

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук  
Образовательная программа «Компьютерные науки и анализ данных»

**Отчет о программном проекте на тему:**  
**Мобильное приложение для трэвел-блогеров**

**Выполнил студент:**

группы #БКНАД221                      Голубкова Анастасия Ярославовна

**Руководитель проекта:**

Голубков Ярослав Владимирович  
Директор ООО "Датахауз"

# Содержание

<b>1</b>	<b>Введение</b>	<b>4</b>
<b>2</b>	<b>Обзор литературы</b>	<b>6</b>
<b>3</b>	<b>SWOT-анализ</b>	<b>7</b>
<b>4</b>	<b>Описание разработки</b>	<b>9</b>
4.1	Часть взаимодействия с пользователем . . . . .	9
4.2	Бекэнд-сервер . . . . .	10
4.3	База данных . . . . .	10
4.4	База данных . . . . .	10
<b>5</b>	<b>Angular</b>	<b>11</b>
5.1	Страница авторизации . . . . .	11
5.2	Главная страница приложения . . . . .	12
5.3	Функция Переименовать . . . . .	12
5.4	Функция Удалить . . . . .	12
5.5	Функция Добавить описание . . . . .	14
5.6	Функция Добавить фото . . . . .	14
5.7	Функция Просмотр фото . . . . .	14
5.8	Добавление маршрута . . . . .	14
5.9	Просмотр маршрута . . . . .	15
5.10	Функция Поделиться . . . . .	15
<b>6</b>	<b>ASP.NET</b>	<b>16</b>
6.1	Контроллеры . . . . .	16
<b>7</b>	<b>Удаленный сервер</b>	<b>16</b>
<b>8</b>	<b>Заключение</b>	<b>16</b>

## Аннотация

Разработанное по теме курсового проекта «Мобильное приложение для тревэл-блогеров» решение представляет собой мобильное приложение, обеспечивающий его работу бэкэнд-сервер и базу данных для хранения информации. Мобильное приложение использует для отображения картографической информации широко известный open source проект OpenStreet Map, который я адаптировала для работы в мобильном приложении. На карте в приложении можно отмечать места и добавлять к ним описания и фотографии. Места объединяются маршрутом (или целью) поездки. Данные из мобильного приложения передаются на бэкэнд-сервер, который структурирует их и записывает в базу данных для долговременного хранения. Бэкэнд-сервер обеспечивает возможность поиска и обмена информацией, а также синхронизирует использование нескольких устройств пользователя для одного аккаунта. Для документирования REST API был использован стандарт OpenApi 2 и Swagger, обеспечивающий не только документирование методов, но и удобный интерфейс для разработчика. В процессе разработки использовался Docker Compose.

## Ключевые слова

Мобильное приложение, Angular, Ionic, OpenStreetMap, Geocoding, .Net, C, EF Core, REST API, PostgreSQL, Docker

# 1 Введение

В последнее время все большую известность набирает профессия блогеров. Популярным направлением являются путешествия. Специально для трэвел-блогеров и их подписчиков разработано мобильное приложение. Также им сможет пользоваться любой желающий.

В этом приложении можно просматривать карту, отмечать, сохранять любимые места и добавлять к ним описание и фотографии. Все сохраненные данные будут отображаться на карте. Блогер сможет составлять из своих сохраненных точек маршрут для туристов и фоловеров, а после этого делиться им со своими подписчиками. Это поможет наладить общение блогеров с аудиторией и повысить вовлеченность пользователей приложения.

Благодаря использованию сервера и базы данных, пользователи могут использовать несколько устройств для одного аккаунта, а все данные будут надежно защищены и сохранены. Также наличие базы данных обеспечивает возможность реализовывать гораздо больший функционал, чем при ее отсутствии.

На карте работает функция геолокации пользователя, что облегчает простановку геолокации точки. Стоит отметить, что основной целью данного проекта является изучение мобильной разработки, работы с картами, разработки с Ionic, Angular, HTML, .Net, а также PostgreSQL.

Для реализации данного проекта я выбрала стандартную клиент-серверную архитектуру для обеспечения взаимодействия клиентов с базой данных.

