

Разработка мобильного приложения для поиска спортивных
площадок и организация совместных тренировок

Пояснительная записка к курсовому проекту

ПМ.01 Разработка модулей программного обеспечения для компьютерных
систем

МДК.01.03 Разработка мобильных приложений

НАТКиГ.201000.010.000 ПЗ

Разработал:
студент группы ПР-23.106
Мешков Александр Андреевич

Содержание

Введение.....	3
Исследовательский раздел	4
1.1 Описание предметной области	4
1.2 Образ клиента	5
1.3 Сценарии	5
1.4 Сбор и анализ прототипов	6
Проектирование приложения.....	9
1.5 UI/UX дизайн проекта.....	9
2.2 Выбор технологии, языка и среды программирования.....	16
3 Разработка мобильного приложения.....	18
3.1 Описание API.....	18
3.2 Разработка мультимедийного контента.....	18
3.3 Разработка используемых библиотек	20
3.4 Описание разработанных классов	21
3.5 Описание разработанных процедур и функций.....	22
4 Тестирование	28
4.1 Протокол тестирования дизайна приложения	28
4.2 Протокол тестирования функционала приложения	32
Заключение	36
Библиография	37
Приложение А	38
Приложение Б	45

					НАТКиГ.201000.010.000 ПЗ			
Изм.	Лист	№ докум	Подпись	Дата				
Разраб		Мешков А.А.			Разработка мобильного приложения для поиска спортивных площадок и организации совместных тренировок	Литера	Лист	Листов
Пров		Климова И. С.				у	2	45
Н. Контр		Тышкевич Е. В.				ПР-23.106		
Утв		Тышкевич Е. В.						

Введение

В современном мире, где здоровый образ жизни и физическая активность становятся неотъемлемой частью человеческого развития, мобильные приложения играют важную роль в организации спортивного досуга. Одной из ключевых проблем, с которой сталкиваются любители спорта, является поиск доступных спортивных площадок и единомышленников для совместных тренировок.

Актуальность темы обусловлена растущим спросом на удобные цифровые решения, помогающие людям находить места для занятий спортом, планировать и собирать совместные тренировки.

Целью курсового проекта является разработка мобильного приложения для поиска спортивных площадок и организации совместных тренировок.

Задачами курсового проекта в связи с указанной целью являются:

- изучение предметной области;
- рассмотрение приложения с точки зрения пользователя для выявления необходимых функций приложения;
- разработка и реализация дизайна приложения;
- разработка функционала приложения;
- тестирование полученного продукта.

Продукт исследования – мобильное приложение с встроенной навигационной картой для поиска спортивных площадок, а также с возможностью создания событий на добавленной площадке.

Исследовательский раздел

1.1 Описание предметной области

В современном обществе спорт перестал быть рутиной, превратившись в увлекательное хобби и неотъемлемую часть досуга. Если раньше физическая активность ассоциировалась исключительно с изнурительными тренировками ради поддержания здоровья, сегодня она стала способом самовыражения, социализации и поиска единомышленников. Люди все чаще посвящают свободное время не просто занятиям в зале, а созданию сообществ – будь то футбольные команды, группы для утренних пробежек или йога-марафоны в парках. Организация совместных событий – от товарищеских матчей до масштабных турниров – помогает не только поддерживать форму, но и выстраивать социальные связи, обмениваться опытом и заряжаться мотивацией. Спорт становится языком, на котором говорят люди разных возрастов и профессий.

Сейчас многие люди хотят заниматься спортом, но сталкиваются с проблемами, которые мешают им начать. Одна из главных сложностей – найти подходящую площадку. Даже если открыть карту, не всегда понятно, есть ли рядом нормальное поле для футбола, тренажеры во дворе или бесплатный корт. Часто информация на картах, устаревшая: непонятно, работает ли площадка, есть ли там освещение или сетка на воротах. Особую проблему представляет отсутствие возможности узнать посещаемость данного объекта. Отсутствие инструментов для отслеживания посещаемости создает информационный вакуум: потенциальные пользователи не могут прогнозировать загруженность объекта, что приводит к конфликтам расписаний, снижению мотивации и, как следствие, отказу от активностей.

Одним из способов решить данные трудности, является оптимизация этих процессов и объединение в единую структуру в виде цифрового продукта, позволяющую пользователям добавлять объекты известные им и создавать на них события для оповещения других пользователей.

Такое решение поможет в структуризации данных о объекте и посещаемости им других пользователей.

В рамках данного курсового проекта решено разработать мобильное приложение для платформы Android, которое поможет пользователям находить удобные для себя площадки и устраивать на них совместные тренировки.

1.2 Образ клиента

Разрабатываемый сервис идеально подойдет для подростков и людей среднего возраста, которые хотят быстро находить спортивные площадки и организовывать совместные тренировки.

Основные пользователи приложения – люди, которые регулярно занимаются спортом и стремятся оптимизировать процесс поиска подходящих мест для тренировок. Они хотят иметь под рукой надежный инструмент, позволяющий не только находить ближайшие площадки, но и быстро собирать команду единомышленников.

Приложение предоставляет возможность легко организовать спортивный досуг в любом районе города. Это делает сервис незаменимым помощником для тех, кто ценит свое время и хочет проводить его с пользой для здоровья, независимо от места нахождения.

1.3 Сценарии

Артем любит настольный теннис, но во дворе стол постоянно занят. В сервисе он видит свободные столы в соседних районах, выбирает ближайший и создает событие «Турнир для новичков». К назначенному времени собирается компания для игры. Теперь они играют каждую пятницу, а прохожие, видя объявление в приложении, иногда присоединяются.

Ольга хочет укрепить спину, но в её районе нет подходящих площадок. В сервисе она находит событие Виктора — тренера, который год назад добавил самодельную площадку в лесопарке. Ольга присоединяется к его

тренировкам «Силовая подготовка на свежем воздухе». Группа растёт, участники вместе улучшают площадку.

Катя занимается йогой, но одной дома не хватает мотивации. В сервисе она находит поляну в парке, куда другие пользователи ходят для медитации. Она создаёт событие: «Йога на рассвете. Все уровни подготовки!». На первое занятие приходят три человека, а со временем группа расширяется. Сервис помогает Кате найти единомышленников и превратить одиночные тренировки в регулярные встречи.

1.4 Сбор и анализ прототипов

При анализе существующих решений для поиска спортивных площадок и организации мероприятий обнаруживается небольшое количество прямых аналогов. В качестве прототипа для картографического функционала разумно рассмотреть популярные сервисы, предлагающие работу с картами. Хотя тема достаточно узкоспециализированная, некоторые приложения и платформы предоставляют схожие возможности.

Однако основная часть таких сервисов концентрируется либо на навигации, либо на бронировании, не интегрируя эти функции с удобным инструментарием для планирования событий. Современные платформы по поиску спортивных объектов активно используют API картографических сервисов, таких как Google Maps, Яндекс.Карты и OpenStreetMap, что обеспечивает точное отображение локаций и доступ к актуальной информации для пользователей.

1.4.1 Карта – сервис для навигации от FindSport

Приложение FindSport отличается лаконичным дизайном с интуитивно понятной навигацией. Основной экран, представленный на рисунке 1, содержит карту с отмеченными спортивными площадками и кратким описанием их характеристик. Цветовая гамма выдержана в ярком стиле с акцентами на контрастных элементах для удобства восприятия.

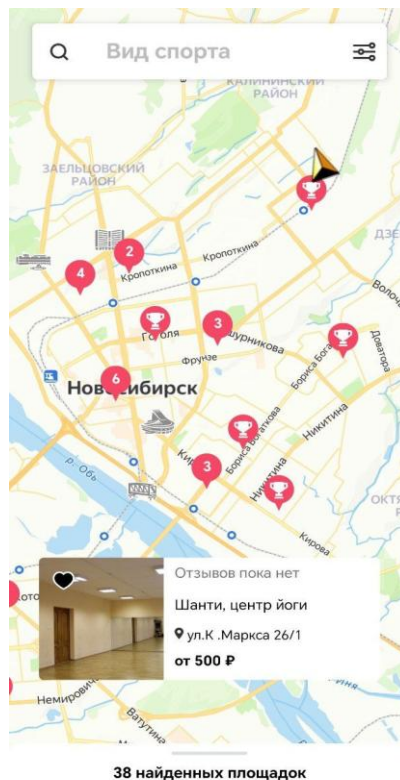


Рисунок 1 – Страница навигации

О приложении:

Главной особенностью FindSport является возможность не только поиска, но и организации мероприятий. Система автоматически учитывает геопозицию, что упрощает планирование совместных тренировок и игр. В отличие от аналогов, здесь реализован комплексный подход, объединяющий навигацию, бронирование и социальный функционал. FindSport требует постоянного доступа к геолокации для корректной работы, что может влиять на энергопотребление.

