Information System for Finding Inclusive Places for People with Disabilities

Vasyl Andrunyk and Dmytro Shostak

Lviv Polytechnic National University, S. Bandera Street, 12, Lviv, 79013, Ukraine

Abstract

As a result of work on this paper, an information system for finding locations suitable for people with disabilities is created in the form of browser and mobile versions of the application, which allows you to search for such sites and access information about places.

This work describes in detail the process and result of creating a backend of the resource, its user interface and the principles of interaction of system elements with each other and characterizes the difference between a browser and mobile versions of the service. Also, the existing analogues of services with similar goals and targets were analyzed, their advantages and disadvantages were identified, and the results of this analysis were taken into account before the development phase. The frontend part of the information system is implemented using the usual web development tools such as HTML / CSS and Javascript. Specific technologies used to create the user interface are React and jQuery. The server part of the application is developed using Node.js technology. The most important factors that were taken into account when designing and developing the information system were flexibility, speed and stability, the overall efficiency and relevance of the technologies used, and the prospect of long-term product support that does not require much resources and effort. Regarding the user interface configuration, the main conditions for development were its accessibility and convenience in providing users with uninterrupted access to all of the systems' functions.

Keywords 1

Information system, user interface, software product, mobile version, frequently asked question, support team, web application, detailed location information, web page, mobile application, web version, inclusive place, Ukrainian user, people with disabilities

CEUR-WS.org/Vol-2917/paper37.pdf

1. Introduction

In Ukraine, there is a pretty numerous category of citizens in need of special conditions for everyday life, such as people with disabilities. According to official data, this population category includes about 2 million 600 thousand people [1], which is about 6% of the total population. Among them are more than 240 thousand people with disabilities of the 1st group, more than 900 thousand people with disabilities of the 2nd group, and more than 1.3 million people of the 3rd group. These data indicate the presence in Ukraine of a relatively large group of people for whom the developed information system will be relevant and valuable. This work aims to develop an information system designed for use by people with special needs and aimed in its main features to improve access for users to relevant information about the surrounding infrastructure and their state of adaptation to the unique needs of people in wheelchairs. The task set before the developer of the information system was to create, design, and maintain a software product in the form of browser and mobile versions of the application to find and obtain information about the suitability of places to visit people with disabilities.

The main idea of the created information system is to provide users with access to information about the suitability of various locations and infrastructure elements to provide services to visitors with special

MoMLeT+DS 2021: 3rd International Workshop on Modern Machine Learning Technologies and Data Science, June 5, 2021, Lviv-Shatsk, Ukraine

EMAIL: vasyl.a.andrunyk@lpnu.ua (V. Andrunyk); dmytroshostak@gmail.com (D. Shostak)
ORCID: 0000-0003-0697-7384 (V. Andrunyk); 0000-0002-7438-1382 (D. Shostak)

© 2021 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

needs. With the help of the application, users will conveniently and quickly access up-to-date and verified information presented in an organized form through the user interface of the mobile or browser version of the application.

The current level of development of most private and public institutions and infrastructure elements of Ukrainian settlements does not sufficiently or not meet the needs of citizens with special needs. It is due to both the obsolescence of the infrastructure in many cases and the disregard for the needs of people in wheelchairs when building new elements or renovating existing locations.

Despite the overall positive dynamics in this direction, the problem is still very acute. Efforts by international, governmental, and volunteer organizations to raise awareness of people with disabilities in Ukraine to improve their living conditions and function as full members of society are currently insufficient, as are the resources allocated to address the problem.

At the same time, the spread of information technology, its more profound penetration into citizens' daily lives, and the gradual decline in the cost of electronic devices and software for them make access to such technologies increasingly accessible to all segments of the population.

Therefore, in the future, such software solutions as what is done in this work will become more popular over time and, accordingly, will help more people in need.

Despite attempts by various teams and companies to implement the idea of a service for finding inclusive locations designed to facilitate travel and visits to places by citizens with special needs, after analyzing existing analogues, it was recognized that none of them could provide practical activities based on local characteristics. also make the service fully accessible to users. During the design of the information system, opportunities were laid to expand the service's coverage of a more extensive list of locations that are constantly growing, as well as to add new features to the software product. Since the primary purpose of the information system is to provide users with access to information on the inclusiveness of various locations, it can be considered that the object of study is information on the inclusiveness of places, ways to identify and present them using information tools and tools. The subject of the study, respectively, is the analysis, processing of data on the inclusiveness of locations and their presentation to the user. To implement such a system, it was necessary to identify and analyze in detail factors such as some aspects of infrastructure in institutions, such as ramps or specially equipped toilets or specific characteristics of the sites themselves. Then, you determine the level of influence of these various factors and their combinations, which may have different importance for different categories and types of locations, institutions' suitability, and the fair reflection of such places.

2. Related works

2.1. Analysis of existing products, description of their functionality

Before implementing the software product, an analysis of the existing developed information systems for finding inclusive places for people with special needs and strategies partially solves this problem. However, a significant issue was also that in the domestic market of information systems, systems sharpened for this problem are not, or they are not very popular. Therefore, the selected methods for analysis list the main functions that solve similar problems with those that formed the basis of product creation while working on this work and future improvement.

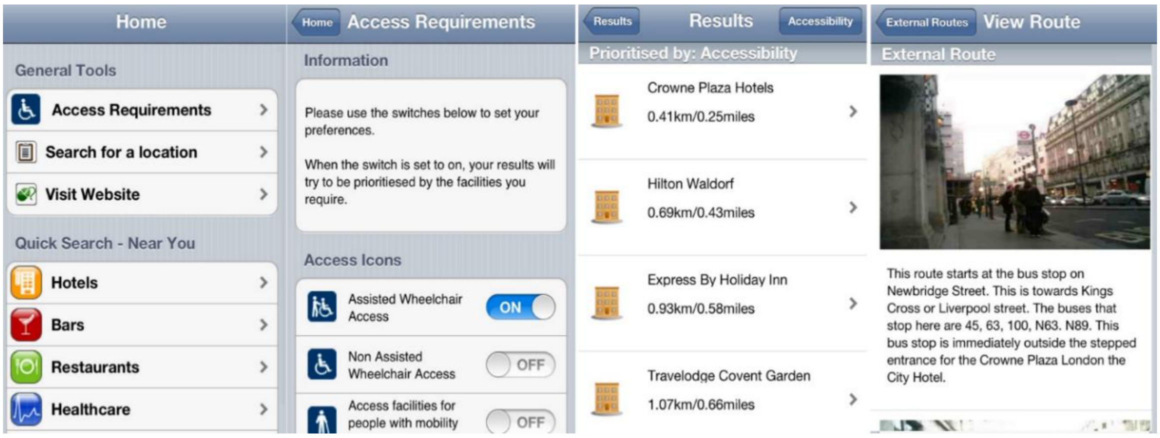
Table 1 lists the applications found that are designed to help people with disabilities in their daily lives. In most of them, the aspect is the comfortable movement of people with disabilities.

Table 1
List of information systems selected for analysis

The name of the application	Country developer	Availability today
Inclusive Britain	Great Britain	–
Wheelmate	Germany	+
AccessAble	Great Britain	+
Handiplanet	France	+
Wheelmap	Germany	+

2.1.1. Inclusive Britain

This service was developed for people with inclusive needs by developers from the UK. An important feature is that it was one of the first services of this kind. Also, it was an innovative development that contained the best-designed resource among such applications at the time of its creation. Fig. 1 shows the interface of this application.



Home

General Tools

Access Requirements

Search for a location

Visit Website

Quick Search - Near You

Hotels

Bars

Restaurants

Healthcare

Home Access Requirements

Information

Please use the switches below to set your preferences.

When the switch is set to on, your results will try to be prioritiesed by the facilities you require.

Access Icons

Assisted Wheelchair Access ON

Non Assisted Wheelchair Access OFF

Access facilities for people with mobility OFF

Results Results Accessibility

Prioritised by: Accessibility

Crowne Plaza Hotels 0.41km/0.25miles

Hilton Waldorf 0.69km/0.43miles

Express By Holiday Inn 0.93km/0.58miles

Travelodge Covent Garden 1.07km/0.66miles

External Routes View Route

External Route

This route starts at the bus stop on Newbridge Street. This is towards Kings Cross or Liverpool street. The buses that stop here are 45, 63, 100, N63. N89. This bus stop is immediately outside the stepped entrance for the Crowne Plaza London the City Hotel.

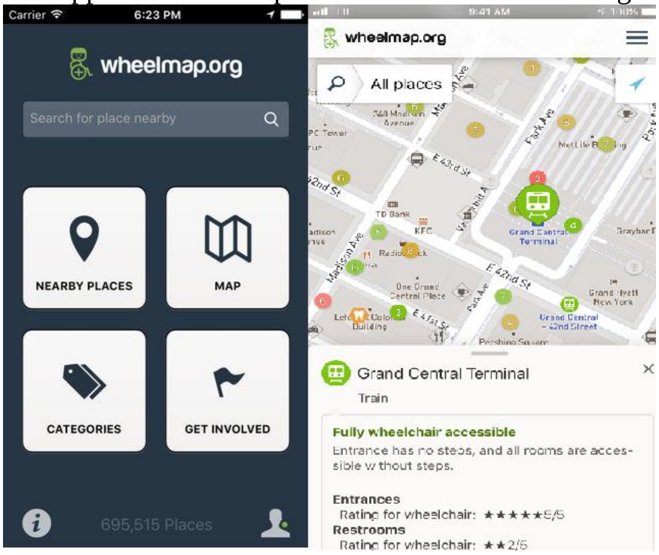
Figure 1: Type of interface Inclusive Britain.

A feature of the application was the ability to select the priority of the search, for a person with disabilities, in the case of accompanying people. If people with disabilities are accompanied, you can specify this in the application, and the search for locations will include places where you can use it. Similar to the previous one, it was possible to choose the search for places where you can go without accompanying persons or persons who use crutches or have the opportunity to move independently. In addition, there is a function to add feedback and route description from an individual user to other users of the application. On the downside, the app was designed to meet users' needs in the UK and was filled with geolocation information. There is also no search and viewing of recreation facilities on the interactive map of the area. Still, it is possible to choose the categories of establishments to search, view the remaining distance to overcome before arriving at a particular location. Functionality was also available, allowing visits to a public institution with pets and the ability to search for toilets, which provided the opportunity to visit them in a wheelchair. Moreover, the ability to access through the browser and mobile applications of this application is currently unavailable for download. Given the above shortcomings in this application, it still deserves consideration due to the functionality that was implemented in it.

2.1.2. Wheelmap

This service is an online map of the world that allows you to find and mark places accessible to people with disabilities, developed by the German non-profit organization Sozialhelden. Based on OpenStreetMap [2], the map was created in 2010 by social entrepreneur Raoul Krauthausen to help people with wheelchairs or wheeled walkers plan their days more efficiently. Parents pushing a stroller can also use Wheelmap information. Currently, the map shows almost six hundred thousand public places around the world. About three hundred new sites are added every day. Wheelmap is available on the website and as an application for iOS, Android and Windows Phone. The interface of the application on the phone is shown in Figure 2.

In this application, anyone can mark such places as bus stations, restaurants, cinemas, museums, banks, government agencies, etc., according to the accessibility for people with wheelchairs. A simple system of three traffic light colours, red, orange and green, assesses wheelchair space availability. Places that have not yet been wheelchair accessible are marked in grey. Anyone can quickly and easily observe them without prior registration. Unregistered users can add photos or comments with additional information about wheelchair space availability. Registered users can add images to the location and write comments with further details about wheelchair space availability. This app focuses on relevant accessibility information and is not limited to wheelchair spaces such as therapy centres or restrooms. Instead, wheelchair users should be involved in everyday life, focusing on public places of daily life, including cafes, clubs, theatres and essential public services, including public transport, government agencies and banks. Like Wikipedia, the user enters information so that everyone helps develop this tool from the bottom up. But there are also negative features. The application itself has a somewhat cumbersome implementation, which affected the performance of the service and significantly lower processing speed, which is an essential factor for the user. However, many elements of this application are currently being tested for errors in the system, which is necessary when working with the application. The user may lose data with which he worked or did not finish working due to an error. It is also worth noting that the developed application, namely its interface, has a complex implementation, which puts novice users in a position where they can not take full advantage of its capabilities. But the most crucial disadvantage of this portal is the poor localization for users from countries that are different from the government of the application developer and different from the English localization.



Grand Central Terminal

Train

Fully wheelchair accessible

Entrance has no steps, and all rooms are accessible without steps.

