономические требования к элементам графического пользовательского интерфейса изложены в ГОСТ Р ИСО 9241-161-2016:

– визуальное оформление: интерфейс должен иметь четкое и контрастное отображение информации. Цветовые схемы должны быть подобраны таким образом, чтобы не вызывать зрительного утомления. Размеры шрифтов должны быть достаточными для легкого чтения без напряжения глаз;

– информативность и предсказуемость: элементы управления должны быть интуитивно понятными, а их реакция на действия пользователя – предсказуемой. Система должна предоставлять своевременную обратную связь о выполняемых операциях и возникающих ошибках;

– доступность: хотя данный аспект в большей степени относится к разрабатываемому продукту, согласно ГОСТ Р 52872—2019, разработчик также выигрывает от использования инструментов, поддерживающих принципы доступности, так как это часто коррелирует с общим удобством использования.

Соблюдение перечисленных требований к условиям труда, организации рабочего места и используемым графическим интерфейсам позволяет создать безопасную и продуктивную среду для разработчика, снизить риск возникновения профессиональных заболеваний и повысить общую эффективность его работы над проектом «Спортивная карта города».

4.2 Экологичность

Деятельность по разработке программного обеспечения, такая как создание веб-приложения «Спортивная карта города», на первый взгляд, может показаться экологически нейтральной. Однако использование офисной техники, потребление электроэнергии и образование различных видов отходов неизбежно оказывают определенное воздействие на окружающую среду. Ответственный подход к минимизации этого воздействия является важным аспектом современной профессиональной деятельности.

Основными источниками отходов при работе разработчика являются:

– вышедшая из строя или устаревшая оргтехника: персональные компьютеры, мониторы, принтеры, сканеры, клавиатуры, мыши и другое периферийное оборудование. Эта техника содержит различные материалы, включая пластик, металлы (в том числе драгоценные и цветные), стекло, а также потенциально опасные компоненты (например, в старых мониторах или батареях);

– отработанные расходные материалы: картриджи для принтеров, батареи, аккумуляторы;

– макулатура: бумага, использованная для печати документов, чернови-ков, упаковочные материалы;

– люминесцентные лампы: если используются для освещения помеще-ний, содержат ртуть и требуют особого обращения;

– офисная мебель: столы, стулья, шкафы, пришедшие в негодность.

В соответствии с действующим законодательством Российской Федера-ции, в частности, Федеральным законом № 89-ФЗ «Об отходах производства и потребления» и СанПиН 2.1.3684-21 «Санитарно-эпидемиологические тре-бования к содержанию территорий городских и сельских поселений, к водным объектам, питьевой воде и питьевому водоснабжению населения, атмосфер-ному воздуху, почвам, жилым помещениям, эксплуатации производственных, общественных помещений, организации и проведению санитарно-противо-эпидемических (профилактических) мероприятий», отходы, образующиеся в процессе деятельности организаций, подлежат сбору, накоплению, транспор-тированию, обработке, утилизации, обезвреживанию и размещению таким об-разом, чтобы минимизировать их негативное воздействие на окружающую среду и здоровье человека.

Рассмотрим организацию обращения с отходами в типичной организа-ции:

Накопление и временное хранение:

– оргтехника и электронное оборудование: списанная техника накапли-вается в специально отведенном месте (складское помещение или выделенная зона), недоступном для несанкционированного доступа. Важно предотвратить ее разукomплектование лицами, не имеющими соответствующей квалифика-ции, так как это может привести к выбросу вредных веществ;

– люминесцентные лампы: отработанные ртутьсодержащие лампы должны храниться в герметичной таре (специальных контейнерах), исключая их повреждение и утечку ртути;

– батарейки и аккумуляторы: собираются отдельно от других отходов в специальные контейнеры, предназначенные для их последующей передачи на переработку или утилизацию;

– картриджи от принтеров: могут накапливаться для последующей перезаправки (если это предусмотрено) или передачи специализированным организациям для утилизации;

– макулатура: собирается в отдельные контейнеры или связывается в кипы для удобства транспортировки;

– офисная мебель: крупногабаритные отходы, такие как мебель, могут временно складироваться на специально отведенной площадке до момента вывоза.

Передача на утилизацию/переработку:

Все вышеперечисленные отходы, особенно содержащие опасные компоненты, не могут быть выброшены вместе с твердыми коммунальными отходами (ТКО). Организация должна заключать договоры со специализированными компаниями, имеющими лицензию на деятельность по сбору, транспортированию, обработке, утилизации, обезвреживанию, размещению отходов I–IV классов опасности.

