

Научная статья

УДК 004.942+004.584+528.9

ББК 32.973+32.972+26.190

М 52

DOI: 10.53598/2410-3225-2024-2-341-73-77

**Система индикации приближения пользователя к заранее  
выбранному географическому объекту при движении по маршруту**  
(Рецензирована)

**Анастасия Владимировна Меркулова<sup>1</sup>, Индар Мадинович Напсо<sup>2</sup>**

<sup>1-2</sup> Адыгейский государственный университет, Майкоп, Россия

<sup>1</sup> [napso@adygnet.ru](mailto:napso@adygnet.ru)

<sup>2</sup> [anastasia.merkulova.work@gmail.com](mailto:anastasia.merkulova.work@gmail.com)

**Аннотация.** Разработан сервис, который определяет индикацию приближения пользователя к предварительно выбранной локации. Используя Telegram Bot API для определения текущих координат пользователя в реальном времени, система оповещает о приближении к географическому объекту. С помощью технологий геокодирования и Overpass API удается находить конечную точку маршрута. Полученные результаты в работе с API OpenStreetMap и другими геолокационными технологиями могут быть применены в различных областях, включая транспорт, логистику, туризм и многие другие, где навигация играет важную роль.

**Ключевые слова:** прикладное программирование, прикладное программное обеспечение, индикация приближения, навигация, карты, маршрут, Telegram Bot, OpenStreetMap, геокодирование

**Для цитирования:** Меркулова А. В., Напсо И. М. Система индикации приближения пользователя к заранее выбранному географическому объекту при движении по маршруту // Вестник Адыгейского государственного университета. Сер. : Естественно-математические и технические науки. 2024. Вып. 2 (341). С. 73–77. DOI: 10.53598/2410-3225-2024-2-341-73-77

Original Research Paper

**System for indicating the user's proximity to a pre-selected  
geographical object when traveling along a route**

**Anastasiya V. Merkulova<sup>1</sup>, Indar M. Napso<sup>2</sup>**

<sup>1-2</sup> Adyghe State University, Maykop, Russia

<sup>1</sup> [napso@adygnet.ru](mailto:napso@adygnet.ru)

<sup>2</sup> [anastasia.merkulova.work@gmail.com](mailto:anastasia.merkulova.work@gmail.com)

**Abstract.** A service that determines the indication of the user's approach to a pre-selected location is developed. Using the Telegram Bot API to determine the current coordinates of the user in real time, the system notifies about approaching a geographical object. Using geocoding technologies and Overpass API, it is possible to find the end point of the route. The results obtained in the work with the OpenStreetMap API and other geolocation technologies can be applied in once-personal areas, including transport, logistics, tourism and many others, where navigation plays an important role.

**Keywords:** application programming, application software, proximity indication, navigation, maps, route, Telegram Bot, OpenStreetMap, geocoding

**For citation:** Merkulova A. V., Napso I. M. System for indicating the user's proximity to a pre-selected geographical object when traveling along a route // The Bulletin of the Adyghe State University. Ser.: Natural-Mathematical and Technical Sciences. 2024. Iss. 2 (341). P. 73–77. DOI: 10.53598/2410-3225-2024-2-341-73-77

Для пешего туризма и экскурсий важна возможность свободно ориентироваться на местности и оперативно получать информацию о близлежащих объектах, выявлять наиболее интересные и значительные для человека объекты в процессе перемещения, такие, как заправки, парковки, места питания и другое. А также дополнять привычные карты новыми и интересными функциями, возможностями и опциями, например, такими, как геомедийная реклама.

Исходя из всего вышеперечисленного, была поставлена цель – разработать систему индикации приближения пользователя к необходимой локации, и при этом система должна быть интуитивно понятна и удобна. Помимо основной задачи – исследовать инновационные технологии, применяемые в сфере картографии и навигации, были поставлены следующие задачи:

– Определение текущих координат пользователя. Необходимо обеспечить их работу в режиме реального времени, чтобы пользователь мог получать актуальные данные о своем местоположении относительно выбранного объекта. Определение в режиме реального времени текущих координат пользователя, чтобы пользователь мог получать актуальные данные о своем местоположении относительно выбранного объекта;

– Разработка сервис оповещения, которая будет информировать пользователя о приближении к географическому объекту при помощи уведомлений, сообщений или других сигналов;

– Определение конечной точки пользователя из двух возможных вариантов: конкретная точка, к которой пользователь хочет пойти, или ближайший к его местоположению объект.