1.4.2 Карта – сервис для навигации от GoSport

Приложение GoSport выполнено в современном минималистичном стиле с упором на удобство навигации. Основная цветовая палитра построена на сочетании насыщенных синих и белых тонов, что создает ощущение динамичности и соответствует спортивной тематике. Главный экран, представленный на рисунке 2, включает в себя карту с отображением ближайших спортивных объектов, а также быстрый доступ к фильтрам и категориям активностей.

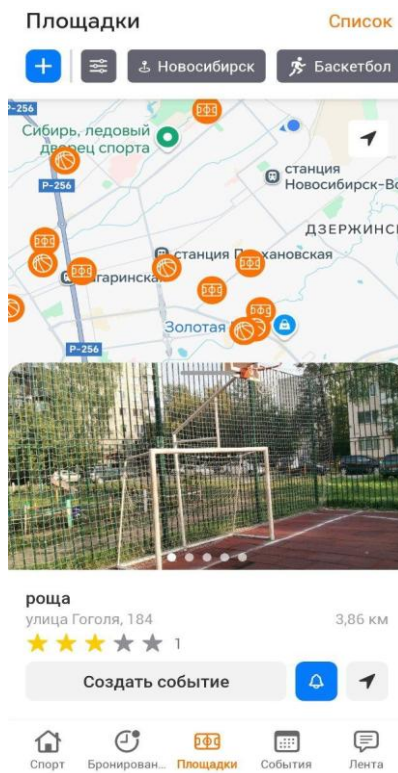


Рисунок 2 – Страница навигации

GoSport ориентирован на поиск и бронирование спортивных площадок, включая футбольные поля, теннисные корты, залы для единоборств и другие объекты. Приложение делает больший акцент на коммерческих локациях, предлагая пользователям не только информацию о доступности, но и возможность онлайн-оплаты.

Интеграция с Google Maps обеспечивает точное позиционирование и построение маршрутов.

Проанализировав данные приложения и выделив положительные и отрицательные качества, можно сделать вывод, что при разработке планируется использовать положительные качества.

Положительными качествами разрабатываемого приложения должны быть:

- оснащенность необходимыми функциями;
- правильная цветовая гамма.

Минусы приложения, которые стоит избегать при разработке:

- непонятный интерфейс.

Проектирование приложения

1.5 UI/UX дизайн проекта

Дизайн проекта разработан в программе Figma, так как рабочее пространство Figma максимально удобно и интуитивно понятно. Когда иные программы для разработки дизайна направлены на аудиторию, имеющую опыт в данной сфере. В Figma все гораздо проще, поэтому освоить ее на базовом уровне не составила большого труда.

Для проекта были определены основные экраны:

- приветствие;
- окно регистрации и авторизации;
- карта;
- список площадок;
- список событий;
- подробная информация о площадках;
- добавление площадок и событий;
- профиль пользователя.

При выборе цветового решения для приложения учитывались современные тенденции и данные о пользовательских предпочтениях. Исследования показывают, что темные интерфейсы становятся все популярнее, особенно на устройствах с OLED-экранами, где черный фон не только выглядит стильно, но и помогает экономить заряд батареи. Кроме того, темная тема снижает нагрузку на глаза при длительном использовании, что особенно важно для спортивных приложений, которыми люди могут активно пользоваться в течение дня. В качестве основного цвета был выбран глубокий черный, создающий эффектный контраст с контентом. Чтобы добавить интерфейсу выразительности и сохранить удобную читаемость, в качестве акцентного цвета используется синий – он ассоциируется с надежностью и технологичностью, что хорошо соответствует тематике приложения. Такое сочетание позволяет создать лаконичный и современный дизайн, который

хорошо работает как при ярком солнечном свете, так и в условиях слабого освещения, обеспечивая комфортное взаимодействие с приложением в любой ситуации.

Ниже на рисунке 3 представлен логотип «PlayPlace», на котором изображены инициалы названия в выбранной цветовой гамме.

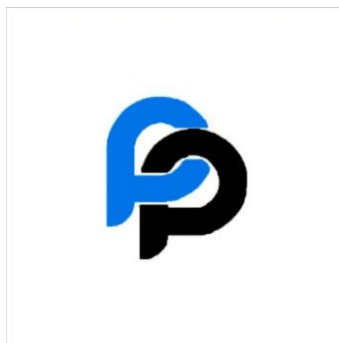


Рисунок 3 – логотип «PlayPlace»

На рисунке 4 ниже представлена страница приветствия



Рисунок 4 – Страница «Приветствия»

На данной странице мобильного приложения представлена возможность увидеть окно, встречающее пользователя.

На рисунке 5 ниже представлена страница с профилем пользователя.

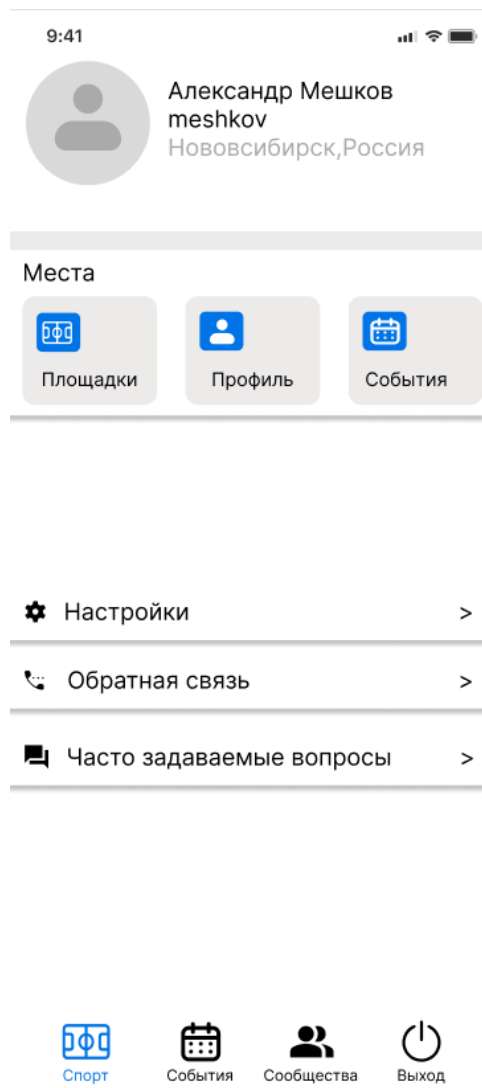


Рисунок 5 – страница профиля пользователя

На данной странице представлена главная информация о пользователе в виде фотографии пользователя, имя, фамилия, отчество и место проживания, возможность перехода на страницу настроек, обратной связи и часто задаваемых вопросов. В нижней части экрана имеется навигационная панель для удобного перехода между главными активностями приложения

На рисунке 6 ниже представлены страницы с выбором входа в приложение, а также страница регистрации с необходимыми полями для ее успешного выполнения.

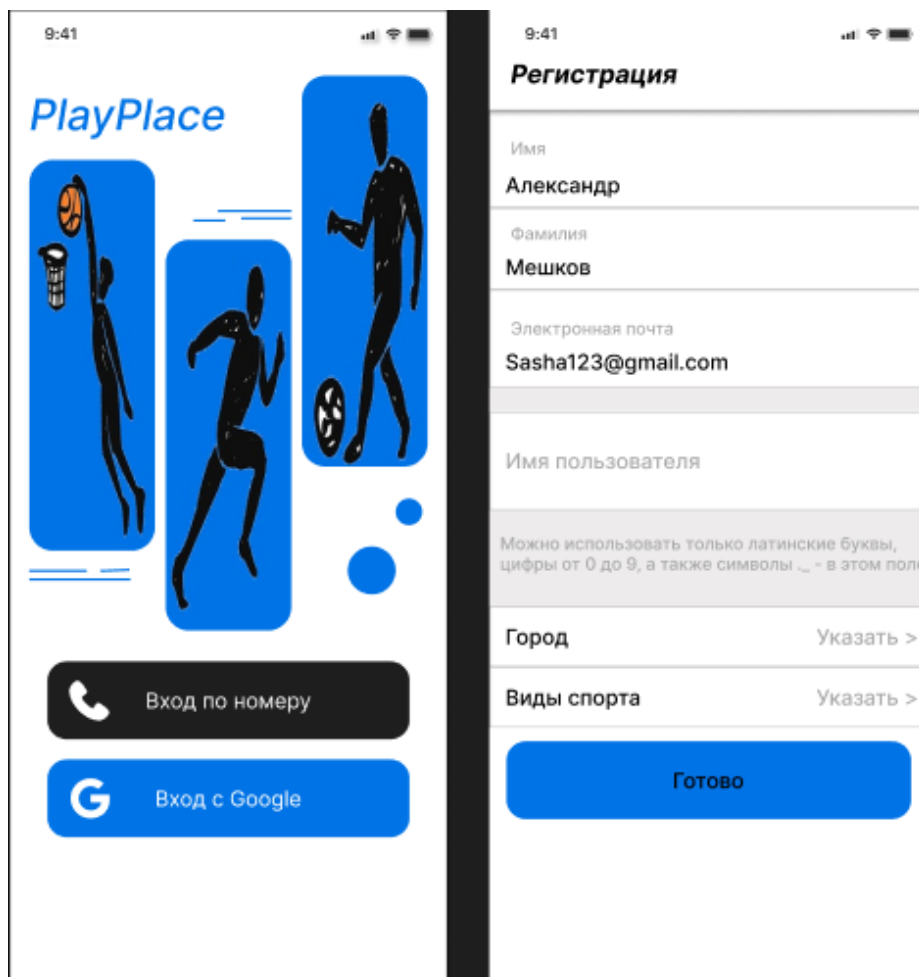


Рисунок 6 – Страница авторизации и регистрации

На данных страницах, пользователю предоставляется возможность выбора входа в приложении с переходом на соответствующую активность, где предоставлены необходимые поля для заполнения и пример для их заполнения. Экраны выполнены в единой цветовой схеме и логичному расположению элементов интерфейса. Подсказки и примеры заполнения полей помогают пользователю правильно и быстро заполнить необходимую информацию, что улучшает пользовательский опыт и делает процесс авторизации и регистрации более удобным.

