ISSN 0798 1015

REVISTA ESPACIOS

Vol. 39 (Number 45) Year 2018. Page 11

# Study of practical implementation of IoMT: creating a web application for distant high-risk group patient’s heart-rate tracking

## Estudio de la aplicación práctica de IoMT: Creación de una aplicación web para el seguimiento distante del ritmo cardíaco de grupos de pacientes de alto riesgo

Tatiana V. SKRYL 1; Vladimir S. OSIPOV 2; Arseny S. PARAMONOV 3

Received: 15/05/2018 • Approved: 01/07/2018

### Contents

**ABSTRACT:**

Background. Over the past decade, modern medicine has risen to an unattainable level. In parallel, one of the main directions in the Internet of things is the development of medicine and public health. In the health care system, the issue of monitoring patients, tracking their location and health status (monitoring temperature, pressure, palpitation and other physical parameters) is especially acute. Due to the development of the Internet of things, the digital transformation in health care is proceeding at a high pace. Objective. Creation of a special web application for distant high-risk group patient’s heart-rate tracking with a fitness bracelet (fitness tracker). Methods. Based on the two existing main approaches to creating a web application, the authors determine the effectiveness of using it in the healthcare system for a high risk group of patients. Results. The results showed that the described method of creating native hybrid applications is the fastest way, perfectly suitable for rapid prototyping and initial launch of the project. However, for a more serious and heavily loaded application, it is worth choosing React Native. Conclusions. This study demonstrates the

**RESUMEN:**

Durante la última década, la medicina moderna se ha elevado a un nivel inalcanzable. En paralelo, una de las direcciones principales en el Internet de las cosas es el desarrollo de la medicina y la salud pública. En el sistema de atención de salud, el problema del monitoreo de pacientes, el seguimiento de su ubicación y estado de salud (monitoreo de temperatura, presión, palpitaciones y otros parámetros físicos) es especialmente importante. Debido al desarrollo del Internet de las cosas, la transformación digital en el cuidado de la salud está avanzando a un ritmo elevado. El Objetivo de este estudio es la creación de una aplicación web especial para el seguimiento del ritmo cardíaco del paciente distante de alto riesgo grupal con un brazalete de ejercicios (rastreador de ejercicios). Sobre la base de los dos enfoques principales existentes para crear una aplicación web, los autores determinan la efectividad de su uso en el sistema de atención médica para un grupo de pacientes de alto riesgo. Los resultados mostraron que el método descrito para crear aplicaciones híbridas nativas es la forma más rápida, perfectamente adecuada para la creación rápida de implementation of IoT and IoMT based on web application. The application of any approach in public health services should be based on the questions posed to the developer and the medical institution.

**Keywords:** Digital transformation, health care, development, IoT, IoMT, technology.

prototipos y el lanzamiento inicial del proyecto. Sin embargo, para una aplicación más seria y más cargada, vale la pena elegir React Native. Este estudio demuestra la implementación de IoT y IoMT en función de la aplicación web. La aplicación de cualquier enfoque en los servicios de salud pública debe basarse en las preguntas planteadas al desarrollador y la institución médica.

**Palabras clave:** La transformación Digital, el cuidado de la salud, el desarrollo, IoT, IoMT, tecnología.

# 1. Introduction

Today, we are witnessing the development of several trends in information technologies that will potentially determine the further vector of the future industry. Firstly, the field of the Internet of things (IoT) is experiencing a real boom of popularity around the world that has been confirmed by a number of studies. For example, in September 2017, the company HewlettPackard Enterprise (HPE) published the results of a survey of a number of IT professionals in 20 countries around the world. According to the report, approximately 57% of the respondent companies have already implemented various IoT technologies in one way or another; at the same time, the overwhelming majority of them (88%) have already estimated the positive economic effect from the introduction [1]. According to the forecast of the IDC consulting company, the progress achieved in the industry is only the beginning of the global digital transformation - according to the results of its research, in 2017 the costs of implementing projects related to IoT devices will grow by 17%, and by 2021 aggregate expenses will reach the level of $ 1.4 trillion [2]. As shown in Figure 1, the application of IoT devices in healthcare is not enough.