В настоящее время существует большое количество разнообразных сервисов, обрабатывающих геоданные. Одним из наиболее популярных является OpenStreetMap. OpenStreetMap (дословно «открытая карта улиц»), сокращенно OSM – некоммерческий веб-картографический проект по созданию подробной свободной и бесплатной географической карты мира силами сообщества участников – пользователей Интернета [1]. Данная работа связана с использованием Overpass API, который, в отличие от основного API, оптимизирован для потребителей данных, которым требуется просмотреть сразу большое количество элементов за несколько секунд по каким-либо критериям поиска, например, местоположение, тип объектов, свойства, близость или их комбинации. Также использован сервис GeoPy [2] – это библиотека Python с открытым исходным кодом, специализирующаяся на добавлении местоположения к данным с помощью сервисов геокодирования, таких как Google Maps, OpenStreetMap и другие. Задачи определения текущего местоположения пользователя и вывод оповещений о приближении пользователя к географическому объекту при помощи сообщений будет решать Telegram Bot API [3].

В OpenStreetMap [4] каждый объект на карте имеет свои критерии. Первый используемый – это вид объекта: точка (node), линия (way) и полигон (area). На данном этапе ограничимся точкой. Второй и самый важный критерий – это категория. Они разделяются по характеристикам: это могут быть дороги, преграды, жилые зоны, здания, покрытие дороги, инфраструктура, энергетика и другое. Исходя из того, что необходимо населению, понимаем, что это общественные места – кафе, школы, банки, аптеки, туалеты, телефоны, автостоянки и другое. Общественные места и инфраструктура носят название amenity. Из всех существующих мест в категории amenity были отобраны 50 наиболее важных значений в сфере общественного питания, образования, транспорта, финансов, здоровья, развлечений, а также и некоторые другие, не относящиеся ни к каким категориям, например, рынок, магазин одежды / обуви и другое.

Взаимодействие с Telegram Bot будет происходить по следующей схеме 1:



Схема 1. Логика построения запросов  
Scheme 1. Query construction logic

После старта бот просит пользователя поделиться своей live-геолокацией. Если live-location получена, то на выбор пользователю предлагается 2 кнопки – найти ближайшее какое-либо место либо ввести конкретный адрес. Если ему нужен адрес, он вводит адрес. Если же он хочет получить ближайшее место, то ему выпадает список отобранных amenuty, он выбирает нужную категорию и желаемую точку. По итогу в обоих случаях выводится карта конечной точки и сообщение о том, насколько пользователь от нее далеко.

Архитектура системы выполнена в таком паттерне проектирования как слоистая (схема 2). Имеется главное приложение и его конфигурацию, которые лежат в пакете cmd, 3 слоя и 3 дополнительные библиотеки. Main – это входная точка приложения, Config является своего рода пучком, который собирает воедино все части приложения и предоставляет единственную собирательную функцию для main. В Config находится YAML [5] файл, который позволяет изменять категории, изменять токен бота, менять настройки Redis, не рассматривая содержимое кода. YAML парсит yaml документ и преобразует в Python-объект, а DataClass\_Factory [6] помогает распарсить объект YAML в необходимую DataClass конфигурации (OSM, Redis [7], Telegram Bot).

Слой Controllers отвечает за пользовательские запросы. Он получает запрос и выбирает, какую часть из сервисов ему нужно вызвать или использовать. Слой контроллера представлен в виде Телеграм-бота, который позволяет обрабатывать live-location пользователя, показывать ему все категории, возвращать список всех заведений в определенном радиусе и посчитать расстояние до него от текущего местоположения.

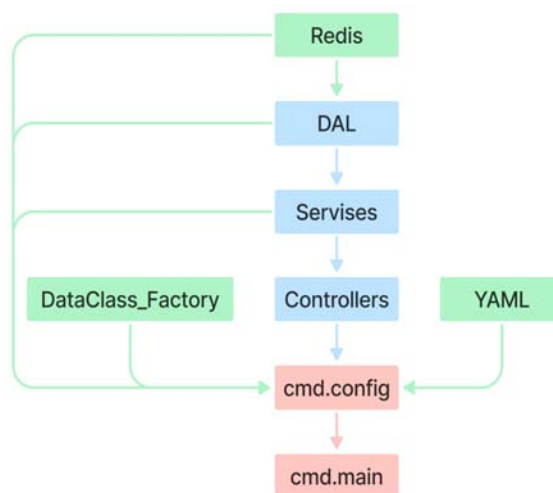


Схема 2. Архитектура системы индикации приближения  
Scheme 2. Architecture of the proximity indication system