Entrances
Rating for wheelchair: ★★★★★5/5
Restrooms
Rating for wheelchair: ★★2/5

Figure 2: View of the Wheelmap interface.

2.1.3. It’s Accessible

With all its advantages, this application is such as a user-friendly interface that often uses a computer, as well as the availability of functionality that is quite useful. A key feature is the availability of a geolocation map, which allows you to view locations with a detailed description from other users, useful for people with special needs. But some shortcomings significantly affect the user's decision to continue using this application. For example, It’s Accessible was developed only for the iOS platform, making it impossible for users with limited financial capabilities and users of other devices with a different operating system. The vast majority of Ukrainian users use devices with the Android operating system, not to mention that this application was not designed to be accessed from ordinary personal computers using a web browser. Another major problem that makes new users reluctant to continue using this application is that the information provided in it has not been updated for a long time and is not currently relevant. We can assume that the database update does not occur or was not designed to keep all available information up to date.

2.1.4. Wheelmate

This information system is designed and created on a commercial basis. It is a negative factor because to use all its features, and the average user will have to pay for a certain period. The interface of this application is shown in Fig. 3.

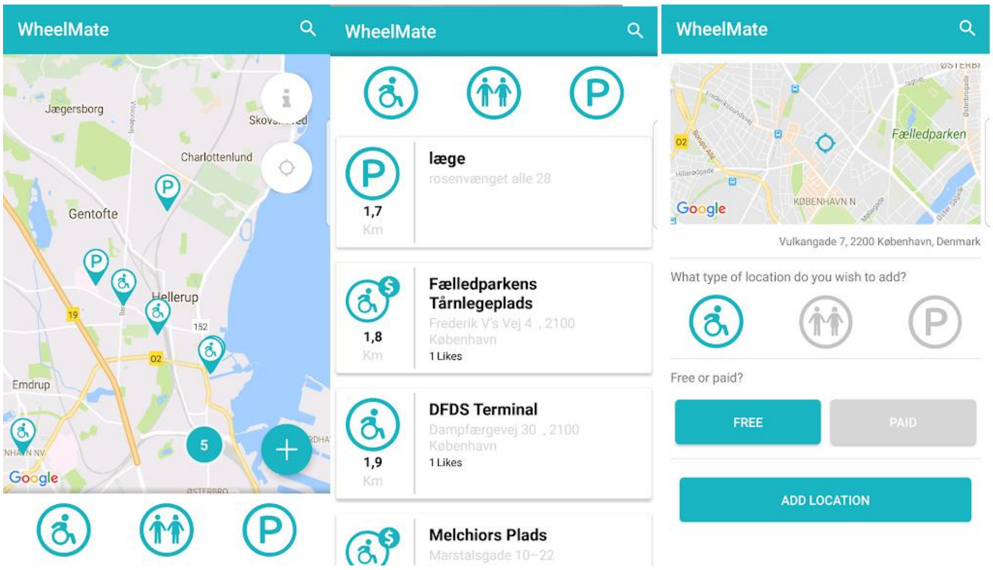


Figure 3: View of the Wheelmate interface.

WheelMate provides an instant overview of nearby wheelchair-accessible toilets and parking spaces on an interactive map. The application is designed and works for wheelchair users who independently add and check each place, ensuring that it works properly. The application is easy to use, but it does not implement the Ukrainian interface. WheelMate currently has more than 35,000 seats in forty-five countries, and new ones are added daily. The application was downloaded by about seventeen thousand people. Main features and benefits:

- Instant inspection of the nearest toilet and parking spaces suitable for wheelchairs;
- Ability to get directions from the current location to the desired location;
- Ability to help other users by adding toilets and parking spaces in the application;
- Ability to evaluate the response of the place indicated by others;
- No registration is required to work with the application.

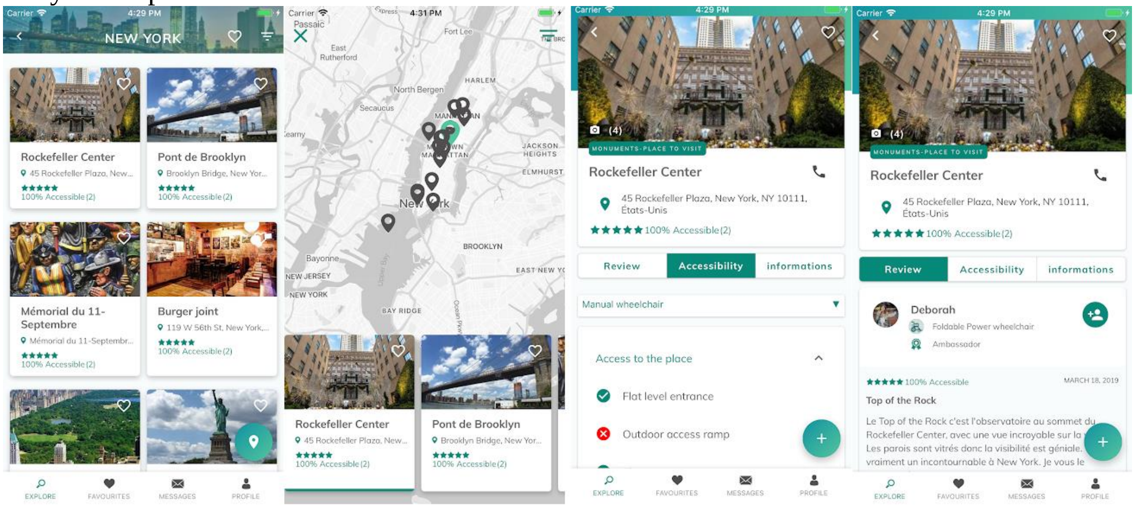
But for the peculiarities of the modern Ukrainian mentality, we can assume that the policy of the developers of this application at the expense of paid subscription for use is a failure in our country for the next few years. The traditional reluctance of citizens to use the purchased software in most cases is replaced by the ability to search for free analogues. There is also no system to prevent the distribution of software versions, which bypasses the step of paying for the use of software for free, negatively affecting developers' financial performance in our country, leading to partial or complete refusal to maintain the software in good condition for Ukrainian users. An additional problem to keep in mind, which is familiar to most Ukrainian users with disabilities, is that most of them are financially vulnerable. As a result, purchasing a paid version of the software product is one of the last to be funded due to the mandatory more essential costs. This software product is implemented as a version for viewing in a browser and as an application of a mobile device of the Android operating system.

2.1.5. Handiplanet

This application is designed as a travel guide for people with disabilities to organize, plan and prepare trips through reviews of the availability of future places to visit. As in the previous appendices, the information is filled in with the help of tips from people in the same situation.

On this platform, the user can find information about the availability of places such as bars, restaurants, hotels, museums and others from the available database. Depending on the user's location, you can view places in the area and photos and reviews shared by others. In addition, the application has a unique feature that allows a user on the social network handiplanet to create a community and ask them questions about a particular trip to help.

Figure 4 shows the graphical interface of the application on android, a geolocation map with the specified points, an example of user feedback, and functionality that allows you to assess a specific place by a new person.



Le Top of the Rock c'est l'observatoire au sommet du Rockefeller Center, avec une vue incroyable sur la Les parois sont vitrés donc la visibilité est géniale. vraiment un incontournable à New York. Je vous le

Figure 4: View of the Handiplanet interface.

But for the Ukrainian user, the application has such shortcomings as the lack of Ukrainian localization, the lack of ability to translate the comments of other users who have essential accessibility information. Also negative is that there is no version of viewing the application from the browser. However, on the plus side, this app is entirely free to use.

2.2. List and description of technologies used for development

The development of the graphical user interface of this system was carried out using modern technologies and software tools based on the existing Internet resources listed in this chapter.

Git. When developing a software product, developers often have a situation where it is necessary to restore the state of the program to the state where it was a certain period ago. To solve this problem, a standard solution was used today, and it is create conditions for future expansion of the product both in width and users who did not start developing it from the beginning, it was decided to use Git [3].

Management system: It is in most cases declared that Git is a content tracking tool. Git can be used to store content – it is mainly used to store code due to other features it provides. Version control system: The code stored in Git is constantly changing when more code is added. Also, many developers can add code in parallel. Therefore, the version control system helps to solve this problem by keeping a history of changes. In addition, Git provides features such as branches and mergers. Distributed version control system: Git has a remote repository stored on the server and local storage stored on each developer's computer. It means that the code is not just stored on a central server, but a full copy of the code is present on all developer computers. Git is a distributed version control system because the code is current on each developer's computer. I will explain the concept of remote and local storage later in this article. Real-life projects usually have several developers working in parallel. Therefore, a version control system such as Git is needed to ensure no code conflicts between developers.

In addition, the requirements in such projects often change. Thus, the version control system allows developers to go back and return to the previous version of the code.

To ensure that the sequence of changes is not overwritten or that changes have occurred are not deleted, the git version system uses cryptographic-based methods that use cryptographic keys to obtain cryptographic keys that are unique to specific files. Thus, changing files affects the critical change. Also, because the methods involve cryptographic encryption, the electronic signature of the developer with the binding of the created committees and tags can be used as the encryption key.

The program is designed for free use under the GNU GPL 2 version license.

The system has several developed access interfaces, such as git-gui and gitk, which are distributed as the basis of Git. Remote control of Git repositories is supported by the git-daemon, an ongoing SSH or HTTP server. Git-daemon has a TCP service that is part of the Git software package and is at the same time SSH is a widely used program for accessing git. The HTTP connection method, despite several limitations, is often used in monitored networks because it supports several existing filter settings of this network.

GitHub is currently a subsidiary of Microsoft. This service is an American web service for hosting versions of user files on the Internet for remote access and change and management of arrangements using Git. In most cases, the service is used to store program code. It implements the basics of git and the capabilities of distributed version control of source code. In addition to the git functionality, GitHub implements a number of its functions. The service provides access control and several cooperation functions for several programmers. For example, it is monitoring the owners of created errors, requesting functions, task management and the ability to design an information page for each project separately. In addition, GitHub provides separate plans for the team, corporate, accessible, and professional accounts. In most cases, such statements are used to cover implemented projects by groups or individuals for all available review and free access. At the beginning of January 2019, GitHub implemented an essential function of giving all types of users the ability to create an unlimited number of private repositories, including users who use this service for free.

HTML – Hypertext Markup Language is a markup language currently considered the main for creating web applications and web pages. Together with Cascading Style Sheets (CSS) and JavaScript, it forms the three technologies considered the basis for the modern World Wide Web [4]. Web browsers download HTML documents from servers or hard drives and display them as multimedia pages that can modify or influence the content, unlike regular ones. Using the semantics of HTML, the programmer can only describe the page's structure, namely the appearance of the document, only once and then use this structure to generate excellent content.

HTML tags are the basic logical units of HTML used to describe pages. Using them, the developer can add to the page elements such as images, videos, audio files, and other objects with which the user can interact with the web page as a current program window. Also, the markup language allows and creates a means by which the structure of documents can be organized to fill with similar content and create static directories such as Wikipedia. HTML is not a programming language. For example, it cannot create dynamic functionality. Instead, it allows you to organize and format documents, similar to Microsoft Word. The browser reads the HTML file and plays its contents so that Internet users can view it, so it is essential to follow the structure in the file because if you make a mistake, the browser will not be able to submit the file for viewing to the user.

Typically, the average website includes several different HTML pages. For example, home pages, about pages, contact pages would have separate HTML documents.

CSS is cascading style sheets. These are style sheets that are used to describe where specific blocks should be in a document and what they should look like. Usually used in conjunction with a markup language, for example, HTML [5]. Today, CSS is a critical element of the World Wide Web, par with JavaScript and HTML. It can also be used with any XML document, including regular XML, SVG and XUL. CSS is used in conjunction with HTML and JavaScript to create user interfaces for web applications and user interfaces for many mobile applications. Before making CSS, tags such as font, colour, background style, element alignment, borders, and dimensions had to be repeated on each webpage, along with descriptions of all their types. It was a very long process. For example: If you design an extensive website where fonts and colour information are added to each page, it will be a long and expensive process. CSS was created to solve this problem. It was a W3C recommendation [6]. CSS style definitions are stored in external CSS files, so you can change the entire website by changing only one file. CSS provides more detailed attributes than regular HTML to determine the appearance of a website.

Javascript is one of the high-level programming languages that meet the ECMAScript specification. Similar to CSS and HTML, JavaScript is one of the three leading web development technologies today. JavaScript provides the ability to create web pages in which the user can modify the content and is an integral part of web applications. Most websites use Javascript, and the most popular browsers have a unique JavaScript mechanism for its execution.

