**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**Учреждение образования  
«Национальный детский технопарк»**

**Образовательное направление** «Информационные и компьютерные технологии»

(полное название образовательного направления)

«Вед-дизайн»

(название учебной программы)

**Исследовательский проект**

**«Веб-ресурс для обмена впечатлениями о туристических местах Беларуси»**

Исполнитель:

Клещукевич Артём Александрович,

учащийся

Шешко Максим Александрович,

учащийся

УО «Национальный

детский технопарк»

Руководитель проекта:

Андрейчук Ольга Николаевна,

Середа Ирина Александровна,

Сицко Владимир Александрович

Минск, 2025

**СОДЕРЖАНИЕ**

|  |  |
| --- | --- |
| ВВЕДЕНИЕ | 3 |
| 1. Актуальность темы | 3 |
| 1. Новизна | 3 |
| 1. Цель, объект, предмет, задачи исследования, гипотеза исследования | 3 |
| ТЕОРЕТИЧЕСКАЯ ЧАСТЬ | 4 |
| 1. Глава 1. Постановка задачи | 4 |
| * 1. Обзор существующих аналогов   2. Функциональное назначение | 4  7 |
| 1. Глава 2. Проектирование задачи | 8 |
| 2.1 Описание инструментов разработки | 8 |
| ПРАКТИЧЕСКАЯ ЧАСТЬ | 12 |
| 1. Глава 3. Программная реализация | 12 |
| 3.1 Описание разработанных классов | 12 |
| 3.2 Описание разработанных функций | 15 |
| 1. Глава 4. Тестирование | 16 |
| 1. Глава 5. Применение | 18 |
| 5.1 Назначение и условия применения | 18 |
| 5.2 Руководство пользователя | 18 |
| ЗАКЛЮЧЕНИЕ | 20 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ | 21 |
| ПРИЛОЖЕНИЯ | 22 |
| Приложение 1. Исходный код проекта | 22 |
| Приложение 2. Скриншоты интерфейса | 23 |

# **ВВЕДЕНИЕ**

## 

## **Актуальность проекта**

Актуальность проекта состоит в том, что представленный веб-service позволяет путешественникам по Беларуси обмениваться впечатлениями и местами, которые стоит посетить.

## 

## **Новизна проекта**

Новизну проекта можно обосновать в том, что существуют сложности выбора мест для посещения, обусловленные отсутствием достаточного количества информации, собранной в едином интернет-ресурсе.

Значимость этого проекта состоит в разработке веб-сервиса, объединяющего путешественников в сообщество для обмена впечатлениями и рекомендациями.

## **Цель, объект, предмет, задачи исследования, гипотеза исследования**

Цель исследования – разработка и оценка эффективности веб-ресурса, предназначенного для обмена фотографиями и опытом, полученным в ходе путешествий, с целью популяризации внутреннего туризма.

Объект исследования – веб-ресурс с функционалом публикации постов и интерактивной картой с метками локаций.

Предмет исследования – эффективность веб-ресурса в популяризации малоизвестных туристических мест Беларуси и удобстве в обмене опытом.

Гипотеза – разработка веб-ресурса, способного содействовать популяризации туристических мест Беларуси и способствовать обмену опытом.

Перед началом разработки данного исследовательского проекта было поставлено несколько задач:

* изучить фреймворк веб-разработки Bootstrap;
* провести поиск аналогов;
* ознакомиться с библиотекой создания интерактивных карт Leaflet;
* изучить фреймворк разработки приложений Express;
* исследовать базовые принципы работы с SQLite;
* создать веб-ресурс, который позволит делиться опытом своих путешествий и способствовать популяризации туристических мест;
* провести анализ эффективности разработанного веб-ресурса;

# **ТЕОРЕТИЧЕСКАЯ ЧАСТЬ**

## **1. ГЛАВА 1. ПОСТАНОВКА ЗАДАЧИ**

### **1.1 Обзор существующих аналогов**

По результатам поиска существующих аналогов было найдено несколько существующих веб-ресурсов.

Google Maps (рисунки 1.1 – 1.2) – это веб-картографический сервис, разработанный компанией Google, предоставляющий спутниковые снимки, панорамные изображения улиц, маршрутизацию, поиск мест и бизнесов, а также API для интеграции в сторонние приложения. Однако данный веб-ресурс не предоставляет возможностей для создания постов и не имеет социального взаимодействия.

Wikiloc (рисунки 1.3-1.4) – это онлайн-платформа для создания, хранения и обмена GPS-треками, ориентированная на туристов и любителей активного отдыха. Сервис позволяет пользователям загружать маршруты в форматах GPX, KML и других, дополняя их фотографиями, описаниями и метками. Несмотря на это данный ресурс обладает недостатками, такими как: низкая модерация, практически нет контента по Беларуси, сложный интерфейс для обычных пользователей.

Также веб-ресурсы Google Maps и Wikiloc не нацелены на популяризацию туристических локаций.