На рисунке 7 ниже представлена страница с картой, на которой отмечены места в виде маркеров.

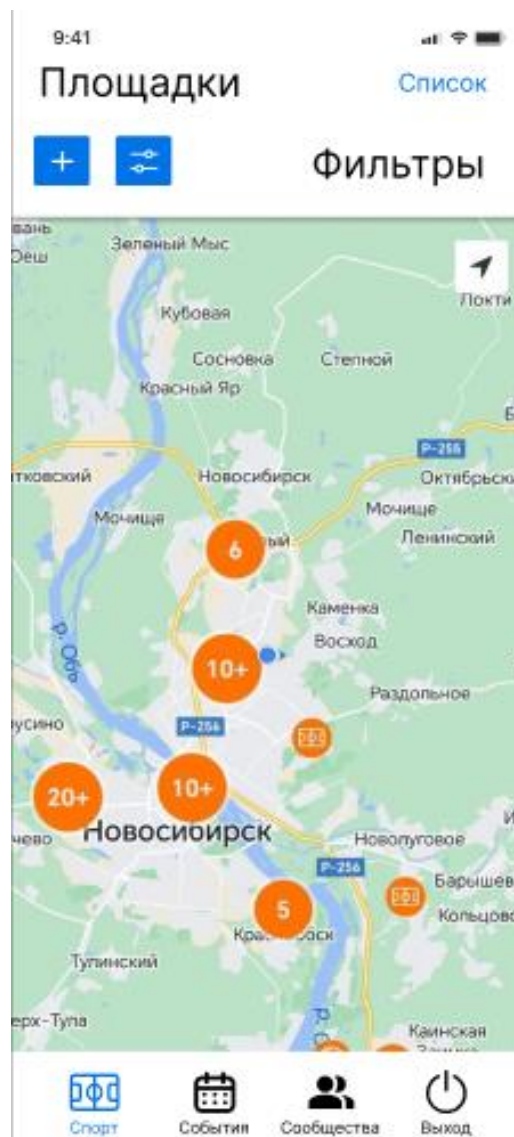


Рисунок 7 – страница «Карта»

На данной странице представлена карта, с отмеченными на ней местами в виде маркеров, а также возможность переход к страницам добавления площадки, фильтрации, вывода площадок в виде общего списка.

Экран с картой площадок предоставляет пользователю удобный и наглядный способ поиска и добавления спортивных площадок. Использование маркеров на карте позволяет быстро оценить количество площадок в различных районах, а кнопки управления и навигационные элементы обеспечивают легкий доступ к дополнительным функциям приложения. Общий дизайн экрана соответствует единому стилю приложения.

На рисунке 8 ниже представлена страница, на которой показан вывод всех площадок в виде списка.

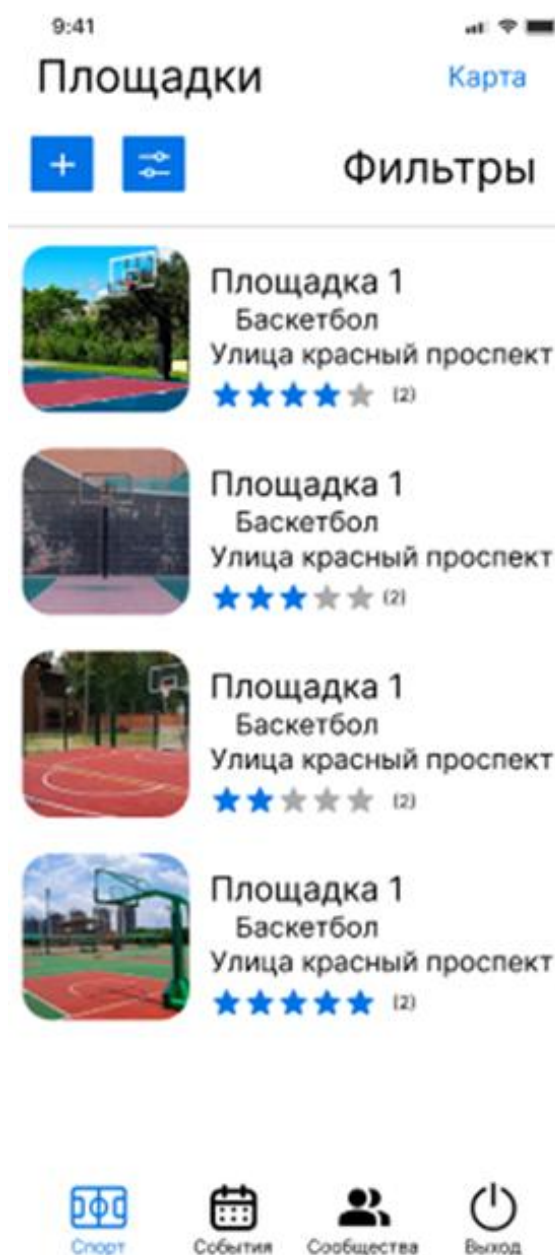


Рисунок 8 – страница списка доступных площадок

На данной странице продемонстрирован вывод всех площадок, выполненных в виде списка, где каждое поле содержит информацию о площадке, а именно название, описание, адрес и фотографию. В верхней части есть возможность вернуться к карте, а также переход к активности добавления и фильтрации.

На рисунке 9 ниже представлена страница для просмотра событий



Рисунок 9 - страница для просмотра событий

Экран "События" в приложении предоставляет пользователю удобный интерфейс для поиска и создания новых событий. Использование иконки лупы в центре экрана символизирует поиск и предлагает пользователю начать взаимодействие с приложением. Кнопки управления и навигационные элементы обеспечивают легкий доступ к дополнительным функциям приложения. Общий дизайн экрана соответствует единому стилю приложения.

На рисунке 10 ниже представлены страницы для добавления событий и площадок с полями для заполнения.

Рисунок 10 – страницы добавления площадки и события

Экраны добавления события и площадки предоставляют пользователю удобный интерфейс для ввода необходимых данных. Поля для заполнения параметров события и площадки расположены логично и удобно, что облегчает процесс добавления новой информации. Кнопка "Создать" завершает процесс и сохраняет введенные данные, обеспечивая легкий доступ к дополнительным функциям приложения. Общий дизайн экранов соответствует единому стилю приложения.

2.2 Выбор технологии, языка и среды программирования

Средой разработки клиентской части выбрана Visual Studio 2022 с поддержкой .NET MAUI (Multi-platform App UI). Это универсальная платформа, позволяющая создавать кроссплатформенные приложения для

Android, iOS, Windows и macOS из единой кодовой базы. .NET MAUI предоставляет встроенные инструменты для адаптации интерфейса под различные устройства: смартфоны, планшеты и настольные компьютеры. Среда включает эмуляторы для тестирования приложений на разных разрешениях экрана и соотношениях сторон, а также горячую перезагрузку (Hot Reload) для ускорения разработки UI.

Для создания пользовательского интерфейса используется XAML — декларативный язык разметки, который позволяет гибко проектировать адаптивные макеты. Интеграция с MVVM (Model-View-ViewModel) упрощает разделение логики и отображения, что повышает поддерживаемость кода.

Бэкенд-часть реализована на Python с использованием FastAPI — современного фреймворка для создания высокопроизводительных RESTful API. FastAPI обеспечивает автоматическую генерацию документации (Swagger/OpenAPI), асинхронную обработку запросов и встроенную валидацию данных через Pydantic.

Для работы с базой данных применяется:

- SQLAlchemy в качестве ORM (Object-Relational Mapping) для удобного взаимодействия с СУБД на уровне объектов Python;
- Alembic для управления миграциями базы данных, что позволяет безопасно изменять схему БД без потери данных.

Интеграция карт реализована через внешние API OpenStreetMap с использованием HTML/JavaScript для отображения интерактивных карт в гибридном режиме через WebView в .NET MAUI

3 Разработка мобильного приложения

3.1 Описание API

API OpenStreetMap (OSM) предоставляет доступ к геоданным в форматах XML и JSON, что упрощает работу с картографической информацией, включая дороги, здания, природные объекты и другие элементы инфраструктуры.