In most cases, JavaScript is either embedded in a web page or included in a .js file. JavaScript is also a language mainly performed on the client-side rather than the server-side, although these figures have been close to equalization over the last five years. The script in the JS language is downloaded to the computers of site visitors and then processed. Adding JavaScript code to a web page is a relatively simple process and standard if you perform HTML and CSS encoding. You can add JavaScript directly to the page code using the <script> tags and giving them the type text/javascript attribute. The process of adding JavaScript is similar to adding CSS to a site. You can also add JavaScript to the page as a separate header file with a .js extension. It is usually done if it is code that needs to be included on multiple pages at once. The script is then downloaded and processed in each user's web browser, turning it into dynamic objects and effects that they see on their screen. Because different browsers handle JavaScript, the user may be able to disable JavaScript at the end. Sites that use JavaScript must have a backup plan to avoid crashing when this happens. JavaScript syntax uses dynamic typing and curly braces other than regular OOP but focuses on writing code as objects using prototypes. The fact that the script is on an HTML page means that the scripts written by the programmer can be seen and copied by anyone who views the page. However, this openness is a great advantage, as the downside is that the script can be viewed, learned and used in any other JavaScript script and used on another page.

Client JavaScript is specifically designed for use in a web browser along with HTML pages. It has certain security implications. The user requests a specific HTML page without knowing if it contains JavaScript. The HTML page is delivered to the browser, including scripts. Typically, scripts run automatically when a page loads or when a user performs specific actions. There is nothing the user can do to stop the hands. The user can disable JavaScript, but few end-users know how to do it. When a user receives a page that includes JavaScript, his browser's JavaScript interpreter launches and tries to execute the script. Now the main problem is that different browsers use their interpreter and that sometimes browser providers choose not to implement a bit of JavaScript. Their reasons were usually related to the advantages of business over competitors. Hence the fleeting browser incompatibility. In addition, each new version of the browser understands more JavaScript and allows more and more parts of the HTML page to change scripts. [7] It leads to even greater incompatibility.

It is best to solve compatibility issues in each case. JavaScript allows the use of imperative along with prototype and object-oriented programming styles and also supports functional types. This language supports form and field APIs for fast interface building. It is used to process data received from the client in text and processed using arrays, date conversions, regular expression checking and visualization of DOM results. [8] The language itself uses a host environment that is a browser in which it is built to enable the performance of its primary functions [9].

jQuery is a fast, small, multi-platform and multi-functional JavaScript library. It is designed to simplify client-side HTML scripts. It makes crawling and manipulating HTML documents, animation, event handling, and AJAX easy with an easy-to-use API that runs on many different browsers [10]. The primary purpose of jQuery is to provide an easy way to use JavaScript on your website to make it more interactive and attractive. It is also used to add animation.

jQuery is a small, easy, and fast JavaScript library. It is cross-platform and is supported in various types of browsers. It's also called writing less to do more because many everyday tasks require many lines of JavaScript code, and the methods they implement can be reached by a single line of code when needed. It is also instrumental in simplifying many complex things with JavaScript, such as AJAX calls and DOM manipulation [11]. The key benefits of jQuery are listed below:

- jQuery is a small, fast and easy JavaScript library;
- jQuery is platform-independent;
- jQuery implements the principle of writing less, doing more;
- jQuery simplifies AJAX and DOM call handling.

The following are important jQuery features:

- HTML manipulation;
- DOM manipulation;
- DOM element selection;
- CSS manipulation;
- Effects and animation;
- Asynchronous AJAX requests;
- Short syntax for calling HTML event methods;
- Extensibility due to plugins;
- Built-in JSON parsing.

These advantages allow developers to implement abstractions for animation and interaction, creating complex effects from combining simple results and creating widgets with a complex logical structure [12]. In addition, the plug-in approach used to build jQuery allows you to create dynamic web applications and pages on the World Wide Web.

React. This JavaScript library is written in open source to build user interfaces or interface components. It is supported by Facebook and the community of individual developers and companies. React can be used as a basis for developing one-page or mobile applications. However, React only provides data to the DOM, so the creation of React programs usually requires the use of additional libraries developed by it or third-party developers for status management and routing [13].

React does not try to provide a complete library of programs in a single file but implements a gradual addition. It is explicitly designed to build user interfaces, so it does not include many tools that some developers may consider necessary to create an application. It allows you to choose which library the developer prefers to perform tasks such as providing network access or local storage. As the library matures, typical usage patterns emerge. React.js has become one of the technologies used to develop the client side of this application. This library specializes in helping developers create user interfaces or interfaces. For websites and web applications, user interfaces are on-screen menus, search bars, buttons, and anything else that users interact with to use a website or application. [14].

Before the development of React, developers were stuck in creating user interfaces manually using the Vanilla Js API, in which the developer independently prescribed states and handlers in response to user action. Or did it with less UI-oriented React predecessors, such as jQuery. It created many opportunities to make errors in programs and develop programs with low performance. In 2011, Facebook engineer Jordan Walk began React JS specifically to improve interface design. In addition to providing reusable React library code that saves development time and reduces the risk of coding errors, React has two key features that appeal to JavaScript developers such as JSX and virtual Dom

Since ReactJS is a new technology and has not yet fully taken root as one of those used without thinking, the disadvantages and advantages that led to the choice of this library for the development of this project were described in detail below. List of benefits of ReactJS [15]:

- This library is easy to learn and use. Furthermore, this library is not difficult to study and easy to use due to its extensive documentation, which is constantly updated and translated into Ukrainian.
- ReactJS recommends that developers use the MVC pattern (Model View Controller stands for Model View Controller) when developing a commonly used software design template for implementing user interfaces.
- ReactJS is also an open-source library of freely available examples of project implementation on which there are enough to master it.
- Building web pages that are not inferior to programs become much more accessible. For example, creating a web application interface that implements a dynamic display of information with HTML strings without this library was difficult because it required many interconnected codes. Still, with React JS, the problem was solved if we take its approach as a basis for development. This approach allows less coding and provides much more functionality.

React's virtual DOM improves user interaction, and the developer speeds up work. DOM is a logical structure of documents in HTML, XHTML or XML formats. Web browsers use layout mechanisms to convert or parse the HTML syntax into an object model document that we can see in browsers. The main concern with the traditional DOM design is how it handles changes, such as user inputs, queries, etc. The server constantly checks for the difference caused by these changes to provide the required response. To answer correctly, he also needs to update the DOM trees in the entire document, which is not ergonomically valid because the DOM trees are quite large today, containing thousands of elements. Using React was able to increase the speed of updates using a virtual DOM. Unlike other frameworks that work with Real DOM, ReactJS uses its abstract copy – Virtual DOM. It updates even the minimalist changes applied by the user but does not affect other parts of the interface. It is also possible due to the isolation of React components. It makes updates very fast, allowing you to create a highly dynamic interface. Using react, the user can notice how he writes in a Facebook chat and sees a simultaneous update of the news feed. Moreover, in React, developers do not have to tie the DOM to the functionality in the interface because the React elements are already connected to it. It also allows you to transfer react programs to other platforms. This approach allowed developers to work faster with UI objects and use hot reboots – applying changes in real-time. [16] It not only increased developer productivity but also accelerated team programming.

The **API** stands for Application Programming Interface, or a set of functions and procedures that allow you to create applications that access the parts or data of the operating system, program, or other services. To put it simply, an API is a software reseller that allows two different applications to communicate with each other. [17]. Commercial websites often provide developers with specifications or protocols for service requests or data exchange with the company. The code they share is called an API, and the tools they produce are called applications. Large organisations often create APIs for their customers or internal use, and the main goal is the external exchange of information. Walgreens provides a photo printing API that allows someone to create a mobile app so users can print photos directly from their phones in Walgreens. Companies of all sizes can use the API to analyze websites, project and team management tools, online payment systems, and many other operational solutions.

For individual cases, ways of modifying and presenting data, the API can implement many interfaces written on predefined specifications for routines, classes that use specific objects, data structures. Windows API, POSIX, and ASPI are examples of different forms of APIs.

There are three approaches to API release policy. Private in which the API is intended for internal use only. It gives companies the most control over their API. Affiliate – The API is shared with specific business partners. It can provide additional revenue streams without compromising quality. The public API is available to everyone. It allows third parties to develop applications that interact with the service server API and can be a source of innovation [18-21].

The modern API has acquired some characteristics that make them extremely valuable and helpful:

- Modern APIs adhere to standards (usually HTTP and REST), user-friendly, easily accessible and understandable;
- They are treated more as products than as code. They are designed for consumption for a specific audience (for example, mobile device developers), they are documented and created so that users can have certain expectations about its service and life cycle;
- Because they are highly standardized, they have much stronger discipline in security and management and are controlled and managed by results and scale;
- Like any other piece of production software, the modern API has its software development lifecycle (SDLC) design, testing, construction, management and versioning. In addition, modern APIs are well documented for consumption and version installation.

JSON – JavaScript Object Notation is an open standard data format or exchange for semi-structured data. It is based on text and can be read by humans and machines. In JSON, semi-structured data coming from various sources and devices, including mobile phones, web browsers, servers, and IoT devices, is collected as messages, called «events,» logically organized into packets and then transmitted to the data platform via API data interface. It can be used in many applications but is especially common for transferring data between servers and web applications or Internet-connected devices. These programs can often only receive data as text, and JSON uses human-readable text transmission.

During the development of this format, there were several shortcomings [22]:

- Clients cannot check whether the generated response text from the server includes all the necessary data. Instead, they just need to use it.
- Frontend developers can and should receive metadata through more clearly defined endpoints. A set of access keys to receive data from the server through the API is already provided through oauth2 or API authorization servers. Information about the capabilities of the authorized user is provided through oauth2. It can also be obtained through more public endpoints, such as distributing the received data according to the action performed by the user through the API. Additional data can be provided much more quickly through API endpoints than by increasing the access key even more in which user data is encrypted.
- Self-encoded access tokens are now quite massive and must be stored in the client's cookies. The receipt must also be sent with each request. A typically generated token can reach tens of bytes, while a typical JWT token can easily reach high hundreds. It requires different types of requests to be used according to the user's actions.

But in contrast to the disadvantages of this format, there are advantages:

- The data format is relatively simple, easy to read and write. All formats are concise, low busy bandwidth;
- Easily parsed, client-side JavaScript can simply read JSON data via the js function JSON.pares();
- Support for multiple languages, including server-side scripting languages such as ActionScript, c, c #, ColdFusion, java, javascript, Perl, PHP, python, ruby, facilitates server-side parsing.
- PHP-JSON and JSON-PHP have appeared in the world of PHP. It is preferable that the program after serialization of PHP was called directly. Objects, arrays, etc., on the server-side. Php can directly generate a JSON format that is easy to access and remove from the client;
- Because the JSON format can be used directly with server-side code, the scope of server-side and client-side code development has been dramatically simplified, and tasks have been performed consistently and easy to maintain.

TypeScript is a programming language whose source code is available for public access by Microsoft [19]. However, its development does not stand still, and user support constantly adds new features to the IDE, which are used to develop. The language itself is a syntactic set of JavaScript, which adds rules to the language in strict typing of variable types when writing programs.

TypeScript inherits JavaScript's main advantages and offers additional benefits from static typing and other concepts specific to TS. They are instrumental in a large code base and distributed teams working on a single project. TypeScript detects 15 per cent of common compilation errors. Far from a 100 per cent result, this amount is still significant enough to save developers time and allow them to focus on correcting errors in logic – instead of catching common mistakes. Pushing code through the compiler also reduces quality assurance and testing.

By adding strict types and other elements that make the code more expressive, the programmer can see the idea of the developers who originally wrote the code. It is essential for distributed teams working on a single project. The code, which speaks for itself, can compensate for the lack of direct communication between team members. Type information makes editors and integrated development environments (IDEs) much more helpful. They can offer features such as code navigation and autocomplete, providing accurate suggestions. The developer also gets feedback when typing: the editor marks errors, including types, as soon as they occur. All this helps developers to write supported code and leads to a significant increase in productivity. TypeScript supports such concepts of object-oriented programming like classes, interfaces, inheritance, etc. The OOP paradigm helps build well-organized scalable code, and this advantage becomes more apparent as the project increases in size and complexity. TypeScript typing files have been created for widely used libraries such as jQuery, MongoDB, and D3.js [20].