– оргтехника передается на аффинаж (извлечение драгоценных металлов) и последующую утилизацию компонентов;

– люминесцентные лампы, батарейки и аккумуляторы передаются на обезвреживание и переработку;

– макулатура сдается на переработку, что способствует сохранению лесных ресурсов;

– изношенная мебель вывозится на полигоны ТКО или, по возможности, передается на переработку (например, древесные компоненты).

Соответствие современному законодательству:

– современное российское законодательство в области обращения с отходами направлено на предотвращение их образования, сокращение объемов и максимальное вовлечение в хозяйственный оборот в качестве вторичных

материальных ресурсов. Организации обязаны вести учет образующихся отходов, разрабатывать паспорта опасных отходов (при их наличии) и представлять соответствующую отчетность. Для того чтобы деятельность по обращению с отходами соответствовала законодательству, организация должна:

- иметь утвержденные инструкции по обращению с различными видами отходов;
- назначить ответственных лиц за сбор, хранение и передачу отходов;
- обеспечить отдельный сбор отходов;
- заключать договоры только с лицензированными организациями на вывоз и утилизацию отходов;
- вести документацию, подтверждающую факт передачи отходов (акты приема-передачи, накладные).

Снижение экологического воздействия проекта:

Хотя непосредственная разработка веб-приложения не генерирует большого количества промышленных отходов, разработчик может способствовать снижению общего экологического следа организации, придерживаясь принципов «зеленого офиса»:

- энергосбережение: выключать компьютер и периферийные устройства по окончании рабочего дня, использовать энергосберегающие режимы.
- экономия бумаги: по возможности использовать электронный документооборот, печатать с двух сторон листа;
- продление срока службы оборудования: бережное отношение к технике может отсрочить момент ее списания;
- участие в программах отдельного сбора: активно участвовать в сборе макулатуры, пластика (если организовано в офисе).

Таким образом, экологичность разработки веб-приложения «Спортивная карта города» косвенно связана с общей экологической политикой организации, в рамках которой ведется разработка. Соблюдение требований законодательства по обращению с отходами, образующимися от использования

офисной техники и других материалов, является ключевым фактором минимизации негативного воздействия на окружающую среду.

4.3 Чрезвычайные ситуации

Работа в офисном помещении, где осуществляется разработка веб-приложения «Спортивная карта города», сопряжена с потенциальными рисками возникновения чрезвычайных ситуаций (ЧС) техногенного характера. Наиболее вероятной и опасной из них является пожар. Поэтому в организации должны быть предусмотрены и реализованы меры по предупреждению и ликвидации таких ЧС, направленные на обеспечение безопасности сотрудников и сохранности имущества.

Возможные причины возникновения пожара в офисном помещении:

– неисправность электрооборудования и электропроводки: короткое замыкание, перегрузка сети, использование неисправных электроприборов (компьютеры, принтеры, обогреватели, чайники), повреждение изоляции проводов;

– неосторожное обращение с огнем: курение в неустановленных местах, использование открытого огня вблизи горючих материалов (хотя для офисов это менее характерно, но полностью исключать нельзя);

– нарушение правил пожарной безопасности при проведении ремонтных или иных работ: использование оборудования, не соответствующего требованиям безопасности, несоблюдение мер предосторожности;

– поджог: умышленные действия;

– внешние факторы: пожар в соседних помещениях или зданиях, распространившийся на офис.

Меры по предупреждению пожара в организации:

Для предотвращения возникновения пожара и минимизации его последствий в организации должен быть реализован комплекс организационных и технических мероприятий в соответствии с Федеральным законом № 123-ФЗ «Технический регламент о требованиях пожарной безопасности» и Правилами противопожарного режима в Российской Федерации.

Организационные мероприятия:

– назначение ответственных лиц: назначение приказом руководителя лиц, ответственных за пожарную безопасность в организации и в отдельных помещениях.

– разработка документации: разработка и утверждение инструкций о мерах пожарной безопасности, планов эвакуации людей при пожаре.

– обучение и инструктажи: проведение регулярных инструктажей по пожарной безопасности (вводного, первичного, повторного, внепланового, целевого) со всеми сотрудниками. Обучение персонала действиям при пожаре, включая использование первичных средств пожаротушения и порядок эвакуации.

