

*В.В. Ковылина, К.О. Москвина, А.Д. Казаков, Р.М. Якубов,
В.Д. Янков, С.Л. Бугаева, Н.М. Гаврилова*

Тюменский государственный университет, г. Тюмень

УДК 004.9

РАЗРАБОТКА СЕРВИСА ДЛЯ ПОМОЩИ В УТИЛИЗАЦИИ МУСОРА

Аннотация. В данной статье описывается идея, принцип работы и архитектура разработанного мобильного приложения для поиска ближайших пунктов приема мусора в зависимости от категории сдаваемых отходов, построение оптимального маршрута от указанного места до выбранного пункта.

Ключевые слова: мобильное приложение, переработка мусора, утилизация, экология, Flutter, Django, API.

Введение

На сегодняшний день в России действует более 300 комплексов по переработке мусора различных мощностей [1]. Однако лишь 7,5% твердых коммунальных отходов (далее – ТКО) перерабатывается, оставшаяся часть отправляется на полигоны для отходов – санкционированные и несанкционированные [2]. Это происходит из-за того, что заводам не хватает отсортированного сырья, по разным данным, их загруженность составляет от 30% до 60%. Причем большую часть отходов составляют промышленные и импортное вторсырье, и лишь малая часть приходится на отдельно собранный мусор.

Проблема заключается в том, что, несмотря на положительный взгляд большинства россиян на отдельный сбор мусора, и то, что многие готовы уже сейчас этим заниматься, сталкиваясь с такими сложностями, как непонятные правила сортировки, необходимость следить: где, что и как именно принимают, люди часто бросают это дело, находя его чересчур утомительным [3]. Эти сложности возникают по той причине, что пункты отдельного сбора отходов существуют чаще как частные инициативы небольшого масштаба, а потому требования в каждом пункте разнятся.

По официальным данным бытовой мусор занимает не очень большую территорию – лишь 2,3% от всей загрязненной территории России. Но ежегодно количество бытового мусора увеличивается, а место для его хранения – уменьшается (рис. 1). В 32 регионах уже через 5 лет не будет места для его складирования, поэтому уже сейчас нужно предпринимать действия для сокращения производства бытовых отходов.

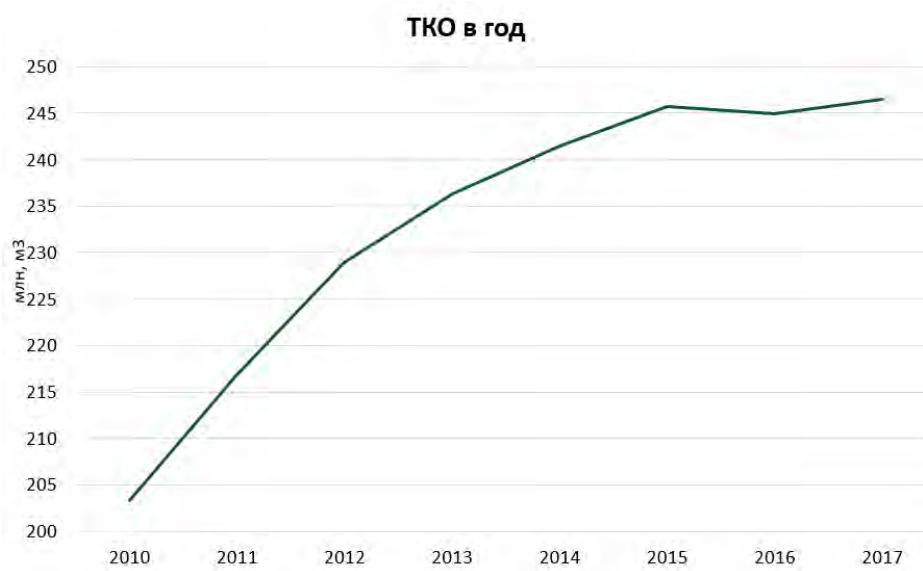


Рис. 1. График роста бытовых отходов

Идея решения

В качестве решения поставленной проблемы необходимо разработать клиент-серверное мобильное приложение для облегчения доступа к актуальной информации о местах сбора бытовых отходов в целях повышения уровня экологической грамотности населения и стимулирования жителей г. Тюмени к разделительному сбору мусора.

Серверная часть приложения должна будет хранить и, при поступлении соответствующего запроса, выдавать информацию о пунктах приема мусора.

Клиентская часть будет отображать загруженную из серверной или локальной базы данных информацию о местах возможной сдачи мусора, причем места должны удовлетворять выбранным пользователем категориям сдаваемого мусора. Предоставленная информация должна отображаться в виде точек на карте. Пользователь сможет выбрать желаемый пункт приема мусора из

доступных, и приложение построит маршрут до него от местоположения пользователя или от другой заданной точки.