Adobe Photoshop. This software is widely used for bitmap editing, graphic design and digital art. It uses layers to provide depth and flexibility in the design and editing process and offers powerful editing tools that combined can do just about anything. Photoshop has several tools that users can use to manipulate and edit images – the cloning tool duplicates part of the image; the shape tool creates rectangles, circles and lines; the pen tool creates precise paths that use anchor points. Other tools include trimming, slicing, magic wand, lasso, rubber band, and tent.

Android. The platform used by most Ukrainians in mobile devices is Android. Android Studio is an IDE, an integrated development environment with an interface where a developer can enter code and access all the various tools needed for development. In addition, Android Studio allows you to access libraries and APIs from the Android SDK, thus providing access to your operating system features. The IDE also can embed your program in the APK using Gradle, test it with a virtual device – an emulator and debug the code at startup.

In this chapter of the explanatory note to the work, an overview, analysis, identification of key functions and comparisons of currently developed information systems for finding inclusive places for people with disabilities were conducted. Their ability to be used by ordinary Ukrainian citizens who do not speak a foreign language has been compared. Availability for the wallet of the average Ukrainian was also analyzed. Based on these data, it was concluded that foreign applications are not adapted to Ukrainian realities. They do not support Ukrainian localization, and some of the information systems are developed on a commercial basis and require payment for use. With the summarized results, this work will implement the functionality that includes the development of other information systems and in some places ahead of them, namely:

- during use, the information service must immediately and effectively provide information to its users;
- the graphical user interface will be implemented with the requirements of clarity and ease of use;
- the system will implement functions designed for Ukrainian users, namely the provision of information about the possibility of using the specified routes by people with disabilities;
- will not have a commercial basis in the created information system. Its main goal will be to help people with disabilities, who are often financially disadvantaged;
- the necessary database is filled with information on the accessibility of public institutions and all possible places in the city of Lviv for the possibility of visiting them by people with disabilities;

The information system, which will operate on the territory of Ukraine, should embody these functional features, and the critical task is to help people, not financial enrichment.

3. Materials and methods

3.1. System analysis and substantiation of the problem

This chapter of the work describes and characterizes the content of the information system, the scope and purpose of its elements, their design and placement in the interface, as well as detailed information about all implemented functions and parts of this software product.

To analyze the information system and determine the general purpose of the operation, it was necessary to develop a tree of project objectives, which is shown in Figure 4.

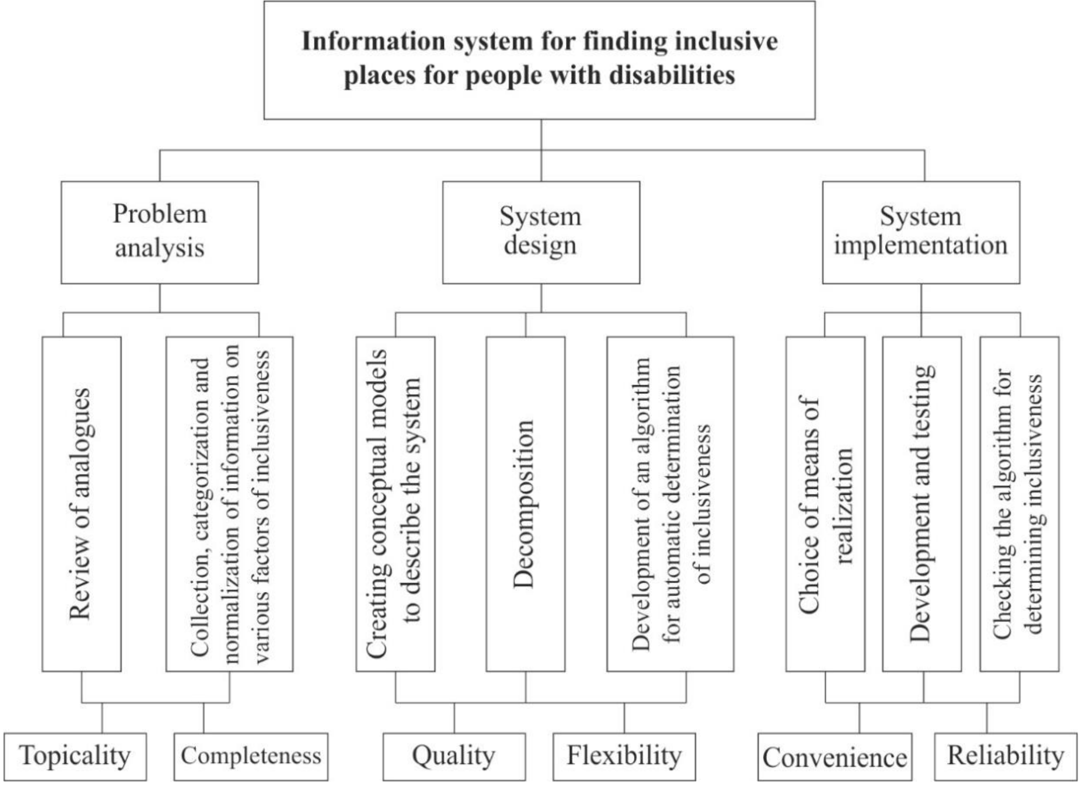

Information system for finding inclusive places for people with disabilities
Problem analysis
System design
System implementation
Review of analogues
Collection, categorization and normalization of information on various factors of inclusiveness
Creating conceptual models to describe the system
Decomposition
Development of an algorithm for automatic determination of inclusiveness
Choice of means of realization
Development and testing
Checking the algorithm for determining inclusiveness
Topicality
Completeness
Quality
Flexibility
Convenience
Reliability

Figure 5: The goal tree

Even though the development of a detailed plan of the information system implementation process was initially quite time-consuming and resource-intensive, it still paid off, as long-term planning allowed to work much more carefully and effectively on the application. When planning the stages of system development, all actions related to the implementation of work directly on the design, development, testing, placement and documentation of software solutions, as well as a preliminary analysis of the usefulness, relevance and prospects of the system, were taken into account. In the process of direct planning of the work process, a study was conducted on the system's feasibility and identified its main features, a balanced assessment of available resources for the most accurate and unbreakable time frame for all stages of software development. Before the direct result of the application, a detailed project of the software product was created, which contained a description of the system, its components, their interaction with each other and with the user, models of algorithms for subsystems and design sketches of the client part of the application. The following is a detailed description and description of the interaction of critical functions of the user interface of the application in a sequence that corresponds to the logical architecture of the information system. The main aspects of the overall goal according to the goal tree: Analysis of issues; Architecture design; System implementation. According to the goal tree, the main elements of the purpose of the information system are relevance, completeness, quality, flexibility, convenience and reliability. This system can be considered an information retrieval system. The general purpose of the information system is to provide users with disabilities with convenient access to information about the inclusiveness of locations.

3.2. System architecture and modules

3.2.1. Main page and main menu

This chapter provides a detailed description of the primary and auxiliary components of the main page of the user interface, describes their appearance and design as the elements themselves and their placement on the page and relative to each other, the functions of these elements, their interaction and their role in the system in general. When you first launch a web application or its mobile version, the new user of the resource must give the program special permissions within the operating system to obtain information about this user's current location and display additional pop-up notifications. To grant the system the necessary permissions, the user must confirm the required permissions in a separate pop-up window the first time you use the program, or provide them later in the Internet browser settings menu or mobile device settings for the web version and mobile application, respectively. If the required settings are not changed as described before using the application, the user will still have access to most system functions but will not perform specific interactive actions. Also, the information system will not provide the user with information based on the current location because it has not been granted access to the geolocation. Figure 6 shows the design of the main page of the application. The main menu of the information system is located directly on the top panel of the browser version of the application in buttons containing the name of the corresponding submenu. In the mobile version of the application, you can view these components by calling the list by pressing the main menu button, which is located in the upper right corner of the screen on the top panel of the interface. Regardless of the application version, the system's main menu contains the following functions: Change the language; Add a location; Contacts; Help or Frequently Asked Questions; Return to the map (mobile version only). Figure 7 shows the main application menu.

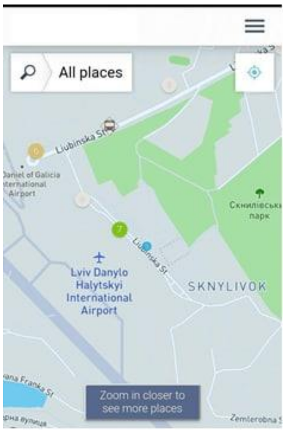

All places
Liubinska St
Daniel of Galicia International Airport
Скнилівський парк
Liubinska St
Lviv Danylo Halytskyi International Airport
SKNYLIVOK
Zoom in closer to see more places
Zemlerobna

Figure 6: Home page view.

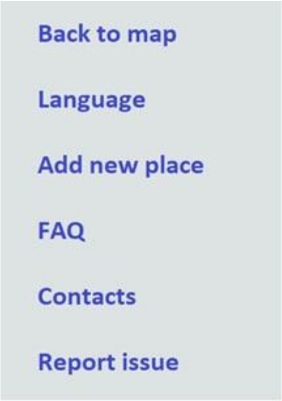


Figure 7: View of the main menu.

3.2.2. Map

The map is the most critical component of the whole system regarding the client and server part. The vast majority of system algorithms and their functions interact with the map to some extent, as almost all actions that users can perform or initiate by their commands, with some exceptions, are directly or indirectly related to locations. The map is also the dominant design element of the home page and opens by default as the initial view of the application. This design solution was dictated by the apparent need to display the map on as large an area of the display as possible to provide an adequate representation of its elements and tools for interaction with the map and provide users with access to map interaction and interaction. This chapter describes both the map itself and additional tools for performing operations with it. The map entered into the system using API technology, in addition to the topographic images themselves, contains the following elements and functions [26-33]:

- Determining the current location;
- Centring;
- Scaling;
- Search by name;
- Search by category;
- Indicating a marker on the map;
- View detailed location information.

The current location feature uses GPS technology, and geolocation services are available on the application's device. Thanks to this feature, the system is able to immediately display the area of the map in which the user is when marked by the user's location in the centre of the map field on the screen, which speeds up the visual search of places and routes and provides ease of map and application. The main reason for assigning such a configuration is the default location function or the initial location. In addition to loading the map with such an initial display, a particular UI element has also been implemented, namely a button located in the upper right corner of the map, which returns the map to the default position to simplify the process of navigating the map.

If the user refuses the application to provide access to the geolocation services of the device, the map will be displayed centred on the zero coordinates of the map. Also, in addition to location, scaling is an essential parameter of the map taken into account when setting the default display.

If you grant access to the device's geolocation, the map will be displayed close up, where the display radius and the initial scale depend on the screen resolution. On the other hand, suppose you refuse to provide access to the device's location services. In that case, the map will be displayed in full size with the maximum display radius, regardless of the device's characteristics.

Map scaling can be performed by the user using standard interface tools. For the web version, these are buttons marked with “+” and “-” symbols that zoom in and out on the map, respectively, or using a computer mouse wheel or laptop touchpad. In the mobile version, scaling is performed using common gestures on the mobile device's touch screen. The map also has a standard function for viewing detailed information about the place. Calling this function is triggered by clicking on the symbol of the selected location on the map, which results in opening a window containing information about the place, such as its official full name, address, hours of operation, and the level of inclusiveness.

Each of the location categories is represented on the map by its unique icon. All types are discussed in detail and described later in this chapter. The location search function on the map is represented in the user interface by a button with a search symbol in the form of a magnifying glass in the mobile version of the application and a text entry bar with a similar symbol web version. In both versions, the location of the search button is unchanged, namely in the upper left corner of the screen. When you click on the search button in the mobile version of the program or activate the search box in the browser version, an additional window allows you to search with other filters, divided into two chapters, level of inclusion and category. In addition, you can search by place name in the search text box. If the location database does not contain the desired location, the user will see a window showing that the system failed to find the desired location because it is not in the database. The interface will also display an option for the user to suggest a new place to add to the database using a specially designed form, which will get the user data already structured and simplify checking and adding recent locations.

3.2.3. Types of locations

This chapter examines and categorizes the types of locations present in the database structure, algorithms, and application user interface. In addition, this chapter describes each of the categories, shows the appearance of icons for all these types of locations, and a list of their subtypes and examples of such places. At the stage of designing the elements of the program, its functions, and creating a design, it was decided to distribute the locations according to their purpose and main activity. The icons of the categories are shown in Table 2.