База данных хранится на удаленном сервере. На нем также хранится бэкенд-приложение.

Запросы отправляются не напрямую на сервер, а с помощью прокси-сервер Nginx. Он обеспечивает безопасность взаимодействий, так как был настроен на отправку запросов на сервер через определенный порт. Таким образом, запросы с других портов не будут приниматься сервером, и его работа не будет нарушена.

Мой проект реализован использованием Angular, так как он позволяет разработать и веб-приложение, и мобильное приложение практически не меняя кода программы. Он позволяет работать не с JavaScript, а с TypeScript - языке, разработанном Microsoft с учетом всех недостатков JavaScript.

Я использовала Ionic, так как этот фреймворк стандартизирован, более простой для изучения, чем Kotlin Multiplatform, и более строгий, чем React. Технология надежнее работает и не имеет ошибок, характерных для новых фреймворков.

Приложение реализовано под Android, однако для того чтобы оно работало на IOS можно легко добавить несколько команд в терминал. К сожалению, у меня не было возмож-

ности тестировать приложение на IOS.

Для разработки я использовала Docker, потому что он позволяет использовать готовые образы - PostgreSQL и pgAdmin.

## 2 Обзор литературы

### TRIP ADVISOR

Наверное, самый известный сервис для путешественников. Работает с 2000 года и позволяет заранее не только спланировать путешествие, но и заказать отели и трансферы, а также арендовать машину и посмотреть описание достопримечательностей. Содержит до 900 млн отзывов реальных путешественников об объектах и является основным конкурентом. Но информация подается с минимальной персонализацией, блогер не может вести полноценный блог в рамках приложения.

### MAPS.ME

Некоторые блогеры, пользователи этого приложения, делятся маршрутом (а точнее точками на карте, собранными в одну группу) по достопримечательностям в каком-то городе.

### Visit a City

Собраны руководства и тревел-блоги от реальных путешественников для планирования поездки, экскурсий и посещения достопримечательностей. Есть подборка наиболее интересных мест для каждого города, которые стоит посетить, если мало времени. Информация слишком жестко структурирована, не очень удобная возможность для обсуждения или оценки.

### Guides by Lonely Planet

Есть подробная информация про отели, рестораны, достопримечательности для многих стран, более чем 9500 городов доступны в приложении. Можно купить платные гайды по стране или городу. Нет возможности вести личный блог и делиться маршрутами и впечатлениями от путешествий

### Klook

Есть подробная информация про отели, рестораны, достопримечательности, страховку, скидки. Нет возможности вести личный блог и делиться маршрутами и впечатлениями от путешествий

**Общий вывод** из анализа рынка такой: приложение стоит серьезно развивать только в случае, если будет найден стратегический инвестор, так как много конкурентов высокого уровня. Однако, возможно, что ориентация на блогинг станет той фишкой, которая позволит приложению взлететь.

### 3 SWOT-анализ

Сильные стороны	<ul style="list-style-type: none"><li>• Наличие платежеспособной целевой аудитории (блогеров, путешественников)</li><li>• Выход на глобальный рынок</li></ul>
Слабые стороны	<ul style="list-style-type: none"><li>• Незнание на рынке</li><li>• Функциональность начального уровня</li><li>• Требуется интернет</li><li>• Отсутствие финансирования</li></ul>
Возможности	<ul style="list-style-type: none"><li>• Рост вместе с ростом рынка блогинга/путешественников</li><li>• Быстрая адаптация к требованиям целевой аудитории</li></ul>
Угрозы	<ul style="list-style-type: none"><li>• Отсутствие патентной защиты</li><li>• Трудно прогнозируемый «рынок впечатлений»</li></ul>

	Возможности	Угрозы
Сильные стороны	<ul style="list-style-type: none"> <li>• Прямые продажи</li> <li>• Простота перевода на другие языки</li> <li>• Доступность для большой аудитории</li> </ul>	<ul style="list-style-type: none"> <li>• При попадании в тренд «рынка впечатлений» высокая скорость привлечения клиентов затруднит «догоняющим» конкурентам набор клиентской базы</li> </ul>
Слабые стороны	<ul style="list-style-type: none"> <li>• Высокая стоимость входа в рынок затрудняет продвижение</li> <li>• Органический рост не позволит захватить рынок</li> </ul>	<ul style="list-style-type: none"> <li>• Отсутствие финансирования затруднит и продвижение на рынке и развитие продукта</li> <li>• Требуется инвестор</li> </ul>