Реализация

На рис. 2, 3 представлены схемы работы сервиса.

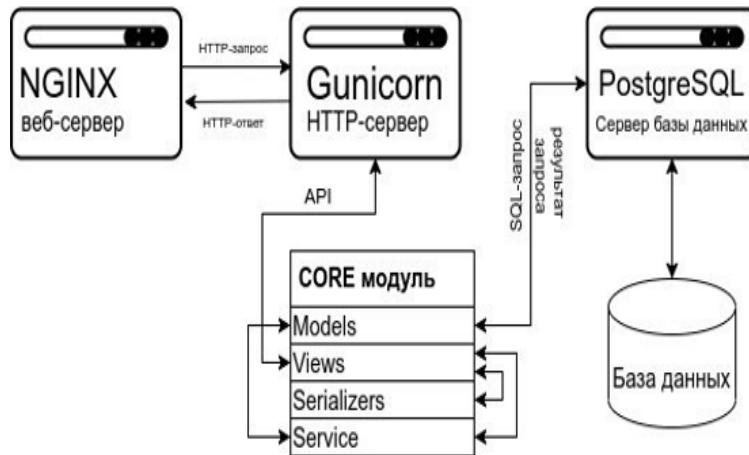


Рис. 2. Серверная часть схемы работы сервиса

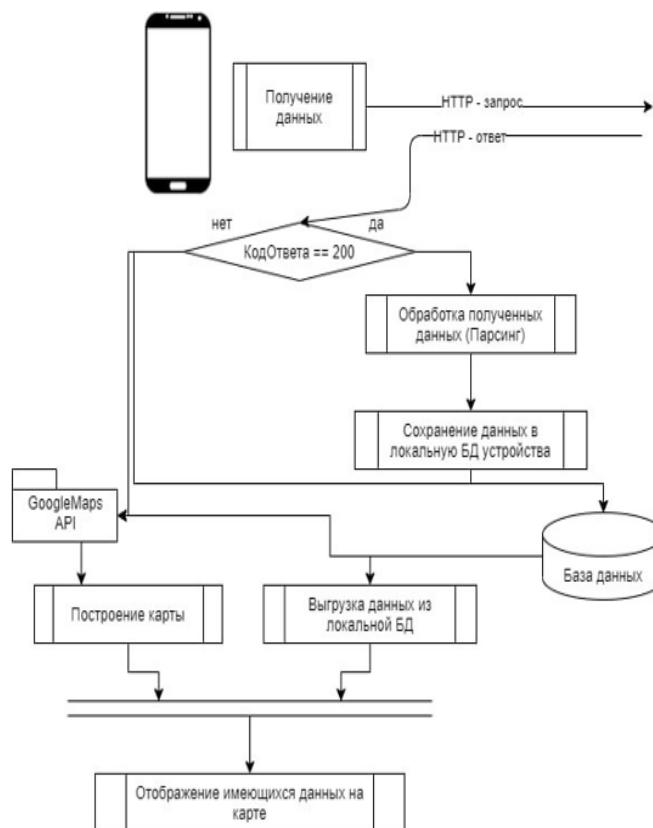


Рис. 3. Мобильная часть схемы работы сервиса

Серверная часть

В серверной части нашего решения реализовано *горизонтальное масштабирование*, т.е. увеличение производительности проекта и разделение его архитектуры путем добавления нескольких серверов. Схема базы данных серверной части представлена на рис.3.

Веб-сервер (NGINX) принимает запросы извне, обрабатывает их и отправляет ответ в соответствии со стандартом протокола на HTTP-сервер.

HTTP-сервер (Gunicorn) работает с Django-приложением, содержащим его “ядро” – модуль CORE.

Модуль CORE содержит:

- модели (models), которые выступают источниками информации о наших данных;
- представления (views), которые реализуют HTTP-методы;
- сериализаторы (serializers), преобразующие данные моделей, описанных на языке программирования Python, в JSON;
- сервис (service), содержащий бизнес-логику приложения.

Серверная база данных (PostgreSQL) принимает SQL-запросы, обрабатывает их, используя базу данных, представленную на рис.4, и отправляет результаты обратно HTTP-серверу.

