УДК 004.42+528.7

РАЗРАБОТКА И ПРИМЕНЕНИЕ МОБИЛЬНОГО ПРИЛОЖЕНИЯ ПО СБОРУ СТАТИСТИКИ ГОРОДСКИХ ТРАНСПОРТНЫХ ПОТОКОВ

И. Ю. Зайцев, И. Н. Даниленко
Сургутский государственный университет, ilia.zaitsev@outlook.com, vice1@surgu.ru

Получение статистических данных об интенсивности движения городских транспортных потоков сопряжено с рядом сложностей. Подобного рода статистика не всегда ведется муниципальными властями, и зачастую для ее получения необходимо иметь доступ к установленным на перекрестках камерам видеонаблюдения. В статье приводится описание процесса разработки клиентской и серверной частей мобильного приложения по сбору статистики автомобильного передвижения пользователя и оценивается возможность применения собранных данных для анализа интенсивности городского движения.

Ключевые слова: транспортные потоки, мобильная разработка, Python, геолокация.

DEVELOPMENT OF MOBILE APPLICATION FOR THE URBAN-TRAFFIC STATISTICS GATHERING

I. Yu. Zaitsev, I. N. Danilenko
Surgut State University, ilia.zaitsev@outlook.com, vice1@surgu.ru

A process of collecting urban-traffic density statistics is connected to a range of complications. Municipal authorities do not always gather such statistics. Due to various reasons, access to street cameras is not always possible. The authors describe the development of mobile application intended to track citizens' automotive movement geolocations and possibilities to use the collected data for analysis of urban traffic density.

Keywords: traffic streams, mobile development, Python, geolocation.

Введение. Построение любой модели управления городскими транспортными потоками для последующего повышения эффективности светофорного регулирования на дорожных перекрестках связано с необходимостью получения статистики о скорости передвижения автомобилей на различных участках транспортной сети в различные интервалы времени и дни недели. Подобного рода статистика не всегда ведется муниципальными властями и для ее получения необходимо иметь доступ к видеозаписям, полученным с установленных на перекрестках камер видеонаблюдения, после чего применить алгоритмы распознавания образования образов и детектирования автомобилей для преобразования видеоряда в количественные характеристики. В следующих разделах рассматривается разработка альтернативного подхода к сбору информации о загруженности городской транспортной инфраструктуры, который заключается в установке на мобильные устройства горожан приложения, анонимно собирающего геолокационные данные пользователей в моменты их передвижения на автомобиле или автобусе. Такой способ сбора информации позволяет собирать статистику даже в тех случаях, когда камеры наблюдения за дорожным движением или же какие-либо иные средства контроля транспорта отсутствуют или же получение доступа к ним ограничено.

Методика сбора статистки транспортного движения с применением датчиков геолокации

Для решения задачи моделирования городских транспортных потоков необходимо наличие статистики автомобильного движения по улицам города в различное время суток.

В общем случае получение подобного рода сведений может осуществляться различными способами, такими как:

1) получение статистики дорожного движения в структурированном формате от городских автотранспортных служб;

2) расчет пропускной способности городской транспортной сети посредством обработки видеозаписей с камер слежения за дорожным движением, установленных на перекрестках;

3) установка дополнительного оборудования на перекрестках или применение других специальных средств [1, 2].

Однако каждый из указанных выше способов связан с разного рода ограничениями:

- структурированная статистика может отсутствовать;
- не все участки дорожной сети могут быть снабжены видеорегистраторами;
- качественная обработка видеозаписей движения транспортных потоков сама по себе представляет нетривиальную задачу;
- оборудование дополнительными датчиками всех перекрестков дорожной сети крупного города требует значительных экономических вложений и юридических согласований.

Исходя из указанных соображений, авторами было принято решение предпринять качественно иной подход к определению пропускной способности городской транспортной сети. Предлагаемая авторами методика заключается в разработке мобильного клиент-серверного приложения, в котором используются датчики определения геолокации и характера передвижения пользователя, установленные во многих современных мобильных устройствах, и которое предназначено для сбора сведений о перемещении отдельных автомобилей в пределах городской транспортной сети.