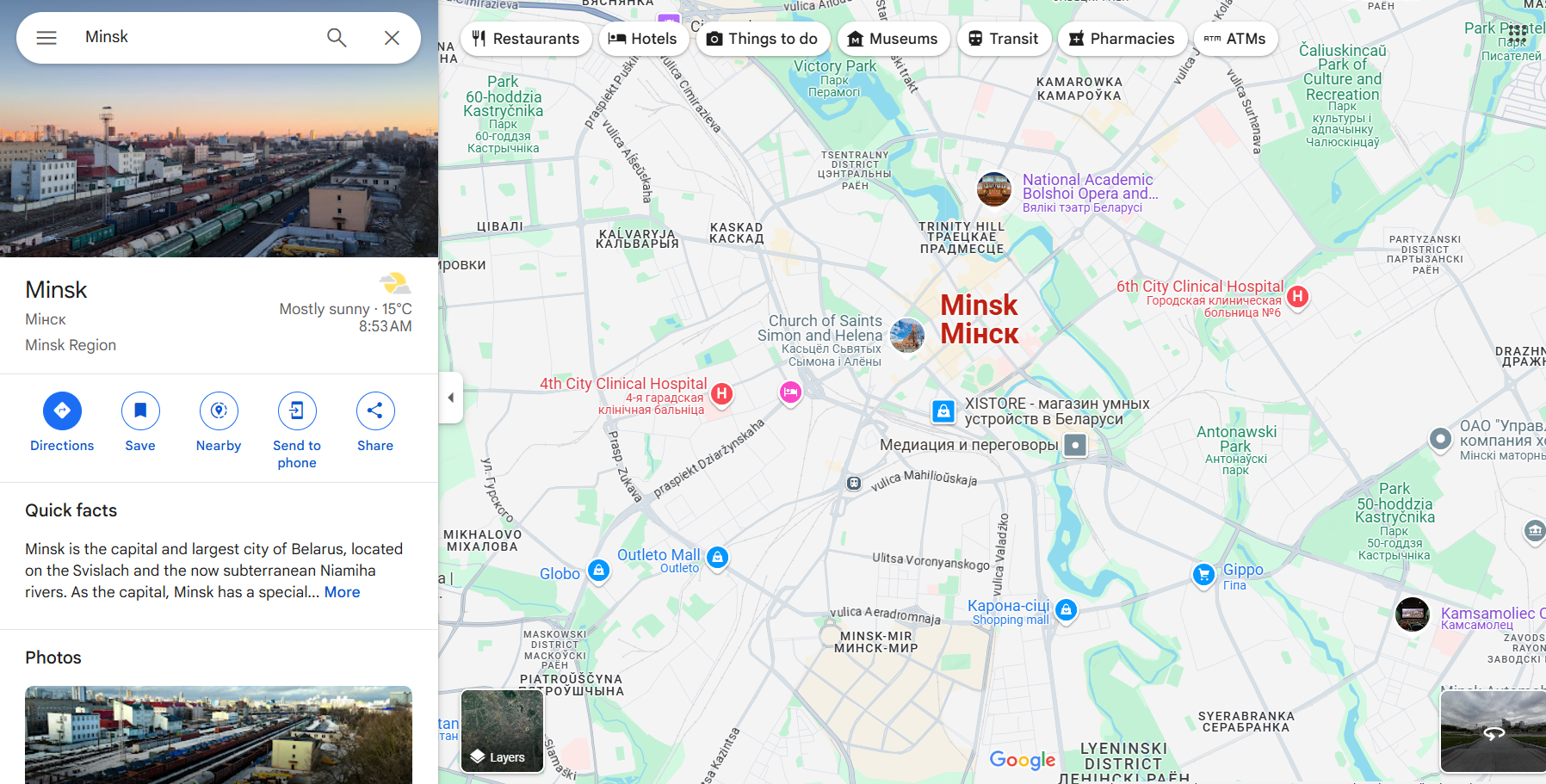


Рис. 1.1 – Информация о веб-ресурсе Google Maps и его интерфейс

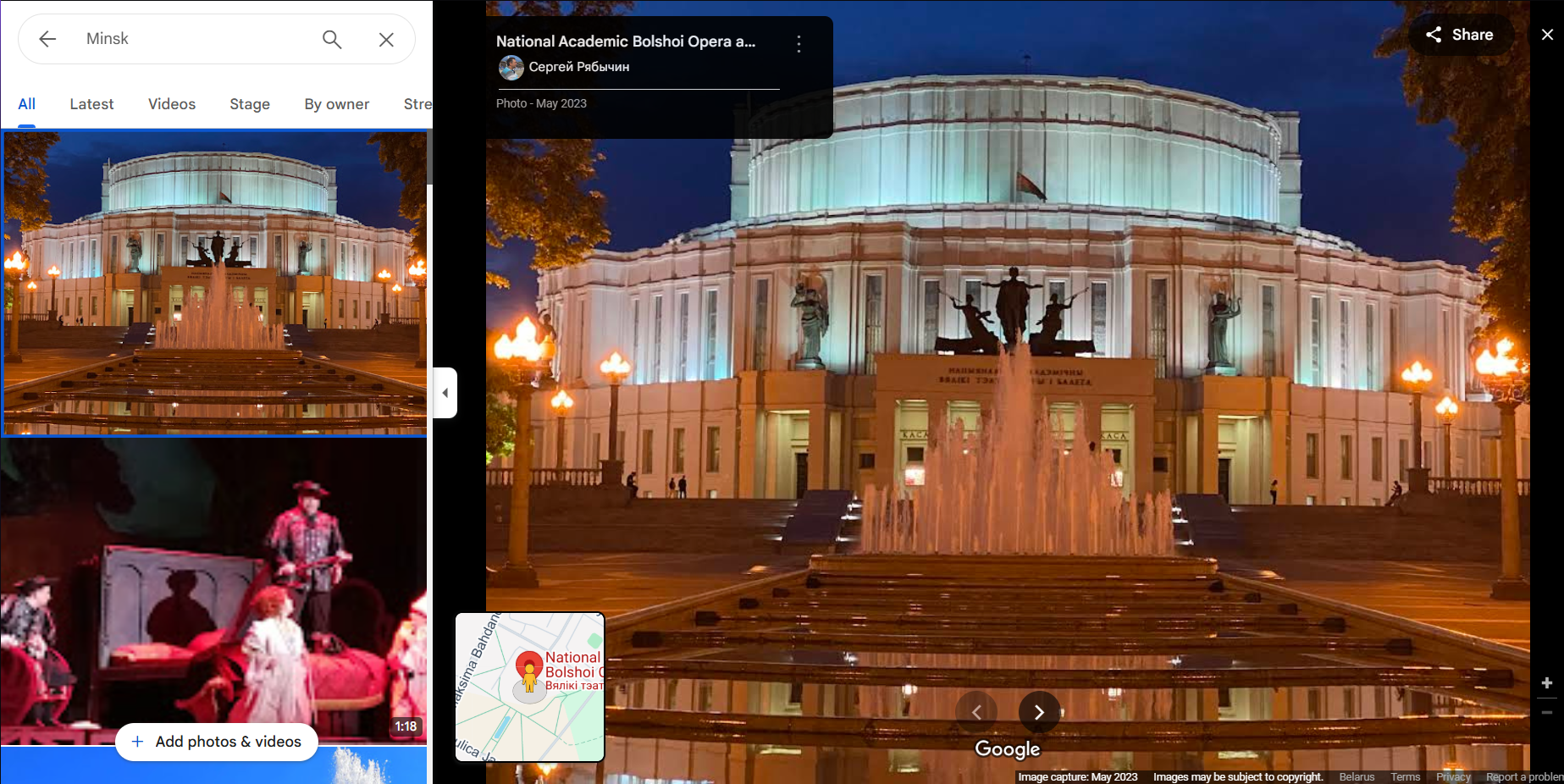


Рис. 1.2 – Дополнительные возможности веб-ресурса Google Maps

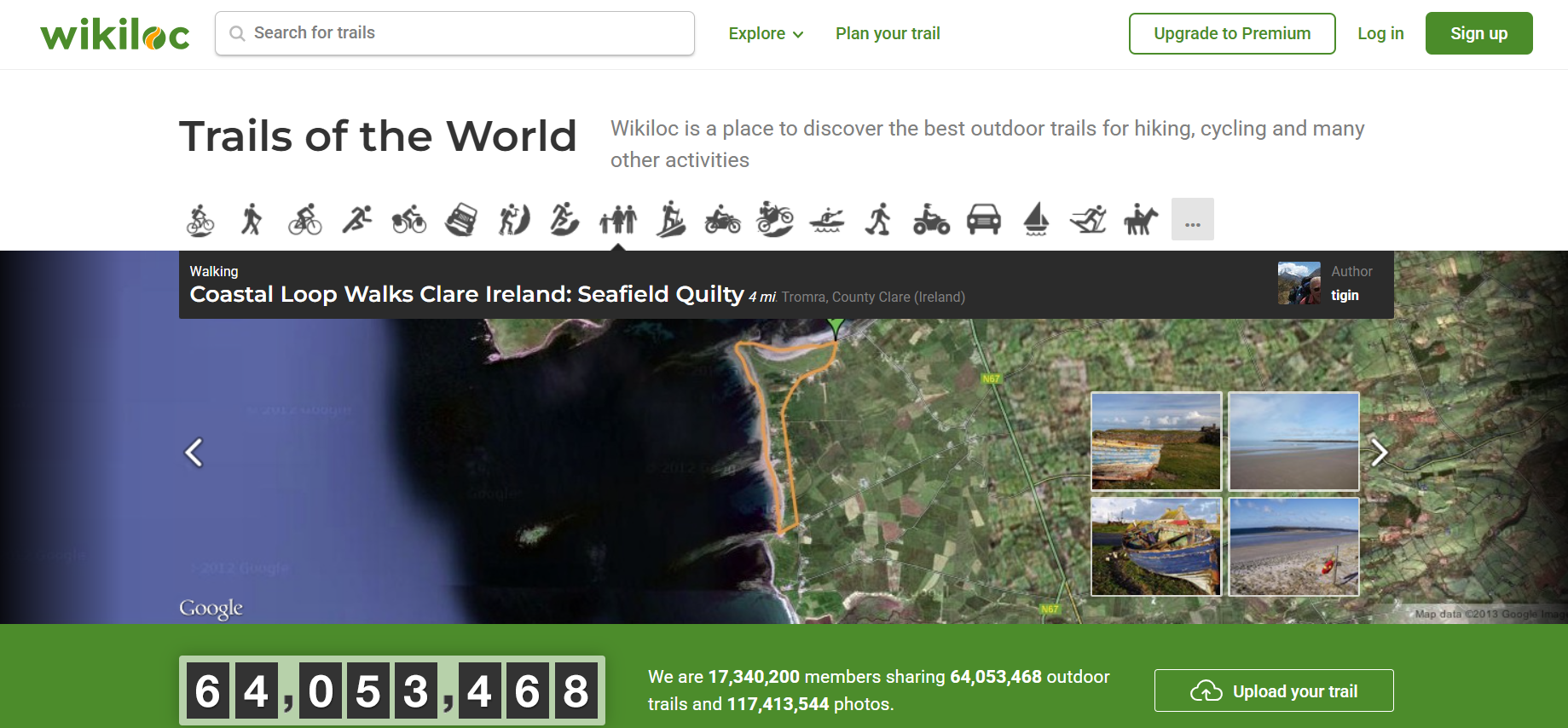


Рис. 1.3 – Информация о веб-ресурсе Wikiloc и его интерфейс

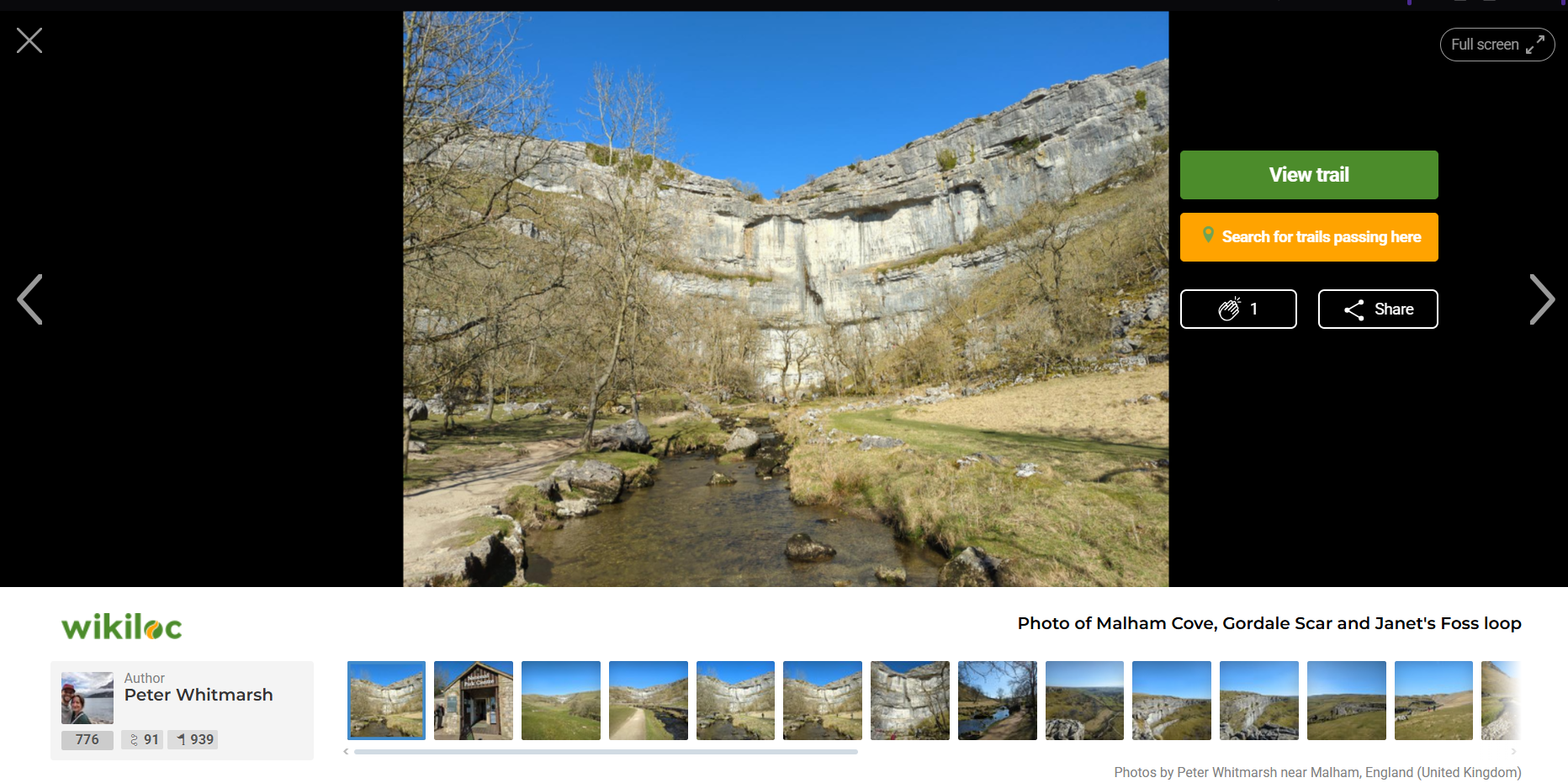


Рис. 1.4 – Дополнительные возможности веб-ресурса Wikiloc

На основе анализа аналогичных веб-ресурсов были выявлены основные требования к функциональному назначению.

### **1.3 Функциональное назначение**

Функциональное назначение данного проекта представлено следующими возможностями веб-ресурса «BeloGo»:

* просмотр фотографий, опубликованных другими пользователями;
* создание своих постов;
* \*сохранение понравившихся постов в избранное:
* добавление меток на карту;
* просмотр карты с метками;
* возможность настройки профиля;

## **2. ГЛАВА 2. ПРОЕКТИРОВАНИЕ ЗАДАЧИ**

### **2.1 Выбор и обоснование инструментов разработки**

Bootstrap — популярный фреймворк для создания сайтов и веб-приложений с открытым исходным кодом. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Bootstrap использует современные наработки в области CSS, JavaScript и HTML, поэтому использование данного инструмента не всегда может быть правильным решением при использовании старых браузеров.

Leaflet — библиотека с открытым исходным кодом, написанная на JavaScript, предназначенная для отображения карт на веб-сайтах. Поддерживает большинство мобильных и стационарных платформ из числа тех, что поддерживают HTML5 и CSS3.

Leaflet позволяет разработчику, не знакомому с ГИС, легко отображать растровые карты, состоящие из маленьких фрагментов — тайлов, с, возможно, дополнительными слоями, накладываемыми поверх основного. Слои могут быть интерактивными, например, отображать подсказку при клике по маркеру.

Leaflet позволяет создавать интерактивные метки на карте, поэтому именно эта библиотека использовалась в научно-исследовательском проекте.

JavaScript – мультипарадигменный язык программирования. Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией спецификации ECMAScript.

Node.js — программная платформа, основанная на движке V8 (компилирующем JavaScript в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения. Node.js добавляет возможность JavaScript взаимодействовать с устройствами ввода-вывода через свой API

Express.js — это быстрый, непринужденный и минималистичный веб-фреймворк для Node.js, который облегчает создание веб-приложений и API.