Серверная база данных состоит из следующих таблиц:

1. *place* – содержит информацию о точках, которые располагаются на карте: их адрес, краткое описание, время работы и координаты (широта и долгота);
2. *place_type* – содержит типы мусора (всего было выделено 15): макулатура, стекло, пластик, металл, текстиль, дерево, опасные отходы, батарейки, лампочки, бытовая техника, крышечки, резина, тетрапаки, благотворительность и “прочее”, то есть то, что нельзя отнести ни к одной из перечисленных выше категорий;
3. *place_subtype* – содержит информацию о том, чем является точка, отмеченная на карте: пунктом переработки мусора, пунктом его сбора

или же местом проведения какого-либо мероприятия экологической тематики;

4. *place_place_type* – связующая, так как между первыми двумя таблицами существует отношение “многие ко многим”: в каждом пункте можно сдать/утилизировать несколько типов мусора, и каждый тип мусора может находиться в нескольких точках сбора/утилизации по городу.

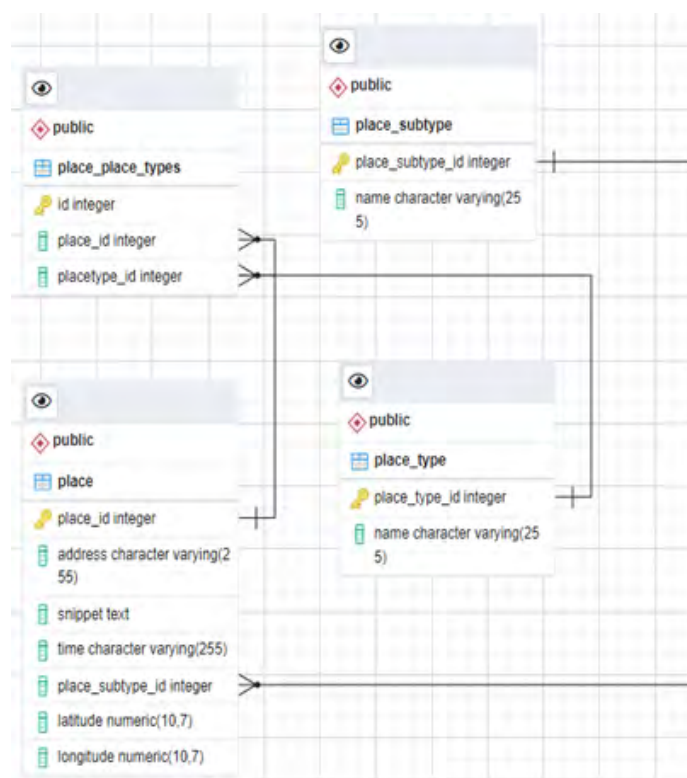


Рис. 4. Схема серверной базы данных

Мобильное приложение

Мобильное приложение было разработано на языке программирования Dart. Стек используемых технологий:

Flutter – фреймворк для написания кроссплатформенных приложений:

SQLite3 – СУБД, которая использовалась для сохранения на мобильном устройстве уже полученной с серверной БД информации;

Google Maps API – инструмент для добавления карты в мобильное приложение.

На рис.4 представлен прототип мобильного приложения. При запуске приложения, в результате обмена сообщениями с сервером посредством интерфейса API, происходит загрузка данных с сервера, с последующим сохранением в локальную БД телефона.

После запуска приложения пользователь попадает на карту, первоначальным фокусом которой будет местоположение устройства.

После окончания загрузки данных с сервера и построения карты (см. рис.4, правая часть), пользователь начинает взаимодействовать с приложением: ему доступна возможность просматривать карту, обнаруживать маркеры, получать информацию об отмеченных на карте точках (см. рис.5, левая часть), использовать функционал построения маршрута от любой возможной точки (включая текущее местоположение) до искомого маркера.