## 4 Описание разработки

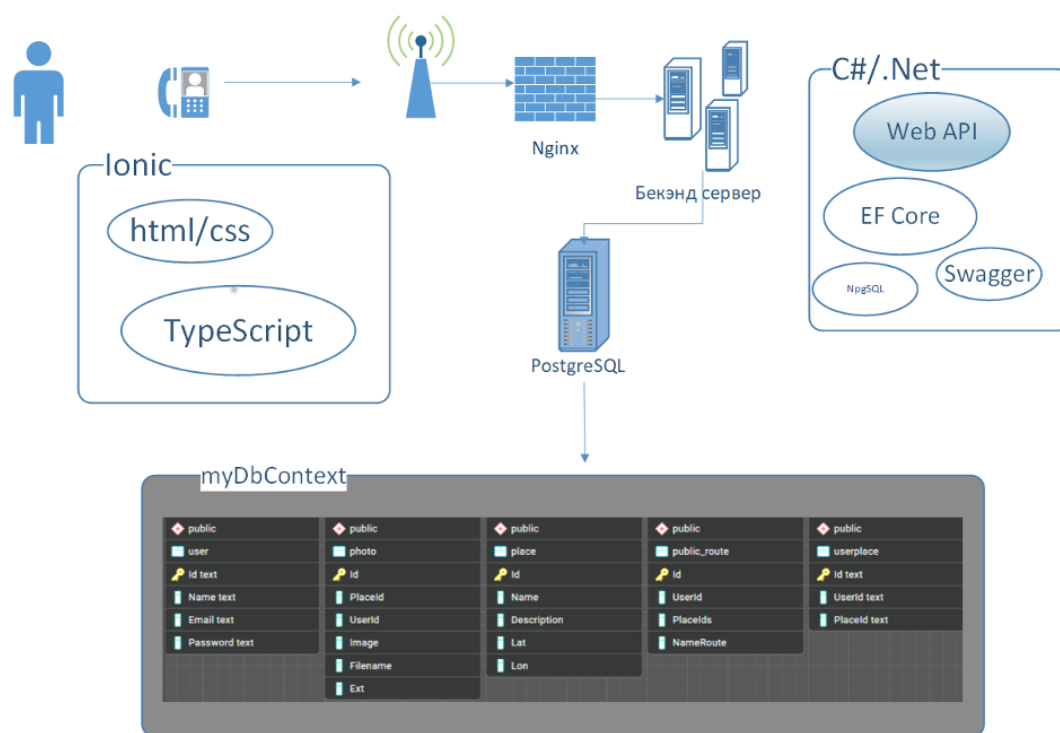


Рисунок 4.1: Схематичное описание работы

### 4.1 Часть взаимодействия с пользователем

Мобильное приложение написано мной на кроссплатформенном фреймворке Ionic 7. Код, относящийся к взаимодействию с пользователем (то есть то, что пользователь видит на экране мобильного приложения), написан с использованием фреймворка Angular, который включает в себя business-layer на TypeScript и presentation-layer в виде HTML и файлов описания стиля css. Также есть специальные встроенные сервисы, например routing, отвечающие за корректную переадресацию между страницами. На языке TypeScript написан http-client, в котором реализованы функции, отправляющие запросы бекэнд-серверу и получающий его ответы. Асинхронное взаимодействие обеспечивается использованием библиотеки RxJs, реализующий асинхронность с помощью паттерна subscribe. Разработка мобильного приложения проходила в 2 этапа: на первом этапе я использовала эмулятор мобильного телефона в Ionic для первоначальной отладки. После того, как мобильное приложение в целом заработало, я перешла к отладке на реальном мобильном телефоне (Android) с использованием Android Studio. Также для разработки мобильного приложения я использовала IDE Visual Studio Code.