– содержание территории и помещений: регулярная уборка помещений от горючих отходов и пыли. Запрет на загромождение эвакуационных путей и выходов мебелью, оборудованием или другими предметами. Оборудование специальных мест для курения (если курение разрешено на территории).

– контроль за электрооборудованием: регулярная проверка состояния электропроводки, электрооборудования, розеток и выключателей. Запрет на использование неисправных приборов и самодельных удлинителей. По окончании рабочего дня все электроприборы, не предназначенные для круглосуточной работы, должны отключаться от сети.

Технические мероприятия:

– системы автоматической пожарной сигнализации (АПС) и оповещения и управления эвакуацией людей при пожаре (СОУЭ): Офисные помещения должны быть оборудованы АПС, способной обнаружить возгорание на ранней стадии, и СОУЭ для своевременного оповещения людей о пожаре и путях эвакуации. Эти системы должны регулярно проверяться и поддерживаться в исправном состоянии специализированными организациями;

– первичные средства пожаротушения: наличие и исправное состояние огнетушителей (порошковых, углекислотных) в соответствии с нормами и их типом, подходящим для тушения возможных классов пожаров (включая

электрооборудование). Размещение огнетушителей в легкодоступных местах, обозначенных специальными знаками. Сотрудники должны быть обучены правилам их применения;

– эвакуационные пути и выходы: обеспечение достаточного количества эвакуационных выходов, их свободное состояние. Наличие на путях эвакуации знаков пожарной безопасности, указывающих направление движения. Двери эвакуационных выходов должны открываться по направлению выхода из здания;

– противопожарные преграды: при необходимости, использование противопожарных дверей, перегородок для ограничения распространения огня;

– внутренний противопожарный водопровод (при наличии): Поддержание в исправном состоянии пожарных кранов, укомплектованность их рукавами и стволами.

Действия персонала при возникновении пожара:

При обнаружении пожара или признаков горения (задымление, запах гари, повышение температуры) каждый сотрудник обязан:

– немедленно сообщить о пожаре;

– принять меры по эвакуации людей;

– приступить к тушению пожара первичными средствами (если это безопасно);

– организовать встречу подразделений пожарной охраны.

Соблюдение всех перечисленных мер по предупреждению пожара и готовность персонала к правильным действиям в случае его возникновения являются залогом обеспечения безопасности сотрудников и минимизации возможных последствий чрезвычайной ситуации в офисе, где ведется разработка веб-приложения «Спортивная карта города».

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы было успешно спроектировано и разработано веб-приложение «Спортивная карта города». Проведен всесторонний анализ предметной области, выявлены потребности пользователей и недостатки существующих решений, что обосновало актуальность проекта.

На этапе проектирования была выбрана клиент-серверная архитектура с использованием технологий Flask (Python) для серверной части и стандартных веб-технологий (HTML, CSS, JavaScript) для клиентской части. База данных спроектирована с использованием СУБД. Особое внимание уделено интеграции с API Яндекс.Карт для картографических функций и API Telegram для взаимодействия с интерфейсом мессенджера. Пользовательский интерфейс спроектирован с учетом удобства использования и адаптивности.

Реализация проекта включила разработку серверного API для обработки запросов, управления данными спортивных объектов, отзывов, избранного и пользовательской информации, а также клиентской части, обеспечивающей интерактивное взаимодействие с картой и данными. Продемонстрирована работоспособность ключевых функций: отображение объектов на карте, фильтрация по оборудованию, просмотр детальной информации, добавление объектов в избранное, оставление отзывов и добавление новых спортивных площадок пользователями.

Разработанное приложение предоставляет пользователям удобный инструмент для поиска спортивной инфраструктуры города и способствуя популяризации здорового образа жизни. Дальнейшее развитие проекта может включать расширение набора фильтров, внедрение системы уведомлений и улучшение социальных функций.