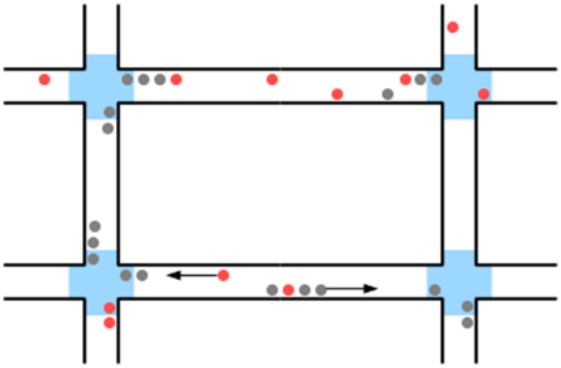

Рис. 1. Схематическое представление фрагмента транспортной сети с отслеживаемыми автомобилями

Разрабатываемое авторами приложение состоит из мобильного клиента и серверного API, предоставляющего несколько точек доступа для анонимной отправки и получения геолокационных данных пользователя.

Разработка мобильного клиента ведется для устройств с операционной системой iOS на языке программирования (далее – ЯП) Swift. Разработка серверной части приложения осуществляется на ЯП Python с применением фреймворка Flask [4], в качестве системы управления базами данных (СУБД) выбрана PostgreSQL [5]. На рис. 2 представлена общая структура разрабатываемого приложения.

Логика сбора геолокаций полностью сосредоточена в мобильном клиенте и основана на применении iOS SDK, что позволяет получить простое, но функциональное решение. Недостатком подобного подхода является зависимость от операционной системы мобильного устройства, которым обладает пользователь. Однако, по мнению авторов, наибольшую сложность представляет преобразование собранных сведений в формат, пригодный для дальнейшего моделирования. Портирование приложения на другие устройства представляет собой проще формализуемую задачу.

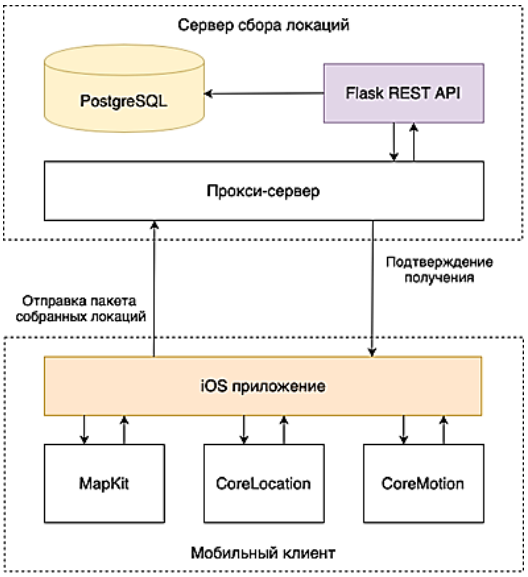

Рис. 2. Схематическое представление общей структуры разрабатываемого приложения

Далее следует более подробное рассмотрение каждой из обозначенных частей приложения. В первую очередь будет рассмотрена схема БД. Затем будет дано краткое описание мобильного клиента. В завершении раздела будет рассмотрена структура серверного API.

База данных (далее – БД). Главная задача разрабатываемого приложения сводится к сбору геопозиций пользователя во время передвижения по городским дорогам в личном автомобиле или посредством общественного транспорта. По этой причине схема БД, необходимой для поддержания работы REST API, содержит весьма ограниченное количество сущностей, как это представлено на рис. 3.

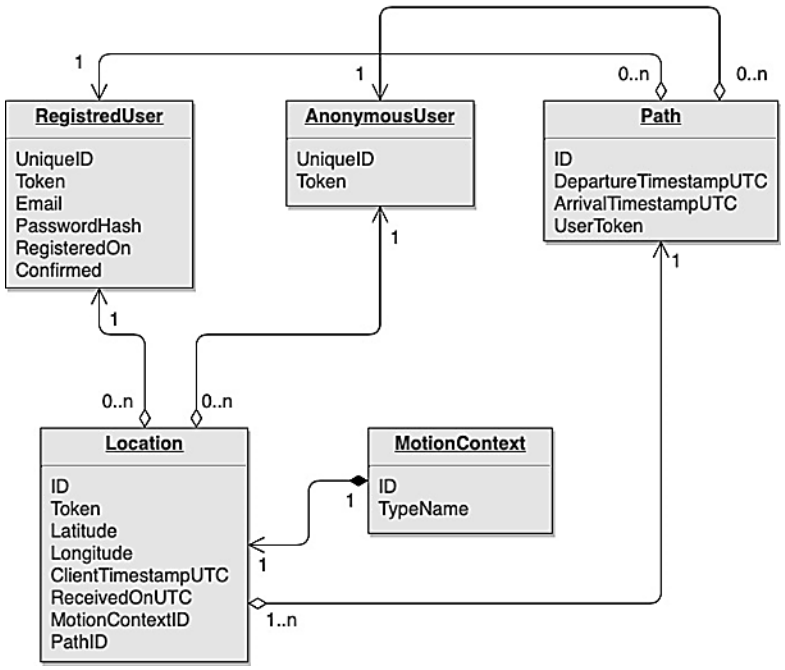

Рис. 3. Базовый вариант схемы БД приложения, используемой для хранения собранных сведений о локациях пользователя