## 4.2 Бекэнд-сервер

Для создания бекэнд-сервера я изучила язык C и платформу .Net 8. Для обеспечения работы с базой данных я изучила и использовала технологию доступа к данным Entity Framework Core (EF Core) и драйвер Npgsql для доступа к серверу БД Postgresql. Бекэнд-сервер реализует различные методы web-api для обеспечения работы мобильного приложения. Взаимодействие происходит следующим образом: пользователь вызывает функцию в своем приложении на телефоне (например, нажав кнопку), мобильное приложение отправляет http-запрос на удаленный бекэнд-сервер, этот запрос первоначально получает специальный сервер Nginx, который перенаправляет запрос на бекэнд-сервер, на котором работает серверная часть решения, которая при необходимости обращается к базе данных. Ответ возвращается пользователю в обратном порядке. Посредничество через Nginx нужно для обеспечения безопасности передачи данных, балансировки нагрузки (не требовалось в проекте) и безопасности самого бекэнд-сервера. В процессе разработки я использовала код бекэнд-сервера, скомпилированный для работе на Windows 11. Для работы мобильного приложения на телефоне я компилирую и размещаю бекэнд-сервер на удаленном сервере, работающем под управлением Ubuntu 24.04. Такую возможность компиляции одного моего приложения бекэнд сервера для различных операционных систем дает мне кроссплатформенная платформа .Net, именно поэтому она и была выбрана для разработки. EF Core не только дает возможность работать с БД на абстрактном уровне, но и с помощью механизмов миграции позволяет не только первоначально создать все таблицы в базе данных, но и корректировать их структуру в процессе разработки. Для разработки я изучила и использовала IDE Visual Studio 2022 Community Edition

Описание web-api:

## 4.3 База данных

База данных для разработки физически находится на компьютере разработчика, там же, где и бекэнд-сервер, и установлена в Docker. База данных для работы с мобильном приложении физически находится на том же удаленном сервере, где и удаленный бекэнд-сервер, и тоже установлена в Docker. Для взаимодействия с базой данных созданы специальные функции-контроллеры.

## 4.4 База данных

Схема базы данных имеет следующий вид:

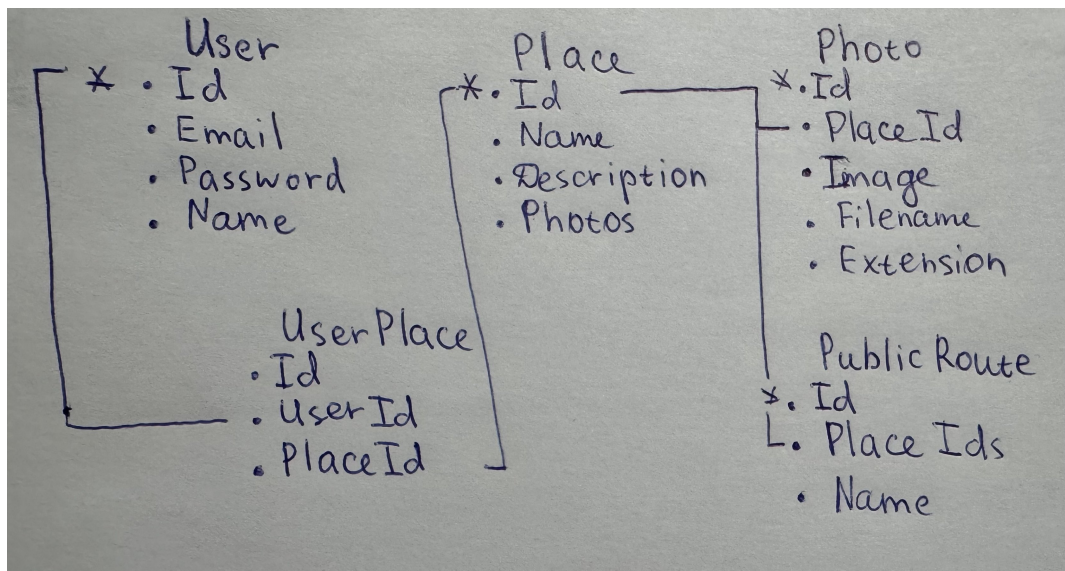


Рисунок 4.2: База данных

Для отладки работы приложения БД была заполнена фейковыми данными с помощью вызовов контроллеров через Swagger. Для работы с БД я использовала PG Admin 4, также установленный в Docker