Данный веб-фреймворк был выбран для реализации серверной части веб-ресурса. Благодаря его простоте и минимализму, знакомство с его возможностями и последующая разработка сервера не составила особого труда.

Для передачи файлов на сервер(фотографий пользователей) использовалось программное обеспечение Multer.

Multer – это промежуточное программное обеспечение слоя Express.js, которое помогает в обработке multipart/form-data форм, в основном используемых для загрузки файлов. Он упрощает процесс загрузки файлов, обеспечивая эффективную обработку файлов и управление ими.

Для сохранения данных о пользователе между его запросами к серверу использовался Express session.

Express session – это механизм в веб-приложении, основанном на фреймворке Express.js, который позволяет сохранять данные о пользователе между его запросами к серверу. Это позволяет веб-приложению сохранять информацию о пользователе и его действиях в течении определённого промежутка времени(сессии).

В данном проекте используются следующие модули языка программирования JavaScript:

* Bcrypt JS – библиотека, реализующая алгоритм bcrypt для надёжного хеширования паролей;
* Path – встроенный модуль Node.js для работу с путями к файлам и каталогам;
* Fs (File System) – встроенный в Node.js модуль, предоставляющий функциональность для работы с файловой системой операционной системы;
* Swiper JS – библиотека, используемая для создания слайдеров, каруселей и других интерактивных элементов на веб-страницах;
* jQuery – библиотека для удобного взаимодействия с DOM-элементами;
* SQLite3 JS – модуль, позволяющий использовать базу данных SQLite в Node.js;

Асинхронное программирование – концепция программирования, суть которой заключается в ожидании одной функцией ответа от какого-либо источника (например, сервера), и запуске выполнения другой функции, следующей в очереди. Таким образом нарушается стандартный порядок выполнения функций, однако такой подход позволяет ускорить работу программы, особенно при взаимодействии с удаленными источниками.

Visual Studio Code – автономный редактор исходного кода, работающий в Windows, macOS и Linux. Лучший выбор для многих разработчиков, поддерживающий многие языки программирования [4].

Данный редактор был выбран в качестве инструмента работы с кодом благодаря его низкому порогу входа и функциональности, а также сбалансированному минимализму (рисунок 2.1).