**Figure 1**
The main areas of application of IoT devices [3]

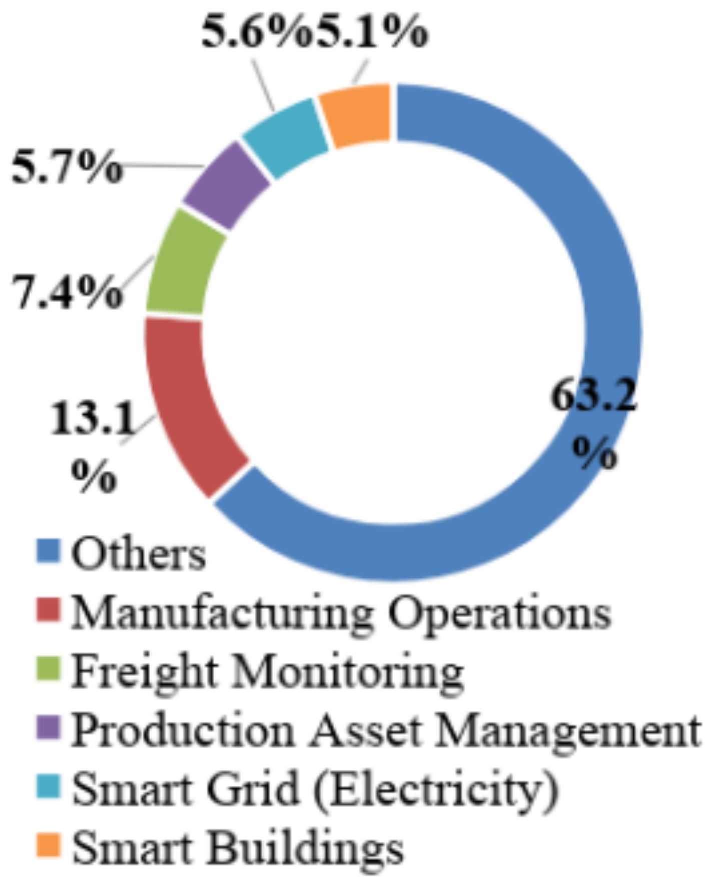
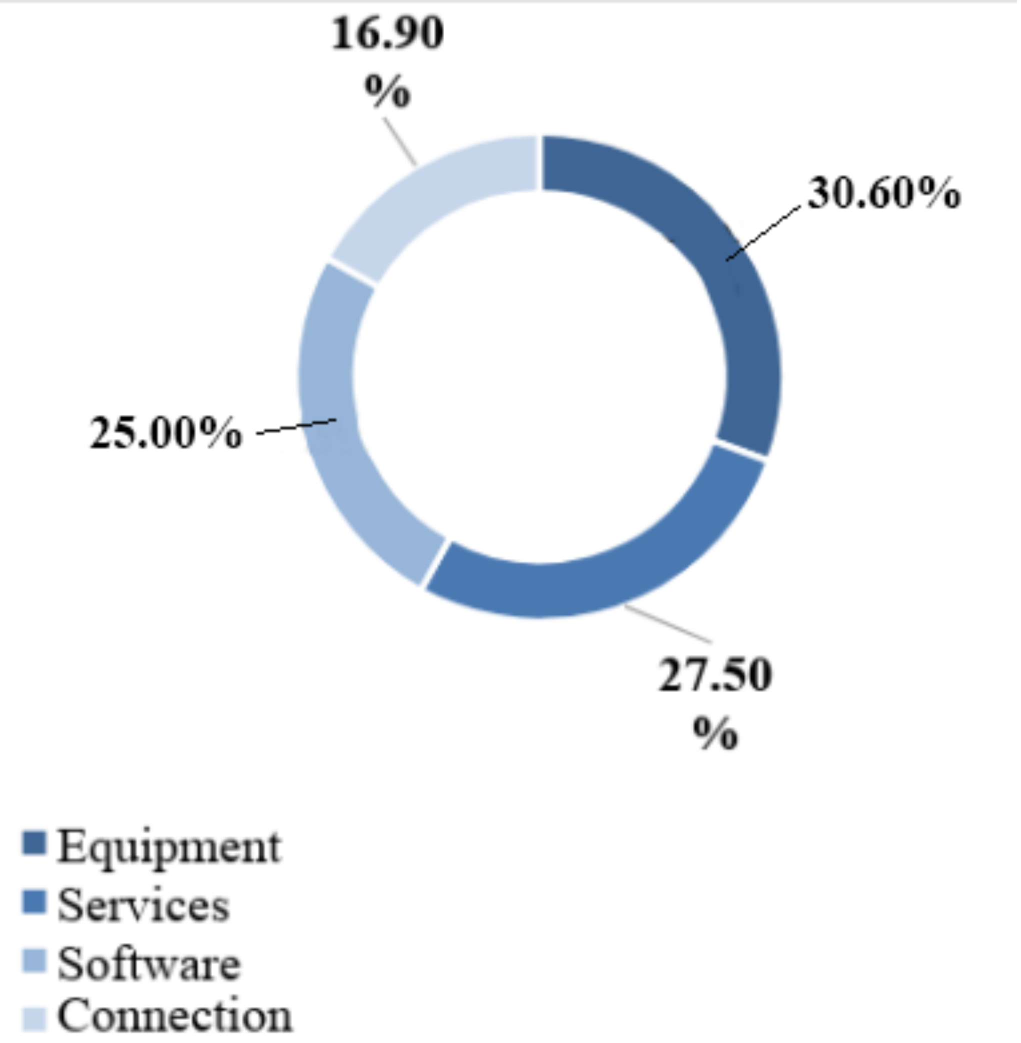


