

# Использование веб-интерфейсов для сервисов на основе сетевой близости

П.М. Волосникова, Д.Е. Намиот

**Аннотация**—В данной статье рассматривается применение современных стандартов веб-интерфейсов для реализации приложений сервисов, основанных на концепции сетевой близости. Сетевая близость - это свойство пространственной близости, которая определяется видимостью одного сетевого устройства другим. В работе исследуется понятие сетевой близости, проводится анализ и сравнение технологий беспроводной передачи данных малого радиуса действия. Проводится анализ необходимых для реализации прикладных сервисов программных интерфейсов. Далее в работе исследуется текущее состояние программных интерфейсов для работы с беспроводными сетями из языка JavaScript, которые реализуются Mozilla Developer и Chromium. В работе описывается модельное веб-приложение для задач сетевой близости. Помимо демонстрации элементов работы с интерфейсами беспроводных сетей из JavaScript, это приложение имеет и собственный прикладной аспект - автоматизация контроля посещаемости занятий учащимися.

**Ключевые слова**— Network proximity, Web API, Bluetooth, Wi-Fi.

## I. ВВЕДЕНИЕ

В настоящее время интернет стал неотъемлемой частью нашей жизни. Интернет-технологии используются в различных сферах деятельности и жизни человека, а их новое поколение, получившее название Интернета вещей, формирует глобальную информационную инфраструктуру цифровой экономики и информационного общества.

Под термином Интернет вещей понимается система объединенных компьютерных сетей и подключенных к ним физических предметов (вещей) с встроенными в них датчиками, выполняющим функции сбора и обмена данными, оснащенная сервисами и приложениями для хранения и обработки больших объемов данных, а также для удаленного контроля и управления в автоматизированном режиме устройствами и процессами без участия человека.

Широкие возможности Интернета вещей предопределили интенсивное развитие индустрии приложений, направленное на создание всеобъемлющего спектра сетевых сервисов и

приложений, поддерживающих масштабную цифровизацию экономики и социума.

Один из таких классов приложений характеризуется использованием сетевых технологий, основанных на понятии пространственной локальности, или сетевой близости. Данные технологии позволяют решать задачи геолокации устройств внутри экранированных помещений, дополняя их возможностями предоставления контекстно-зависимой информации. Примерами такого класса решений могут служить «умные» шахты, некоторый класс платежных систем, сервисы взаимосвязи в локальных пространствах с членами групп социальных сетей и др.

В связи быстрым развитие индустрии приложений разработка и развитие инструментальных средств, технологий и методов создания высокоэффективных сетевых сервисов и приложений для разного класса применений, представляет собой актуальную научно-техническую задачу.

Именно такой актуальной теме применительно к технологиям сетевой близости и посвящена данная работа. В работе исследуются и сравниваются технологии беспроводной передачи данных малого радиуса действия, используемые для построения сервисов и приложений на основе концепции сетевой близости. Далее анализируются возможности стандартов веб-интерфейсов для создания удобных в использовании сервисов и приложений сетевой близости. С целью отработки методики проектирования приложений указанного класса предлагается типовая архитектура таких приложений и разрабатывается на ее основе модельное приложение, на примере которого фактически демонстрируются принципы создания приложений данного класса. Созданное в работе приложение может использоваться в качестве инструмента преподавателя, предназначенного для автоматизации контроля посещаемости занятий учащимися.

## II. СЕТЕВАЯ БЛИЗОСТЬ

Сетевая близость - это свойство пространственной близости, которая определяется видимостью одного сетевого устройства другим. Радиус действия сетевого устройства ограничен, поэтому его видимость другими устройствами является некоторой характеристикой местоположения этих устройств. В зависимости от этого местоположения пользователю предоставляется

Статья получена 9 мая 2020.

П.М. Волосникова – МГУ имени М.В. Ломоносова (email: pvolosnikova98@gmail.com)

Д. Е. Намиот – МГУ имени М.В. Ломоносова (email: dnamiot@gmail.com).

контекстно-зависимая информация и дополнительная функциональность. Системы, основанные на принципе сетевой близости, используют сетевые интерфейсы мобильных устройств как датчики расстояния. Устройство способно определять либо нахождение другого устройства в некоторой его окрестности (способность принять и обработать сообщение, посланный другим устройством), либо получать некоторую полезную информацию из полученных характеристик других устройств. В первом случае функция сетевой близости будет являться бинарной, и возвращать 1, если объект находится в зоне видимости, и 0 иначе. Во втором случае функция не является бинарной и может возвращать любое значение.