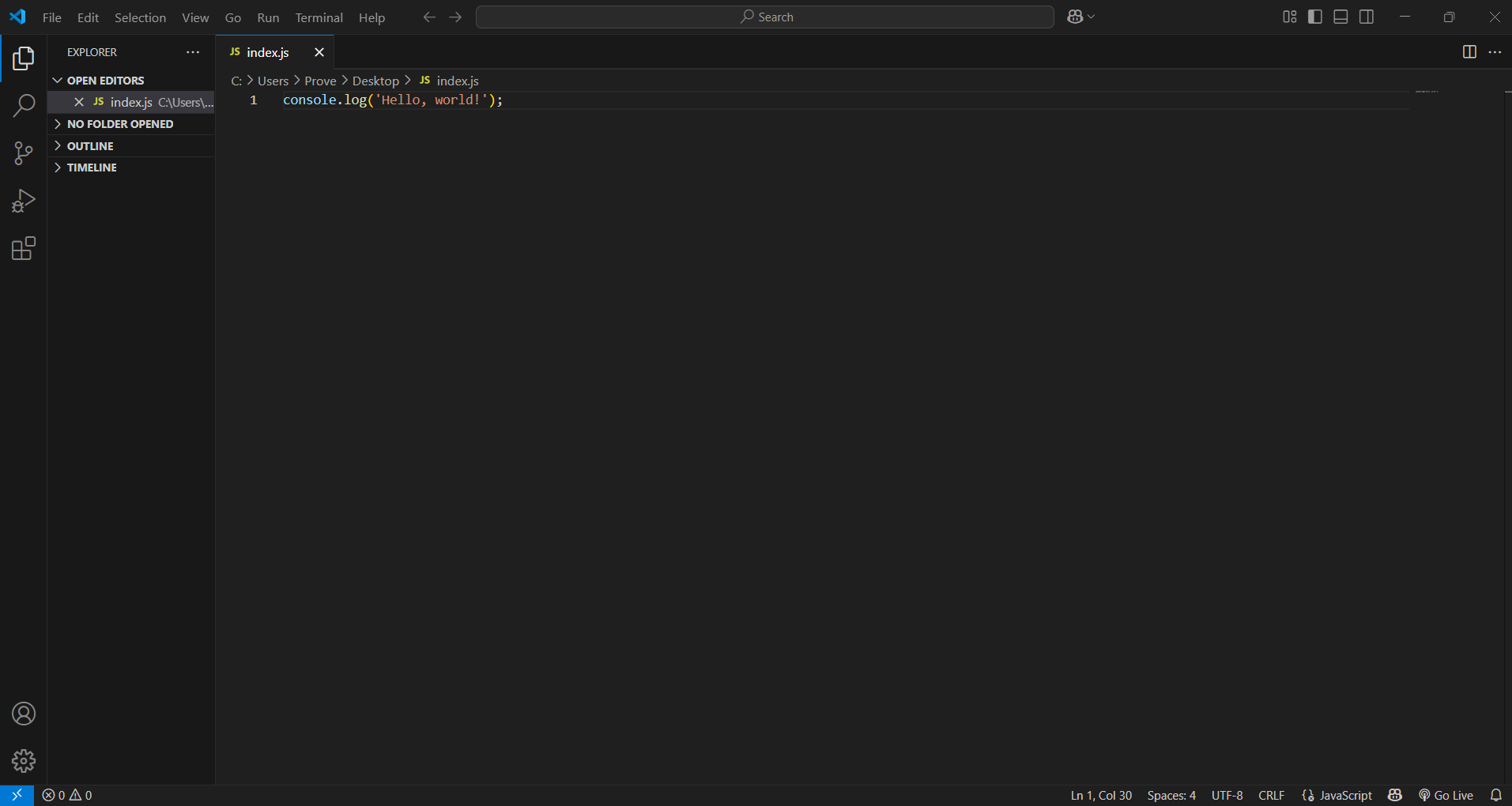


Рисунок 2.1 – Пример программы «Hello, world!» на языке программирования JavaScript в Visual Studio Code

Необходимо отметить, что очень важным преимуществом Visual Studio Code перед остальными инструментами работы с кодом является наличие большого количества расширений, в той, или иной степени упрощающих разработку. Visual Studio Code предлагает возможности для использования сразу нескольких языков программирования в одном проекте.

При разработке веб-ресурса «BeloGo» использовались следующие расширения:

* Russian Language Pack for Visual Studio Code – расширение, для перевода интерфейса Visual Studio Code на русский язык;
* JavaScript (ES6) code snippets – расширение, содержащее фрагменты кода для JavaScript в синтаксисе ES6, поддерживающее как JavaScript, так и TypeScript;
* Live Server – расширение для запуска локального сервера с функцией перезагрузки статических и динамических страниц;
* HTML and CSS Support – расширение, добавляющая поддержку в создании пользовательских интерфейсов, используя язык разметки HTML и язык стилей CSS;

SQLite3 – модуль, позволяющий использовать базу данных SQLite в JavaScript-средах, таких как Node.js. Он, предоставляет API для взаимодействия с SQLite, тем самым позволяя управлять базами данных на JavaScript, не используя Систему Управления Базами Данных напрямую и с использованием дополнительных возможностей языка. Такой подход позволяет при необходимости создавать, изменять и выполнять запросы к базе данных.

SQLiteStudio – браузер и редактор баз данных, использующих SQLite. При помощи данной программы можно без использования SQL-запросов создавать, удалять, просматривать и редактировать базы данных. Это очень удобно при разработке программного обеспечения (в контексте данного научно-исследовательского проекта – «BeloGo»), так как позволяет быстро получить необходимые данные в процессе тестирования.

# **ПРАКТИЧЕСКАЯ ЧАСТЬ**

# **3. ГЛАВА 3. ПРОГРАММНАЯ РЕАЛИЗАЦИЯ**

### **Реализация адаптивной веб-страницы**

ОПИСАТЬ как делали адаптив для веб страниц(медиазапросы, сетка бутстрап и тд)

### **3.2 Описание разработанных классов**

В рамках исследовательского проекта было разработано большое количество классов, поэтому в данном разделе будет их перечисление и краткое описание.

Классы, разработанные и использованные в «Absence Control», можно условно разделить на 4 группы, которые тесно взаимодействуют между собой.

Служебные классы – классы, которые необходимы для функционирования других частей программы. Таких класса в данном проекте можно выделить 4:

* Class Datetime\_Handler – класс, который содержит набор методов для работы с датой;
* Class Middleware – класс, который служит для описания внутреннего Middleware;
* Class Xlsx\_Writer – класс, который используется для создания файлов Microsoft Excel;
* Class Pdf\_Writer – класс, который используется для создания PDF файлов.

Классы состояний – классы, которые наследуются от класса aiogram.fsm.state.StatesGroup и хранят в себе несколько переменных-состояний (aiogram.fsm.state.State()). Таких классов в данном исследовательском проекте было разработано 10:

* Class Set\_Utc\_Offset – класс, который описывает состояние пользователя, когда он указывает смещение UTC;
* Class Send\_Feedback – класс, который описывает состояние пользователя, когда он отправляет отзыв о боте его владельцу;
* Class Create\_Group – класс, который описывает состояние пользователя, когда он создает группу;
* Class Add\_Members – класс, который описывает состояние пользователя, когда он добавляет участников в группу;
* Class Delete\_Group – класс, который описывает состояние пользователя, когда он удаляет группу;
* Class Remove\_Members – класс, который описывает состояние пользователя, когда он удаляет участников из группы;
* Class Assign\_Reports\_Recipient – класс, который описывает состояние пользователя, когда он назначает получателя отчетов для группы;
* Class Create\_Report – класс, который описывает состояние пользователя, когда он создает отчет об отсутствии участников группы;
* Class Get\_Statistics – класс, который описывает состояние пользователя, когда он получает статистику об отсутствии участников группы;
* Class Get\_Reports\_File – класс, который описывает состояние пользователя, когда он получает файл отчетов об отсутствии участников группы.

Классы моделей таблиц в базе данных – классы, которые предназначены для создания и описания таблиц в базе данных. При помощи данных классов можно настроить поля базы данных, их отношения, а также создавать и удалять таблицы. Кроме того, объекты данного класса представляют из себя запись базы данных. Эта запись может быть либо получена из базы данных и обработана в коде, либо создана в коде и добавлена в таблицу базы данных. У объектов таких классов можно получить свойства записи как атрибуты экземпляров классов. Всего таких классов в данном исследовательском проекте 4:

Class Base – базовый (родительский) класс для управления всеми остальными моделями (к примеру созданием). Все классы моделей должны быть унаследованы от него;

Class User – класс для описания модели таблицы с информацией о пользователях Telegram-бота – «users». Содержит в себе следующие поля:

* Id – первичный ключ целочисленного типа, который позволяет задать каждому пользователю уникальный номер (идентификатор) в базе данных;
* Tg\_id – поле типа BigInteger (особый тип SQLAlchemy, позволяющий хранить значения целочисленного типа, которые не помещаются в стандартную 8-байтную переменную), хранящее Telegram-id пользователя. Используется в отправке сообщений пользователю;
* Utc\_offset – поле, хранящее смещение UTC пользователя в секундах. Может принимать только целочисленные значения. Используется для корректного указания дат в отчетах / статистике / файлах отчетов относительно каждого пользователя «Absence Control»;
* Feedbacks\_cnt – поле, для хранения количества отправленных отзывов пользователем за текущий день. Необходимо для ограничения количества отправленных отзывов пользователем в размере 5 с целью предотвращения спама. Данное поле сбрасывается каждый день в 04:00 UTC.

Также в данном классе присутствуют отношения, связывающие его с другим классом – class Group.

Class Group – класс для описания модели таблицы с информацией о группах, созданных пользователями Telegram-бота – «groups». В данной таблице присутствуют следующие поля:

* Id – первичный ключ целочисленного типа, который позволяет задать каждой группе неповторимый номер (идентификатор) в базе данных;
* Creator – поле, которое является внешним ключом и ссылается на поле id в таблице «users». Оно указывает создателя (пользователя «Absence Control») каждой группы. Предназначение данного поля – создание различимости групп по создателю;
* Name – поле, которое содержит строковое значение имени группы. Служит для создания различимости групп по имени. Таким образом в совокупности с полем creator создает различимость групп по 2 полям – создателю и названию;
* Reports\_recipient – поле, которое является внешним ключом и ссылается на поле id в таблице «users». Оно указывает получателя отчетов (пользователя Telegram-бота) группы. Необходимо для отправки отчетов специально заданному пользователю;
* Members - строковое поле, содержащее в себе имена участников группы, разделенные сочетанием символов «;\n». Используется в работе с участниками групп.

Необходимо также отметить, что в данном классе присутствуют отношения, связывающие его с другими классами – class User и class Report.