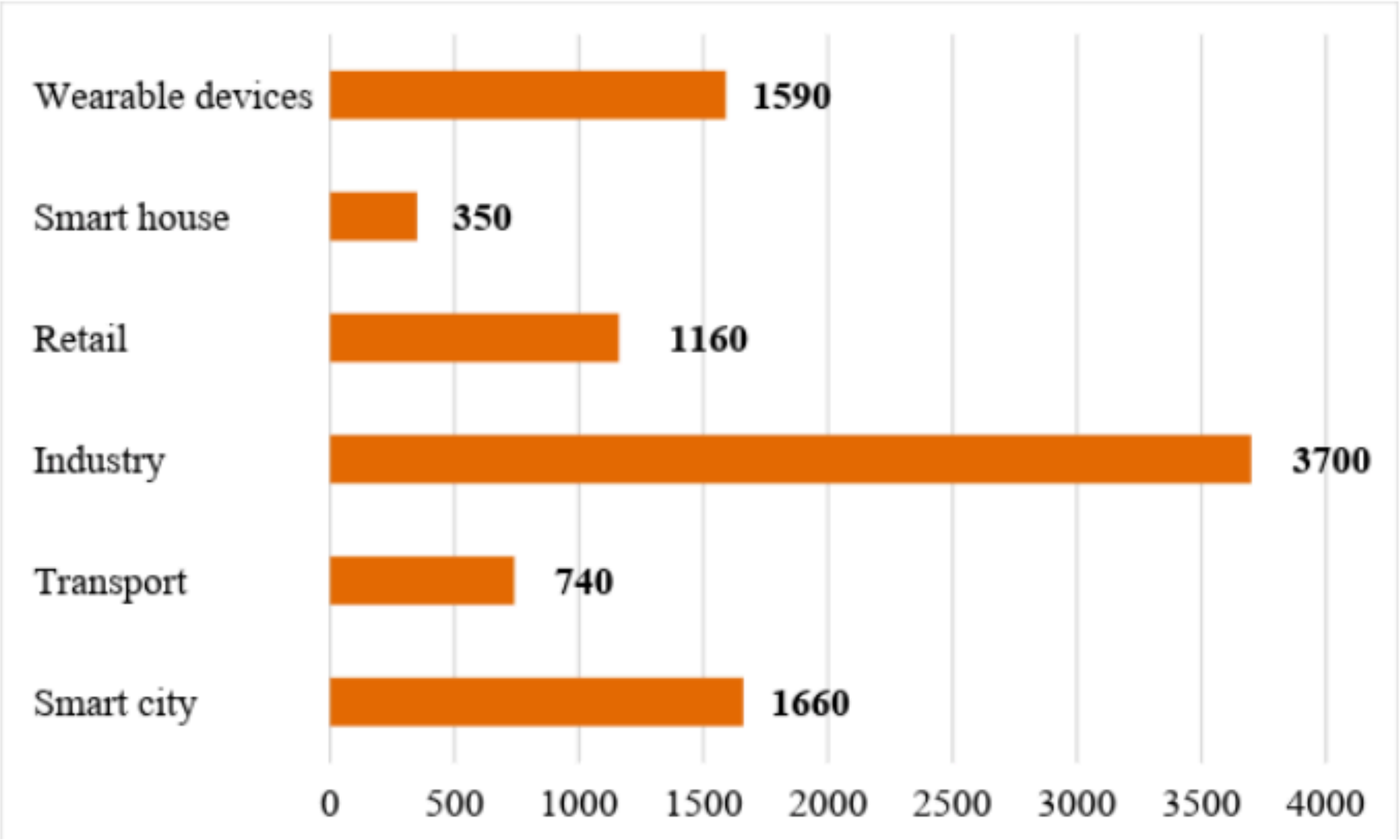
Figure 2 distinguishes the production of equipment, provision of services and software development among the most attractive investment articles.

**Figure 2**
The structure of the world market of Internet of things [3]



According to the data offered in the report of the company McKinsey, the greatest economic value in monetary equivalent will be the introduction of IoT in the industry, but on the third place is different devices that are used to control and monitor the life quality (see Figure 3).

**Figure 3**
The volume of economic value created by IoT in industries by 2025, $ bn [3]



According to Juniper Research estimates, by 2021 the total market volume in numerical terms will grow three times compared to 2016 [4], however, such an avalanche growth in the popularity of the considered solutions inevitably leads to new difficulties, from the degree of successful adaptation to which the success of the industry as a whole depends.

Secondly, solutions related to the rejection of native applications and the orientation towards web applications are gaining immense popularity. An example of this can be the company Patagonia, which completely refused to support the application in favor of optimization of the site [5]. The growth in popularity of Javascript (JS), as a universal development language, contributes to this trend (we can consider this the third of the examined trends). According to the research of Github and Stackoverflow portals, in 2016 this language had already come out on the first place in terms of the number of projects implemented and the created repositories [6-7].

It is necessary to consider the main reasons for such an avalanche growth in the popularity of the programming language, the scope of application of which a few years ago was limited only to the creation of animations and other details on the client. This fact is actually underestimated in the healthcare system due to a digital transformation and the implementation of IoT [8-10].

The main goal of this study is to create the web application that should help the group of high-risk patients to monitor their heart-rate with a fitness bracelet (fitness tracker) and compare its effectiveness in real life.