3.2 Разработка мультимедийного контента

Карта производится через HTML/JS в WebView с использованием OpenStreetMap, что позволяет получить бесплатное картографическое решение с полным контролем над интерфейсом и функционалом.

Для удобства использования карты разработаны следующие функции: четыре типа карт, включая стандартную, темную, спутниковую и топографическую, что позволяет пользователям выбирать наиболее подходящий вид отображения в зависимости от их предпочтений и условий использования. Режим добавления новых маркеров с подтверждением координат обеспечивает точное указание местоположения новых объектов, таких как спортивные площадки, гарантируя актуальность и точность данных на карте. Отображение спортивных площадок с фотографиями и описаниями позволяет пользователям легко находить и выбирать площадки для занятий спортом, получая дополнительную информацию прямо на карте. Интерактивные элементы карты, такие как маркеры и всплывающие окна с информацией, обеспечивают удобство использования и делают процесс поиска и добавления площадок более интуитивным и эффективным. Эти функции обеспечивают пользователям удобный и функциональный инструмент для работы с картой, делая процесс поиска и добавления спортивных площадок более простым и эффективным.

Ниже на Листинге 1 представлен код, который отвечает за изменение типа карт.

Листинг 1 – Изменение типа карты

```
const baseLayers = {
  "Стандартная":
    L.tileLayer('https://{s}.tile.openstreetmap.org/{z}/{x}/{y}.png', {
      maxZoom: 19
    }),
  "Тёмная":
    L.tileLayer('https://{s}.basemaps.cartocdn.com/dark_all/{z}/{x}/{y}{r}.png', {
      maxZoom: 19,
      subdomains: 'abcd'
    }),
  "Спутник":
    L.tileLayer('https://server.arcgisonline.com/ArcGIS/rest/services/World_Imagery/MapServer/tile/{z}/{y}/{x}', {
      maxZoom: 19
    }),
  "Топографическая":
    L.tileLayer('https://{s}.tile.opentopomap.org/{z}/{x}/{y}.png', {
      maxZoom: 17
    })
};

baseLayers["Стандартная"].addTo(map);

L.control.layers(baseLayers, null, {
  position: 'topright',
  collapsed: false
}).addTo(map);
```

Здесь идет настройка нескольких вариантов карт (стандартная, тёмная, спутник, топографическая) и добавляет переключатель слоёв в правый верхний угол.

Листинг 2 – Добавление новых маркеров

```
function handleMapClick(e) {
  if (!addMarkerMode) return;

  if (clickMarker) {
    map.removeLayer(clickMarker);
  }

  clickMarker = L.marker(e.latlng, {
    icon: L.divIcon({
      className: 'new-marker-icon',
      iconSize: [32, 32]
    }),
    draggable: true
  }).addTo(map)
    .bindPopup('Новая метка<br>Перетащите для точного позиционирования')
    .openPopup();

  selectedCoordinates = e.latlng;
  document.getElementById('confirmButton').style.display = 'block';
}
```

Эта функция обрабатывает клик по карте, когда включён режим добавления маркера. Сначала она проверяет, активен ли этот режим, затем создаёт новый маркер в точке клика с зелёной круглой иконкой, делает его перетаскиваемым и добавляет всплывающую подсказку. После этого сохраняет координаты клика и показывает кнопку подтверждения выбора.

Листинг 3 – Отображение спортивных площадок

```
function updatePlacemarks(platforms) {

    platforms.forEach(platform => {
        const marker = L.marker([platform.lat, platform.lon], {
            icon: L.divIcon({ className: 'custom-icon', iconSize: [32, 32] })
        });

        let popupContent =
`<h3>${platform.title}</h3><p>${platform.address}</p>`;
        if (platform.image) {
            popupContent += ``;
        }

        marker.bindPopup(popupContent).addTo(map);
        markers.push(marker);
    });

    if (markers.length > 0) {
        map.fitBounds(new L.featureGroup(markers).getBounds());
    }
}
```

Эта функция показывает спортивные площадки на карте. Она получает список площадок и для каждой создает метку в указанных координатах. К каждой метке добавляется всплывающее окно с названием площадки, адресом и фотографией.

3.3 Разработка используемых библиотек

Ниже в таблице 3.3 представлены названия библиотек и их версия, использующиеся для создания серверной части.

Таблица 3.3 – Используемые библиотеки

Название библиотеки	Версия
Alembic	1.13.1
Asyncpg	0.30.0
Psycopg	3.1.18
Psycopg – binary	3.2.6

Продолжение таблицы 3.3

Sqlalchemy	2.0.27
Fastapi	0.109.0
Pydantic	2.6.0
Pydantic – settings	2.1.0
Uvicorn	0.27.0
Bcrypt	4.3.0
Black	23.11.0
Psycpg2	2.9.10
Asyncio	3.4.3
Yarl	1.18.3
Pyjwt	2.10.1
Python – multipart	0.0.20

Для разработки использованы такие библиотеки, как Alembic, который управляет миграциями БД, работая с SQLAlchemy (ORM для взаимодействия с PostgreSQL через asyncpg и psycpg). FastAPI (на базе Pydantic для валидации и uvicorn для ASGI-сервера) обеспечивает API с асинхронной обработкой запросов. Bcrypt хэширует пароли, PyJWT реализует JWT-аутентификацию, а python-multipart обрабатывает загрузку файлов. Black форматирует код, yarl работает с URL, а asyncio лежит в основе асинхронности.

3.4 Описание разработанных классов

В серверной части реализованы следующие классы, которые представлены в таблице 4.

Таблица 3.4 – Методы приложения

Метод	Описание
1	2
get_profile	Возвращает профиль пользователя по указанным параметрам (id, email)
get_user_by_login	Находит пользователя по логину
get_user_by_id	Находит пользователя по ID
register	Создаёт нового пользователя, проверяет уникальность логина/email, хэширует пароль.

Продолжение таблицы 3.4

Authorize	Проверяет логин/пароль, возвращает пользователя или ошибку аутентификации.
create_platform	Создает новую площадку с привязкой изображения.
get_platform	Возвращает площадку по ID
get_all_platforms	Возвращает список всех площадок.
update_platform	Обновляет данные площадки
delete_platform	Удаляет площадку и связанное изображение.
get_event_by_id	Находит событие по ID
get_all_events	Возвращает список всех событий из БД.
create_event	Добавляет новое событие в БД
update_event	Обновляет данные события

Все методы спроектированы с учетом требований безопасности и целостности данных. Реализованная функциональность обеспечивает полный цикл работы приложения — от аутентификации пользователей до управления основными сущностями системы. Это позволяет поддерживать актуальность и надежность данных, обеспечивая пользователям удобный и безопасный доступ к функциям приложения. Кроме того, использование хэширования паролей и проверка уникальности логинов и email-адресов способствуют защите пользовательских данных и предотвращению несанкционированного доступа. В совокупности, эти методы обеспечивают стабильную и эффективную работу серверной части приложения, поддерживая его функциональность и безопасность на высоком уровне.

3.5 Описание разработанных процедур и функций

В приложении разработаны следующие функции:

- авторизация и регистрация пользователя;
- изменение фото в профиле;
- добавление площадок и событий;
- вывод всех площадок и событий.

Ниже на Листинге 1 представлен код, который отвечает за авторизацию пользователя в приложении.

Листинг 1 – Авторизация пользователя

```
public async Task<string> LoginAndGetToken(string username, string
password)
{
    var me_data = CreateAuthFormContent(username, password);
    var response = await ExecuteAuthRequest(me_data);
    return ParseTokenFromResponse(response);
}

private FormUrlEncodedContent CreateAuthFormContent(string username,
string password)
{
    return new FormUrlEncodedContent(new[]
    {
        new KeyValuePair<string, string>("username", username),
        new KeyValuePair<string, string>("password", password),
        new KeyValuePair<string, string>("grant_type", "password")
    });
}

private async Task<HttpResponseMessage>
ExecuteAuthRequest(FormUrlEncodedContent content)
{
    var response = await _httpClient.PostAsync("/v1/token", content);
    response.EnsureSuccessStatusCode();
    return response;
}

private string ParseTokenFromResponse(HttpResponseMessage response)
{
    using var jsonDoc =
JsonDocument.ParseAsync(response.Content.ReadAsStreamAsync().Result).Result;
    return
jsonDoc.RootElement.GetProperty("access_token").GetString();
    public async Task<UserData> GetUserData(string token, string userId)
    {
        SetAuthHeader(token);
        var response = await _httpClient.GetAsync($" /v1/users/{userId}");
        return await ProcessUserDataResponse(response);
    }
}
```

Пользователь вводит логин и пароль, которые отправляются на сервер для получения токена. Система формирует запрос с введённым паролем и вызывает конструкцию на сервер. В ответе извлекается токен пользователя, который добавляется в заголовки запросов. Этот токен позволяет получать защищённые данные пользователя через API. Этот процесс обеспечивает безопасный и эффективный способ авторизации пользователей и получения их данных, используя современные методы аутентификации и обработки данных.