БИБЛИОГРАФИЧЕСКИЕ ССЫЛКИ

- 1 Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс. Пер. с англ. М.: Вильямс, 2018. 1088 с.
- 2 Гринберг М. Разработка веб-приложений с использованием Flask на языке Python. Пер. с англ. М.: ДМК Пресс, 2019. 288 с.
- 3 Дакетт Д. HTML и CSS. Разработка и дизайн веб-сайтов. Пер. с англ. М.: Эксмо, 2021. 496 с.
- 4 Документация API Яндекс.Карт [Электронный ресурс]. URL: <https://yandex.ru/dev/maps/jsapi/doc/2.1/quick-start/index.html> (дата обращения: 01.04.2025).
- 5 Документация Telegram Mini Apps [Электронный ресурс]. URL: <https://core.telegram.org/bots/webapps> (дата обращения: 01.04.2025).
- 6 Кардаш Т. А. Эргономика рабочих мест служащих и инженерно-технических работников, оснащенных ПЭВМ: учеб. пособие АмГУ, ИФФ. Благовещенск: Изд-во Амур. гос. ун-та, 2002. 60 с.
- 7 Круг С. Не заставляйте меня думать! Веб-юзабилити и здравый смысл. Пер. с англ. СПб.: Питер, 2021. 216 с.
- 8 Массе А. Проектирование веб-API. Пер. с англ. СПб.: Питер, 2021. 368 с.
- 9 Норман Д. А. Дизайн привычных вещей. Пер. с англ. М.: Манн, Иванов и Фербер, 2021. 384 с.
- 10 Оуэнс К. SQLite. В кратчайшие сроки. Пер. с англ. М.: ДМК Пресс, 2019. 272 с.
- 11 Пособие по безопасной работе на персональных компьютерах. Разраб. В. К. Шумилин. М.: НЦ ЭНАС, 2005. 28 с.
- 12 Шумилин В. К. ПЭВМ. Защита пользователя. М.: Охрана труда и социальное страхование, 2001. 214 с.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Бадд, Э. Flask для профессионалов / Э. Бадд; пер. с англ. – СПб.: Питер, 2021. – 304 с.
- 2 Головач, В. В. Дизайн пользовательского интерфейса. Искусство мыть слона / В. В. Головач. – М.: ДМК Пресс, 2020. – 488 с.
- 3 Гримальдо, А. Python для сложных задач. Наука о данных и машинное обучение / А. Гримальдо; пер. с англ. – Санкт-Петербург: Питер, 2023. – 688 с.
- 4 Гронлунд, М. Flask. Веб-разработка на Python / М. Гронлунд; пер. с англ. – Санкт-Петербург: БХВ-Петербург, 2020. – 352 с.
- 5 Дейт, К. Дж. Введение в системы баз данных / К. Дж. Дейт; пер. с англ. – 8-е изд. – Москва: Вильямс, 2019. – 1328 с.
- 6 Документация Flask [Электронный ресурс]. – Режим доступа : <https://flask.palletsprojects.com/>. – 01.04.2025.
- 7 Ларман, К. Применение UML и шаблонов проектирования / К. Ларман; пер. с англ. – М.: Вильямс, 2006. – 736 с.
- 8 Лутц, М. Изучаем Python: в 2 т. Т. 1 / М. Лутц; пер. с англ. – 5-е изд. – Санкт-Петербург: Диалектика, 2021. – 800 с.
- 9 Макконелл, С. Совершенный код. Мастер-класс / С. Макконелл; пер. с англ. – Москва: Русская Редакция, 2021. – 896 с.
- 10 Мартин, Р. Идеальный программист. Как стать профессионалом разработки ПО / Р. Мартин; пер. с англ. – СПб.: Питер, 2021. – 224 с.
- 11 Мартин, Р. Чистый код: создание, анализ и рефакторинг / Р. Мартин; пер. с англ. – Санкт-Петербург: Питер, 2022. – 464 с.
- 12 Маттиас, К. Docker для разработчиков / К. Маттиас, Ш. П. Кейн; пер. с англ. – М. : ДМК Пресс, 2022. – 304 с.
- 13 Орлов, С. А. Технологии разработки программного обеспечения: Учебник для вузов / С. А. Орлов. – 4-е изд. – СПб.: Питер, 2018. – 608 с.
- 14 Рамальо, Л. Python. К вершинам мастерства / Л. Рамальо; пер. с англ. – Москва: ДМК Пресс, 2022. – 768 с.

15 Соммервилл, И. Инженерия программного обеспечения / И. Соммервилл; пер. с англ. – 10-е изд. – М.: Диалектика, 2019. – 768 с.

16 Стандарт организации. Оформление выпускных квалификационных и курсовых работ (проектов): СТО СМК 4.2.3.21-2018 / АмГУ; разработ. Л. А. Проказина, Н. А. Чалкина, С. Г. Самохвалова. – Благовещенск, 2018. – 75 с. – Режим доступа : https://irbis.amursu.ru/DigitalLibrary/AmurSU_Edition/9702.pdf – 01.04.2025.

17 Фаулер, М. Рефакторинг. Улучшение проекта существующего кода / М. Фаулер; пер. с англ. – 2-е изд. – М.: Вильямс, 2019. – 448 с.