## 5 Angular

### 5.1 Страница авторизации

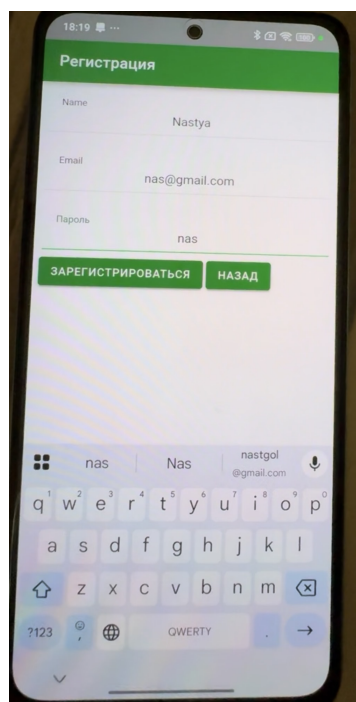


Рисунок 5.1: Страница регистрации

При открытии приложения пользователь попадает на страницу авторизации. На ней предлагается либо заполнить email и пароль, либо зарегистрироваться. Естественно, при вводе несуществующего в БД email или пароля приложение выдаст предупреждение - появится сообщение о неправильном вводе данных. При нажатии на кнопку Зарегистрироваться происходит переход на другую страницу. На ней нужно ввести Имя Пользователя, email и пароль. Эти данные передадутся в БД, и пользователь может использовать их для входа

## **5.2 Главная страница приложения**

После авторизации перед пользователем открывается карта (см. рис. 5.2). Если геопозиция выключена, приложение запросит доступ к ней. При включенной геопозиции она отобразится красной точкой на карте. При нажатии на карту один раз появится уведомление с вопросом создать новый маркер. При положительном ответе маркер с синей иконкой появится на карте. При передвижении этого маркера по карте (то есть в момент отпускания пальца с маркера, функция DragEnd) появится уведомление с вопросом, хочет ли пользователь добавить маркер в эту точку (см. рис. 5.3). Если нет, то маркер исчезнет, если да, то появится окно для ввода названия места. После добавления названия маркер поменяет иконку с синей на зеленую (зеленый листок), а также при нажатии на него появится всплывающее окно с названием и функциями: переименовать, Удалить, Добавить описание, Добавить Фото, Просмотр Фото. Ниже представлены подробности о каждой из функций.

## **5.3 Функция Переименовать**

Переименовать появится окно для ввода, которое будет заполнено текущим названием, то есть пользователь сможет его редактировать. После ввода нового имени маркера название поменяется (и в БД тоже).

## **5.4 Функция Удалить**

При нажатии на эту кнопку приложение запросит подтверждение удаления маркера с карты. При положительном ответе маркер удалится с карты, а все данные о нем сотрутся из БД.

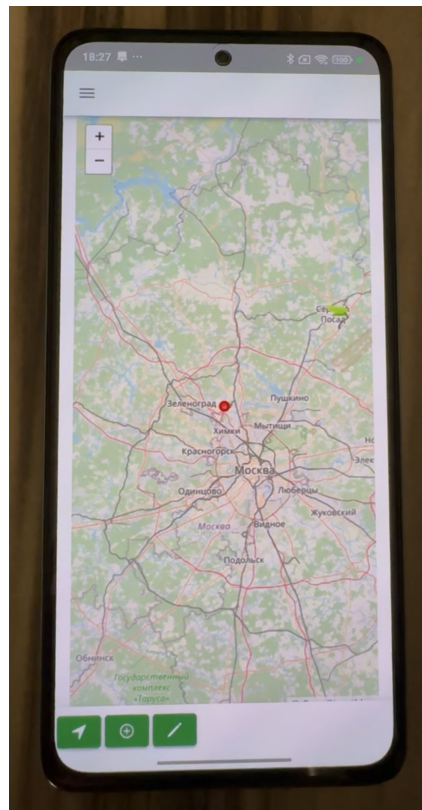


Рисунок 5.2: Главная страница. На ней красная точка геопозиции и зеленый маркер.