Рис.1. Сетевая близость

Можно выделить ряд преимуществ сервисов, основанных на сетевой близости:

- *Системы позиционирования* [2]. Системы глобального позиционирования плохо работают в помещениях - качество сигнала ухудшается при проходе через бетонные стены и металлические укрепления. Помимо этого, они имеют довольно большую погрешность, которая не позволяет применять их для сервисов, функциональность которых требует высокую точность. Сетевые технологии ближнего действия позволяют решить эти проблемы. Применяя методы спутникового позиционирования для систем, основанных на технологиях малого радиуса действия, и при достаточной плотности сетевых элементов системы, можно добиться необходимой точности позиционирования.

- *Стоимость и распространенность*. Разработка сетевых технологий ближнего действия имеет относительно низкую стоимость, что увеличивает их распространенность. Такие технологии, как Bluetooth и Wi-Fi, на сегодняшний день являются повсеместными. Распространенность снижает общую стоимость систем, так как требуется меньшее количество дополнительных затрат на оборудование - люди сами приобретают устройства, оснащенные Bluetooth и Wi-Fi.

- *Перемещаемость*. Одно из главных преимуществ сетей ближнего действия - это мобильность. В сравнении с глобальными системами, их можно легко перемещать, менять относительное положение компонентов.

- *Гибкость*. Есть возможность создавать системы, компоненты которой в течение времени могут менять сценарий функционала.

Сервисы, основанные на сетевой близости решают следующие задачи:

- Навигация внутри закрытых помещений [3].
- Предоставление контекстной информации внутри помещений [4], [5].
- Предоставление доступа к некоторым объектам и их управлению, требующего нахождение человека в некоторой окрестности.
- Объединение пользователей, находящихся в одной локации.
- Разработка систем умного дома и умного города.
- Идентификация и др.

Сервисы, основанные на сетевой близости, используют сетевые технологии ближнего действия. К основным технологиям относятся Wi-Fi, Bluetooth, ZigBee и NFC.

### III. СЕТЕВЫЕ ТЕХНОЛОГИИ

#### 1. Wi-Fi

Wi-Fi — технология беспроводной локальной сети на основе стандартов IEEE 802.11 [6].

Можно выделить следующие преимущества Wi-Fi:

- *Расширяемость*. Данную сеть довольно легко расширить до любых размеров, увеличивая количество клиентов и геометрию сети установкой дополнительных точек доступа.

- *Глобальная совместимость*. Wi-Fi гарантирует совместимость устройств благодаря обязательной сертификации оборудования.

К недостаткам можно отнести:

- *Достаточно высокое энергопотребление*.

#### 2. Bluetooth

Bluetooth - это технология беспроводной передачи данных между устройствами на основе стандарта IEEE 802.15.1 [7].

К достоинствам Bluetooth можно отнести:

- *Помехоустойчивость и надежность*. Алгоритм скачкообразной перестройки частоты, используемый в bluetooth, позволяет устройствам быстро изменять несущую частоту в широком диапазоне рабочих частот. Это позволяет избежать переполнения в рабочем частотном диапазоне и сделать подключение более устойчивым к помехам.

- У технологии Bluetooth относительно *низкая стоимость*.

• *Совместимость*. На сегодняшний день Bluetooth является одним из распространенных сетевых технологий, присутствует во многих устройствах и позволяет обмениваться информацией вне зависимости от типа такого устройства.

• *Низкое энергопотребление* для *Bluetooth LE*.

Недостатками является:

*Энергопотребление*. Для версий Bluetooth 3.x и ниже потребляемая энергия в течение одной передачи данных незначительна, но на многих устройствах Bluetooth постоянно включен и это тратит значительный заряд батареи, а также ощутимо снижает срок ее службы.

### 3. Сравнение технологий

Таблица 1. Сравнение технологий.