18 Флэнаган, Д. JavaScript. Подробное руководство / Д. Флэнаган; пер. с англ. – 7-е изд. – Санкт-Петербург: Диалектика, 2022. – 720 с.

19 Фримен, Э. Изучаем программирование на Python / Э. Фримен, Э. Робсон; пер. с англ. – Санкт-Петербург: Питер, 2020. – 720 с.

20 Хант, Э. Программист-прагматик. Путь от подмастерья к мастеру / Э. Хант, Д. Томас пер. с англ. – Санкт-Петербург: Питер, 2021. – 368 с.

21 Хоган, Б. Веб-разработка с применением Python и Django / Б. Хоган, Д. Массе пер. с англ. – Москва: ДМК Пресс, 2021. – 576 с.

22 Цекова, Е. Веб-дизайн. Книга Джесса Гарретта. Элементы опыта взаимодействия / Е. Цекова, Д. Гарретт; пер. с англ. – Санкт-Петербург: Символ-Плюс, 2019. – 192 с.

23 Чирт, К. SQLite для профессионалов / К. Чирт; пер. с англ. – М.: ДМК Пресс, 2022. – 350 с.

ПРИЛОЖЕНИЕ А

Техническое задание

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Полное наименование системы

Веб-приложение «Спортивная карта города».

Условное обозначение: SportMapCity.

1.2 Плановые сроки:

- начало работ: 20.02.2024 г;
- Окончание работ: 10.06.2025 г.

2 НАЗНАЧАНИЕ И ЦЕЛИ СИСТЕМЫ

2.1 Назначение системы

Платформа предназначена для предоставления пользователям удобного инструмента поиска и получения актуальной информации о спортивных объектах (площадках, залах, секциях и т.д.) в городе. Система обеспечивает отображение объектов на интерактивной карте, фильтрацию по различным параметрам (тип объекта, наличие оборудования, доступность), а также возможность добавления и оценки объектов пользователями.

2.2 Цели создания системы

- создать централизованный и актуальный источник информации о спортивной инфраструктуре города.
- упростить поиск спортивных объектов, соответствующих индивидуальным потребностям пользователей.
- повысить информированность граждан о доступных местах для занятий спортом.
- способствовать вовлечению населения в активный образ жизни.
- предоставить пользователям платформу для обмена мнениями и оценками спортивных объектов.

Продолжение ПРИЛОЖЕНИЯ А

Критерии оценки успеха:

- увеличение базы данных спортивных объектов на 20 % за первые 6 месяцев эксплуатации за счет пользовательского контента;
- достижение 100+ активных пользователей в неделю к концу первого года эксплуатации;
- средняя оценка качества информации об объектах (по отзывам пользователей) не ниже 4 из 5;
- снижение среднего времени, затрачиваемого пользователем на поиск подходящего спортивного объекта с определенным набором оборудования, на 30 %;
- количество добавленных пользователями отзывов и оценок объектов достигает не менее 50 в месяц.

3 ТРЕБОВАНИЯ К СИСТЕМЕ

3.1 Требования к системе в целом

- структура и функционирование: Система должна состоять из клиентской части (Telegram Mini App) и серверной части (API на Python/Flask);
- клиентская часть: отображение карты (API Яндекс.Карт), интерфейсы для поиска, фильтрации, просмотра информации об объектах, добавления объектов, отзывов, управления избранным;
- серверная часть: обработка запросов от клиента, взаимодействие с базой данных (SQLite3), бизнес-логика (фильтрация, расчет рейтингов, аутентификация, управление пользовательским контентом);
- численность и квалификация персонала: для модерации контента и управления системой требуется администратор с базовыми навыками работы с ПК и веб-интерфейсами;
- надежность: система должна обеспечивать стабильную работу при нагрузке до 500 одновременных пользователей. Время простоя системы не должно превышать 4 часов в месяц (за исключением плановых работ);

Продолжение ПРИЛОЖЕНИЯ А

- безопасность: защита данных, передаваемых между клиентом и сервером, с использованием протокола HTTPS. Аутентификация пользователей через Telegram (проверка подлинности initData). Защита от основных веб-уязвимостей (например, SQL-инъекции, XSS) на серверной стороне. Проверка загружаемых файлов (фотографий) на допустимые типы и размеры;

- эргономика: интуитивно понятный, адаптивный и визуально согласованный с интерфейсом Telegram пользовательский интерфейс.