Сущность Location хранит отправленную клиентом геопозицию, тип движения, временные штампы отправки и получения, а также уникальный идентификатор приложения, от-

правившего данные. Собранные локации организуются в более «крупные» сущности-пути, (сущность Path), каждая из которых представляет собой массив локаций с момента начала движения в транспортном средстве до момента остановки и выхода из него. Объединение отдельных локаций в пути осуществляется на стороне сервера на основе собранных временных штампов и уникальных идентификаторов.

Другими необходимыми сущностями являются Registered User и Anonymous User. Обе сущности необходимы для разбиения всего массива полученных геолокаций на релевантные пути, соответствующие перемещениям определенного пользователя.

Необходимо отметить, что идентификация пользователя через электронный адрес не является необходимым условием использования приложения. Более того, на текущий момент клиент приложения не предусматривает возможность регистрации. Каждый пользователь идентифицируется через уникальный iCloud-токен [6], генерируемый при каждой установке приложения. Наличие дополнительной сущности в БД для хранения неанонимных пользователей предусмотрено для дальнейших итераций разработки и поддержки «тестовых» аккаунтов для однозначной идентификации устройства и более точных измерений.

Серверная часть. Основная функциональность серверного API сосредоточена в получении локаций от мобильного клиента, проверке уникального iCloud-токена пользователя и сохранении собранных локации в БД. Помимо этого API предусматривает ряд «служебных» точек доступа, предназначенных для управления собранными данными.

На рис. 4 схематически представлено взаимодействие клиента и сервера при сборе геолокаций. Для уменьшения частоты и объема пересылаемых данных клиент кэширует собранные локации и отправляет их пакетами с частотой меньшей, чем частота получения обновлений от трекера движения.