Class Report – класс, описывающий модель таблицы, которая содержит в себе отчеты, созданные пользователями Telegram-бота – «reports». В данной таблице присутствуют следующие поля:

* Id – поле, содержащее уникальный целочисленный идентификатор отчета. Служит для идентификации каждого отчета отдельно;
* Group – поле, являющееся внешним ключом и ссылающееся на поле id таблицы «groups». Данное поле содержит целочисленный идентификатор (id) группы, для которой был создан отчет. Используется для работы с отчетами определенной группы;
* Date – поле, которое содержит значение даты, когда был создан отчет. Служит для получения отчетов, принадлежащих определенному промежутку времени, при создании статистики / получении файла отчетов;
* Members – строковое поле, которое содержит список имен участников определенного отчета, разделенных сочетанием символов «;\n». Необходимо для предоставления статистики / файла отчетов.

В данном классе, помимо всего прочего, присутствует отношение для связи с другим классом – class Group.

Классы запросов в базу данных – классы, каждый из которых отвечает за запросы в определенную таблицу базы данных. Всего таких классов 3:

* Class User\_Requests – класс, отвечающий за запросы в таблицу «users» базы данных. Данный класс содержит следующие методы;
* Class Group\_Requests – класс, отвечающий за запросы в таблицу «groups» базы данных;
* Class Report\_Requests – класс, отвечающий за запросы в таблицу «reports» базы данных.

Таким образом в данном исследовательском проекте было разработано 19 классов, которые позволяют структурировать код и разделить его на обособленные части, что заметно улучшает его качество.

### **3.2 Описание разработанных функций**

В данном проекте было разработаны множество различных функций. Их можно условно разделить на несколько групп.

Служебные – функции для корректной работы программы;

* Async def main() -> None – основная функция для запуска программы, которая запускается через asyncio.run();
* Async def create\_models() -> None – функция для создания базы данных и моделей таблиц в ней.

Создатели разметок – функции для создания разметок клавиатур по заданным параметрам:

* Async def create\_reply\_markup(data: list[str], add\_stop\_button: bool = False) -> ReplyKeyboardMarkup – функция для создания reply-разметки, каждая кнопка которой содержит текст из списка data. Также есть возможность указать добавлять ли в конец кнопку с текстом «Стоп»;
* Async def create\_request\_markup() -> ReplyKeyboardMarkup – функция для создания reply-разметки, которая содержит одну кнопку, для выбора Telegram-пользователя;
* Async def create\_report\_markup(members: list[str]) -> InlineKeyboardMarkup – функция, которая по списку участников отчета, создает для него inline-разметку с кнопкой с текстом «Отправить» и настроенными данными для обратного вызова.

Обработчики – функции, которые обрабатывают отправку сообщений пользователями. Данные функции способны обрабатывать как определенные команды, так и сообщения пользователя в определенном состоянии.

Команды в данном боте можно условно разделить на 3 группы, таким же образом можно разделить и функции, которые обрабатывают эти команды:

* Служебные обработчики – обработка команд «/start», «/help», «/cancel», «/setutcoffset», «/feedback» и их состояний;
* Конфигурационные обработчики - обработка команд «/creategroup», «/addmembers», «/deletegroup», «/deletemembers», «/assignreportsrecipient» и их состояний;
* Обработчики работы – обработка команд «/createreport», «/getstatistics», «/getreportsfile» и их состояний.

С помощью данного набора функций полностью реализуются требования к функциональному назначению.

### **4. ГЛАВА 4. ТЕСТИРОВАНИЕ**

Тестирование – процесс поиска и исправления ошибок в программном продукте. Наиболее интересными методами тестирования являются:

* функциональное тестирование – тестирование программного продукта на выполнение определенных функций, которые необходимы заказчику
* регрессионное тестирование – тестирование ранее исправленных ошибок и основного функционала. Суть данного тестирования заключается в проверке того, что исправление каких-либо ошибок не влечет за собой появление других
* тестирование белого ящика – тестирование программного продукта, при котором тестировщик имеет доступ к исходному коду продукта и может добавлять свои надстройки во время тестирования
* тестирование черного ящика – тестирование программного продукта, при котором тестировщик не имеет доступа к исходному коду продукта. То есть тестировщик при данном методе тестирования ставится на роль обычного пользователя

На разных этапах разработки программного обеспечения могут изменяться способы тестирования, однако на последних этапах наиболее правильным является тестирование черного ящика. Так как при таком раскладе тестировщик может быть полностью на месте пользователя, так как весь функционал уже написан, и большая часть основных ошибок исправлена. При нахождении некоторых неисправностей в работе программы тестировщик может, как бы в роли пользователя, отправить отзыв и данная ошибка будет не исправлена, при этом он не будет вмешиваться в внутреннюю работу кода.

Далее представлена таблица тестирования «Absence Control» методом черного ящика.

Таблица 4.1 Тестирование «Absence Control»

|  |  |  |  |
| --- | --- | --- | --- |
| Название операции | Входной параметр | Ожидаемый результат | Фактический результат |
| 1 | 2 | 3 | 4 |
| Начало диалога с ботом | Отправка команды «/start» | Бот ответит сообщением-приветствием | Бот отвечает сообщением приветствием |
| Получение справочной информации от бота | Отправка команды «/help» | Бот выдаст сообщение со справочной информацией | Бот выдает сообщение со справочной информацией |
| Отмена выполнения текущей команды | Отправка команды «/cancel» | Бот прервет выполнение текущей команды | Бот прерывает выполнение текущей команды |

Продолжение таблицы 4.1

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| Указание смещения UTC | Отправка команды «/setutcoffset» | Бот запросит смещение UTC и установит его | Бот запрашивает смещение UTC и устанавливает его |
| Отправка отзыва | Отправка команды «/feedback» | Бот запросит отзыв и отправит его владельцу | Бот запрашивает отзыв и отправляет его владельцу |
| Создание группы | Отправка команды «/creategroup» | Бот запросит название группы и создаст ее. | Бот запрашивает название группы и создает ее. |
| Добавление участников в группу | Отправка команды «/addmembers» | Бот запросит название группы, выдаст reply-клавиатуру со списком групп, запросит участников и добавит их в группу. | Бот запрашивает название группы, выдает reply-клавиатуру со списком групп, запрашивает участников и добавляет их в группу. |
| Удаление группы | Отправка команды «/deletegroup» | Бот запросит название группы, выдаст reply-клавиатуру со списком групп, запросит подтверждение, выдаст reply-клавиатуру для подтверждения и удалит группу. | Бот запрашивает название группы, выдает reply-клавиатуру со списком групп, запрашивает подтверждение, выдаст reply-клавиатуру для подтверждения и удаляет группу. |
| Удаление участников из группы | Отправка команды «/removemembers» | Бот запросит название группы, выдаст reply-клавиатуру со списком групп, запросит участников, выдаст reply-разметку со списков участников и удалит их из группы. | Бот запрашивает название группы, выдает reply-клавиатуру со списком групп, запрашивает участников, выдает reply-разметку со списков участников и удалит их из группы. |

По итогам проведенного тестирования в исходном коде «Absence Control» отсутствуют программные ошибки, и он полностью готов к работе.

## **5. ГЛАВА 5. ПРИМЕНЕНИЕ**

### **5. 1 Назначение и условия применения**

Для использования данного программного продукта необходимо иметь на устройстве установленное приложение Telegram. Данного бота можно найти по имени «Absence Control» и свободно им пользоваться каждый день, но в рамках времени с 7:00 до 10:00. Данное временное ограничение связано с хостингом данного бота на сервере PythonAnywhere через поллинг – принцип работы Telegram-бота, когда он посылает серверам запросы. При таком способе работы бота он будет со временем выключатся примерно через 3 часа после запуска.

Также есть возможность запустить аналог данного бота, используя свой личный токен, который можно получить в Telegram-боте BotFather. Данный бот не является высоконагруженным и сможет запуститься и работать даже на средних по мощности устрой­­­ствах.

## 5.2 Руководство пользователя

При первом запуске приложения пользователю необходимо зарегистрироваться (рисунок 5.1).



Рисунок 5.1 – Регистрация пользователя

После этого пользователь попадёт на экран входа, где ему необходимо ввести логин и пароль, указанные при регистрации (рисунок 5.2).



Рисунок 5.2 – Вход пользователя

После входа пользователь попадёт на главный экран (рисунок 5.3).