Характеристика	Wi-Fi	Bluetooth	Bluetooth LE	ZigBee	NFC
Частота	2.4/5 ГГц	2.4 ГГц	2.4 ГГц	2.4 ГГц	13.56 МГц
Расстояние	100м	10м	40м	20м	10см
Стоимость	Высокая	Средняя	Низкая	Низкая	Низкая
Энергопотребление	Высокое	Среднее	Низкое	Низкое	Низкое
Поддержка веб-интерфейсами	+	+	+	-	-

Каждая из технологий имеет свои достоинства и недостатки. Для каждой задачи необходимо выбирать свой стандарт, максимально удовлетворяющий требованиям. Для общего случая подойдут Wi-Fi, Bluetooth и Bluetooth LE. Они являются самыми распространенными технологиями и поддерживаются всеми смартфонами. Далее стоит руководствоваться необходимым радиусом действия технологии для конкретной задачи. Технологию ZigBee стоит выбирать для систем, датчики которой распределены на небольшой территории и не требуют перемещения внутри нее. Технология NFC используется в системах, требующих близкий контакт.

### 4. Web API

Приложения сервисов, основанных на принципе сетевой близости, могут поддерживать следующие функции:

1. Включение и выключение сетевого модуля
2. Запуск сканирования
3. Получение характеристик

- SSID / Имя устройства
- MAC-адрес
- сила сигнала

4. Изменение характеристик

- SSID / Имя устройства

Отсутствие какой-либо функции в приложении влечет за

собой необходимость в выполнении этой функции пользователем вручную (например, через настройки мобильного устройства) и возможного ручного ввода полученной им информации в приложение. Некритичными функциями являются включение/выключение и изменение характеристик (например, SSID, который будет содержать полезную информацию: ссылку, идентификатор в системе и т.д.), которые требуют единоразового действия от пользователя.

Продвижением интерфейсов для работы с сетевыми устройствами на языке JavaScript занимаются две группы разработки - Mozilla Developer и группа разработки Chromium. Международной стандартизацией новых возможностей для веб-разработки занимается Консорциум Всемирной паутины W3C.

#### A. Wi-Fi Web API

• *Mozilla Developer* [8]. Пример:

```
var wifi = navigator.mozWifiManager;
my_mac_address = wifi.macAddress;
var request = wifi.getNetworks();
request.onsuccess = function () {
  var networks = this.result;
  networks.forEach(function (network) {
    network_ssid = network.ssid;
    network_bssid = network.bssid;
    ...
  })
}
```

• *Chromium* [9]. Пример:

```
chrome.networking.onc.requestNetworkScan("WiFi");
chrome.networking.onc.getNetworks({"networkType": "WiFi"}, function(networks) {
  networks.forEach(function (network) {
    network_ssid = network.WiFi.SSID;
  })
})
```

#### B. Bluetooth Web API

• *Mozilla Developer* [10]. Пример:

```
navigator.bluetooth.requestDevice({
  filters: [{
    services: ['heart_rate'],
  }]
}).then(device => device.gatt.connect())
.then(server =>
server.getPrimaryService('heart_rate'))
.then(service => {
  chosenHeartRateService = service;
  return Promise.all([
service.getCharacteristic('body_sensor_location')
  .then(function(characteristic) {
    ...
  })
]),
```

```

service.getCharacteristic('heart_rate_measurement')
    .then(function(characteristic) {
        ...
    });
});
});

```

- *Chromium* [11]. Пример использования данного интерфейса будет рассматриваться ниже.

В данных интерфейсах не поддерживается функционал включения, выключения и изменения характеристик. Это не является критичным и позволит уже сейчас применять эти веб интерфейсы для некоторых сервисов. Включение и выключение сетевого модуля предоставляется самому пользователю через настройки мобильного устройства. Изменение характеристик можно поддержать с помощью веб-сервера и хранилища. Полезная информация, которую можно было бы вложить в характеристику, сохраняется в хранилище, а изменяется и предоставляется другим пользователям через веб-сервер.

#### IV. ВЕБ-СЕРВИСЫ, ОСНОВАННЫЕ НА СЕТЕВОЙ БЛИЗОСТИ

Отметим, что отличительной чертой таких сервисов является распространенность, поэтому при выборе технологий для некоторого сервиса нужно руководствоваться поддержкой данных технологий устройствами и веб-интерфейсами.

Веб-сервисы, основанные на сетевой близости, построены на общей модели. В такой модели обязательно присутствуют как минимум два сетевых устройства, являющихся сканерами и передатчиками, и веб-приложение, функциональность которого основана на работе с информацией, полученной от сетевого устройства.