Table 2
The icons of the categories

Name	Icon	About category
Transport	🚌	This category includes all facilities and institutions that specialize in logistics and transportation.
Food	🍴	This category includes all catering establishments. This group of places includes all premises registered in the database of locations as catering establishments.
Trade	🛒	The Trade category includes all types of locations and establishments that specialize in trade.
Dressing room	WC	Public toilets were placed in a separate category to display on the map, with a corresponding icon. This type of location has no subcategories.
Medicine	✚	This type of location includes all public and private institutions that provide health care services.
Finances	💵	This category includes all institutions and establishments that provide various financial services and conduct foreign exchange or banking transactions
Tourism	🏰	The category of Tourism includes the types of places that have a primary focus on visiting tourists as assistance or specific services and attractions that have a recognized historical or cultural value. The category does not include establishments, accommodation or catering, as they are not focused exclusively on tourism.
Residence	🛏	This category of locations includes locations and facilities that offer temporary accommodation services to their visitors. The type does not include companies specializing in the sale or lease of real estate.
Entertainment	🎭	Institutions specialising in organizing and conducting entertainment events and providing services of this nature are included in this type of location.
Administration	🏢	Local and state authorities, administrative institutions and government agencies to provide services to the population belong to this category.
Education	🎓	This category includes higher, secondary, primary and preschool education institutions and specialized educational institutions and libraries.

Sport	⚽	The category called Sports includes elements of infrastructure that provide access to recreational or recreational sports for the population and all institutions engaged in activities in the field of physical culture.
Miscellaneous	◉	This category includes all locations that do not fall under the characteristics of any of the previously listed and described places because they do not correspond to their direction but are not widely represented enough to create separate categories of locations for them.

3.2.4. Place information

Each location is displayed on the map as a symbol of the category to which the place belongs. To view detailed information about a place, the user needs to click on the icon on the map, bringing up a window containing the available information about this place. The location description window is divided into logical blocks according to the subject matter of the information provided in it or the options available for that location. The top block contains the category icon in the upper left corner of the window and the institution's name and the name of the subcategory to which the institution belongs. The next block shows one of three levels of inclusiveness, indicating whether the location is fully adapted for use by people in wheelchairs, partially or not at all adapted to the needs of people with disabilities. Suppose there is no information on the inclusiveness of the place. In that case, this block will allow you to specify the level of suitability of the area for visitors' needs in wheelchairs by selecting one of the three options listed above. Options are presented in the form of three buttons with the name of the level of inclusiveness indicating the appropriate colour, red - for wrong places, orange for partially adapted and green for locations fully adapted to the needs of citizens with disabilities.

The third block displays the available photos of this location. If the system does not have pictures of this location, the interface displays an option that prompts users to add images independently. The choice is shown as a button with a standard camera icon and a text pointer. Clicking this button brings up a new window with a warning about the conditions that new photos uploaded by the user must meet.

List of these conditions:

- There are no identifiable images of people in the photo.
- Photos were taken personally by the user, and it is not allowed to use images uploaded from the Internet;
- Photographs provide information that is relevant to the novelty of the photo. For example, it does not give outdated data and helps assess the inclusiveness of the location.

If the user decides to upload their photos and agrees with the requirements they must meet, you need to click on the «Continue» button located at the bottom of the window. Clicking this button opens the file manager program, which is installed by default in the operating system of the device used. In this menu, the user can select the desired photos. To complete the process of downloading these images, the user must confirm the selection and download by the actions provided by the configuration of the device's file manager. Selected photos uploaded to the information system are displayed in the detailed location information window after refreshing the page with that location. Suppose the user does not want to upload photos and close the initial window to add images. In that case, you must click on the «Cancel» button, which is also located at the bottom of the window, for adding photos of the location. You can also cancel the process of adding pictures of a particular place by performing the action provided by the file manager's settings of the operating system of the device on which the operation is performed. The last block of this menu contains additional options for functions related to this location.

The first option is to open information about a given place in an alternative map and geolocation system. Examples of such systems are Bing Maps, Google Maps, OpenStreetMap and others. If you select this option, the specified location will be opened in the alternative map service.

When using the web version of the application, a new tab will open with the web version of the service or application representing such a service. In the mobile version, the corresponding mobile application will be opened on the device. In any case, an alternative system will open the page of the specified location. The next option displayed the institution's contact phone number and implemented the ability to call directly from this window, using which the user will be directed to the window of the Phone application on the mobile device. The address provided by the owners or the institution's administration as the official website of this place is provided below. Selecting this option will open a new tab with the specified site address. For both of the above options, these fields will not be displayed at all if information such as a website address or contact phone number is missing.

If desired, users can suggest location information in the Report a Problem menu and select the appropriate option. Then there is an option that allows you to spread links to the location page in the information system using third-party communication services such as social networks, including Facebook and Twitter, or through the assistance of messenger Telegram or Whatsapp. It is also possible to distribute data via e-mail. Each of the distribution options is presented in the form of a button, which shows the logo of the service and the name of the resource under it.

To exit the distribution menu, click the button with the arrow icon on the left side of the block.

At the bottom of the location, details menu is the error message option. With it, the user can complain about inconsistent or incomplete information or suggest additions to it.

3.2.5. Contact support

To call up the error message feature or contact customer support form, you must select the error message option in the location details window or go to the Contacts page of the application's main menu. When calling the error message function through the menu of detailed location information, the user will receive a window. You will need to select the error category that the user wants to report. The error classification options offered in this menu have the following names:

- The location has an “unknown” inclusive status.
- I have location information.
- This location does not exist.
- The wrong location address is specified.
- My problem is not listed.

When pressed, the first option opens a menu with icons and a description of the three levels of equipment of the location for use by people with particular problems, where you need to select one of them and confirm the selection. The second item opens a link to a form to provide additional information or suggest changes to existing data. The non-existent location option, similar to the previous paragraph, prompts the user to fill out a form to change the information about the place in the system. The ability to report an incorrect location address allows you to suggest a speech change in a particular field. The last of these options is to redirect the user to the communication menu with the support team. The user can get contact information for feedback, describe the problem for consideration by the support service and get the results of the review, research and solutions by e-mail. All menu options are implemented in the form of buttons containing text with the corresponding description of possibilities without using special symbols or icons. When the functionality of the contacts menu is called up, the user is redirected to the communication page by the support team and the resource developers using the corresponding button of the interface's main menu. This page contains forms for providing information about malfunctions in the information system and sending a non-error message or question to the representatives of the resource support team. The standard message form for the information system support team contains separate text fields in which the user must enter his full name, e-mail address, feedback, and the text of the message itself. The forms for sending a standard message and a system failure message differ. The latter has fields required to fill in technical details of the device, environment and application such as application version, browser or operating system for mobile applications and browser versions of this application, respectively. After entering the required information in the appropriate fields of the contact form, it is necessary to consent to the user's data by the system.

This agreement is a necessary tool to ensure a secure and transparent process of transmission and use of user data by developers and managers of information systems by national and international legislation on user protection, such as the General Data Protection Regulation or GPDR.

All fields that are required to enter before sending a message are marked with the appropriate symbol. If any of the required fields were left blank when entering, it would not be possible to send a message, and the corresponding note will be displayed in the window. To complete the form and send the message, click the Send button at the bottom of the system contact form.

3.2.6. Adding a new location

To open the form of adding a new place, you need to click on the button with the appropriate name in the application's main menu that redirects to the page with the form. The location add-in interface looks like a window that displays a sequence of conditions in which the user needs to choose from suggested options, enter information in appropriate text boxes, or upload files. To switch between the stages of filling in the form for adding information about the new location, you need to use the Forward and Back buttons located at the bottom of this window. First of all, the window displays a request to the user to choose how the information on the level of inclusiveness of this place was analyzed among the options offered by the form, including personal observation and location inspection or conversation with employees. In the next window, you need to specify the location of the new location by selecting it on the map located in the window or by entering the desired address in the search column and selecting it from the suggested options. Also, the opportunity to choose the user's current location is available to choose a point. The following window in the order of adding a new place contains a field for entering text, in which the user must enter the name of the new location. After filling in the name and moving on in providing information about the place, the user needs to select from the list the appropriate category to which the place will be assigned and its subcategory. Next, we move on to the next logical block of location questions, which do not relate to general information about the place, as in the previous paragraphs. Instead, further questions are created to determine the specific characteristics of the inclusiveness of the area according to the needs of visitors with special needs. The next block of questions involves choosing between two answers, «Yes» and «No», so the questions themselves are formulated accordingly. List of issues related to the suitability of the location for the needs of visitors in wheelchairs:

- Is there an entrance?
- Is there a staircase in the facility?
- Is there a ramp?
- Is it allowed to enter with a guide dog?
- Do the staff of the institution interact with people with special needs?
- Is there enough wheelchair space in the room?
- Is there a toilet in the facility?
- Is the toilet suitable for people in wheelchairs?
- Do the main elements of the location provide access to the trolley?

The last two questions are exceptions, as they also provide an additional answer, «Partially».

Given the answers to the previous questions, the user is also asked to answer additional clarifying questions about the characteristics of the location. Such clarifying questions concern the number and height of steps and the aspects of the ramp and the entrance to the room. In addition, the user is also able to upload photos of location infrastructure elements that affect the determination of the level of inclusiveness, such as, for example, the entrance to the room, stairs or ramp. Additional questions are displayed to the user as soon as the answer to the question that causes them is answered. Accordingly, the order in which these questions are displayed changes due to the solutions provided. In the last window, you need to click on the Send button to send the provided information for processing by the system and then add space to the database and display this institution on the map in the future.

3.2.7. Certificate

When you go to this item in the application's main menu, you are redirected to the help page. This part of the information system interface contains a list of frequently asked questions divided into blocks according to the subject matter of the questions. The default view is to display a list of questions on the page, the answers to which are hidden unfolded manually by the user.

The answers to the questions are displayed only when you click on the question. To close the field with the answer, you need to click on the text of the question again.

Logical blocks into which the list of questions was divided to simplify page navigation:

- General service questions.
- Questions about using the card.
- Questions about using search.
- Questions about the algorithm for adding locations.
- Questions about the algorithm for editing existing places.
- Common errors and known issues.

The block of general questions contains an available description of the purpose of the information system, its origin, history, creators, as well as other facts that are not directly related to the functions and algorithms of the system or technical information about the software product as such but allow you to better understand the ideas and basic concept of this service. The points related to the use of the map answer questions about the algorithm of using the map directly, navigating its interface, its additional elements, such as detailed location information or searching for the current location, and explaining the symbols used on the map. The following block of questions and answers provides a detailed description of using the search function and its additional elements. Here the algorithms of using filters and directly the field of search of locations are explained. Also, this chapter lists and explains the features and limitations of using search functions that have been implemented in this information system. The help menu chapter on adding new locations to the system provides answers to questions related to this process. This chapter also discusses the features and limitations that users may often encounter when using this application functionality. Answers to questions about changing detailed location information already entered in the system explain the process of changing certain blocks of information such as description, address, contact information or location photos.

The latter indicates a block that contains a list of known problems and errors in the system and difficulties and questions that users most often turn to the information system support service.

3.2.8. Indication of the level of fitness

When assigning a specific location to the appropriate level of adaptation to the needs of people with disabilities, certain features are considered to determine the desired level of inclusiveness of the place. Assignment of the level of location availability is performed directly by the system algorithm based on information provided by third parties or based on feedback from users of the resource, obtained through a specially designed application interface functionality, in which users offer data in a form that allows their processing algorithm on the information provided. In case of incorrect testing of the algorithm, the system allows the developer to change the current value manually. The plan defines the following distributed levels of location inclusiveness: Fully customized; Partially adapted; Unadapted; Equipped toilet available; No data. Table 3 show the symbols that correspond to specific groups of adaptation of places used to indicate locations on the map.

Table 3
The symbols that correspond to certain levels of adaptation of places

Name	Icon	About symbol
Fully inclusive location	🟢	The location, which is fully adapted for use by people with special needs, meets all the criteria defined in the algorithm for assigning inclusiveness.
Partially adapted location	🟠	Facilities that are only partially adapted to provide services to visitors in wheelchairs are marked according to the algorithm's criteria as somewhat inclusive.
Unsuitable place	🟥	Locations and institutions that are impossible or very difficult for people with special needs to access are considered unsuitable.

		As a result, all or almost all of the inclusive needs for this place are not met.
The location has an equipped toilet	+ WC✓	A separate point was also highlighted to indicate that, despite the inconsistency of the location in general with the conditions for visits by people with special needs, the existing toilet is adapted for use by visitors in wheelchairs.
No location data	+	Suppose the information system does not have enough data to determine the level of inclusiveness of the location. In that case, it will display the area as one about which there is not enough information and, accordingly, the level of fitness is unknown.