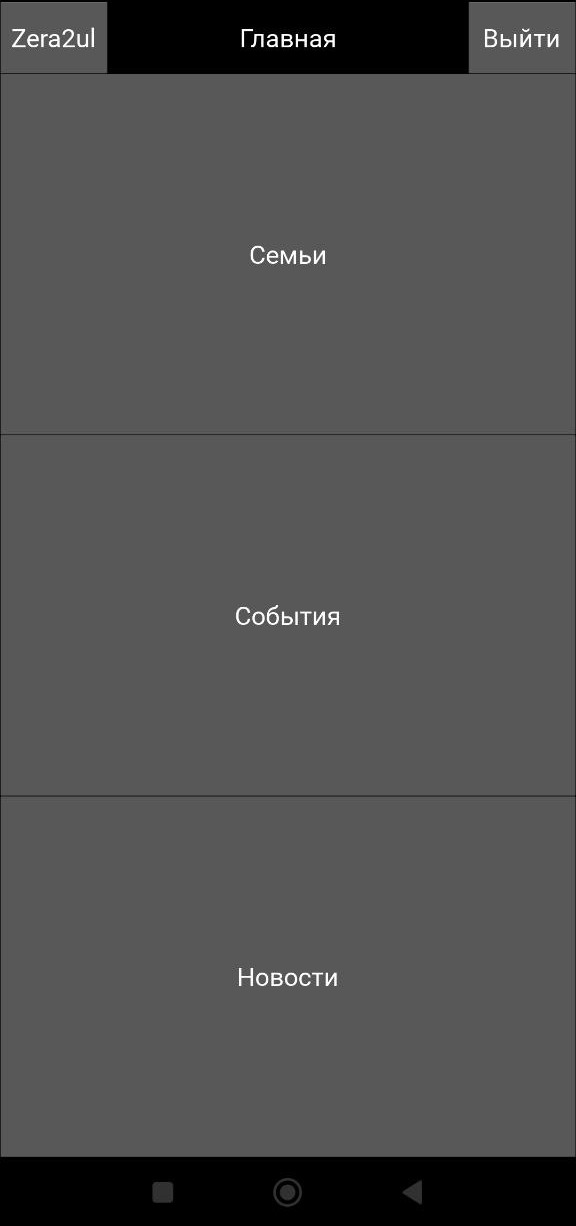


Рисунок 5.3 – Главный экран

При нажатии на кнопку, содержащую логин, пользователь попадёт на экран редактирования персональной информации (рисунок 5.4), а при нажатии на кнопку «Выход», пользователь вернётся к экрану входа.



Рисунок 5.4 – Редактирование персональных данных

Используя кнопку «На главную», пользователь вернётся на главный экран.

Нажатие на кнопку «Семья» приведёт пользователя к экрану с функциями управления семьями (рисунок 5.5).

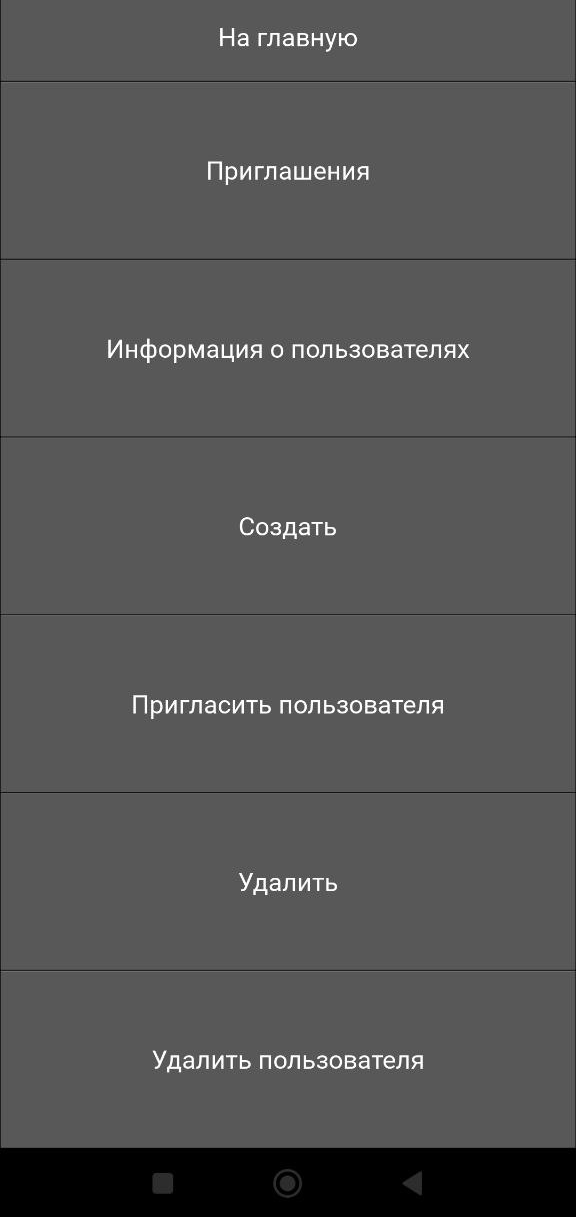


Рисунок 5.5 – Функции управления семьями

Нажатие кнопки «Приглашения» позволит пользователю просмотреть и обработать приглашения в семьи (рисунок 5.6).

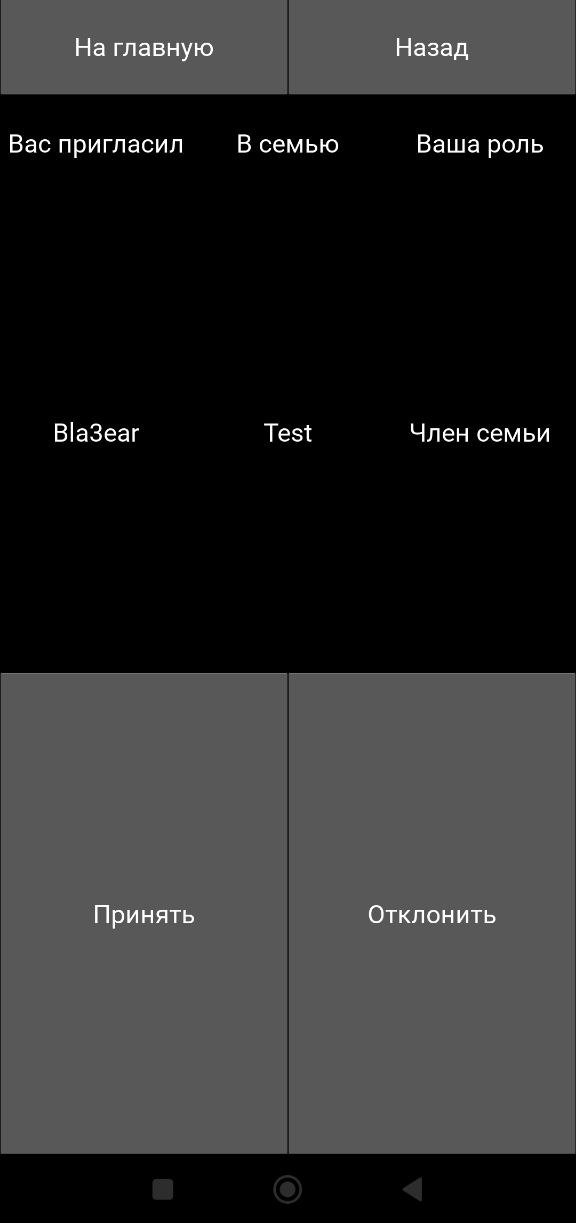


Рисунок 5.6 – Экран обработки приглашений

Используя кнопку «Назад», пользователь сможет вернутся на предыдущий экран.

Кнопка «Информация о пользователях» позволит увидеть данные других пользователей в семье (рисунок 5.7).



Рисунок 5.7 – Получение информации о пользователях в семье

Нажатие кнопки «Создать», позволит пользователю создать семью (рисунок 5.8).

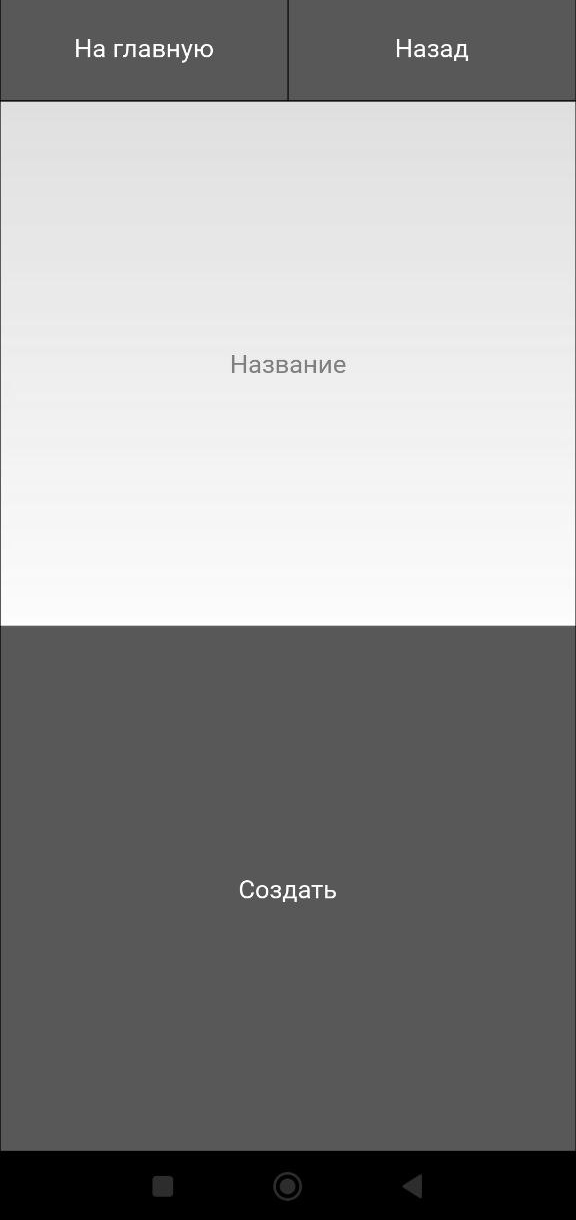


Рисунок 5.8 – Создание семьи

Нажатие кнопки «Пригласить пользователя» позволит пригласить в семью других пользователей (рисунок 5.9).