Servises – это слой бизнес-логики. Своего рода посредник. Он получает какой-либо запрос от слоя контроллера, обрабатывает его и делает CRUD-запрос в DAL, и возвращает ответ в контроль. Этот слой представлен сервисом OSMService, который осуществляет поиск по интерфейсному / абстрактному Geo-сервису, который в свою

очередь помогает получать все точки на карте по категории, выбранной пользователем.

DAL (Data Access Layer) – это слой работы с данными. Он ничего не знает о бизнес-логике, он выполняет стандартные CRUD операции. CRUD – это 4 основные функции обработки данных – Create, Read, Update и Delete. Слой DAL представлен в виде двух классов: первый – OSMBroker, который делает запрос в OverPass API и возвращает точки – node; второй – Redis-Helper, который позволяет хранить локацию пользователя и точку, к которой он движется, относящийся к определенному чату (пользователю).

В результате проведенной работы реализован Telegram Bot, который не только строит маршрут, но и уведомляет о приближении пользователя к выбранному им географическому объекту. Интерфейсная реализация бота представлена на рисунках 1 и 2.

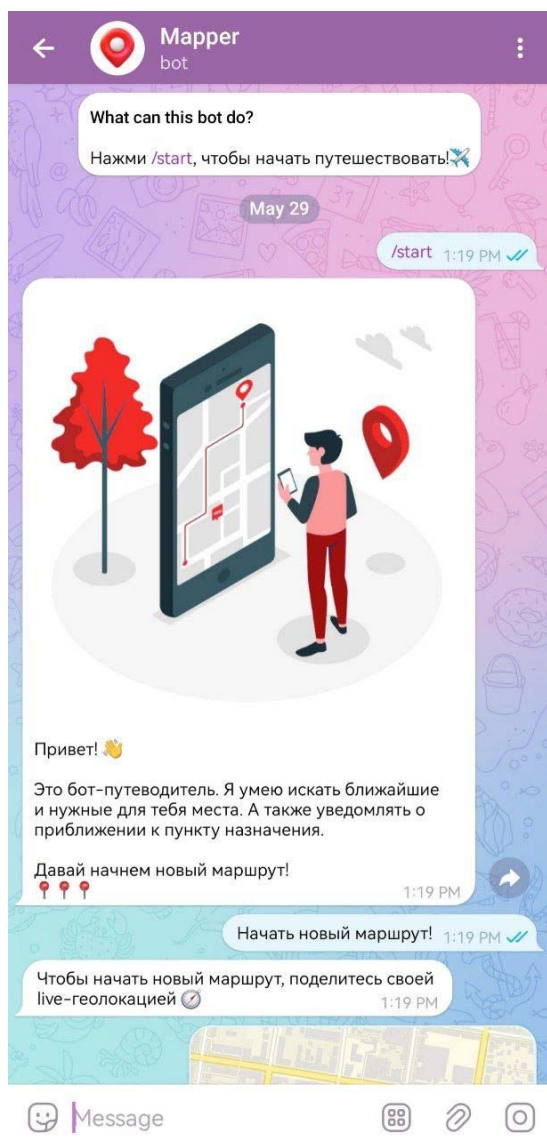


Рис. 1. Команда / start и приветствие Telegram Bot

Fig. 1. Telegram Bot command / start and welcome message

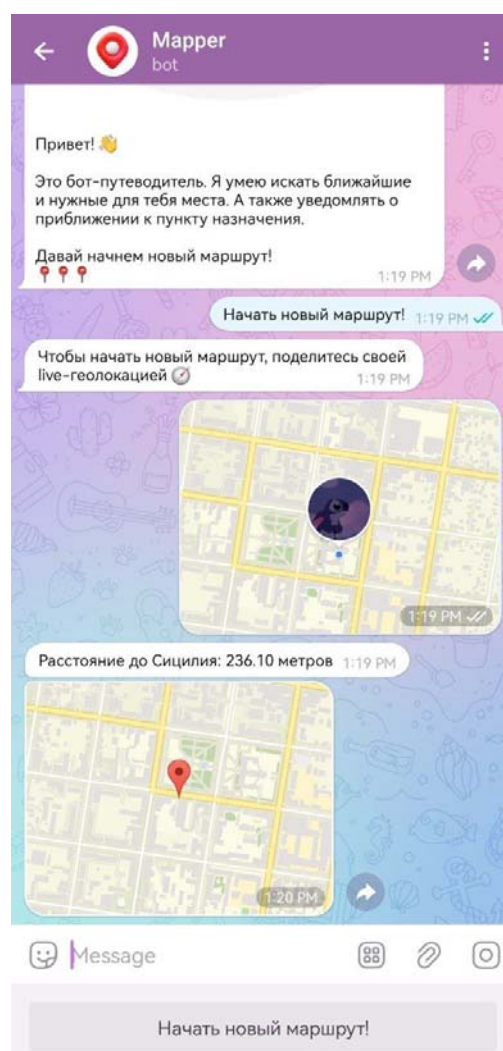


Рис. 2. Уведомление геолокации и сообщение о приближении к выбранному месту

Fig. 2. Share geolocation and notification of proximity to the selected location

Безусловно, способы обработки геоданных и результаты работы с ними могут меняться в зависимости от конечной цели. Полученные в ходе работы с API OpenStreetMap и другими геолокационными технологиями данные могут быть применены в

широком диапазоне областей, включая транспорт, логистику, туризм и другие, где навигация является критически важной составной частью предоставляемого сервиса.

### Примечания

1. Overpass API // Wiki : сайт. 2024.  
URL: [https://wiki.openstreetmap.org/wiki/RU:Overpass\\_API](https://wiki.openstreetmap.org/wiki/RU:Overpass_API) (дата обращения: 21.01.2024).
2. GeoPy Documentation // GeoPy : сайт. 2006–2018. URL: <https://geopy.readthedocs.io/> (дата обращения: 21.01.2024).