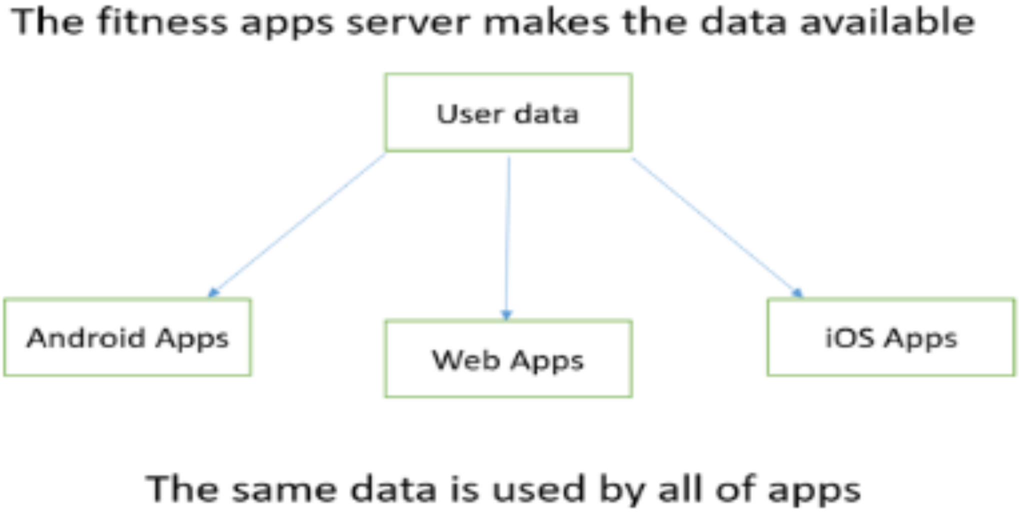
# 2. Methodology

## 2.1. The basic approaches to the creation of web-applications

The community of developers is very practical in approaching the use of a programming language in the development of products and services – the most convenient tool is often used, which fully meets the requirements of the assigned task. Thus, rapid increase in projects written in JS is mainly explained by the emergence of server implementation of language - Node.js, that largely unifies the development process[11]. In general, two mutually complementary approaches to the design of similar applications, which will be considered below, can be distinguished[12-14].

The objective of the study is to develop a web application that can use data obtained from a fitness tracker for remote monitoring of the condition of a patient from a risk group. In general, the schematic diagram of obtaining data is shown in Figure 4:

**Figure 4**
The scheme of obtaining data from the fitness tracker


The fitness apps server makes the data available
User data
Android Apps
Web Apps
iOS Apps
The same data is used by all of apps

The classical approach involves the creation of three different versions of the app for different platforms – one by one for Android, iOS and the web[15]. However, there is something irrational in this, from what everyone is trying to get rid of in the first place in the process of development - the actual repetition of the same code, that violates one of the basic principles of writing quality code - DRY (Don’tRepeatYourself)[16].