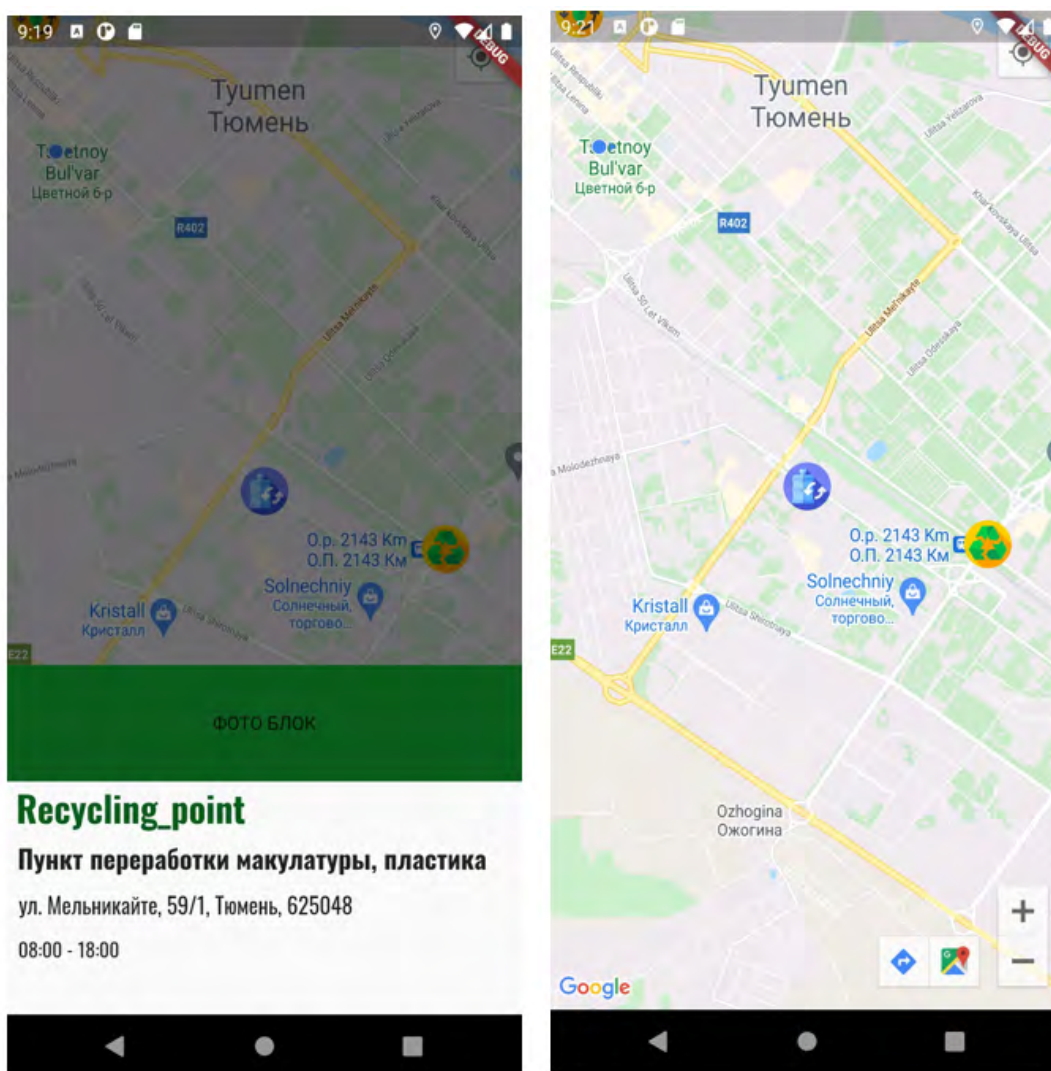


Рис. 5. Прототип мобильного приложения

Заключение

В рамках проведенной работы было разработано мобильное приложение для поиска ближайших пунктов приема мусора, удовлетворяющих заданным категориям, и построения оптимального маршрута до них. Приложение реализовано на языке Dart с использованием фреймворка Flutter. Серверная часть приложения реализована на языке Python с помощью фреймворка Django.

Также были задействованы две СУБД: SQLite, PostgreSQL. SQLite использовалась для сохранения на мобильном устройстве уже полученной с серверной БД информации, что позволило избежать повторных идентичных запросов к серверу, в то время как PostgreSQL принимала SQL-запросы, совершала обработку и возвращала результат HTTP-серверу.

СПИСОК ЛИТЕРАТУРЫ

1. Жерносек А.В. Проблема внедрения системы селективного сбора бытовых отходов в России / А.В. Жерносек, Е.О. Реховская // Материалы VI Международной научно-практической конференции: статья в сборнике трудов конференции. – Омск: Омский государственный технический университет, 2019 – С. 342-345.

2. Харчикова Н.В. Раздельный сбор мусора в России и за рубежом: проблемы и перспективы // Материалы межрегиональных научно-практических конференций. – Калуга: ФГБОУ ВО «Российская академия народного хозяйства и государственной службы при Президенте Российской Федерации», Калужский филиал, 2020 – С. 260-270.

3. Большинство россиян поддерживают идею раздельного сбора мусора – ROMIR.RU [Электронный ресурс]. – URL: <https://romir.ru/studies/bolshinstvo-rossiyan-podderjivaet-ideyu-razdelnogo-sbora-musora> (дата обращения: 20.04.2021).

4. Документация языка Dart: [Электронный ресурс]. – URL: <https://dart.dev/guides> (дата обращения: 01.05.2021)

5. Документация фреймворка Flutter: [Электронный ресурс]. – URL: <https://flutter.dev/docs> (дата обращения: 01.05.2021)