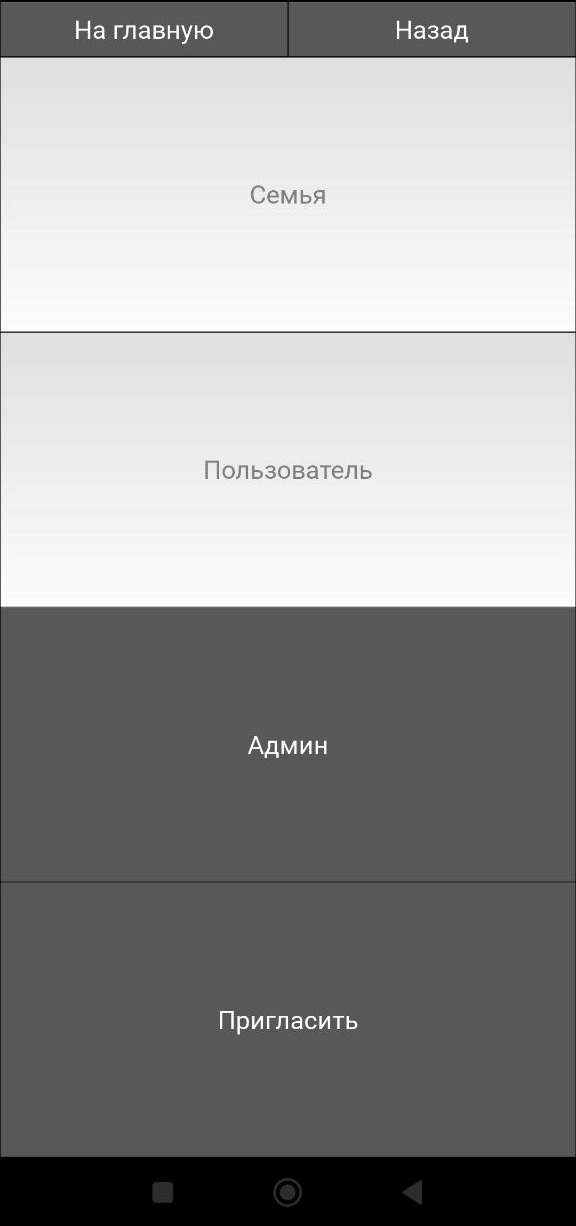


Рисунок 5.9 – Приглашение пользователя в семью

Кнопка «Удалить» приведёт пользователя к экрану удаления семьи (рисунок 5.10).

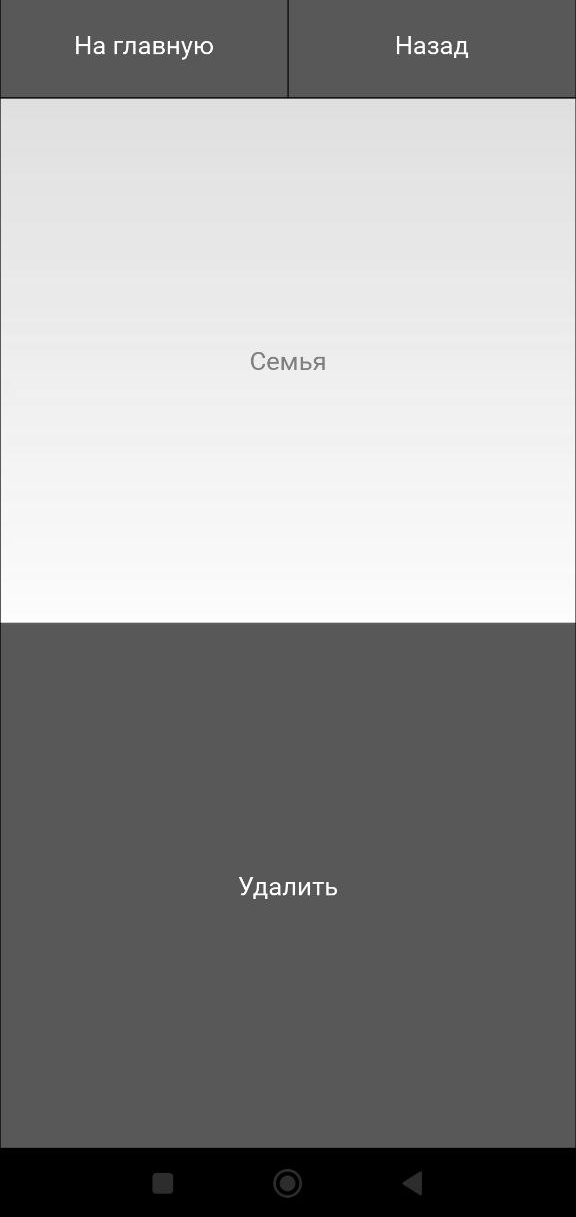


Рисунок 5.10 – Удаление семьи

Кнопка «Удалить пользователя» позволит удалить из семьи других пользователей (рисунок 5.11).

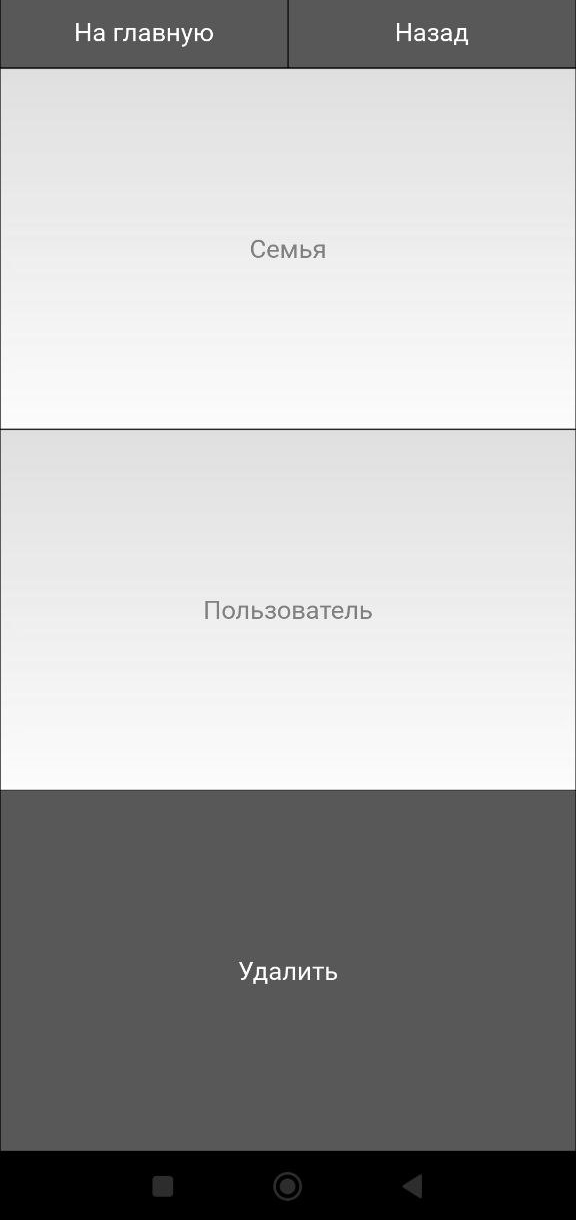


Рисунок 5.11 – Удаление пользователя из семьи

Нажатие кнопки «События» позволит пользователю перейти к функциям событий (рисунок 5.12).