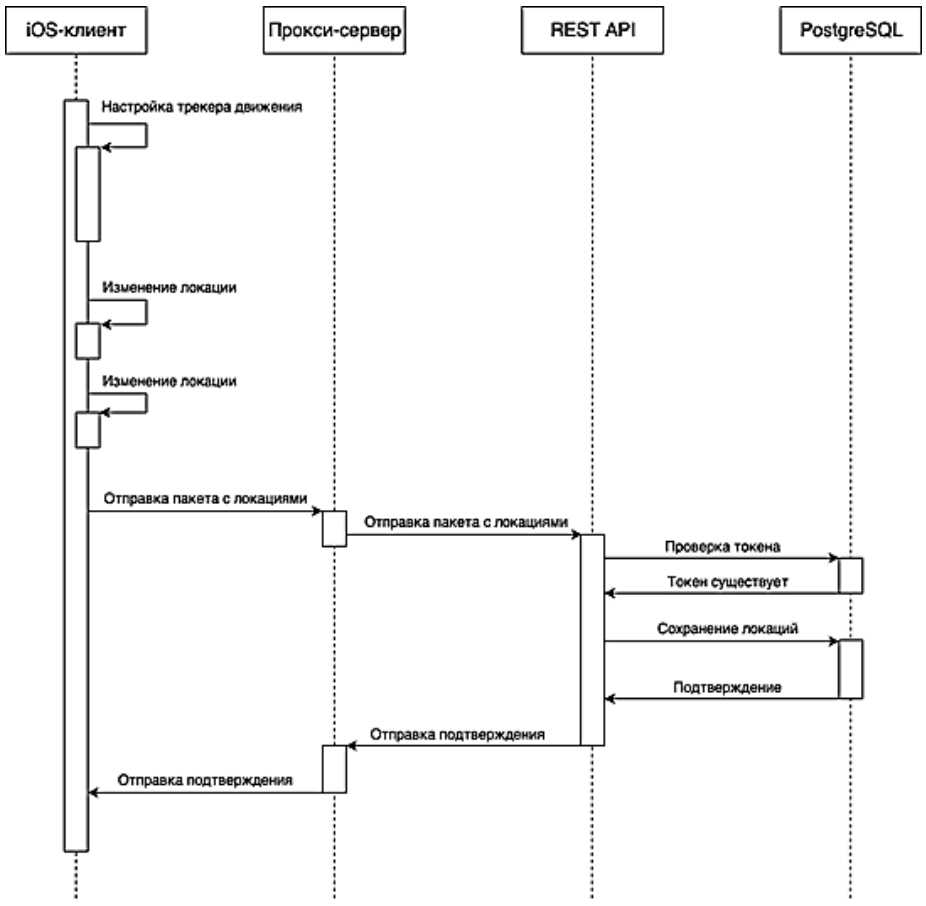

Рис. 4. Диаграмма взаимодействия мобильного клиента и сервера при отправке пакетов с локациями

Как уже было упомянуто ранее, реализация REST API осуществляется при помощи ЯП Python и веб-фреймворка Flask. В качестве прокси-сервера используется nginx [7]. Приложение развернуто на облачном хостинге AWS.

В дальнейшем авторами планируется расширение базового API путем добавления точек доступа, предназначенных для отправки агрегированной статистики на устройство пользователя с целью отображения на дополнительном экране с метриками, которые могут быть интересны пользователю.

Мобильный клиент. Главная задача мобильного приложения заключается в детектировании автомобильного движения и отправке локации пользователей на сервер с определенной периодичностью. По причине своей весьма узкой специализации мобильный клиент очень прост и состоит из группы экранов, объясняющих пользователю принцип своей работы, карты с отметками собранных локаций и экрана с настройками, позволяющими задать расписание для сбора статистики или же полностью отключить слежение в случае необходимости.

Ключевой частью приложения является конечный автомат, определяющий текущее состояние пользователя. Автоматом предусмотрены следующие состояния:

1) инициализация;
2) вне автотранспортного средства;
3) в автотранспортном средстве (в движении);
4) в автотранспортном средстве (остановка).

На рис. 5 представлен ориентированный граф, отражающий структуру переходов между состояниями автомата.

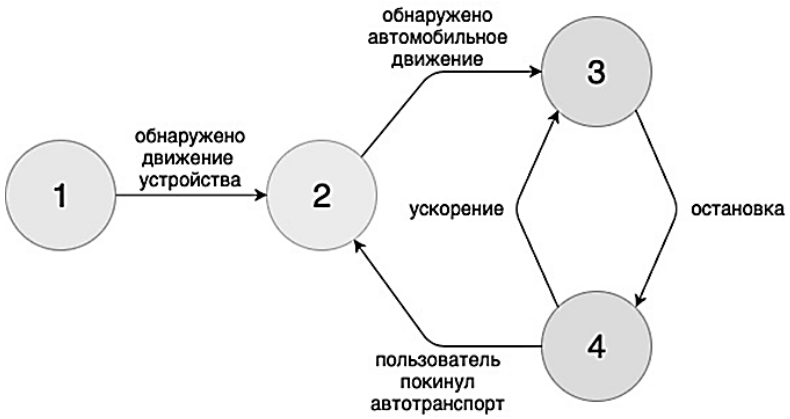

Рис. 5. Конечный автомат состояний приложения при движении пользователя по городу

Автомат находится в состоянии 1 после запуска приложения и до первого детектирования движения. В случае, если движение обнаружено, но не является движением в автомобиле, автомат переходит в состояние 2. Если характеристика движения пользователя предполагает собой нахождение в автомобиле, автомат переходит в состояние 3 или 4 в зависимости от того, двигается автомобиль или же остановился на светофоре. Отправка геолокаций пользователя на сервер осуществляется только в том случае, когда автомат находится в состоянии 3 или 4.

Для получения сведений о локации пользователя и определении типа движения авторы используют CoreLocation и CoreMotion SDK операционной системы iOS [8, 9].

Сведения, собранные с помощью тестовой сборки приложения

В процессе тестирования разработанного приложения авторами была произведена его установка на личные устройства с последующим перемещением по городу на автомо-

биле и в автобусе. На рис. 6 и 7 показаны результаты двух тестовых поездок при запущенном приложении.