3.2.9. Localization

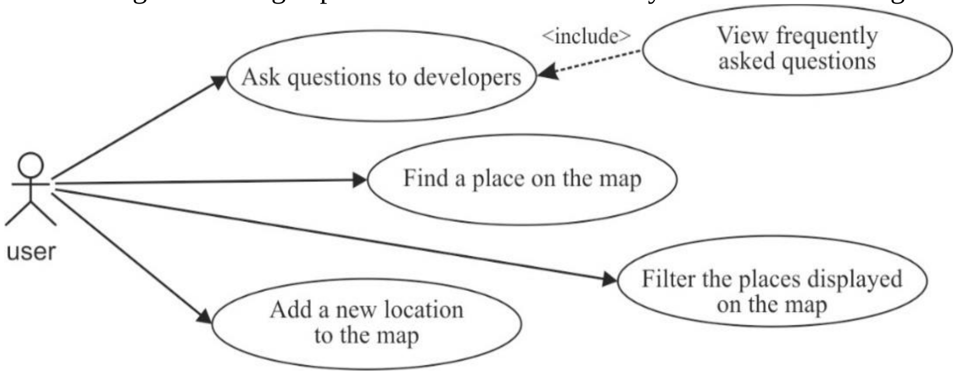
The user interface of the information system allows users to choose the language in which all elements of the application will be displayed. By default, the user interface of this software product is shown in the language that is the primary language of the device on which the application is running, provided the mobile version of the application, or the primary language of the Internet browser when using the web version of the application. If desired, the user can change the display language of the interface by selecting the desired language in the drop-down list, which is displayed when you click on the language change button in the main menu of the interface of this application. The location selection option is also available on other pages of the information system, such as the help menu, contact with the project development and support team, or in the error message window and adding a new location.

Currently, the user has a choice of interface localization from two options: Ukrainian and English translation of the interface elements of the resource.

It is planned to expand the possibilities of localization of the software product in other languages such as Spanish, German, French, and others according to the needs of the project.

3.3. Creating UML charts

The first to develop is a diagram of use cases. The actor of the system indicates all the actions that he can perform in the method suggested by the arrow in the direction from the actor to the action. There are also dependencies between activities that one step cannot be performed without performing a dependent activity. In the usage component diagram, the key actors will be only two people, the system administrator and the system user. The system administrator has only one use case, which is to work with frequent entries. This work includes adding, editing, and deleting frequently asked questions from users. The created diagram of usage options for this information system is shown in Figure 8.



<include>
View frequently asked questions
Ask questions to developers
Find a place on the map
user
Filter the places displayed on the map
Add a new location to the map

Figure 8: Diagram of use cases.

The next step was to develop a class diagram. Because the project is quite large, and the description and creation of all its classes will not fit in one UML diagram, the report is given a diagram of the types responsible for displaying the buttons with icons in the search menu of the main window.

Figure 9 shows the NodeToolbar class, its methods, and its attributes.

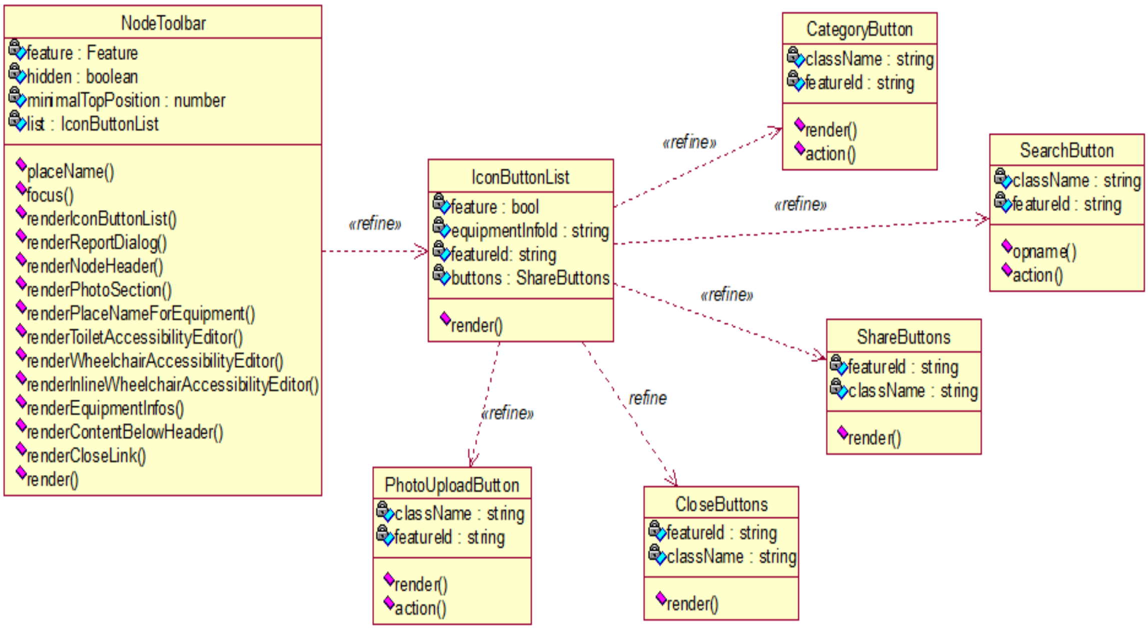


Figure 9: NodeToolbar class.

The peculiarity of the reaction is to reuse the components that were created earlier and build significant parts from a set of small ones, so there is a relationship between the classes of the search box, which is shown in the diagram by a dotted line.

Next, you need to create a user activity chart. To begin with, the starting point is set on the diagram. The next step is a line of parallel actions that divides the activity into different sequence-independent scenarios that the user can perform. The created activity diagram is shown in Figure 10.

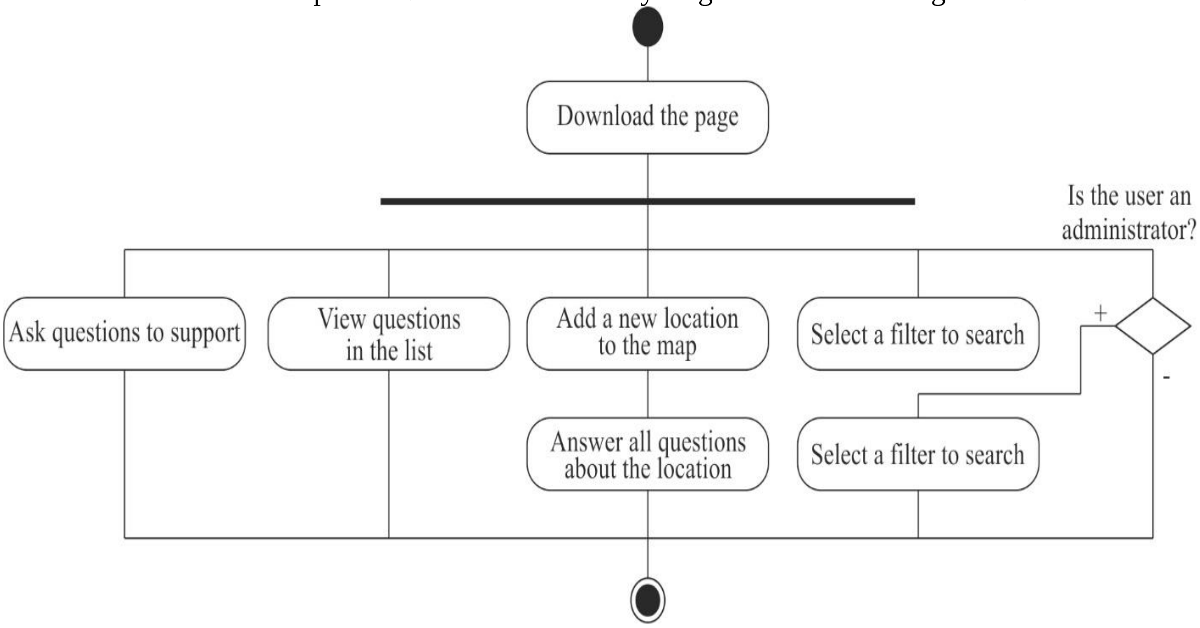

Download the page
Is the user an administrator?
Ask questions to support
View questions in the list
Add a new location to the map
Select a filter to search
+
-
Answer all questions about the location
Select a filter to search

Figure 10: Activity chart

Creating a sequence diagram is required to describe the sequence and see the lifetime and use of the object. In addition, this diagram will help you see during the development process when you need to delete an object or the result of data processing to optimize technical resources. Fig. 11 shows a sequence diagram and all functions, such as requesting data from a client to a database through an intermediate server handler. To create a cooperation diagram in Rational Rose, open the sequence diagram and press the F5 key. The result is shown in Figure 12. This diagram allows you to see what data will be exchanged between the functional components of the information system.

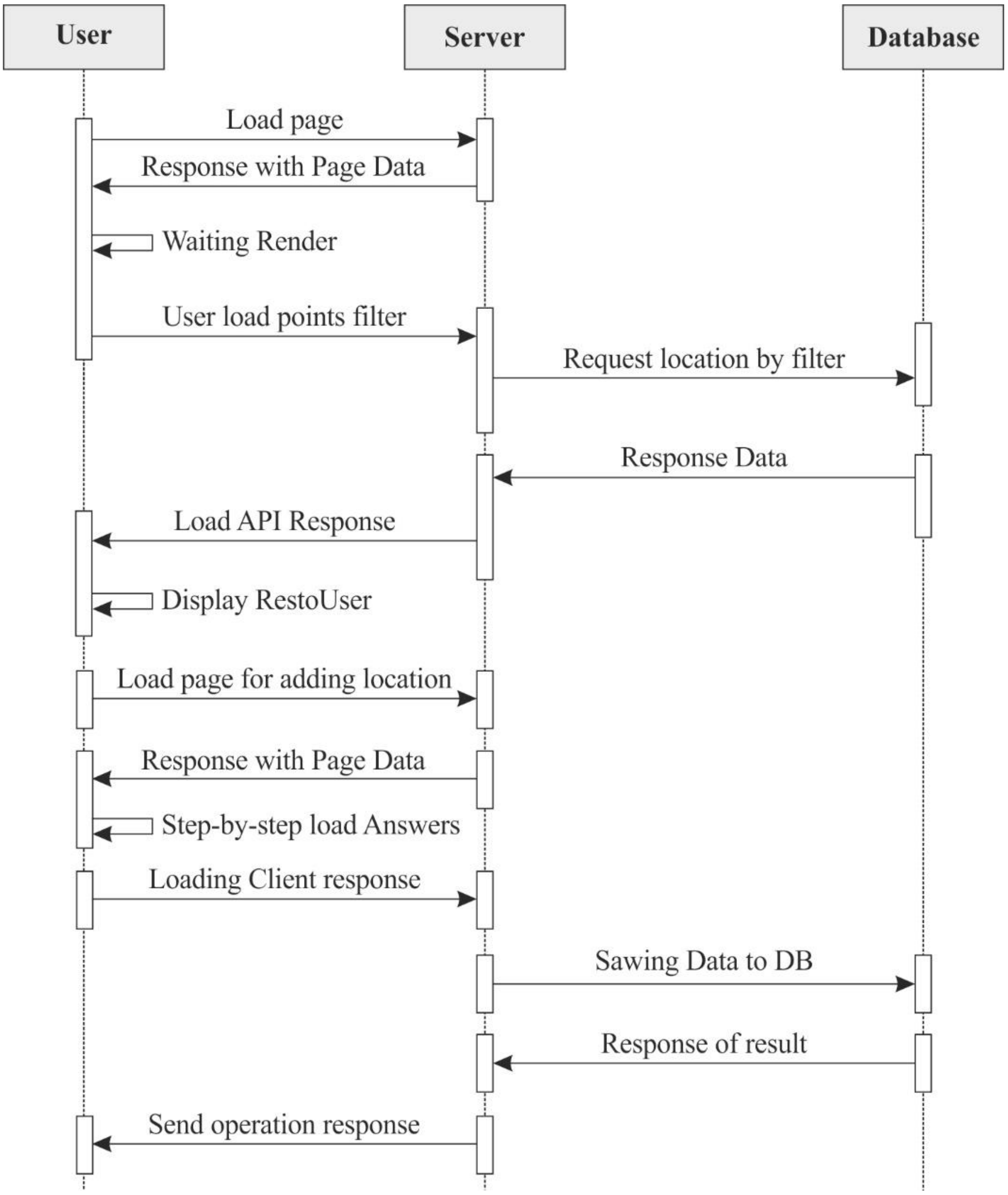


Figure 11: Sequence diagram.

