UDC 004.021; 004.514; 004.55

ELECTRONIC GUIDE FOR PEOPLE WITH DISABILITIES IN NOVOPOLOTSK

A. SOLOTSKY
Polotsk State University, Belarus

The paper presents the development of a web resource with a map and useful information about objects of urban infrastructure for people with disabilities on it. The article discusses the importance of the problem. The analysis of technologies and approaches used for development is provided.

People with disabilities need constant support from other members of society and the state. In the Republic of Belarus today, there are more than half a million disabled people [1]. Modern information technologies can make life easier for such people. The availability of portable, high-performance mobile devices and the reduction in the cost of communication services and Internet access, which is necessary to keep the information up to date, also help make it available to them.

More than two hundred wheelchair users live in Novopolotsk, many of whom cannot lead an active lifestyle due to the lack of information about unhindered access to the city's social infrastructure. Therefore, the development of an electronic guide for people with disabilities in the city of Novopolotsk is an urgent and important task.

Designing a software product of this kind can be divided into several stages. It is necessary to develop a subsystem for storing information about city objects, an administrative interface for managing this information, a user interface for displaying this information to end users in a convenient form, as well as an intermediate infrastructure that allows all components of the system to cooperate.

The relational database management system (DBMS) MySQL 8.0 was chosen as the data warehouse. This DBMS is perfect for working with not very large amounts of data that do not have a complex structure (for example, the structure of the developed database, see Figure 1), providing high performance combined with acceptable resource consumption, sufficient reliability and flexibility. An important fact is its maturity (development has been under way for more than 20 years) and free distribution (distributed under the GNU Public License).

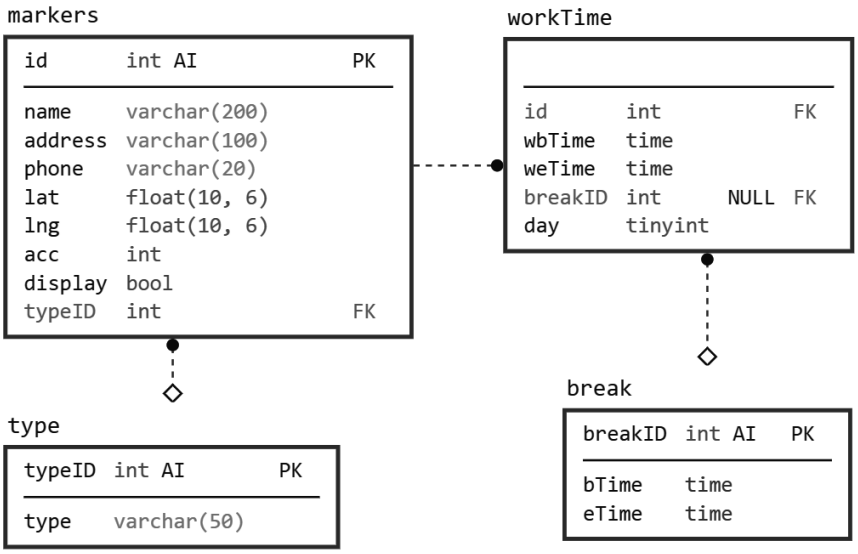

Figure 1. – Database structure

Another open technology which interacts perfectly with MySQL is PHP (PHP: Hypertext Preprocessor). This language was chosen to perform the connecting functions: providing a secure interface for updating information in the database and generating an intermediate representation of data. PHP is very handy for quick and efficient solving problems in web development.

The REST approach (short for Representational State Transfer) was chosen for the overall application architecture. This style of interaction between application components imposes some restrictions on the system, which provide increased productivity, reliability, transparency of interactions, simplification of architecture and ensuring ease of changes and evolution, such as:

– client-server model, including the separation of the database from the client and independent determination by the client how to present data received from the server;

– no client context kept on the server – all the data necessary for an accurate response to the client's request is provided by them;

– caching, which allows in many cases not to process the same request again and again, even if it was received from another client, and immediately return the response;

– uniform interface, which allows independent upgrade the internal implementation and design of components; and also a system of client-server interaction layers to simplify caching and distribution of requests [2].

As a result, the system consists of four components. The administration subsystem is responsible for operations on markers. It interacts with the database and also generates an intermediate JSON file, accessible to any Internet user. In this case, it can be considered as a kind of cache. Using the data from this file, the user subsystem displays markers on the map and information about them and allows filtering. As an important feature, after receiving this information, there is no need to interact with the server, and after the map is loaded, access to the Internet is not needed at all. The page can work completely autonomously, saving user traffic and server resources.