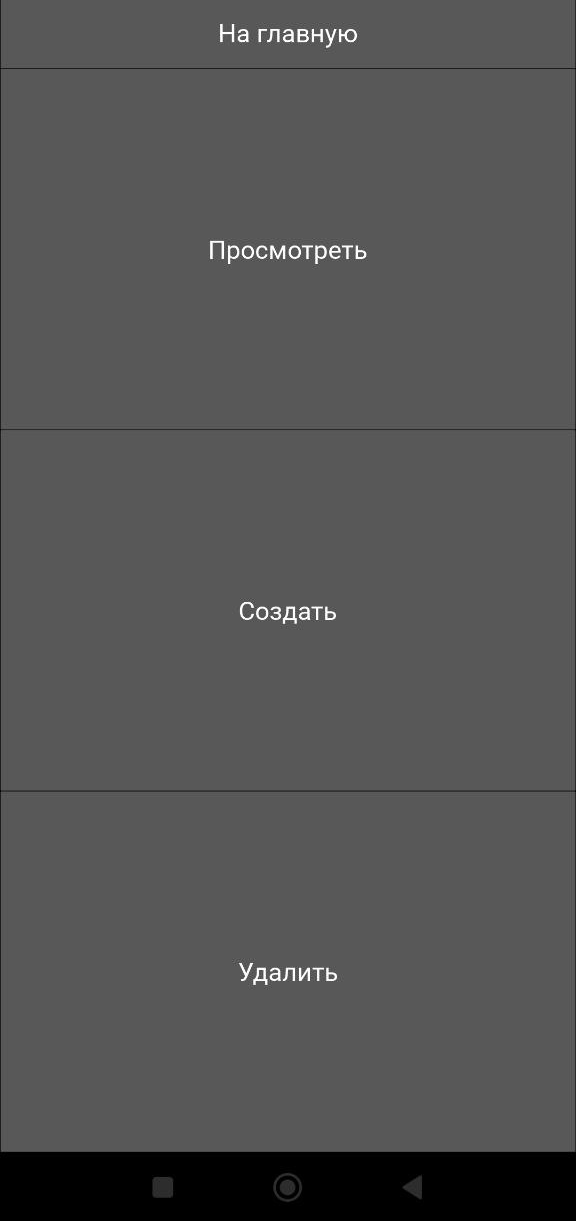


Рисунок 5.12 – Функции событий

Кнопки «Просмотреть», «Создать» и «Удалить» позволят выполнить соответствующие действия с событиями в семьях, в которых состоит пользователь (рисунки 5.13 – 5.15).



Рисунок 5.13 – Просмотр событий



Рисунок 5.14 – Создание события

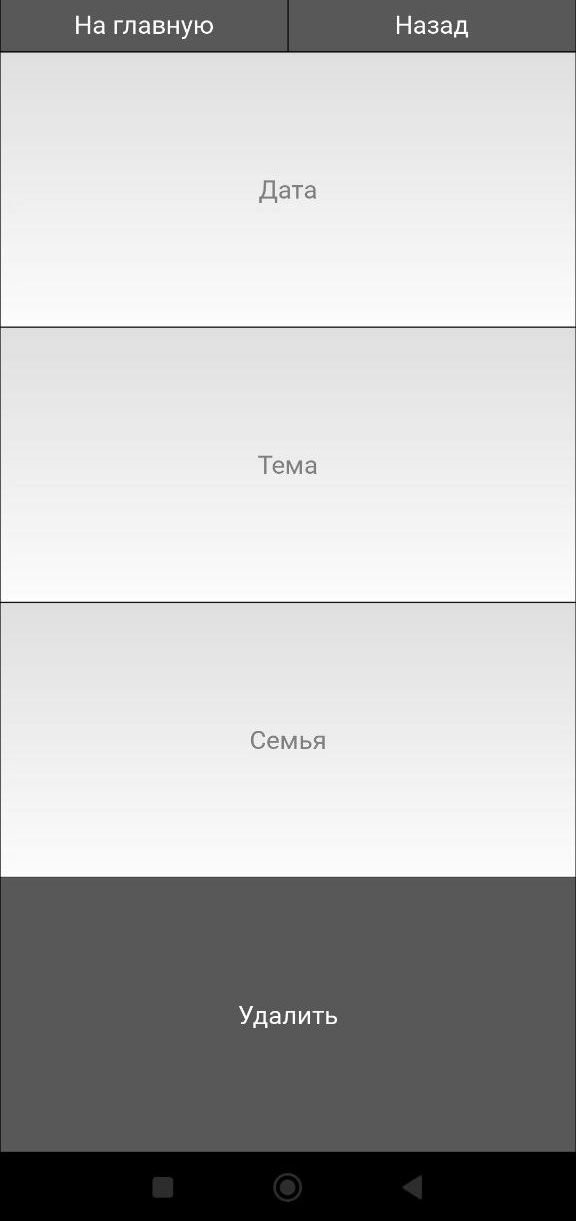


Рисунок 5.15 – Удаление события

Нажатие кнопки «Новости» позволит получить информацию о недавних новостях, связанных с изменением законодательства о браке и семье, при нажатии на кнопки с содержимым которых пользователь будет перенаправлен на веб-страницу с новостью.

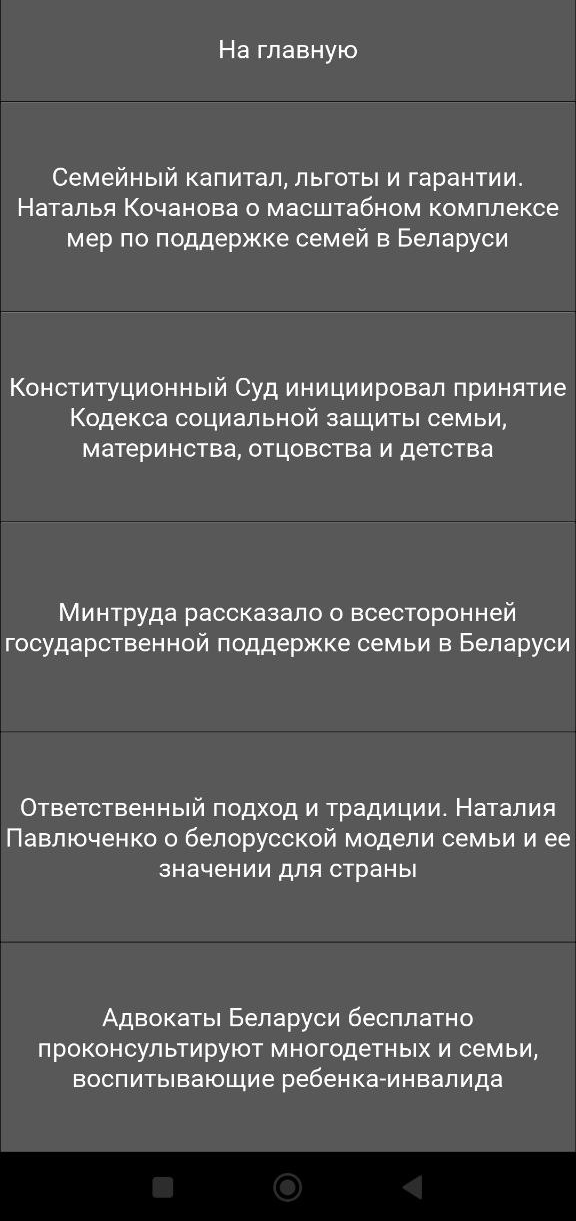


Рисунок 5.16 – Просмотр новостей

# **ЗАКЛЮЧЕНИЕ**

Подводя итоги разработки данного исследовательского проекта можно утверждать, что была достигнута его цель и были выполнены все поставленные задачи, а именно:

* были изучены базовые принципы работы с технологией Git;
* были исследованы основные принципы асинхронного программирования;
* был изучен асинхронный фреймворк Aiogram;
* были изучены базовые принципы работы с ORM SQLAlchemy;
* был создан Telegram-бот, который поможет упростить повседневные задачи;
* Telegram-бот был размещен на удаленном сервере PythonAnyWhere;
* создана пояснительная записка к исследовательскому проекту.

Также были выявлены возможные улучшения данного исследовательского проекта:

* получение файла отчетов об отсутствии участников групп других структур – например таблица, где ряды – имена участников, а столбцы – даты, когда создавались отчеты, на пересечении указывается отсутствовал ли участник в данный промежуток времени;
* добавление объединения групп – сообществ;
* хостинг Telegram-бота через вебхук – принцип работы Telegram-бота, при котором он посылает запросы серверам.

Также в ходе исследования был изучен его предмет и выявлено, что эффективность данного Telegram-бота в упрощении задачи контроля отсутствия участников групп довольно высока. Гипотеза была доказана.

Таким образом данный проект, хоть и был успешно завершен на данной стадии развития, но все же имеет перспективы в будущем.

# **СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Определение Git [Электронный ресурс]. - Режим доступа - https://www.atlassian.com/ru/git/tutorials/what-is-git. - Дата доступа – 18.09.2024.
2. Определение GitHub [Электронный ресурс]. - Режим доступа - https://education.yandex.ru/journal/chto-takoe-github. - Дата доступа – 18.09.2024.
3. Определение Python [Электронный ресурс]. – Режим доступа - https://www.nic.ru/help/osnovy-yazyka-programmirovaniya-python\_11662.html. – Дата доступа – 18.09.2024.
4. Определение Visual Studio Code [Электронный ресурс]. – Режим доступа - https://visualstudio.microsoft.com/ru/. – Дата доступа – 18.09.2024.
5. Определение Aiogram [Электронный ресурс]. – Режим доступа - https://lytkins.ru/index.php?option=com\_content&view=article&id=70&catid=8. Дата доступа – 18.09.2024.

# **ПРИЛОЖЕНИЯ**

## 

## **ПРИЛОЖЕНИЕ 1. ИСХОДНЫЙ КОД ПРОЕКТА**

Ссылка на GitHub репозиторий проекта - https://github.com/zera2ul/Absence\_Control.