Листинг – 2 Регистрация пользователя

```
public async Task<string> RegisterUser(UserRegistration userRegistration)
{
    var content = CreateRegistrationContent(userRegistration);
    var response = await _httpClient.PostAsync("/v1/register",
content);
    return await ProcessRegistrationResponse(response);
}

private StringContent CreateRegistrationContent(UserRegistration
userRegistration)
{
    var json = JsonSerializer.Serialize(userRegistration);
    return new StringContent(json, Encoding.UTF8,
"application/json");
}
```

При регистрации нового пользователя система принимает его данные (логин, пароль и другую необходимую информацию), преобразует эти данные в JSON-формат и отправляет POST-запрос на сервер. Сервер обрабатывает полученные данные, создает новую учетную запись и возвращает клиентскому приложению результат операции, который может содержать информацию об успешной регистрации. Весь обмен данными происходит в формате JSON с использованием UTF-8 кодировки для корректной передачи информации.

Листинг – 3 Изменение фото профиля

```
public async Task<bool> UploadUserPhoto(string token, string userId, Stream
photoStream, string fileName)
{
    try
    {
        SetAuthHeader(token);

        using var content = new MultipartFormDataContent();
        using var fileContent = new StreamContent(photoStream);

        content.Add(fileContent, "photo_file", fileName);

        var response = await
_httpClient.PutAsync($"/v1/users/{userId}", content);
        return response.IsSuccessStatusCode;
    }
    catch (Exception ex)
    {
        Debug.WriteLine($"Ошибка загрузки фото: {ex.Message}");
        return false;
    }
}
```

Пользователь выбирает фотографию со своего устройства, которая передается в метод в виде файлового потока. Система сначала устанавливает авторизованный токен в заголовок запроса для подтверждения прав доступа.

Затем создается составной запрос, куда добавляется файл с оригинальным именем. Этот запрос отправляется на сервер методом PUT.

Листинг – 3 Добавление событий

```
public async Task<bool> CreateEventAsync(EventRequest eventRequest)
{
    try
    {
        var content = new StringContent(
            JsonSerializer.Serialize(eventRequest),
            Encoding.UTF8,
            "application/json");

        var response = await _httpClient.PostAsync("/v1/events/",
content);

        return response.IsSuccessStatusCode;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error creating event: {ex.Message}");
        return false;
    }
}
```

Приложение получает от пользователя данные о событии (название, описание, дату и другие параметры) и формирует из них JSON-объект. Этот объект отправляется POST-запросом на сервер.

Листинг – 4 Добавление площадок

```
public async Task<bool> CreatePlatformAsync(Platform platform, Stream
imageStream, string imageFileName)
{
    try
    {
        if (imageStream.CanSeek)
        {
            imageStream.Position = 0;
        }

        using var content = new MultipartFormDataContent();

        content.Add(new StringContent(platform.Name), "name");
        content.Add(new StringContent(platform.City), "city");
        content.Add(new StringContent(platform.Address), "address");
        content.Add(new StringContent(platform.Latitude.ToString()),
"latitude");
        content.Add(new StringContent(platform.Longitude.ToString()),
"longitude");

        using var memoryStream = new MemoryStream();
        await imageStream.CopyToAsync(memoryStream);
        memoryStream.Position = 0;

        var imageContent = new StreamContent(memoryStream);
        imageContent.Headers.ContentType =
MediaTypeHeaderValue.Parse(GetMimeType(imageFileName));
        content.Add(imageContent, "image", imageFileName);
    }
}
```

```

        var response = await _httpClient.PostAsync("/v1/platforms/",
content);

        if (!response.IsSuccessStatusCode)
        {
            var errorContent = await
response.Content.ReadAsStringAsync();
            Console.WriteLine($"Error response: {errorContent}");
            return false;
        }

        return true;
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error creating platform: {ex.Message}");
        if (ex.InnerException != null)
        {
            Console.WriteLine($"Inner exception:
{ex.InnerException.Message}");
        }
        return false;
    }
}

```

Метод принимает данные площадки название, город, адрес, координаты и изображение. Формируется составной запрос, куда добавляются текстовые данные площадки и изображение. Изображение отправляется с оригинальным именем файла и соответствующим. Запрос отправляется на сервер.

Листинг – 5 Вывод всех событий

```

public async Task<List<EventRequest>> GetAllEventsAsync()
{
    try
    {
        var response = await _httpClient.GetAsync("/v1/events/");
        response.EnsureSuccessStatusCode();

        var content = await response.Content.ReadAsStringAsync();
        return
JsonSerializer.Deserialize<List<EventRequest>>(content) ?? new
List<EventRequest>();
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error fetching events: {ex.Message}");
        return new List<EventRequest>();
    }
}

```

Метод отправляет GET-запрос на сервер для получения списка всех доступных событий. Метод читает содержимое ответа как строку и обрабатывает её в список объектов.

Листинг – 6 Вывод всех площадок

```
public async Task<List<Platform>> GetAllPlatformsAsync()
{
    try
    {
        var response = await _httpClient.GetAsync("/v1/platforms/");
        response.EnsureSuccessStatusCode();

        var content = await response.Content.ReadAsStringAsync();
        return JsonSerializer.Deserialize<List<Platform>>(content) ??
new List<Platform>();
    }
    catch (Exception ex)
    {
        Console.WriteLine($"Error fetching platforms: {ex.Message}");
        return new List<Platform>();
    }
}
```

Метод выполняет GET-запрос на сервер для получения перечня всех зарегистрированных спортивных площадок.

Рассмотренные списки представляют ключевые функциональные возможности мобильного приложения, обеспечивающие взаимодействие пользователя с системой. Разработанные процедуры и функции охватывают основные сценарии работы. Реализованная система аутентификации обеспечивает безопасный доступ пользователей через токены, что соответствует современным стандартам защиты данных. Процесс регистрации упрощен за счет использования JSON-формата, что делает его удобным для интеграции с внешними сервисами. Таким образом, разработанные функции формируют полноценный каркас приложения, охватывающий основные потребности пользователей — от авторизации до работы с контентом. Использование современных технологий (REST API, JSON, асинхронные операции) обеспечивает высокую производительность и удобство взаимодействия с серверной частью.

4 Тестирование

4.1 Протокол тестирования дизайна приложения

Для проведения тестирования дизайна приложение выбрано устройство Honor 90 Lite разрешение экрана 1080×2388, а также Android SDK 34.

Проверка была проведена на:

- оптимальный размер кнопок;
- читабельный размер шрифта;
- корректность выводимых данных;
- корректное расположение элементов.

Примеры проверок отображения элементов на экране представлены на рисунках 11-17, слева расположен экран с Android SDK 34, а справа Honor 90 Lite, на рисунке 11 представлен экран выбора входа в систему приложения.

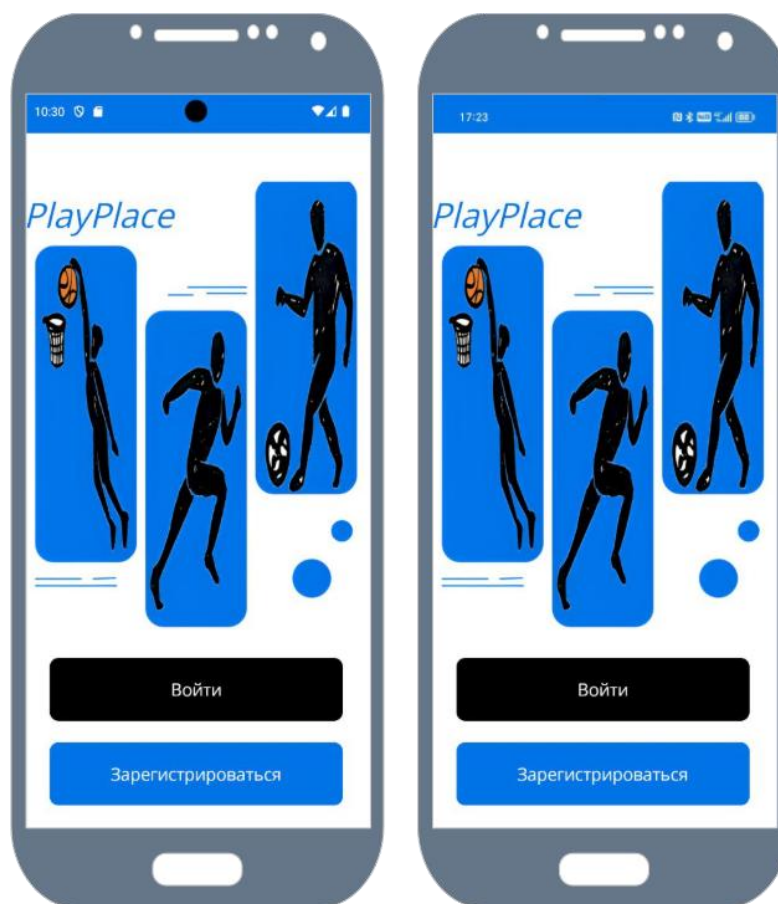


Рисунок 11 – Экран выбора входа

Элементы интерфейса экрана выбора входа в обоих случаях отображаются корректно. Графические элементы, кнопки входа и регистрации стоят на своих местах и правильных размеров.