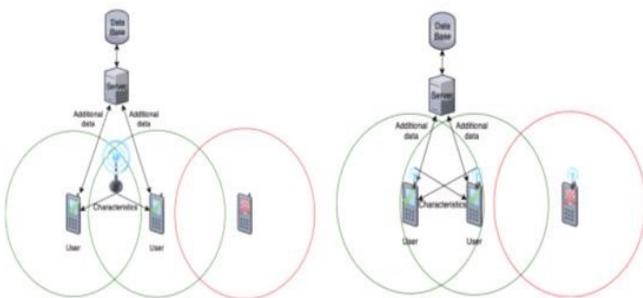


Рис.2. Веб-сервисы на базе сетевой близости

Устройства, которые опрашивают другие устройства в беспроводной сети для их обнаружения, называются *сканерами*. Для таких устройств необходимо, чтобы они были оснащены сетевыми модулями, а также на них должно быть предустановлено веб-приложение для доступа к этим модулям, запуска сканирования и обработки полученной от сетевых модулей информации.

Устройства, которые отвечают на опросы -

*передатчики*. Для передатчиков достаточно наличия сетевого модуля на устройстве. Пакеты, которые отсылают передатчики, должны содержать информацию об идентификации данного устройства. Помимо этого, в них могут содержаться данные для вычисления силы сигнала. Использование в качестве передатчиков устройств на основе сетевых технологий, определяющих свои специфичные протоколы, требует установки дополнительного ПО на считывающие устройства. Важно отметить, что для некоторых сервисов в качестве передатчиков можно использовать устройства-сканеры. В таком случае, есть возможность менять роли данных устройств в течение времени.

Веб-приложение — это клиент-серверное приложение, где клиент — это веб-браузер, а сервер — веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен данными происходит по сети.

Клиентская часть приложения может представлять собой веб-сайт, либо приложение, исполняющееся в браузере. Средства разработки должны иметь интерфейсы для доступа и управления к сетевым модулям устройства-сканера, на котором исполняется данное веб-приложение. Приложение должно считывать информацию, полученную сетевым устройством, и далее либо самостоятельно обрабатывать ее, либо отправлять на сервер, если для обработки необходимы дополнительные данные, не хранящиеся в приложении и на устройстве. Данные, полученные с сетевого устройства могут содержать информацию как о самом устройстве, так и о видимых этим устройством других.

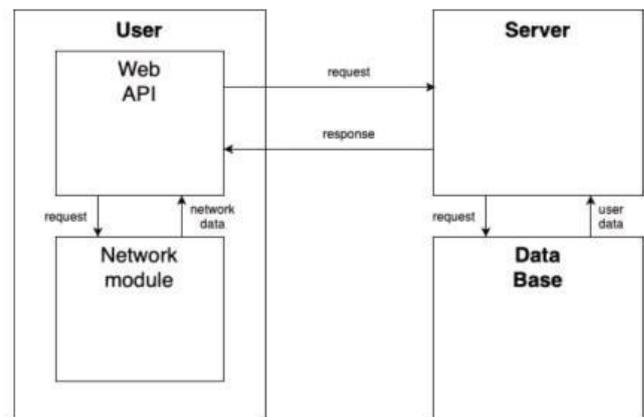


Рис.3. Архитектура веб-приложения

Веб-сервер — это сервер принимающий HTTP-запросы от клиентов по протоколу HTTP или HTTPS.. Данный сервер служит для аутентификации и авторизации пользователей, предоставления данных из хранилища и осуществления тяжелых вычислений по обработке информации. Данный сервер принимает запросы, содержащих адрес (идентификацию) устройства пользователя и адреса (идентификации)

других устройств, находящихся в области видимости. Сервер должен корректно работать как с устройствами, которые он знает, так и с незарегистрированными устройствами. Данные об зарегистрированных устройствах сервер берет из хранилища. Объекты хранилища представляются как записи в формате ключ-значение, где идентификация устройства, является первичным ключом, а информация об устройстве - значением.