# 3. Results and Discussion

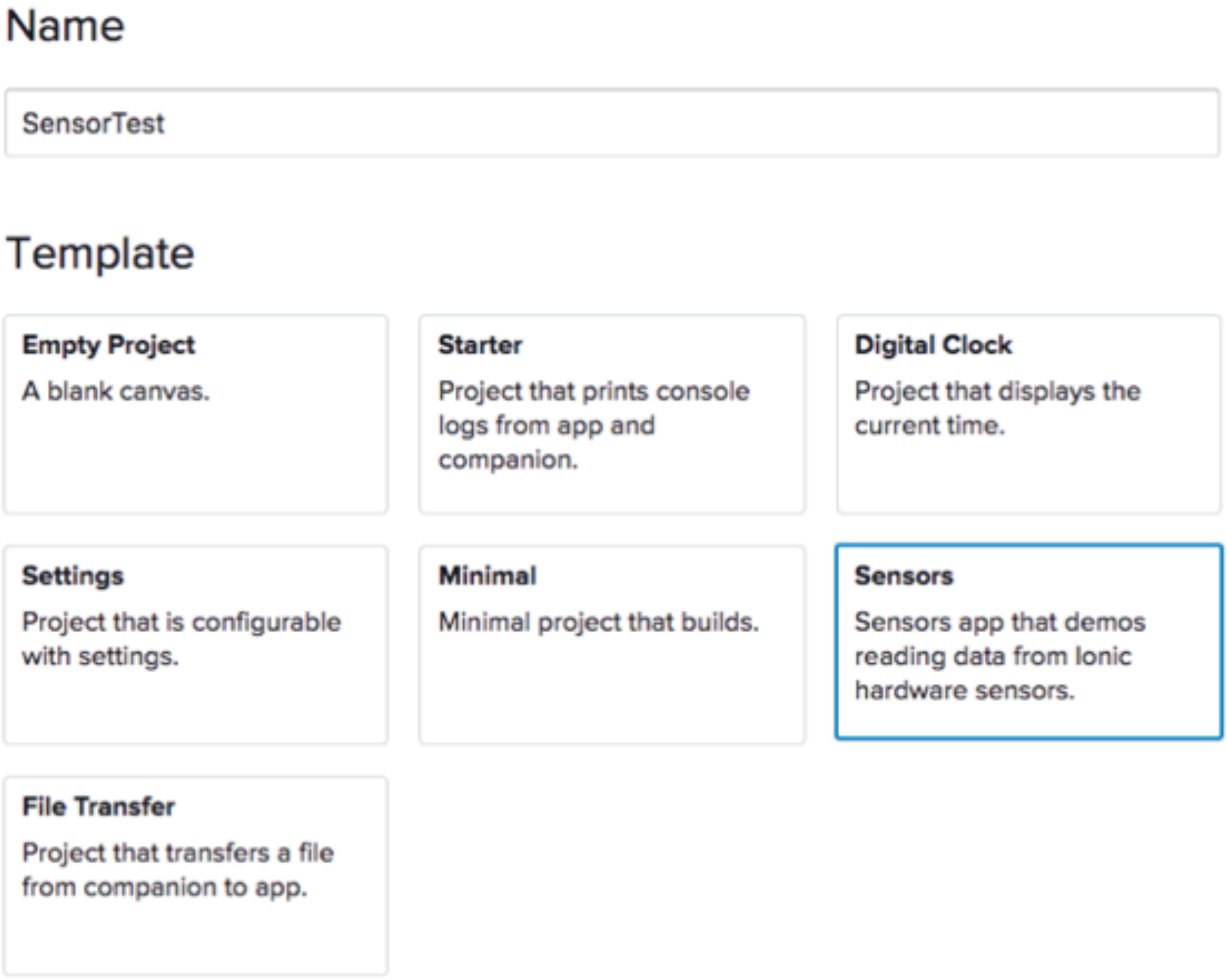
## 3.1. Hybrid application

As an alternative, it is proposed to create a hybrid application that not only deprives us of the need to repeat the same operation three times, but also saves time and money, both necessary for development. It is necessary to start the creation of web application with information about the heart rate from the fitness tracker and gradually turn it into a full-fledged android application.

One of the main problems that are most often encountered when developing software related to devices of the Internet of things is the lack of open API for developers to obtain data[17]. However, there are exceptions to this rule - so, there are solutions both from the giants of the industry, which are a kind of aggregator of medical information (Apple HealthKit, Google Fit), and from private companies that are engaged in creating fitness trackers (FitBit SDK). An example of obtaining data via the FitBit API should be considered, since it is there that everything is implemented in native JS. After registration and authorization on the site FitBit, the FitBit SDK development panel will be entered, where a new project of work with sensors of tracker from the template will be created (please see Figure 5):

**Figure 5**

Creation a new project by template in the FitBit SDK [18]



Name

SensorTest

Template

**Empty Project**
A blank canvas.

**Starter**
Project that prints console logs from app and companion.

**Digital Clock**
Project that displays the current time.

**Settings**
Project that is configurable with settings.

**Minimal**
Minimal project that builds.

**Sensors**
Sensors app that demos reading data from Ionic hardware sensors.

**File Transfer**
Project that transfers a file from companion to app.

Now all necessities to get started and obtain data from the sensors of the device are contained in the file index.js of the template. The following code of output of the current heart rate of owner of the fitness tracker in the console:

**Snippet of code of output of the heart rate in the console**

```
// Import of sensor module of heartbeat
import { HeartRateSensor } from "heart-rate";
// Creation of instance of an object
varhrm = newHeartRateSensor();
// Start of tracking of indicators
hrm.start();
hrm.onreading = functiongetRate() {
varcurrentRate = hrm.heartRate;
  // obtaining of current indicators of heart rate
console.log ("Current heart rate: " + currentRate);
}
// Calling a function getRatec with an interval of 1000 ms
setInterval(getRate, 1000);
```

FitBit API uses the WebSocket API to transfer data between devices, so that our application can easily transfer data on the client. Suppose we already have a simple site, the main function of which is to display the data received from the fitness tracker. How can we turn it into a hybrid web application? First, it is necessary to install Android SDK and create a new project. The basis of all hybrid applications on Android is the WebView element that displays web pages using WebKit, while all our styles and scripts are connected and used as in a normal HTML file. In order to achieve the correct display, we stretch this element overall screen in the layout file:

**Snippet of the layout of the WebView element**

```
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent" >
<WebView
android:id="@+id/WebView"
android:layout_width="fill_parent"
android:layout_height="fill_parent"   />
</LinearLayout>
```

In the file AndroidManifest.xml we will allow access to the Internet:

```
<uses-permission android:name="android.permission.INTERNET" />
```

and add the class code of our activity:

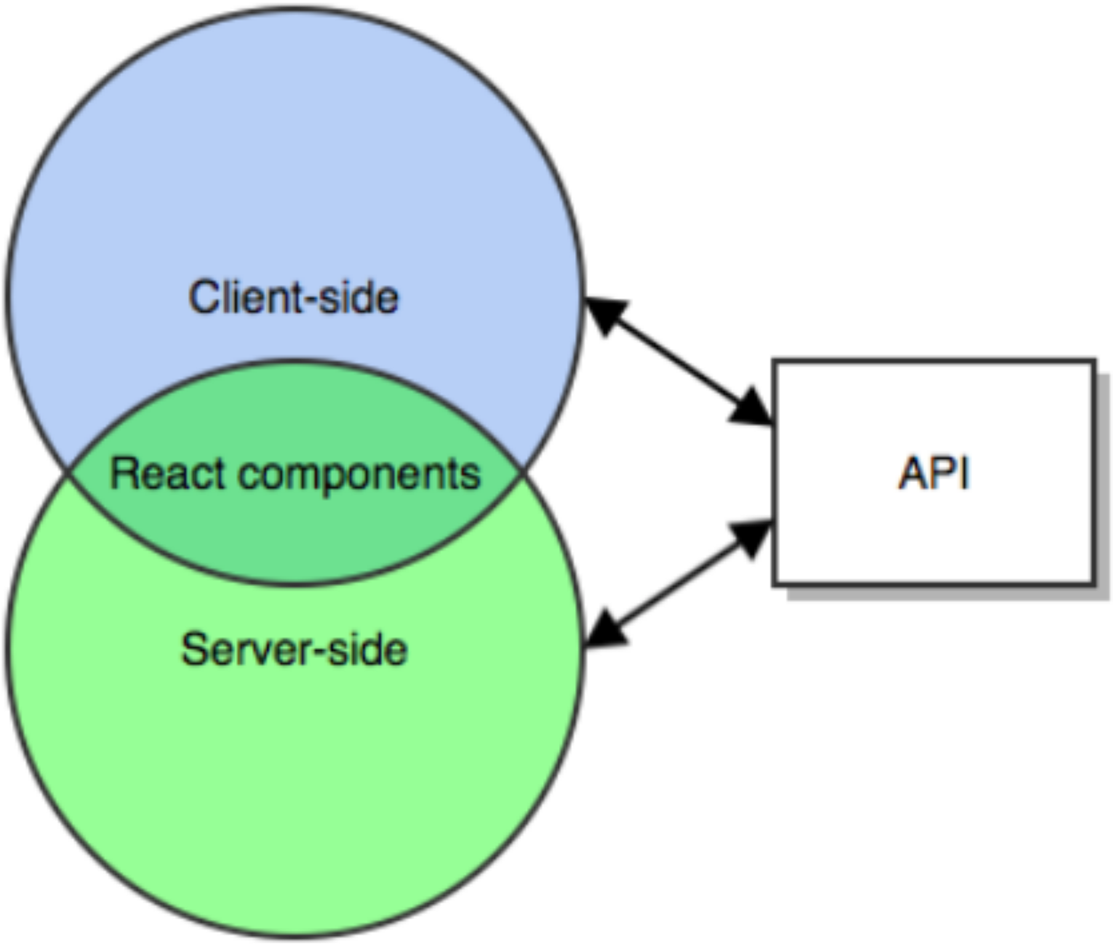
```
packagecom.example.apsoftware;
importandroid.os.Bundle;
importandroid.app.Activity;
importandroid.view.View;
importandroid.view.Window;
importandroid.view.WindowManager;
importandroid.webkit.*;
public class MainActivity extends Activity {
   @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
        /* deploy the application to full screen */
requestWindowFeature(Window.FEATURE_NO_TITLE);
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
/* apply our layout to the current screen */
setContentView(R.layout.activity_main);
/* find the WebView element by its id */
WebViewwebView = (WebView) findViewById(R.id.WebView);
/* create new settings for our WebView element */
WebSettingswebSettings = webView.getSettings();
webSettings.setJavaScriptEnabled(true);
webView.setScrollBarStyle(View.SCROLLBARS_INSIDE_OVERLAY);
/* here we put the URL of site */
webView.loadUrl("http://example.com/");
}
}
```

In fact, the hybrid application is ready - only a few improvements will make it a real tool for monitoring the patient's condition, both by the patient and by the doctor. The universality and reduction of resources that are necessary for launching the project can be called among the advantages of this solution, from minuses - this application will operate slower than native, this should always be remembered when choosing an approach for the implementation of truly high-loaded systems. How can such a solution be improved? Turn it into an isomorphic one, the process of creation of which will be described below.

## 3.2. Isomorphic application

As part of the popularization of JS on the client and on the server, the concept of SPA-application (Single-PageApplication), in which the only HTML document is a shell for all web pages with dynamically loadable styles and scripts, was formed[19]. In fact, it is a native application which is not only for the operating system, but for the browser and which eliminates the classic disadvantages in the form of a long reboot when navigating through the pages and also during the execution of requests to the server. This approach also showed negative aspects - in particular, such sites are extremely poorly indexed by search systems, which is directly related to the order of JS-compilation of content which is not recognizable in results of the search system. However, an appropriate solution was found here too: isomorphic applications in which the code can be simultaneously executed both on the client and server side. The first time the compilation takes place on the server and in the browser, the habitual HTML-document is given, while after the full load other processes function within the framework of the classical SPA-approach. JS libraries and frameworks are actively used to achieve a similar result – for example, ReactJS which is the core of a site like Facebook. Figure 6 shows the scheme of operation of the isomorphic application.

**Figure 6**
The scheme of operation of the isomorphic application [20]



The use of this approach is greatly complicated by the relatively low entry threshold at the start and a rather high entry threshold for creation of more complex multipurpose applications[21]. So, for a quick start for a novice developer who wants to focus on the business logic of the application and not to spend time on the project configuration, it is enough to start using a ready-made template CreateReactApp created by the Facebook[22]. To do this, it is only needed to have a package manager Node – npm and type in the console.