Nginx or Apache HTTP Server can be used as a web server, but the second is preferable, since it does not require an additional layer to generate dynamic content, and the target audience of the developed application is not so extensive as to necessitate the separation of the server system into independent modules provided by the first application.

Thus, all the technologies mentioned are widespread, time-tested and well-described, which simplifies the deployment of the system in any environment. If necessary, adaptation or scaling of the system can be carried out in a short time.

JetBrains PhpStorm was used to develop the application. This is due to the support of all the technologies used: SQL, PHP, HTML, CSS, JavaScript for the development of the user part; the presence of powerful tools for auto-generation, refactoring and auto-completion; deep integration with version control systems in one appication.

The development of the interface of such a system is also a non-trivial task. It is necessary that the content is correctly displayed on various devices. This includes not only the configuration of the blocks on the screen, but also the adjustment of the text size, the location and size of the images, and sometimes the color scheme. In this case, it is desirable to obtain a high-quality and easy-to-support product in the shortest possible time. In this regard, adaptive layout is an effective approach that solves many problems at once.

Responsive web design is an approach to page layout development in which the interface is correctly displayed on devices with different orientations, screen size and clarity, browser window size and, usually, is able to dynamically adjust in case of changes.

The current approach to adaptive interface development has emerged as a result of the interaction of several development strategies:

– Progressive improvement. The strategy provides for the gradual improvement and expansion of the user interface through increasingly sophisticated and modern CSS styles and JavaScript scripts for devices that support them, but continue to provide basic functionality to less modern customers. It was represented by Stephen Champion at the SXSW conference in 2003.

– Responsive design. This includes rubber blocks, stretched images, and the widespread use of CSS media queries. The concept was first introduced by Ethan Marcott in 2010 [3]. Later on, Jeffrey Zeldman proposed to expand this concept for any techniques that allow to achieve the correct display of interfaces regardless of the characteristics of the screen [4].

– Mobile First. A strategy that requires first developing a mobile version of a resource with subsequent adaptation to other devices. Described by Luke Wroblewski in 2011 [5].

It is also important to note Aaron Gustavson, who described a strategy for the simultaneous use of the approaches described above [6].

There are various adaptation methods. It is possible to divide devices into subgroups and develop layout for each separately, from scratch. However, this significantly increases the complexity of developing with a single style, very time-consuming, and is also negatively perceived by search engines in case of placing different versions on different subdomains.

Implementation according to the original plan of Ethan Marcott also has its drawbacks. Often there are difficulties with images and there is free space in the area of the side panels as you scroll the page.

Techniques to load different resource sets depending on the characteristics of the client have been developed (for example, Responsive-Images by filamentgroup or using User-Agent to determine the type of device), but this cannot be considered as a standalone solution.

Rubber layout is implemented by changing the layout of the markup tapes (for example, the Bootstrap framework). Block transfer involves changing the layout structure when changing the screen width. Hiding and panels provides the ability to display only part of the default interface for some devices, but with access to them, for example, by pressing buttons.

In addition to the already mentioned option with the generation of HTML markup on the server, taking into account the characteristics of the user's device, it is possible to use additional files of CSS styles and media queries, as well as JavaScript.

Media queries (@media), as well as support checks (@supports), allow you to apply styles only for types of devices with certain characteristics and support for these specific styles. For example, you can easily develop a printable version of a page or with some aspect ratio (see Figure 2).