3. Python-telegram-bot Documentation : сайт. 2015–2024. URL: <https://docs.python-telegram-bot.org/en/stable/index.html> (дата обращения: 21.01.2024).
4. OpenStreetMap Wiki : сайт. 2024. URL: [https://wiki.openstreetmap.org/wiki/Main\\_Page](https://wiki.openstreetmap.org/wiki/Main_Page) (дата обращения: 27.12.2023).
5. PyYAML Documentation : сайт. 2024.  
URL: <https://pyyaml.org/wiki/PyYAMLDocumentation> (дата обращения: 23.01.2024).
6. Dataclass-factory Documentation : сайт. 2020. URL: <https://dataclass-factory.readthedocs.io/en/latest/quickstart.html> (дата обращения: 21.01.2024).
7. Redis Documentation : сайт. 2024. URL: <https://master--redis-doc.netlify.app/docs/> (дата обращения: 25.01.2024).

### References

1. Overpass API // Wiki : site. 2024.  
URL: [https://wiki.openstreetmap.org/wiki/RU:Overpass\\_API](https://wiki.openstreetmap.org/wiki/RU:Overpass_API) (access date: 21/01/2024).
2. GeoPy Documentation // GeoPy : site. 2006–2018. URL: <https://geopy.readthedocs.io/> (access date: 21/01/2024).
3. Python-telegram-bot Documentation : site. 2015–2024. URL: <https://docs.python-telegram-bot.org/en/stable/index.html> (access date: 21/01/2024).
4. OpenStreetMap Wiki : site. 2024. URL: [https://wiki.openstreetmap.org/wiki/Main\\_Page](https://wiki.openstreetmap.org/wiki/Main_Page) (access date: 27/12/2023).
5. PyYAML Documentation : site. 2024.  
URL: <https://pyyaml.org/wiki/PyYAMLDocumentation> (access date: 23/01/2024).
6. Dataclass-factory Documentation : site. 2020. URL: <https://dataclass-factory.readthedocs.io/en/latest/quickstart.html> (access date: 21/01/2024).
7. Redis Documentation : site. 2024. URL: <https://master--redis-doc.netlify.app/docs/> (access date: 25/01/2024).

*Авторы заявляют об отсутствии конфликта интересов.*

*Статья поступила в редакцию 23.04.2024; одобрена после рецензирования 03.05.2024; принята к публикации 04.05.2024.*

*The authors declare no conflicts of interests.*

*The article was submitted 23.04.2024; approved after reviewing 03.05.2024; accepted for publication 04.05.2024.*

© А. В. Меркулова, И. М. Напсо, 2024