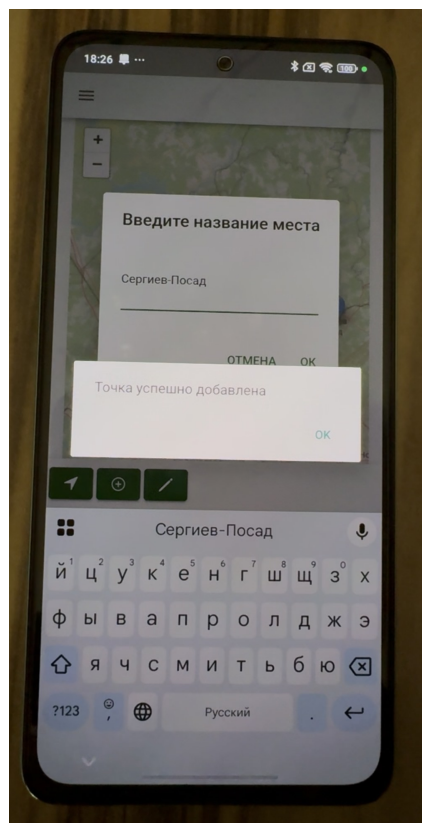


Рисунок 5.3: Функционал добавления маркера

## 5.5 Функция Добавить описание

Эта функция аналогична функции переименовать, только в окне для ввода появляется не имя маркера, а текущее описание. Соответственно, пользователь может отредактировать его и сохранить, а новое описание передастся в БД.

## 5.6 Функция Добавить фото

При нажатии на эту кнопку откроется галерея телефона пользователя. Он сможет выбрать любое фото и сохранить его. Файл фото (типа bytes) передается в БД. Для доступа к галерее пользователя использовалась специальная библиотека Capacitor, позволяющая взаимодействовать с устройством пользователя (эта же библиотека использовалась и для доступа к геолокации).

## 5.7 Функция Просмотр фото

При выборе этой кнопки открывается отдельная страница - на ней посередине расположена первая фотография из всех добавленных фотографий к этому месту, а под ней появляется описание (см. рис. 5.4). При слишком длинном описании страницу можно прокрутить вниз. Также на странице есть кнопка Назад (возвращение к карте) и Ещё фото - при нажатии на нее также появляется новая страница со списком всех фото подряд. Эту страницу также можно пролистать вниз. Вернуться на главную страницу можно, нажав кнопку Назад.

## 5.8 Добавление маршрута

Помимо работы с каждым маркером отдельно, можно объединять группы маркеров в маршруты (эта функция делает приложение актуальным для блогеров). Чтобы перейти к созданию маршрута нужно нажать на кнопку + внизу экрана. На экране появится вопрос, хочет ли пользователь создать новый маршрут, а также нужно ввести для него название. Далее снова откроется карта, где нужно выбрать маркеры. При нажатии на маркер он становится красным (то есть сигнализирует о добавлении его в маршрут), при повторном нажатии - снова зеленым (удаление из маршрута). Когда все нужные маркеры выделены, нужно нажать на кнопку Подтвердить выбор. Тогда все нужные данные передадутся в удаленную БД.

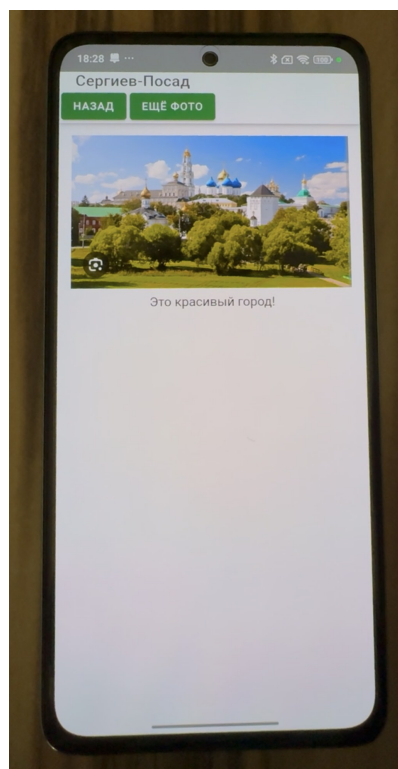


Рисунок 5.4: Страница с фото + описание

## 5.9 Просмотр маршрута

При нажатии этой кнопки в меню появится окно, где нужно выбрать маршрут по названию. После подтверждения выбора маркеры, составляющие данный маршрут, отобразятся на карте.