## V. МОДЕЛЬНОЕ ПРИЛОЖЕНИЕ

В данной работе предлагалось разработать сервис, для учета посещаемости занятий. В приложении могут регистрироваться преподаватели и учащиеся. Преподаватели создают свои курсы, которые привязываются к ним, а учащиеся могут подписываться на них. На занятии преподаватель включает Bluetooth-адаптер, и в приложение открывает в нужном курсе регистрацию на занятие, определяющееся датой и временем. Учащиеся, могут зарегистрироваться на занятие, если курс открыт для регистрации на занятие. Предполагается, что если устройство учащегося находится зоне регистрации (в видимости устройства преподавателя), то учащийся присутствует на занятии и может отправить запрос на регистрацию. Для преподавателя на странице курса доступна полная посещаемость учащихся, а для учащегося доступна только его посещаемость. Для определения присутствия и идентификации будут использоваться интерактивные устройства, оснащенные беспроводной технологией передачи данных Bluetooth.

Данный сервис состоит из трех основных компонентов: клиента, сервера и базы данных.

Требования:

- Радиус действия сетевой технологии должен быть в пределах 10-15 метров.
- Данный интерфейс должен быть широко распространен и поддерживаться максимальным количеством телекоммуникационных устройств.

### 1. Используемые технологии и средства разработки

При создании программного продукта важно выбрать правильные технологии и средства разработки для эффективного решения поставленной задачи, удовлетворению всех ее требований с учетом имеющихся ограничений. А также руководствоваться принципом высокой универсальности и удобством использования.

Для сервиса учета посещаемости в качестве беспроводной сетевой технологии ближнего действия был выбран протокол передачи данных Bluetooth. Он является оптимальным по радиусу действия, а также поддерживается большинством персональных телекоммуникационных устройств.

Клиентская часть представлена приложением Chrome - веб-приложением, которое работает в веб-браузерах,

основанных на браузере с открытым исходным кодом Chromium. Данное приложение представляет наиболее удобный интерфейс работы с Bluetooth, а также поддерживается более 30 веб-браузерами, среди которых Chrome, Opera и Яндекс.Браузер. Для интерфейса пользователя использовался язык гипертекстовой разметки HTML версии 5.

В качестве основного языка программирования клиентской части выбран язык JavaScript. Он используется для придания интерактивности веб-страницам, реализации логики и предоставления доступа к Bluetooth. Одно из ключевых преимуществ данного языка программирования является переносимость и поддержка практически всеми известными и самыми популярными браузерами. JavaScript отлично работает с другими языками и может использоваться в самых разных приложениях. Он также не нуждается в компиляции на стороне клиента, что дает ему определенные преимущества скорости и снижает нагрузку на сервер. JavaScript постоянно развивается, создаются новые библиотеки и фреймворки, которые упрощают и ускоряют разработку приложения, дают новые возможности для функциональности. Для сетевых приложений, JavaScript имеет удобные интерфейсы, дающие доступ к управлению сетевых модулей и получения от них информации. Для взаимодействия с веб-сервером, он предоставляет простой API для отправки HTTP запросов и получения ответов от сервера. Все это позволяет написать веб-приложение, удовлетворяющее всем требованиям к разработке сервиса.

Для реализации серверной части был выбран язык программирования Python3, предоставляющий удобные интерфейсы разработки сервера и взаимодействия с базами данных.

На стороне сервера используется встраиваемая кроссплатформенная система управления базами данных SQLite, поддерживающая достаточно полный набор команд SQL и доступная в исходных кодах.

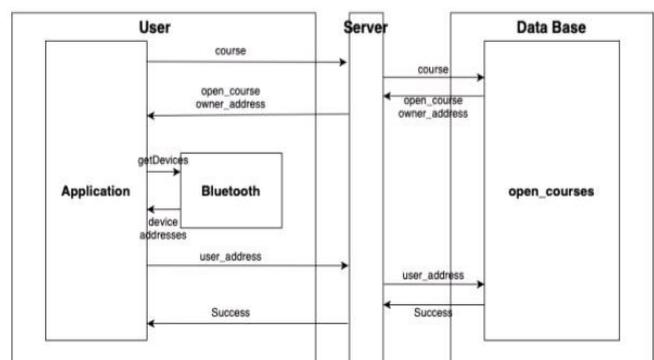


Рис.4 Диаграмма взаимодействия.

### 2. Клиентская часть

В веб-приложении обращение к Bluetooth происходит в двух случаях: при запуске приложения и при регистрации на занятие.