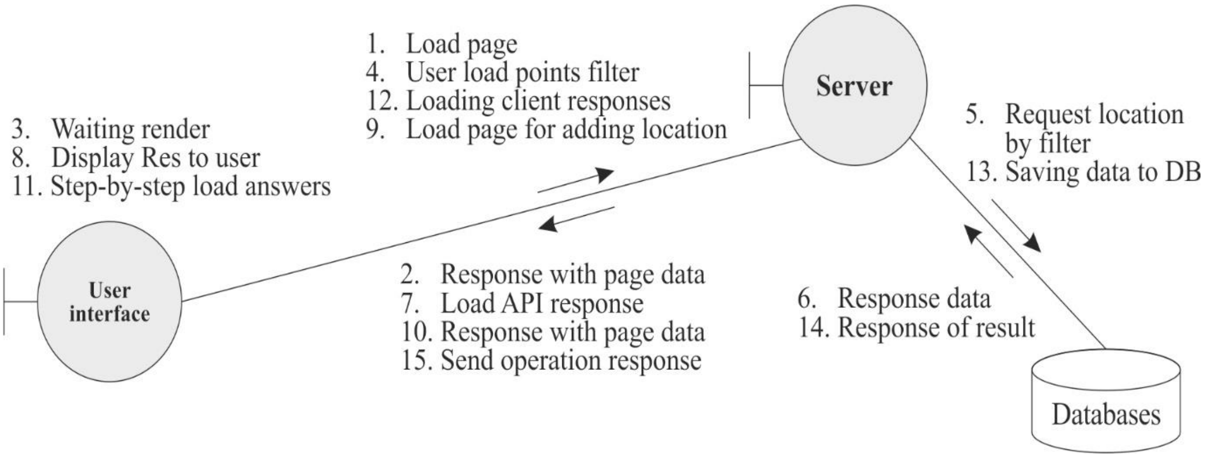

1. Load page
4. User load points filter
12. Loading client responses
9. Load page for adding location
Server
3. Waiting render
8. Display Res to user
11. Step-by-step load answers
5. Request location by filter
13. Saving data to DB
User interface
2. Response with page data
7. Load API response
10. Response with page data
15. Send operation response
6. Response data
14. Response of result
Databases

Figure 12: Diagram of cooperatives.

Figure 13 shows a diagram of the components of this information system.

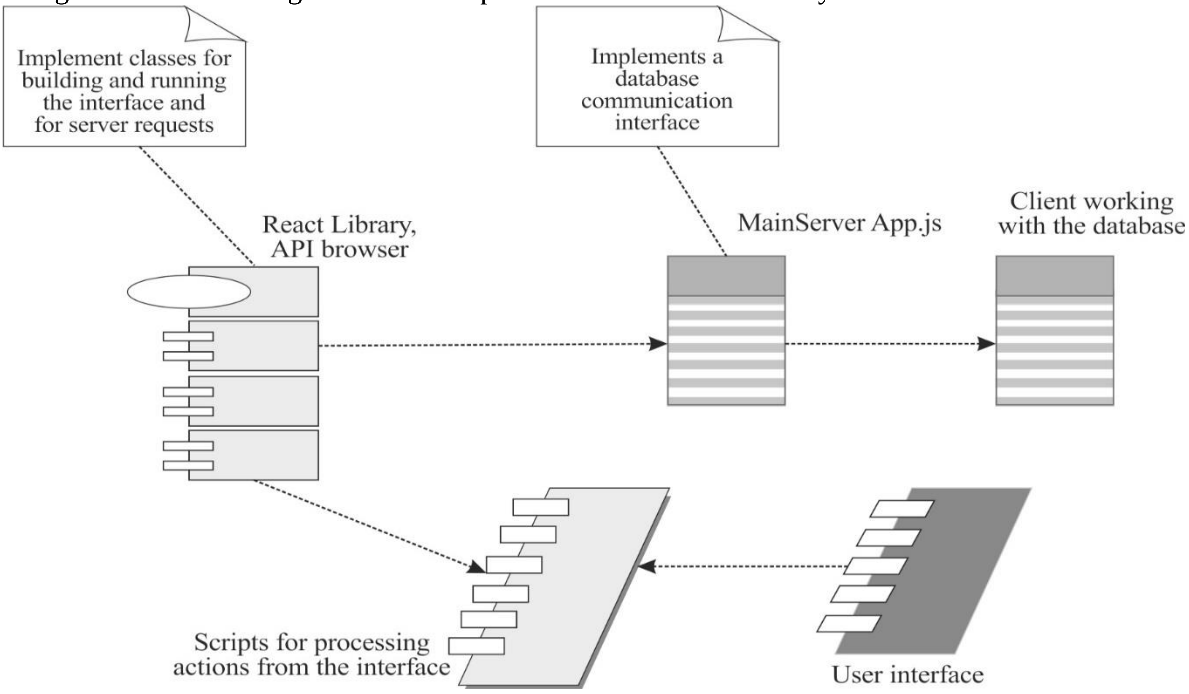

Implement classes for building and running the interface and for server requests
Implements a database communication interface
React Library, API browser
MainServer App.js
Client working with the database
Scripts for processing actions from the interface
User interface

Figure 13: Component diagram.

The diagram of components clearly shows which parts are presented for development and based on which packages and modules it is necessary to implement the planned functionality. The developed charts will help application development consistently and implement the described functionality in the information system. This part of the work contains a detailed description and description of all the main chapters and elements of the user interface of the information system and a justification of the general purpose of the system. All the main components of the interface, such as the main menu, map, help or form of adding a new place, are described separately, with examples of use, images of graphic elements and symbols and a complete description of algorithms and results of actions on aspects of the information system. Use and interaction of parts with each other and directly with the user.

The main components of the user interface of the software product that have been implemented are:

- main page;
- main menu;
- map;
- detailed information about places;
- categorization of locations;
- determining the inclusiveness of areas;
- adding a new place;
- help menu and popular questions;
- contact and developer contact page.

It can be concluded that the implemented application functions meet the objectives and requirements set during the development of the system concept and that the executed application provides convenience and accessibility for its use by users and will perform its functions, which is an indicator of its feasibility.

4. Experiments

4.1. Choice and substantiation of methods of problem-solving

Therefore, to implement the task of creating an information system for finding inclusive places for people with disabilities in the city of Lviv and other Ukrainian cities requires at least two developers, stages of development should organize this work. System development, development, creation, testing, writing technical documentation for the user and further support of the future information system require precise planning of responsibilities. This planning will provide an opportunity to clearly and efficiently allocate time to develop this information system.

The process of implementing the information system took place in successive stages. With this in mind, it was possible to use time resources most efficiently. Therefore, the design, testing, implementation and description process took place without the possibility of assuming that time is used spontaneously in the form of a vaguely formed process, which led to the collapse of the campaign in information system development and inability to fully implement the described functionality or part of the software product. Furthermore, the same lack of time could lead to defective testing of already developed functionality, which would affect the quality of the product received by end-users and the inability to coordinate parts of the program designed by individual developers.

The time was used with the available capabilities of developers with a careful and detailed discussion process, consideration of all comments, features of specific details, analysis of future risks in the development between the developers of this information system. The terms of execution of the planned parts were also agreed with the head of the educational institution appointed to review this work. As a result of coordination, the plan of creating the given information system, which separately for each of the developers is described in detail in this part, was received.

The development process took place following this stage, with the creation of the result that thoroughly performs and corresponds to this work's theme. To begin with, the stages of development of the client part of the web service for finding inclusive places for people with disabilities are described. As a result of planning, the locations given in Table 4 were deduced. Then, similar to creating general stages, it was planned to develop steps separately to implement the client-side for the web browser. The sequence of these stages is shown in Table 5.

Table 4

General stages of software product development

Stage number	Name
1	planning
2	designing
3	development
4	filling
5	testing
6	placing
7	documentation
8	support

Table 5

Stages of development of the client side of the software product

Stage number	Name
1	planning
2	designing
3	creating the layout of the main page
4	card connection
5	search implementation
6	entering accessibility marks in the map
7	entering accessibility marks in the map
8	development of the main menu
9	creating a form to add a location
10	creating a documentation page
11	ensuring interaction between the elements
12	connection to the database
13	design of site elements
14	filling the site

15	testing
16	placing
17	documentation
18	support

The bottom section describes the development of the client-side. The next will tell the development and server.

4.1.1. Site layout and main page

At this stage, the design of the image of the main page of the information system in the scheme and its other subpages, a list of them, and all functions that the description can use. Next, it is necessary to develop the interaction between the pages according to the details of the project. The expanded set of graphic elements of pages describes the list of buttons existing on pages. A page design has been developed for this information system, which shows the location of the controls for the end-user, its list on other pages. Placing elements on the map is a mandatory part of this section and programming their positioning for the end-user. All this must be done before writing the code. To develop this interface, it was decided that the technologies for executing Javascript client-side scripts, which were its react.js library, would be used. Using this library, the solution of dynamic content generation on the client-side was implemented, which significantly accelerated the loading of the client page. With the help of this library, components such as the header and others were created using the principles of the reactor, which allowed their reuse of the developed features and their functionality without high costs for adaptation to new pages. The peculiarity of the reaction allowed for ease of operation to make other components such as a map, menu buttons in separate files. In the file header, the reserved word import describes the files that will be attached to the project to display them on the main page. A feature of the React library allows you to write HTML code together or separately from CSS code in the resulting method. Several functions have been developed for the home page window, such as adapting the preview window to the browser window size, tracking client clicks on the map, following functions of the map by the user, and reactions of event handlers to these actions. To ensure the readability of the code, most of the parts of the available handlers during the development process were divided into categories and made into separate files with the appropriate names. It also allows you to easily navigate the project because not all functions are called immediately when the page loads, but only in response to user action.

4.1.2. Card connection

For this web service, the map is the primary tool that users will use. The main library for working with the map is OpenDataMap. The functionality of connection to the react library is described in the library used in this project called Leaflet. In the map component class during development, handlers handle user actions that directly affect the map or lead to changes in the display scale of the map. In addition, the scripts for downloading data for this map, adding elements to the map and third-party actions that are not affected by the user but provide the functionality of the atomic operation of this map. In addition to the main events, the mechanisms for requesting data about particular objects and the mechanisms for requesting this data through predefined APIs are described.

4.1.3. Filter by category

Detailed information about the types of objects on the map does not need to be implemented and redistributed, as this functionality is provided together with the OpenStreetMap access library. From the received maps, all elements have their own type, which allows you to filter the searched objects according to their type. This feature also adds to the need to attach location object icons to the map at the appropriate location. Accordingly, and the accessibility of these facilities for people with inclusive needs, the following categories of facilities were implemented: trade, food, transport, leisure, housing, tourism, education, administration, health, finance, sports, toilets. In addition to these filters, an add-on to this map is implemented on the server-side in a database to store data about objects on the map according to their accessibility for people with inclusive needs. A more detailed description is provided in the next section. For these, additional filters are implemented in the interface to obtain location data:

- Filter seats partially equipped for the disabled;
- Filter places partially accessible with an accessible toilet;
- Filter seats are fully prepared for the disabled;
- Filter seats are fully accessible with an affordable toilet;
- The filter of areas about which the user can provide information;
- Filter places that are not available.

For these filters to function correctly, an appropriate handler has been created on the server-side.

4.1.4. The output of accessibility marks on the map

According to the data in the previous section, data on the adaptation of public places is not hosted on the OpenStreetMap server. Because of this, it is necessary to implement a functionality that will perform the overlay of area icons corresponding to the above filters, regardless of the already existing and submitted icons or supplementing information about them. To do this, you need to implement a js script which, in addition to general location information from the OpenStreetMap server, received location information from the application server API to supplement it about accessibility for people with disabilities. All places will be displayed on the map as markers. After adding bookmarks to the functionality of the main page, event handlers are added when the user clicks on the titles to download information about the location because this information in connection with the performance tasks will be loaded only as needed.

4.1.5. Add a new location

The page of the form for entering data about the existing location, data about which are currently not available in the service, is designed as a separate page link to which is located in the site's header, namely in the main menu of the main site page. The implementation of this page is made in such a way as to redirect the user to an external Internet resource in which the elements of the form are implemented for entering such data as the name of the place; location of the area; choice of place category; choice of place subcategory; availability of entrance; the presence of stairs; the number of steps; the height of steps; the company of a ramp; sufficient space for visiting with a wheelchair; availability of a toilet; accessibility of this toilet for the disabled; whether the staff is qualified to care for patients with disabilities. After entering all this data, you can add a new location to the information system map.