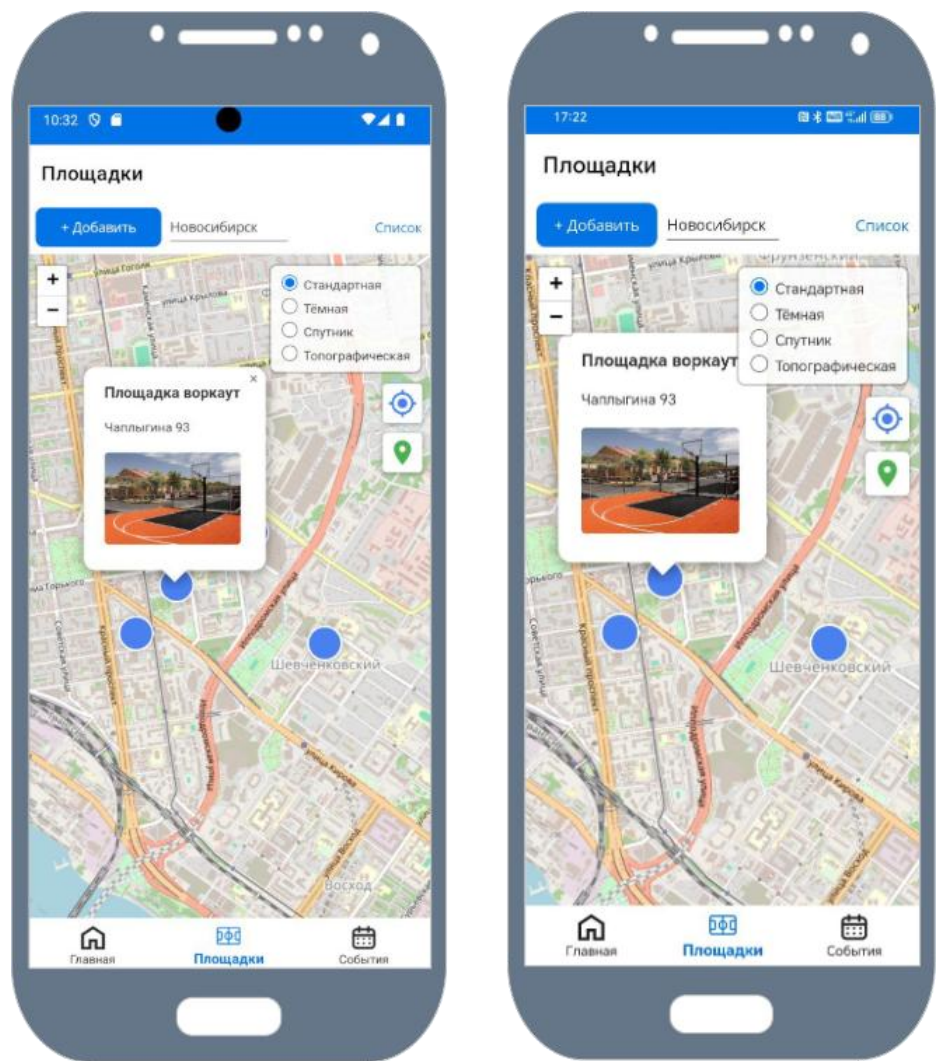


Рисунок 12 – Экран с навигационной картой

Экран навигационной карты так же отображается корректно на обоих устройствах, все элементы интерфейса расположены на своих местах. Карта отображается в оптимальном размер, так чтоб элементы настройки карты были на своих местах. В нижней части экрана находятся навигационные элементы, которые обеспечивают быстрый доступ к различным разделам приложения, таким как главная страница, площадки и события. Это позволяет пользователям легко переключаться между различными функциями приложения, обеспечивая удобство и эффективность использования.

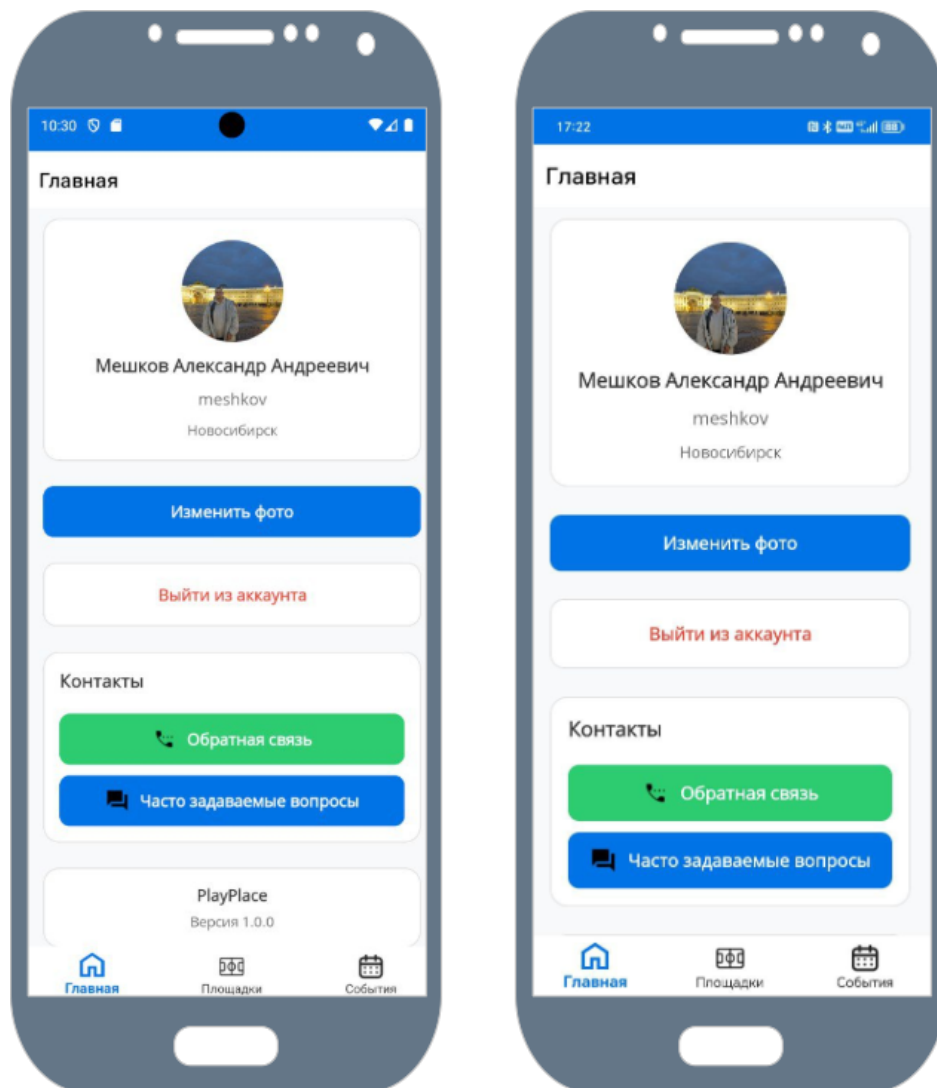


Рисунок 13 –Экран профиля

Экраны профиля тоже отображаются корректно, поля и кнопки не сдвинуты, фотография профиля и данные пользователя отображены корректно. Представлены кнопки для быстрого доступа к страницам обратной связи и часто задаваемым вопросам. Эти кнопки выполнены в зеленом и синем цветах, что делает их легко различимыми и доступными для пользователей.