Рис. 6. Точки передвижения, собранные при движении на автомобиле (синий) и в автобусе (красный)

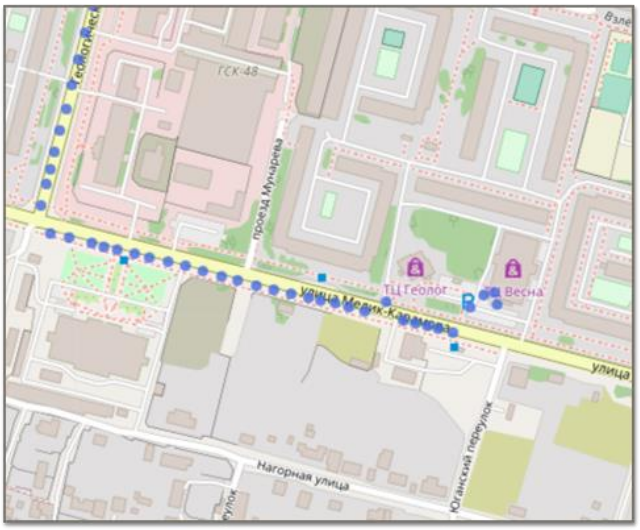

Рис. 7. Задержка детектирования остановки в течение нескольких минут после выхода из автомобиля

Необходимо отметить, что движение на автомобиле сопровождалось более поздним детектированием начала движения, чем при движении на автобусе, что может быть связано с невысокой скоростью движения автомобиля при выезде из жилого сектора на дорогу. Кроме того, рис. 6 демонстрирует заметную задержку перехода в состояние 2 после прибытия на место назначения.

На основе представленных наблюдений авторами делается вывод о необходимости дополнительной обработки собранных сведений для фильтрации аномальных наблюдений, а также дальнейшей доработке системы управления конечным автоматом.

Заключение. Тестирование текущей версии приложения показало успешное детектирование автомобильного движения и сбора статистики передвижения отдельного пользователя.

В качестве следующего этапа своей работы авторы рассматривают усовершенствование алгоритмов детектирования автомобильного движения и интенсивное тестирование мобильного клиента для выявления возможных программных ошибок и обеспечения качества

собираемых сведений. Еще одной задачей является добавление дополнительных экранов к мобильному приложению, мотивирующих пользователя к его использованию, а также разворачивание приложения на большем количестве устройств для сбора подробных сведений об интенсивности движения городских транспортных потоков.

В случае успешного решения задачи сбора геололокационной информации в достаточном объеме авторам представляется возможным построение мультиагентной модели для дальнейшего прогнозирования дорожной ситуации и оптимизации стратегии светофорного регулирования [10, 11].

Литература

1. Кураксин А. А. Совершенствование методов оценки эффективности организации дорожного движения на основе применения технологии мезоскопического моделирования транспортных потоков : автореф. дис. ... канд. техн. наук. Орел, 2017. 18 с.

2. Султанахмедов М. А. Повышение эффективности регулирования городских транспортных потоков на основе моделирования : дис. … канд. техн. наук. Махачкала, 2011. 156 с.

3. Соловьев В. А. Моделирование и оптимизация управления движением транспортных потоков в сети крупного города : дис. … канд. техн. наук. Тюмень, 2013. 118 с.

4. Flask: web development, one drop at a time. URL: http://flask.pocoo.org (дата обращения: 31.10.2018).

5. PostgreSQL: The World's Most Advanced Open Source Relational Database. URL: https://www.postgresql.org (дата обращения: 19.11.2018).

6. UX: iOS Onboarding without Signup Screens // Medium : сетевой журнал. URL: https://medium.com (дата обращения: 20.12.2018).

7. nginx [engine x]: an HTTP and reverse proxy server. URL: https://nginx.org/ru/ (дата обращения: 20.12.2018).

8. Core Location. Obtain the geographic location and orientation of a device. URL: https://developer.apple.com (дата обращения: 20.12.2018).

9. Core Motion. Process accelerometer, gyroscope, pedometer, and environment-related events. URL: https://developer.apple.com (дата обращения: 21.12.2018).

10. Зайцев И. Ю. Средства построения программных моделей, основанных на мультиагентных системах // Наука и инновации XXI века : материалы III Всерос. конф. молодых ученых. Сургут, 1–2 декабря, 2016 г. Сургут : ИЦ СурГУ, 2016. С. 41–48.

11. Зайцев И. Ю., Даниленко И. Н. Мультиагентные системы в решении задачи моделирования городской транспортной инфраструктуры // Север России: стратегии и перспективы развития : материалы III Всерос. науч.-практ. конф. Сургут, 26 мая, 2017 г. : в 3 т. Сургут : ИЦ СурГУ, 2017. Т. II. С. 83–90.