**Snippet of initiation of a new project CreateReactApp:**

```
npm install -g create-react-app
create-react-app my-app
cd my-app/
npmstart
```

Now, we have a ready-made template of the project and can start writing code:

**The structure of the project Create React App:**

```
my-app
├── README.md
├── node_modules
├── package.json
├── .gitignore
├── public
│   └── favicon.ico
│   └── index.html
│   └── manifest.json
└── src
    └── App.css
    └── App.js
    └── App.test.js
    └── index.css
└── index.js
    └── logo.svg
    └── registerServiceWorker.js
```

So, suppose that the main code and business logic of our pulse tracking application with the use of FitBit API were described in the file App.js of our project. At this stage, we clearly share client and server’s rendering and have two independent entry points – client.js and server.js. In the file client.js, we render our component in a common container of the HTML document that we create:

**Snippet of clients’ rendering**

```
            import React      from 'react';
importReactDOM   from 'react-dom';
import App        from 'components/App';
ReactDOM.render(<App />, document.getElementById('react-view'));
```

For the operation of the server part, it is necessary to install a framework Express that greatly facilitates many processes of      development.

After that our file server.js will have the following form:

**Snippet of the file server.js**

```
            //Import of all necessary dependencies
importexpressfrom 'express';
import React    from 'react';
importReactDom from 'react-dom/server';
import App      from 'components/App';
const app = express();
// Rendering of the component by template of the function render()
app.use((req, res) => {
const component = ReactDom.renderToString(<App />);
returnres.end(render(component));
});
            //Template of the component with the container react-view
function render(component) {
return `
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Hello</title>
<link rel="stylesheet" href="myapp/public/assets/styles.css">
</head>
<body>
<div id="react-view">${component}</div>
<script type="application/javascript" src="myapp/public/assets/bundle.js"></script>
</body>
</html>
  `;
}
const PORT = process.env.PORT || 3001;
//Initialization of the server
app.listen(PORT, () => {
console.log(`Server listening on: ${PORT}`);
});
```

The optimization of the site as well as the faster response and overall increase in reliability in case of errors on the server or client will be the result of execution of this code. The general scheme is as follows: The server almost instantly gives the main content while the client code is loaded at the same time. If this does not happen, a normal request to the server is executed. This approach allows to reduce the waiting time by about two seconds, depending on the degree of preliminary minification and optimization. That has a very positive effect on the conversion of commercial sites, as well as automatically improves the chances of a higher place in the results of search systems[23].

It is worth mentioning the fact that now the platform React Native, focused on the development of mobile applications, has been widely developed[24]. Now the technology is not mature and only passes the initial stage of its formation but huge potential is already noticeable - web applications that are written on ReactJS can easily be turned into a mobile cross-platform application, however, there are both advantages and disadvantages in comparison with the hybrid approach described above[25]. An obvious advantage could be the optimized productivity on any platform, while a disadvantage could be the fact that the code will have to be rewritten and modified, since the differences between the ReactJS and the ReactNative are quite large. Thus, the method of creation of native hybrid applications which is described above is the fastest way, excellent for rapid prototyping and initial launch of the project. However, for a more serious and high-loaded application, you should choose ReactNative, as a conceptually more correct paradigm[26].

# 4. Conclusions

During the study, we examined the main approaches of the creation of modern web-based applications which monitor heart rate that work correctly not only on the web, but also as native mobile applications. During the study, the following approaches were highlighted: a classic hybrid application and an isomorphic application ReactJS that work in conjunction with NodeJS and ReactNative. It should be remembered that the application of any of these approaches in real life should be based on the tasks assigned to the developer, hospital and medical company. If the early release of a web application to the market with simultaneous capture of a niche in the area of mobile applications is seen as a priority, then the usual hybrid application seems to be the most optimal choice, since in any case possible problems with productivity will be resolved with the release of the next version (if interest among users will be marked). It should be assumed that this method is also suitable for rapid prototyping. In the case of the initially more serious goals of the creation of high-loaded complex application, attention should be paid to isomorphic applications, supplemented by a separate version of ReactNative, that will significantly improve the fault tolerance, speed of operation and audience coverage. It is already possible to firmly and confidently say that the number of developments in the field of the Internet of things and Internet of medical things will grow every year. This makes the above-described technologies, which allow to work with similar projects on a completely different level, extremely relevant.