• При запуске приложения определяется адрес устройства пользователя для определения типа стартовой страницы, соответствующей типу пользователя:

```
chrome.bluetooth.getAdapterState(function(result) {
    var device_address = result.address;
    ...
})
```

• При регистрации учащегося на открытое занятие, запускается сканирование устройств, находящихся в зоне видимости Bluetooth. Составляется список найденных устройств. Регистрация отправляется на сервер только в том случае, если устройство владельца данного курса, находится в списке найденных устройств:

```
var device_names = {};
var updateDeviceName = function(device)
{
    device_names[device.address] =
device.id;
};
chrome.bluetooth.onDeviceAdded.addListener(updateDeviceName);
chrome.bluetooth.onDeviceChanged.addListener(updateDeviceName);
chrome.bluetooth.getDevices(function(devices) {
    for (var i = 0; i < devices.length;
i++) {
        updateDeviceName(devices[i]);
    }
});
chrome.bluetooth.startDiscovery(function() {
    setTimeout(function() {

chrome.bluetooth.stopDiscovery(function()
) {});
    var check_course_owner_address =
false;
    for (let key in device_names) {
        if (address == key) {
check_course_owner_address = true;
            break;
        }
    }
    if (check_course_owner_address)
{
        ...
    }
}, 3000);
});
```

Для отправки запросов на веб-сервер использовался XMLHttpRequest — API, доступный JavaScript. Использует запросы HTTP или HTTPS напрямую к веб-серверу и загружает данные ответа сервера напрямую в вызывающий скрипт.

Для приложения использовался протокол HTTPS. Обращение к серверу выглядит следующим образом:

```
httpRequest = new XMLHttpRequest();
const url='https://host:port/urlpath';
httpRequest.open("POST", url);
httpRequest.setRequestHeader('Content-
Type', 'text/plain');
httpRequest.send(request_body);
httpRequest.onreadystatechange =
function() {
    if (this.readyState==4) {
        if (this.statusText=="Success")
{
            ...
        } else {
            ...
        }
    }
}
```

### 3. Сервер

Сервер выполняет роль прослойки между веб-приложением и базой данных, а также выполняет некоторые предварительные вычисления для удобного представления данных.

### 4. Хранилище

В данной базе данных содержатся 3 таблицы - информация о пользователях, информация о курсах и список открытых курсов.

Взаимодействие компонент сервиса при регистрации учащегося на занятие:

Исходный код приложения расположен в git-репозитории:

<https://github.com/VPolka/network-proximity-web-service>

## VI. ЗАКЛЮЧЕНИЕ

В рамках данной работы было проведено исследование возможностей беспроводных сетевых технологий, поддерживающих разработку веб-сервисов на основе концепции сетевой близости, и проведен их сравнительный анализ. Беспроводные сетевые технологии Wi-Fi и Bluetooth являются на сегодняшний день повсеместными и самыми удобными в использовании системами, построенными на принципе сетевой близости.

Был проведен анализ функциональных возможностей стандартизованных методов доступа к данным сетевым модулям. На основе данного анализа была предложена типовая архитектура построения веб-приложения для сервисов, основанных на концепции сетевой близости. Данная архитектура подразумевает использование любых сетевых устройств в качестве маячков, а также в качестве сканеров мобильных телефонов или устройств с модулями беспроводной связи малого радиуса действия и с доступом к Web API.

По данной архитектуре было разработано модельное веб-приложение, построенное на концепции сетевой близости и предназначенное для автоматизации

контроля посещаемости занятий учащимися. На этом примере были продемонстрированы принципы создания приложений данного класса и показаны основные элементы взаимодействия компонентов системы. Данное приложение реализовано на языке JavaScript и является open source проектом. Оно может служить отправной точкой при разработке веб-сервисов, на основе концепции сетевой близости.

В ходе реализации данного программного продукта, веб-интерфейс показал себя удобным и практичным средством разработки. Применение Web API для сервисов, основанных на концепции сетевой близости, позволяет расширить сферу их применения и сократить время разработки.