## 5.10 Функция Поделиться

Эта функция реализована двумя разными способами - в меню при нажатии на кнопку поделиться требуется ввести email пользователя приложения, с которым данный пользователь хочет поделиться маршрутом, а также выбрать сам маршрут из предложенного списка. Данные передадутся в БД, и отобразятся на экране другого пользователя после повторной авторизации. Аналогичная функция реализована другим способом, который более актуален для блогеров с большой аудиторией. Для начала нужно получить уникальный код маршрута, нажав на кнопку в меню Получить код. Далее пользователь сообщает этот код другому пользователю. Этот пользователь для получения маршрута на своей карте должен ввести код в окне ввода, которое появляется при нажатии на кнопку записи внизу экрана. Далее, если код правильный, то появится сообщение в котором говорится о необходимости перезагрузки приложения. После повторной авторизации маршрут появляется на карте, а маркеры на нем содержат всю необходимую информацию (название, фото, описание).

## 6 ASP.NET

### 6.1 Контроллеры

Все взаимодействия с БД осуществляются с помощью контроллеров. Эти функции принимают запрос из Angular со всеми необходимыми параметрами и названием самой вызываемой функции и обращаются к БД, которая хранится в переменной `_myDbContext`. Они либо изменяют содержимое БД (HttpPost), либо получают данные оттуда (HttpGet). Для получения объектов из базы данных вызывается метод вида `await _myDbContext.<Имя таблицы>.Where(<условие>).FirstOrDefaultAsync()`. Добавление объекта происходит через вызов метода вида `_myDbContext.<Имя таблицы>.Add()`. В конце каждой функции Post обязательно вызывается метод `SaveChanges()` от переменной `_myDbContext`.

## 7 Удаленный сервер

IP-адрес удаленного сервера указан в файле `environment` в Angular. Как было сказано ранее, посредником является Nginx. Он обеспечивает безопасность, возможность посылать запросы https, а также имеет возможность балансировки нагрузки, если запросы посылает сразу несколько пользователей.

## 8 Заключение

Результатом проекта является работающее мобильное приложение на Android. Приложение работает с бекэнд-сервером и базой данных, содержащей информацию о пользователях, местах, маршрутах и фотографиях. Бекэнд-сервер находится на удаленном сервере и размещен в интернете по адресу: <http://81.19.131.133:4344/swagger/index.html>

В процессе работы над проектом я не только разработала действующее мобильное приложение, бекэнд-сервер и настроила базу данных, но и изучила и опробовала в работе современные средства разработки, которые мы не изучали по учебному плану: Ionic, Angular, TypeScript, HTML, C, .Net, EF Core, Visual Studio Code, Visual Studio 2022.



## Список литературы

- [1] [Angular Documentation](https://angular.io/docs) (дата обр. 25.02.2025) URL: <https://angular.io/docs>
- [2] [Ionic Documentation](https://ionic.io/docs) (дата обр. 15.03.2025) URL: <https://ionic.io/docs>
- [3] [OpenStreetMap](https://www.openstreetmap.org/help) (дата обр. 07.02.2025) URL: <https://www.openstreetmap.org/help>
- [4] [TypeScript](https://www.typescriptlang.org/docs/) (дата обр. 13.02.2025) URL: <https://www.typescriptlang.org/docs/>
- [5] [Pro Angular](#), Adam Freeman, ISBN 978-1-4842-8175-8, 2022
- [6] [Mastering Ionic 2. The Definitive Guide](#), Griffiths J., 2017
- [7] [Learn Ionic 2. Develop Multi-platform Mobile Apps](#), Justin J., Jude J, ISBN 978-1-4842-2616-2, 2017