## Abbreviations

IoT: Internet of things

IoMT: Internet of medical things

HPE: HewlettPackardEnterprise

JS: Javascript

SPA-application: Single-PageApplication

API: Application programming interface

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

# Bibliographic references

[1] HPE The Internet of Things: Today and Tomorrow, 2017, access mode: http://www.arubanetworks.com/assets/eo/HPE_Aruba_IoT_Research_Report.pdf

[2] IDC, Worldwide Internet of Things Software Platform Forecast, 2017-2021, 2017, access mode: https://www.idc.com/getdoc.jsp?containerId=US42035617

[3] IDC, Worldwide Semiannual Internet of Things Spending Guide, 2016, access mode: https://www.idc.com/getdoc.jsp?containerId=IDC_P29475

[4] Internet of Things (IoT) Market: Global Demand, Growth Analysis & Opportunity Outlook 2023, access mode: https://www.researchnester.com/reports/internet-of-things-iot-market-global-demand-growth-analysis-opportunity-outlook-2023/216

[5] Patagonia shuts down its native app, 2016, access mode: https://medium.com/stories-behind-the-screens/patagonia-shuts-down-its-native-apps-fe8156d500fe

[6] Github 15 most popular languages used on Github by opened Pull Request and percentage change from previous period, 2016, access mode: https://octoverse.github.com/

[7] Popular programs technologies of 2017 – study of Stack Overflow, 2017, access mode: https://stackoverflow.com/

[8] S.G. Evsukov, A.V. Sigarev, E.V. Ustyuzhanina, E.V. Zaytseva "Comparative analysis of pricing policies in the market for network goods", Journal of Internet Banking and Commerce, Volume 21, Issue Special Issue 6, p. 17, 2016

[9] Juan Pablo Sáenz, Mónica Paola Novoa, Darío Correal, Bell Raj Eapen "On Using a Mobile Application to Support Teledermatology: A Case Study in an Underprivileged Area in Colombia", International Journal of Telemedicine and Applications, Volume 2018, Article ID 1496941, 2018 https://doi.org/10.1155/2018/1496941

[10]V.S. Osipov, T.V. Skryl, V.O. Evseev "An analysis of economic issues of territories of priority development", Research Journal of Applied Sciences. Volume 11, Issue 9, Pages 833-842, 2016, DOI: 10.3923/rjasci.2016.833.842

[11] R. J. Holden and B.-T. Karsh, "The Technology Acceptance Model: Its past and its future in health care," Journal of Biomedical Informatics, vol. 43, no. 1, pp. 159–172, 2010

[12] Rehan Zaidi "JavaScript Essentials for SAP ABAP Developers: A Guide to Mobile and Desktop Application Development". Apress. 2017

[13] Rick Sturm, Carol Pollard and Julie Craig "Application Performance Management (APM) in the Digital Enterprise. Managing Applications for Cloud, Mobile, Io: T and e: Business" [1st Edition], Morgan Kaufmann. 2017

[14] Kerri Shotts "Mastering PhoneGap Mobile Application Development", Packt Publishing Limited. 2016

[15] Daniel Churchill, Jie Lu, Thomas K.F. Chiu, Bob Fox (eds.) "Lecture Notes in Educational Technology Mobile Learning Design: Theories and Application" [1 ed.], Springer Singapore. 2016

[16] Sougata Mukherjea "Advances in Multimedia and Interactive Technologies Mobile Application Development, Usability, and Security" [1 ed.], IGI Global. 2016

[17] S. S. Niranga “Mobile Web Performance Optimization: Deliver a better mobile user experience by improving and optimizing your website – follow these practical steps for cutting-edge application development”, Packt Publishing. 2015

[18] Dominic Chell “The Mobile Application Hacker’s Handbook” [1 ed.], Wiley, Handbook. 2015

[19] Hazem Saleh “Javascript Mobile Application Development”, Packt Publishing. 2014

[20] Richard Rodger “Beginning building mobile application development in the cloud”, Wiley. 2012

[21] D. Hailey, A. Ohinmaa, R. Roine “Study quality and evidence of benefit in recent assessments of telemedicine”, Journal of Telemedicine and Telecare, Issue 10, Pages 318–324, 2004

[22] M.W. Tsang, M. Mok, G. Kam, et al. “Improvement in diabetes control with a monitoring system based on a hand-held, touch-screen electronic diary”, Journal of Telemedicine and Telecare, Issue 7, Pages 47-50, 2001

[23] Todd Molfenter, Roger Brown, Andrew O’Neill, Ed Kopetsky, Alexander Toy “Use of Telemedicine in Addiction Treatment: Current Practices and Organizational Implementation Characteristics”, International Journal of Telemedicine and Applications, Volume 2018, Article ID 3932643, 7 pages, 2018 https://doi.org/10.1155/2018/3932643

[24] G. Phillips, L. Felix, L. Galli, V. Patel, P. Edwards “The effectiveness of mhealth technologies for improving health and health services: a systematic review protocol”, BMC Res Notes 3(1), p. 250, 2010

[25] Emily Wilson, Lois A. Park, Scott Zeger, Timothy Roberton “Reimagining statistical analysis for evidenced-based policy making: Early experiences using Stats Report”, Journal of Global Health. Issue 2, Article ID 020306, 5 pages, 2017

[26] C.E. King, M. Sarrafzadeh “A Survey of Smartwatches in Remote Health Monitoring”, Journal of Healthcare Informatics Research, 2017 https://doi.org/10.1007/s41666-017-0012-7

1. Department of Economic Theory, Plekhanov Russian University of Economics, Moscow 117997, Russia

2. Department of Risk Management, MGIMO University, Moscow 119876, Russia

3. Faculty of Mathematical Economics, Statistics and Informatics, Plekhanov Russian University of Economics, Moscow 117997, Russia