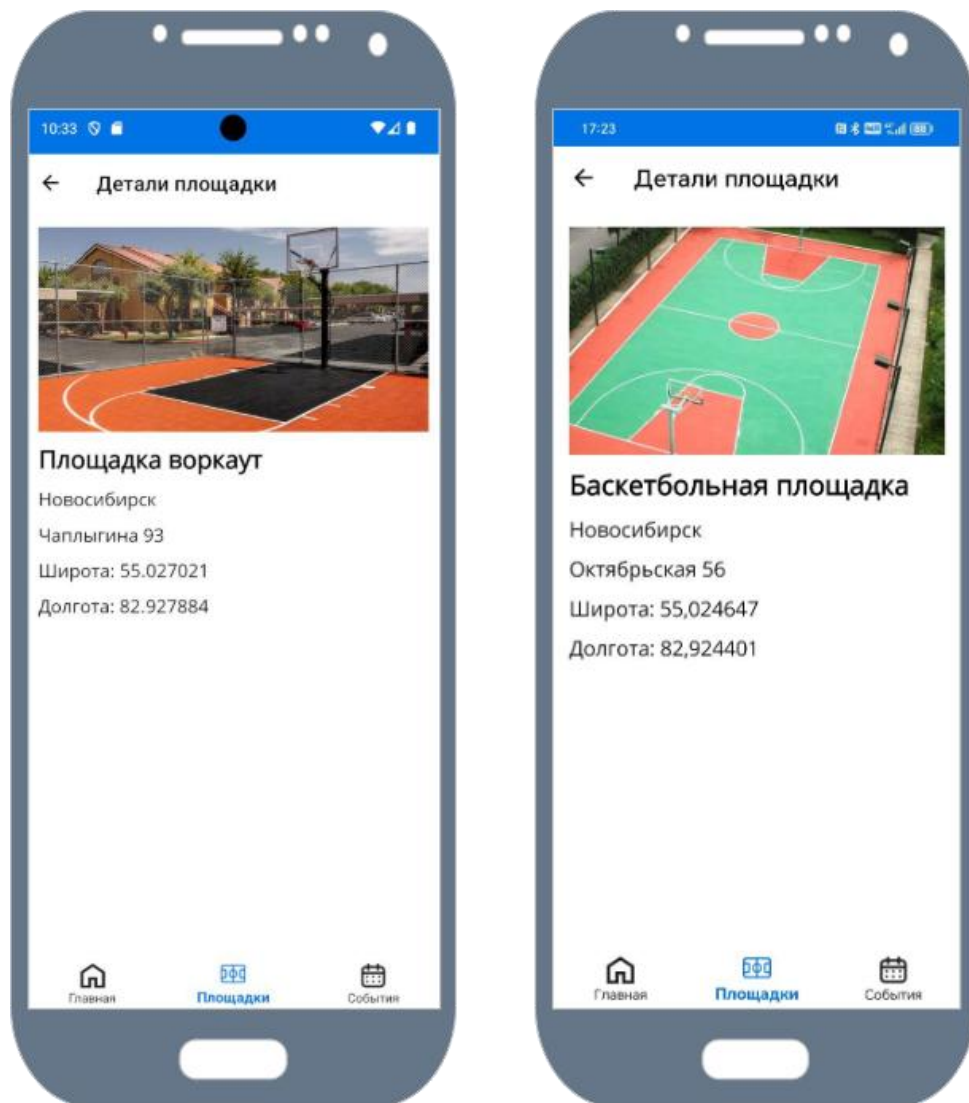


Рисунок 14 – Экран подробной информации о площадке

На экране подробной информации о площадке так же не обнаружено ошибок. Масштаб элементов сохранен. Фотография и данные о площадке отображаются корректно.

В процессе тестирования дизайна приложения на Android SDK 24 и более поздних версиях, в том числе SDK 34, выполнены проверки отображения основных элементов интерфейса на различных диагоналях экранов. Результаты тестирования показали, что все страницы приложения, отображаются корректно и без существенных отклонений на двух устройствах. Элементы интерфейса, такие как NavigationBar и statusBar, сохраняют нормальный размер и расположение. Кнопки, поля ввода и другие элементы на страницах приложения также остаются на своих местах и не имеют смещений или искажений. Масштабирование элементов также

осуществлено правильно, что обеспечивает удобное использование приложения на разных устройствах.

Таким образом, на основании результатов тестирования можно сделать вывод о качественной работе дизайна приложения.

4.2 Протокол тестирования функционала приложения

Для тестирования функционала были разработаны тест-кейсы, каждый из которых описывает небольшую последовательность действий с конкретным результатом. Благодаря тест-кейсам всегда известно, как и что протестировать оптимальным количеством проверок, и не забыть о нюансах, так как записан каждый шаг. Разработанные тест-кейсы представлены в таблицах от 4.2.1 до 4.2.5.

Таблица 4.2.1 –Test Case -1

Идентификатор	ТС001
Приоритет	Наивысший
Название	Вывод добавленной площадки
Актеры	Пользователь
Предусловия	Открыто приложение и пользователь авторизован
Шаги воспроизведения и альтернативные сценарии.	1.Прейдти на активность «Площадки» 2.Нажать на кнопку добавить 3.Ввести необходимые поля тестовыми данными и выбрать место на карте 4.Вернуться на активность карты
Тестовые данные	Город: city Название: platform Адрес: address Фото: img.png
Ожидаемый результат	Площадка добавлена и отображается на карте
Фактический результат	Площадка добавлена и отображается на карте
Постусловие	Площадка добавлена в систему, и пользователь может ее посмотреть

Таблица 4.2.2 – Test Case – 2

Идентификатор	ТС002
Приоритет	Наивысший
Название	Вывод добавленной площадки
Актеры	Пользователь
Предусловия	Открыто приложение и пользователь авторизован
Шаги воспроизведения и альтернативные сценарии.	1.Прейдти на активность «События» 2.Нажать на кнопку добавить 3.Ввести необходимые поля тестовыми данными 4.Вернуться на активность событий
Тестовые данные	Название: event Город: city Адрес: address Время начала: times Время конца: timee
Ожидаемый результат	Событие добавлено и отображается в списке
Фактический результат	Событие добавлено и отображается в списке
Постусловие	Событие добавлено в систему, и пользователь может его посмотреть

Таблица 4.2.3 – Test Case – 3

Идентификатор	ТС003
Приоритет	Наивысший
Название	Изменение фотографии профиля
Актеры	Пользователь
Предусловия	Открыто приложение и пользователь авторизован
Шаги воспроизведения и альтернативные сценарии.	1.Прейдти на активность «Главная» 2.Нажать на кнопку изменить фотографию 3.Выбрать тестовую фотографию на устройстве 4.Вернуться в профиль
Тестовые данные	Фотография: img.png
Ожидаемый результат	Фотография профиля поменялась

Продолжение таблицы 4.2.3

Фактический результат	Фотография профиля поменялась
Постусловие	Фотография в профиле пользователя поменялась на выбранную фотографию с устройства

Таблица 4.2.4 – Test Case – 4

Идентификатор	ТС004
Приоритет	Наивысший
Название	Регистрация
Актеры	Пользователь
Предусловия	Открыто приложение
Шаги воспроизведения и альтернативные сценарии.	1.Открыть приложение 2.Из предложенного перечня выбрать регистрацию 3.заполнить поля тестовыми данными 4.Нажать на кнопку зарегистрироваться
Тестовые данные	Логин: user Пароль: user123 Почта: user@gmail.com Имя: username Фамилия: usersurname Отчество: userpatronymic Номер телефона: 89134563721 Город: city
Ожидаемый результат	Пользователь зарегистрирован
Фактический результат	Пользователь зарегистрирован
Постусловие	Пользователь зарегистрировался в системе

Таблица 4.2.5 – Test Case – 5

Идентификатор	ТС005
Приоритет	Наивысший
Название	Авторизация
Актеры	Пользователь

Продолжение таблицы 4.2.5

Предусловия	Открыто приложение
Шаги воспроизведения и альтернативные сценарии.	1.Открыть приложение 2.Из предложенного перечня выбрать «Войти» 3.заполнить поля тестовыми данными 4.Нажать на кнопку ввойти
Тестовые данные	Логин: user Пароль: user123
Ожидаемый результат	Пользователь авторизован
Фактический результат	Пользователь авторизован
Постусловие	Пользователь авторизовался в системе

Заключение

Курсовой проект был успешно реализован с целью разработки приложения для поиска спортивных площадок и организации совместных тренировок. В ходе выполнения проекта были осуществлены исключительно важные задачи, которые в итоге позволили создать интуитивно понятное приложение для отслеживания физической активности.

Первоначально была проведена тщательная исследовательская работа по изучению предметной области, что позволило углубиться в основы технологий навигационных систем и изучить наиболее актуальные разработки на рынке. Благодаря этому полученное понимание потребностей пользователей позволило определить основные функциональные возможности, которые стоит включить в приложение.

Следующим этапом произведен тщательный анализ приложения с точки зрения конечных пользователей. Получилось выявить ключевые функции, которые пользователи ожидают от приложения, такие как достоверность, простота использования.

На основе собранных данных разработан пользовательский интерфейс, который отличается своей элегантностью и интуитивно понятный дизайн, который позволяет пользователям легко находить спортивные площадки не далеко от своего местоположения и создавать на них события.

Необходимый функционал реализован с высокой степенью надежности, что получилось достичь путем тщательного тестирования приложения. Тесты проведены с целью обеспечить стабильность работы, соответствие заявленным требованиям и общую удовлетворенность пользователей.

В итоге, результатом данного курсового проекта стало создание приложения для поиска спортивных площадок и организация совместных тренировок, которое сочетает в себе функционал полностью отвечающие потребностям пользователей.