3.2 Требования к функциям системы:

- отображение интерактивной карты города с использованием API Яндекс.Карт;

- отображение спортивных объектов на карте в виде интерактивных меток;

- определение текущего местоположения пользователя (с его разрешения) и отображение его на карте;

- поиск объектов по названию или ключевым словам в описании;

- фильтрация объектов по наличию конкретного спортивного оборудования;

- возможность сброса всех фильтров и визуальное отображение активных фильтров;

- просмотр детальной информации об объекте: название, адрес (с возможностью копирования), описание, фотографии (если есть), список доступного оборудования, средний рейтинг, отзывы пользователей;

- возможность добавления новых спортивных объектов пользователями: указание местоположения на карте, заполнение формы (название, адрес, описание, загрузка фото, выбор имеющегося оборудования). Предполагается механизм модерации добавляемого контента;

- возможность оставления текстовых отзывов об объектах;

- возможность выставления рейтинга объектам;

Продолжение ПРИЛОЖЕНИЯ А

- система «Избранное»: возможность для авторизованных пользователей добавлять понравившиеся объекты в личный список, просматривать и удалять объекты из избранного;
- бесшовный запуск приложения внутри мессенджера Telegram;
- использование данных пользователя Telegram для упрощенной авторизации для персонализированных функций;
- соответствие интерфейса цветовой схеме (светлая/темная), выбранной пользователем в Telegram.

3.3 Требования к видам обеспечения:

- математическое обеспечение: Алгоритмы расчета среднего рейтинга объектов; алгоритмы фильтрации и поиска данных.
- информационное обеспечение: База данных SQLite3 для хранения информации о спортивных объектах (координаты, название, адрес, описание, тип, фотографии, оборудование), пользователях (Telegram ID, username), отзывах (текст, оценка, дата, ID пользователя, ID объекта), оборудовании (список типов), избранных объектах (ID пользователя, ID объекта);
- программное обеспечение: Совместимость клиентской части с актуальными версиями мобильных операционных систем (iOS, Android) через приложение Telegram. Серверное программное обеспечение должно быть развернуто на хостинге, поддерживающем Python/Flask приложения и СУБД SQLite3 (или с возможностью подключения к ней);
- техническое обеспечение: Пользовательские устройства: смартфоны или планшеты с установленным приложением Telegram (актуальной версии) и доступом в Интернет.

4 СОСТАВ И СОДЕРЖАНИЕ РАБОТ ПО СОЗДАНИЮ СИСТЕМ

4.1 Этапы работ:

- анализ предметной области.
- проектирование архитектуры веб-приложения.

Продолжение ПРИЛОЖЕНИЯ А

- проектирование базы данных.
- разработка серверной и клиентской части веб-приложения.
- тестирование и отладка.
- ввод в эксплуатацию.

5 ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ СИСТЕМЫ

5.1 Контроль качества:

- проверка соответствия реализованного функционала требованиям настоящего технического задания;
- тестирование пользовательского интерфейса на удобство использования, адаптивность и соответствие макетам/эскизам;
- проверка корректности работы с API Яндекс.Карт (отображение карты, меток, геолокации);
- проверка взаимодействия клиентской и серверной частей через API;
- проверка безопасности (защита от базовых веб-уязвимостей, корректная обработка пользовательских данных и аутентификация).

5.2 Приемка системы:

Система считается принятой после успешного прохождения всех этапов тестирования, демонстрации работоспособности всех заявленных функций и соответствия требованиям, изложенным в данном техническом задании.

6 ТРЕБОВАНИЯ К ПОДГОТОВКЕ ОБЪЕКТА АВТОМАТИЗАЦИИ

6.1 Подготовка персонала:

Для пользователей: интерфейс должен быть интуитивно понятным и не требовать специального обучения. Возможна разработка краткого руководства или подсказок внутри приложения.

Для администраторов: разработка руководства администратора по управлению контентом.

Продолжение ПРИЛОЖЕНИЯ А

6.2 Техническая подготовка

- развертывание серверной части приложения на подходящем веб-хостинге;
- настройка базы данных и, при необходимости, заполнение начальными данными;
- регистрация и настройка Telegram-бота.

7 ТРЕБОВАНИЯ К ДОКУМЕНТИРОВАНИЮ

7.1 Перечень документов:

- техническое задание;
- инфологическая, логическая и физическая модели базы;
- описание API (эндпоинты, параметры, форматы данных);
- схемы и диаграммы, иллюстрирующие архитектуру и логику работы системы;
- программный код.