#### БИБЛИОГРАФИЯ

- [1] World Wide Web consortium. <https://www.w3.org>. [Online; accessed 26-April-2020].
- [2] Namiot D., Sneps-Sneppe M. Geofence and network proximity //Internet of Things, Smart Spaces, and Next Generation Networking. – Springer, Berlin, Heidelberg, 2013. – С. 117-127.
- [3] Namiot D. On indoor positioning //International Journal of Open Information Technologies. – 2015. – Т. 3. – №. 3. – С. 23-26.
- [4] Намиот Д. Е., Шнепс-Шнеппе М. А. КОНТЕКСТНО-ЗАВИСИМЫЕ СЕРВИСЫ НА БАЗЕ БЕСПРОВОДНЫХ ТЕГОВ //Вестник компьютерных и информационных технологий. – 2015. – №. 10. – С. 18-22.
- [5] Намиот Д.Е, Шнепс-Шнеппе М. А. О модели данных для контекстно-зависимых сервисов //Аналитика и управление данными в областях с интенсивным использованием данных: XVII Междунар. конф. DAMID/RCDL. – 2015. – С. 71-77.
- [6] Протокол IEEE 802.11. <http://www.ieee802.org/11/>. [Online; accessed 26-April-2020].
- [7] Протокол IEEE 802.15. <http://www.ieee802.org/15/>. [Online; accessed 26-April-2020].
- [8] WiFi information api. [https://developer.mozilla.org/en-US/docs/Archive/B2G\\_OS/API/WiFi\\_Information\\_API](https://developer.mozilla.org/en-US/docs/Archive/B2G_OS/API/WiFi_Information_API). [Online; accessed 26-April-2020].
- [9] chrome.networking.onc web API. [https://developer.chrome.com/apps/networking\\_onc](https://developer.chrome.com/apps/networking_onc). [Online; accessed 26-April-2020].
- [10] Bluetooth web API. [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Bluetooth\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Bluetooth_API). [Online; accessed 26-April-2020].

# On using web interfaces for network proximity services

Polina Volosnikova, Dmitry Namiot

**Abstract** — This article discusses the use of modern standards of web interfaces for the implementation of service applications based on the concept of network proximity. Network proximity is a property of spatial proximity, which is determined by the visibility of one network device by another. The paper explores the concept of network proximity, analyzes and compares technologies for short-range wireless data transmission. The analysis of software interfaces necessary for the implementation of application services is carried out. Further, the work examines the current state of the software interfaces for working with wireless networks from the JavaScript language, which are implemented by Mozilla Developer and Chromium. The paper describes a model web application for network proximity tasks. In addition to demonstrating elements of working with wireless interfaces from JavaScript, this application also has its own applied aspect - automation of student attendance control.

**Keywords** — Network proximity, Web API, Bluetooth, Wi-Fi.

## REFERENCES

- [1] World Wide Web consortium. <https://www.w3.org>. [Online; accessed 26-April-2020].
- [2] Namiot D., Sneps-Sneppe M. Geofence and network proximity //Internet of Things, Smart Spaces, and Next Generation Networking. – Springer, Berlin, Heidelberg, 2013. – S. 117-127.
- [3] Namiot D. On indoor positioning //International Journal of Open Information Technologies. – 2015. – T. 3. – #. 3. – S. 23-26.
- [4] Namiot D. E., Shneps-Shneppe M. A. KONTEKSTNO-ZAVISIMYE SERVISY NA BAZE BESPROVODNYH TEGOV //Vestnik komp'yuternyh i informacionnyh tehnologij. – 2015. – #. 10. – S. 18-22.
- [5] Namiot D.E, Shneps-Shneppe M. A. O modeli dannyh dlja kontekstno-zavisimyh servisov //Analitika i upravlenie dannyimi v oblastjakh s intensivnym ispol'zovaniem dannyh: XVII Mezhdunar. konf. DAMID/RCDL. – 2015. – S. 71-77.
- [6] Protokol IEEE 802.11. <http://www.ieee802.org/11/>. [Online; accessed 26-April-2020].
- [7] Protokol IEEE 802.15. <http://www.ieee802.org/15/>. [Online; accessed 26-April-2020].
- [8] WiFi information api. [https://developer.mozilla.org/en-US/docs/Archive/B2G\\_OS/API/WiFi\\_Information\\_API](https://developer.mozilla.org/en-US/docs/Archive/B2G_OS/API/WiFi_Information_API). [Online; accessed 26-April-2020].
- [9] chrome.networking.onc web API. [https://developer.chrome.com/apps/networking\\_onc](https://developer.chrome.com/apps/networking_onc). [Online; accessed 26-April-2020].
- [10] Bluetooth web API. [https://developer.mozilla.org/en-US/docs/Web/API/Web\\_Bluetooth\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Web_Bluetooth_API). [Online; accessed 26-April-2020].