Библиография

Нормативно – правовые акты:

1 ГОСТ Р 2.105-2019. ЕСКД. Общие требования к текстовым документам. – Москва: Стандартинформ, 2019. – 36 с

Электронные ресурсы:

1 Android Developers [электронный ресурс]. – Документация Android Studio. – URL: <https://developer.android.com/docs/> (дата обращения: 25.11.2024)

2 Освой Android играючи [Электронный ресурс]. – Уведомления. – URL: <https://developer.alexanderklimov.ru/android/notification.php> (дата обращения: 25.11.2024)

3 OpenStreetMapAPI [Электронный ресурс]: Документация к API – Режим доступа к руководству: <https://www.openstreetmap.org/#map=2/69.6/-74.9> (дата обращения: 20.03.2025)

4 .NET MAUI и C# [Электронный ресурс]: Документация к разработке в .NET MAUI: <https://metanit.com/sharp/maui/1.1.php> (дата обращения: 20.03.2025)

Приложение А

Министерство образования Новосибирской области
ГБПОУ НСО «Новосибирский авиационный технический колледж
имени Б.С. Галушака»

Разработка мобильного приложения для поиска спортивных
площадок и организация совместных тренировок

Техническое задание
НАТКиГ.201000.010.000ПЗ

Выполнил:
Студент группы ПР-23.106
Мешков А.А.

2025

					НАТКиГ.201000.010.000ПЗ	Лист
Изм.	Лист	№ докум	Подпись	Дата		38

Содержание

Введение.....	40
1 Назначения разработки.....	40
2 Требования к мобильному приложению	41
2.1 Требования к функциональным характеристикам	41
2.2 Требования к надёжности	41
2.3 Условия эксплуатации	41
2.4 Требования к составу и параметрам технических средств.....	41
2.5 Требования к информационной и программной совместимости.....	41
2.6 Требования к защите информации	42
2.7 Требования к маркировке и упаковке	42
3 Требования к программной документации	42
4 Техничко-экономические показатели	42
5 Стадии и этапы разработки	43
6 Порядок контроля и приёмки.....	44

Введение

Настоящее техническое задание распространяется на разработку информационной системы «Разработка мобильного приложения для поиска спортивных площадок и организация совместных тренировок», используемого для поиска спортивных площадок и создания спортивных событий.

Наименование приложения: «PlayPlace».

Краткая характеристика области применения: мобильное приложение предоставляет возможность поиска спортивных площадок и создания спортивных событий.

Основанием для проведения разработки является Протокол № Уч-018/4 от 6 февраля 2025 года.

Наименование темы разработки – «Разработка мобильного приложения для поиска спортивных площадок и организация совместных тренировок».

Условное обозначение темы разработки – «PlayPlace».

1 Назначение разработки

Основное назначение приложения заключается в:

- обеспечении удобного и понятного интерфейса приложения для пользователя;
- предоставление навигационной карты для поиска спортивных площадок;
- добавление собственных спортивных площадок.

Лица, которые могут работать с данной системой:

администратор – управляет полностью всем приложением, следит за работоспособностью, следит за данными, добавленными в базу данных;

пользователь приложения – может смотреть добавленные площадки и события, создавать свои собственные площадки и события, фильтровать события и площадки по городу пребывания.

2 Требования к мобильному приложению

2.1 Требования к функциональным характеристикам

Требования к составу выполняемых функций:

- добавление площадок;
- добавление событий;
- просмотр событий;
- просмотр площадок;
- выбор города из доступного списка для событий и площадок;
- изменение фотографии пользователя.

2.2 Требования к надежности

Обеспечение устойчивого функционирования должно выполняться несколькими действиями:

- организация стабильного интернет-соединения.

2.3 Условия эксплуатации

Пользователь должен иметь практические навыки использования мобильного устройства под управлением операционной системы Android.

2.4 Требования к составу и параметрам технических средств

Для работы приложения на устройстве требуется: объем свободной памяти не менее 2 ГБ, версия устройства Android 8 и выше.

2.5 Требования к информационной и программной совместимости

Проектирование взаимодействия с файловой системой должно быть выполнено в рамках разработки курсового проекта. При разработке взаимодействия с файловой системой должен быть использован язык программирования C#.

2.6 Требования к защите информации

Доступ к информации API и серверной части предоставляется только администратору.

2.7 Требования к маркировке и упаковке

Требования к маркировке и упаковке не предъявляются.

3 Требования к программной документации

Состав программной документации должен включать в себя:

- техническое задание;
- пояснительная записка.

4 Техничко-экономические показатели

Экономические преимущества разработки и ориентировочная экономическая эффективность не рассчитывается.

5 Стадии и этапы разработки

Таблица 5 – Стадии разработки

№	Этапы разработки КП	Сроки выполнения	Отчётность
1	Определение цели и задач, объекта и предмета исследования	24.02.2025	Пояснительная записка
2	Описание предметной области	26.02.2025	Пояснительная записка
3	Выбор технологии, языка и среды программирования	01.03.2025	Пояснительная записка
4	Оформление технического задания	05.03.2025	Техническое задание
5	Проектирование UI/UXдизайна	10.03.2025	Спецификации программного обеспечения
6	Разработка мобильного приложения	14.03.2025	Схема структурная системы и спецификации компонентов
7	Выбор API	15.03.2024	
8	Отладка и тестирование приложения	16.03.2025	Тексты программных компонентов
9	Оформление документации	18.03.2025	Программная документация
10	Защита	28.04.2025	

6 Порядок контроля и приёмки

Виды испытаний – защита курсового проекта.

Общие требования к приёмке:

- техническое задание;
- пояснительная записка;
- презентация;
- программный продукт.

					НАТКиГ.201000.010.000ПЗ	Лист
						44
Изм.	Лист	№ докум	Подпись	Дата		

Приложение Б

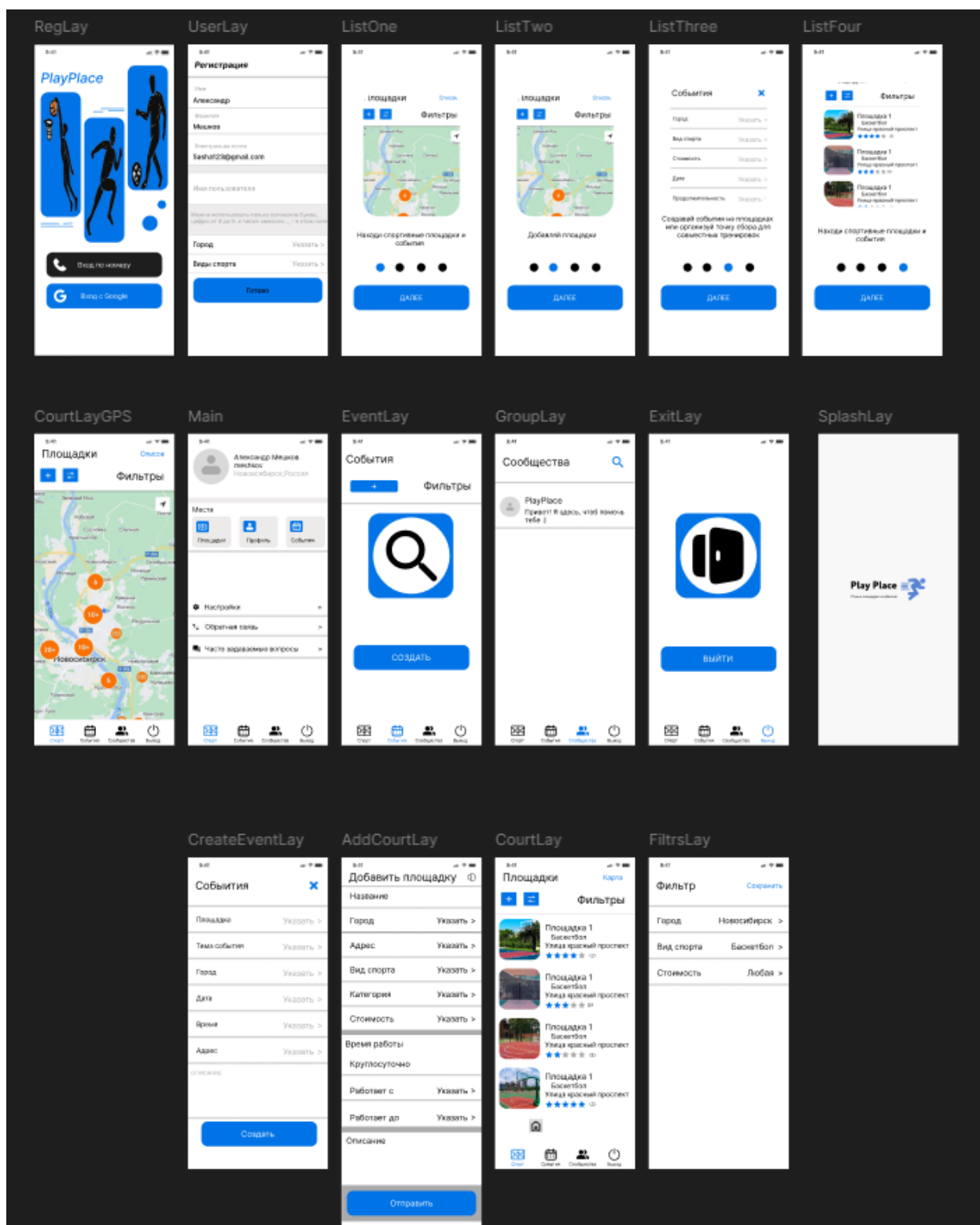


Рисунок 15 – Переходы между экранами в приложении