8 ИСТОЧНИКИ РАЗРАБОТКИ

8.1 Нормативные документы:

ГОСТ 2.105-95. Единая система конструкторской документации. Общие требования к текстовым документам.

ГОСТ 7.32-2001. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления.

СТО СМК 4.2.3.21-2018. Стандарт организации. Оформление выпускных квалификационных и курсовых работ (проектов) / ФГБОУ ВО «Амурский государственный университет».

8.2 Литература и интернет-источники:

- документация API Яндекс.Карт (jsapi);
- документация Telegram Mini Apps;
- документация Python, Flask, SQLite3;

Продолжение ПРИЛОЖЕНИЯ А

- учебные и справочные материалы по веб-разработке (HTML, CSS, JavaScript);
- анализ существующих картографических сервисов и приложений для поиска спортивных объектов.

ПРИЛОЖЕНИЕ Б

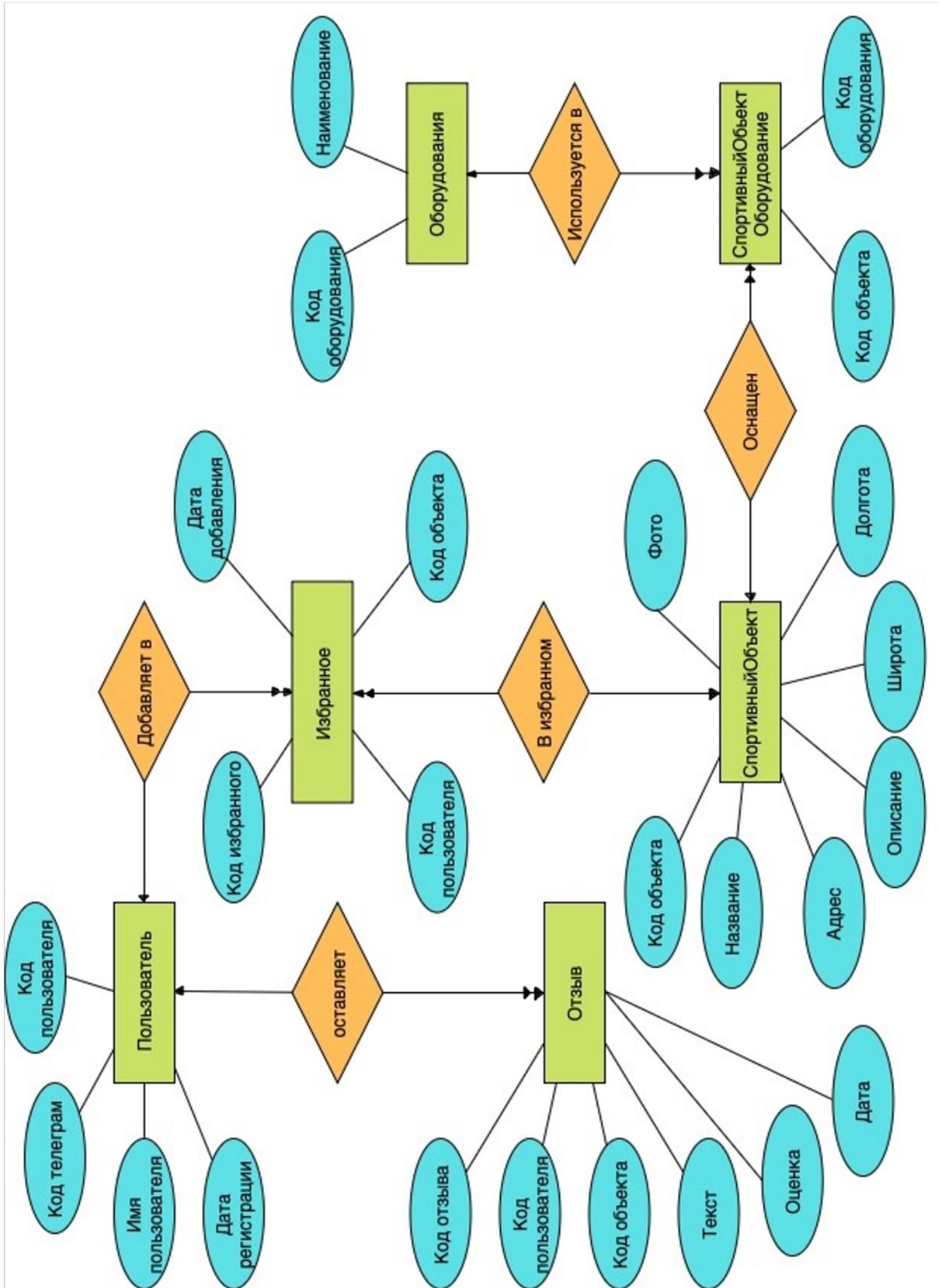


Рисунок Б.1 – Инфологическая модель

ПРИЛОЖЕНИЕ В

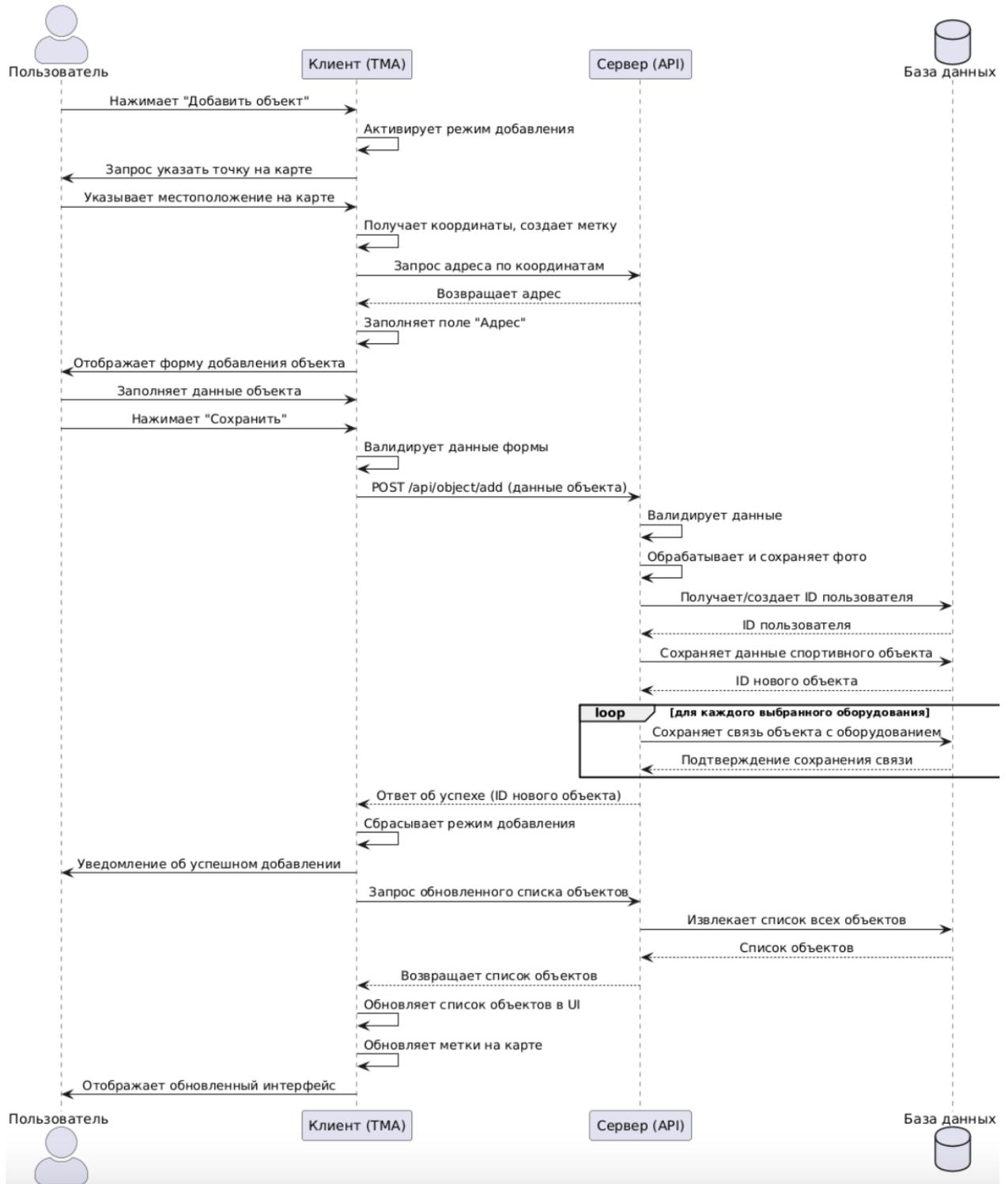


Рисунок В.1 – Логическая структура программы