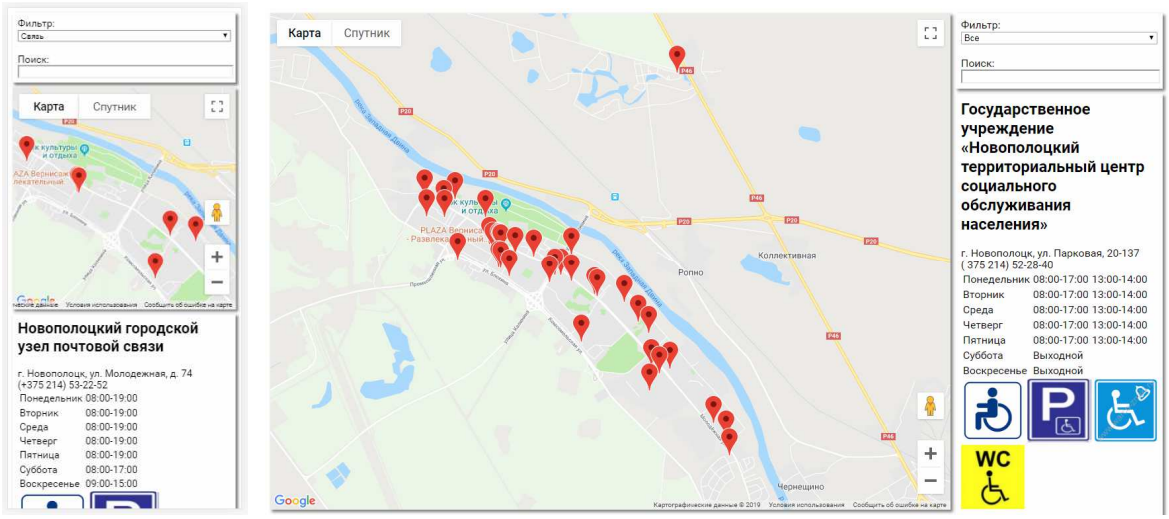

Figure 2. – Adaptive interface of the same page on a mobile device and a personal computer

It is possible to change and create various layout schemes using tables, blocks and more modern flexbox and grids. The last two approaches allow you to create quite complex one- and two-dimensional structures, respectively, using a wide range of restrictions and settings of different types at the same time.

JavaScript provides complete control over the page through the document object model. This allows you to load additional style files, apply styles to elements directly, implement complex reactions that are not provided by HTML and CSS by default, for example, clicking on objects. With its help, it is also possible to realize reactivity, for example, on gradual input or scrolling of a page, various related interactions.

An important parameter in the implementation of adaptability, including for people with disabilities, is the ability of the page to work with a non-standard font size, which can be set through the browser settings. To achieve this, it is important that the vast majority of page element sizes are set not in absolute units (pixels and their derivatives), but in relative: percentages and percentages relative to window size, em and rem (based on font size). Also, the last units allow you to easily and in one place change the size of some nested elements directly on the page, changing the font size of the root element.

The combination of JavaScript and CSS may allow different color schemes for people with visual impairments. It is also important to specify the alt attribute on images, providing a textual description for them. It can also be useful for users with low access speeds or with disabled images. The description should be meaningful, clear, capacious, in the presence of the result of a click - describe it. It's important to use valid HTML tags, such as a hierarchy of headings, as programs for blind rely heavily on them and this speeds up page navigation.

Optimization for mobile devices also includes setting the correct types of input fields, disabling autocorrection for some of them, increasing the size of active interface elements and supporting interaction with

them through touches, the widespread use of autocomplete, refusal to track guidance (replacing it with touches).

The developed application meets all the requirements of the subject area, the tables of the created database meet the requirements of normalization, which ensures the integrity and consistency of information. The most suitable and modern programming languages and development environments, key technologies and approaches for creating adaptive interfaces, including those adapted for use by people with disabilities, were used to create the application.

REFERENCES

1. Численность инвалидов, получающих пенсию в органах по труду и социальной защите | Министерство труда и социальной защиты Республики Беларусь. Официальный сайт [Электронный ресурс]. – Режим доступа: http://www.mintrud.gov.by/ru/chislinv. – Дата доступа: 09.09.2019.
2. Fielding Dissertation: CHAPTER 5: Representational State Transfer (REST) [Электронный ресурс]. – Режим доступа: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm. – Дата доступа: 11.09.2019.
3. Responsive Web Design – A List Apart [Электронный ресурс]. – Режим доступа: https://alistapart.com/article/responsive-web-design/ – Дата доступа: 02.09.2019.
4. Responsive Design. I don't think that word means what you think it means. | Zeldman on Web & Interaction Design [Электронный ресурс]. – Режим доступа: http://www.zeldman.com/2011/07/06/responsive-design-i-dont-think-that-word-means-what-you-think-it-means/ – Дата доступа: 02.09.2019.
5. Вроблевски Л. Сначала мобильные! – М.: Манн, Иванов и Фербер, 2012. – 176 с.
6. Gustafson A. Adaptive Web Design: Crafting Rich Experiences with Progressive Enhancement. – Easy Readers, 2011. – 114 с.