4.1.6. Contact the support team

In case of questions from users of the information system, it is necessary to implement an interface for the possibility of their coverage to the development team to send an e-mail. The KoBo Toolbox was not used to build this page. By the regulations and laws on personal data processing, a warning was added to this page that his personal data would be stored on the information system server after sending the letter. A switch was developed to prevent the user from sending a letter without consent.

4.1.7. Help and FAQ page

New users of the system will have questions about how to use it properly. The listing of questions pages for the help system and such inquiries that often arise for system users is written using HTML tools. Their design is described using cascading tables of CSS styles. The link is added to the site's header on the main page of the information system by the connection with the title contacts. Accordingly, this library was imported into the file for its operation. Thus, the page of answers to the common questions of users was implemented. Also, these questions are in the database, and they can always be supplemented with new solutions.

4.2. Selection and justification of means of solving the problem

4.2.1. Creating graphic elements

During the execution of this work, the performance of the stage of image creation is calculated. When implementing the functional part, it was necessary to create many illustrative materials that would help the client of this system learn to manage and find the required information. As part of the work, button images for all types of filters and subfilters were developed. It was also necessary to create menu backgrounds and survey pages. A significant number of developed icons are used as images of buttons. Also, some of them are used as symbols on the map. For specific cases, existing and designed web programming methods of animation elements were developed or used. One of the leading developed elements is the icon of the page which will be saved and will serve as an icon of the program when the developed page of the information system the user holds in bookmarks. Graphic elements were created using Adobe Photoshop tools. A significant number of images were created without any basis. In contrast, there were images, the cause of which was taken and completed using the capabilities of this raster graphics editor.

4.2.2. Creating a database

To create the MySQL database, we used the database management environment and their development MySQL Workbench due to the proven reliability over time and ease of use. The database itself was hosted by remote MySQL [23-25]. In this environment, you can create a free database, which will be provided without interruption 99.99 per cent of the time per year. The number of databases on this service can be made, but the total volume in the free version should not exceed 250 megabytes.

4.2.3. Aspects of programming language selection

Features of markup and programming languages for work and writing the functionality of this information system are considered in the analytical section of this work. Their choice for implementing specific elements is specified in the section on the choice of methods for solving the problem. It is based on the javascript programming language due to the possibility of reusing client code on the server. The written chapter describes the rationale and type of selected methods and tools for implementing the information system. Particular attention is paid to the software implementation of all elements of this page. Possible missing parts that need to be implemented on the server-side of this application are analyzed. The following details were presented in this information system, the description of methods and means of development of which was given in this chapter: home page menu; interactive map; filter menu to change and search for the desired content; division of locations and presentation on the map of places, if possible, for use by people with inclusive needs; form for adding new locations to the map; page with possible questions from users; a page for questions from the development team. Technologies such as JS, HTML, CSS, JQuery, React, Leaflet were used to implement this functionality of the information system.

5. Results and discussion

5.1. Description of the implementation of the client part

5.1.1. Home page

To display the main page in the reactor, a MainView file was created, which has the general structure of the export class of the reactor component container shown in Listing 4.1.

Listing 1. MainView file structure

```
import React, { Component } from 'react'
export default class MainView  extends Component {
    render() {
        return (
            <div>
                <p> Контент файлу MainView </p>
            </div>
        )
    }
}
```

As a result, after the render keyword, the content will be added that will be displayed to users on the main page. Using the styled-components library, the styles of the main page window have been added. The import keyword added the library to MainWiev, and the code highlighted in Listing 2 added programming to the colour of the links displayed in the main window.

Listing 2. CSS code to the main window

```
const StyledMainView = styled(MainView)`
  a {    color: ${colors.linkColor};    text-decoration: none;  }
> .rai-activity-indicator {
    display: inline-block;
    font-size: 36px;    line-height: 0;
    top: 50%;    left: 50%;
    position: absolute;
    transform: translate(-50%, -50%);  }`;
```

Listing 2 also describes code that specifies the dimensions of the link blocks in the main menu of their animation. The peculiarity of this code is that it is intended for a specific component and is assigned directly to the javascript file that the react library provided. Similarly, all component stylizations were added later. Finally, the section on the choice of methods for solving the problem described the necessary functionality for development. An example of the window resizing function is shown in Listing 3.

Listing 3. Window resizing processing function

```
resizeListener = () => {
  updateTouchCapability();
  this.updateViewportSizeState();  };
function updateTouchCapability() {
const body = document.body;
if (!body) return;
if (isTouchDevice()) {
  body.classList.add('is-touch-device');
} else {    body.classList.remove('is-touch-device');  }
```

Listing 3 also shows the code of the updateTouchCapability handler function, which monitors whether the user has selected an item on the map and, if selected, adds it to the history list of the selected components, which is a separate category filtering component.The given code received a part of the interface of the main window shown in Figure 14.

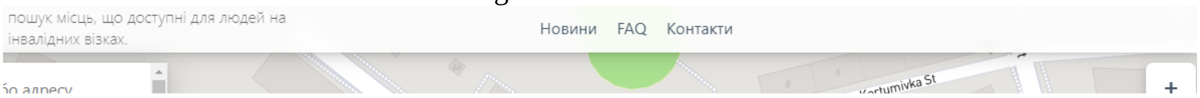

пошук місць, що доступні для людей на інвалідних візках.

Figure 14: View the main page menu.

The full completion of the writing stage of the main window component is not finished. New elements will be gradually added to it, which will expand its functionality by the description of the second section. All other parts are progressively added according to the plan. The next necessary step is to involve the functionality of the adaptive map in the already implemented functionality of event handlers of changes and activation of objects.

5.1.2. Card connection

For the map component and other components, a separate file was created with the class type of components, the reactant functionality of which implements the work with the map. The result of the rendering code for the Map component is shown in Listing 4.

Listing 4 Rendering the map component

```
     return (
         <section
           className={className}
           ref={el => (this.mapElement = el)}        role="main"
           aria-label={t`Map`}      > <div id="mapid"></div>
           {this.renderZoomInfo()}        {this.renderGeolocationError()}
                          <a
               href="http://www.openstreetmap.org/copyright"
               target="_blank"
               rel="noopener noreferrer"
             >           © OpenStreetMap |
             </a>
             &nbsp;
             <a     href="https://www.mapbox.com/map-feedback/"     target="_blank"
rel="noopener noreferrer">
               <strong>Improve this map</strong>
             </a>        </span>     </section>    );
```

After rendering in the block with id mapid in the browser window on the background of this page the interactive card which API code of a call is written in a page root will be inserted.

Similar to the previous component, all the functionality described in the subsection on the maps of the section of substantiation of problem-solving methods is implemented.

5.1.3. Filter by category

To be able to filter objects, a menu has been created to select its type. The item responsible for rendering the menu item made as a react CategoryMenu function item is shown in Listing 5.

Listing 5. CategoryMenu functional element

```
   return (
       <div className={props.className}>
         {map(rootCategories, (category, categoryId) => (
           <CategoryButton
             onFocus={props.onFocus}
             showCloseButton={showCloseButton}
             hasCircle={!showCloseButton &&
!isAccessibilityFiltered(props.accessibilityFilter)}
             accessibilityFilter={props.accessibilityFilter}
             toiletFilter={props.toiletFilter}
             key={categoryId}
             name={category.name}
             isMainCategory={!category.isSubCategory}
             category={categoryId}
             />
         ))}
       </div>
     );
```

For a specific single element, its props, responsible for the properties of specific instances of the reaction classes, prescribe the accessibilityFilter property, which is responsible for a specific filter, which is the id of this type of element. This id is sent to OpenStreetMap servers. Together with the scale, we consider the maps and the selected location to obtain specific details. Thus there is filtering both for one element and for others with corresponding replacement of value id.

Also, implementable scripts are handlers for the operation of additional filters described in the sub-item of filtering by category of the section of substantiation of problem-solving methods.

Changes were made to this react component on the client-side to supplement its data processing handler for these filters. The only key endpoint for the API was a server with a developed database with feedback on the availability of places for people with inclusive needs.

5.1.4. The output of accessibility marks on the map

The project created a MarkerIcon.js file in which the L module was used from the leaflet library, which is responsible for superimposing markers on an existing map. With this marker, the MarkerIcon class was developed, making it possible to create a quality of objects from the information system database on the map for viewing by users. The code of the react functional class that implements the overlay of things on the map is given in Listing 6.

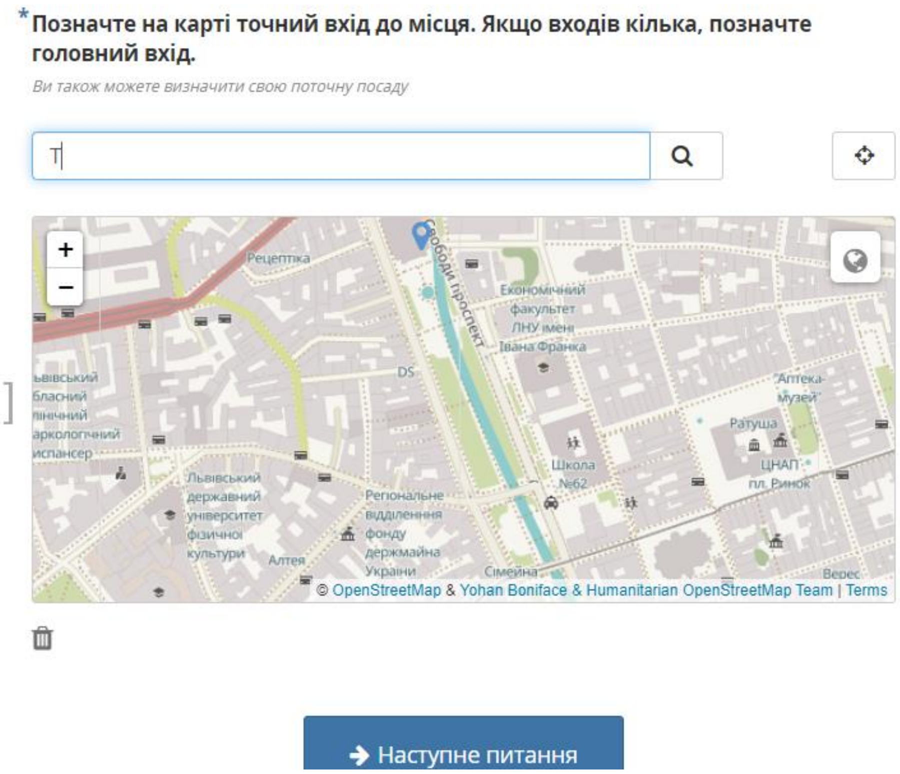
Code Listing 6. MarkerIcon class code

```
  export default class MarkerIcon extends L.Icon {
    constructor(options: Options) {
      // increased tap region for icons, rendered size might differ
      const size = 40;
      const { iconAnchorOffset, onClick, href, highlighted, accessibleName } =
options;
      const leafletOptions = {
        iconSize: new L.Point(size, size),
        iconAnchor: new L.Point(size * 0.5 + iconAnchorOffset.x, size * 0.5 +
iconAnchorOffset.y),        className: `marker-icon${highlighted ? ' highlighted-
marker' : ''}`,    };
      super(leafletOptions);
      this.onClick = onClick;    this.href = href;    this.highlighted = highlighted;
this.accessibleName = accessibleName;  }
    createIcon() {
      const link = document.createElement('a');
      link.href = this.href;

      if (this.iconSvgElement) {
        ReactDOM.render(this.iconSvgElement, link);
      }
      link.style.touchAction = 'none';

      link.addEventListener('click', (event: MouseEvent) => {
        event.preventDefault();
        this.onClick();
      });
      this._setIconStyles(link, 'icon');

      link.setAttribute('aria-label', this.accessibleName);
      return link;
    }
  }
```

The listing describes that an individual icon will be assigned to each new marker by specifying the className: `marker-icon $ {highlighted? 'highlighted-marker': ''} `based on data received from the server, according to its availability type. Specifying the iconSize property will determine the size of the icon on the map in pixels. Specifying the iconAnchor property will specify the coordinates of attaching the point to the map. For each dynamically assigned to.href token, a unique link is created about the characteristics of the object, in which the user can view existing reviews of the location, other people's evaluation of the area, and comments on comments from other users of the service.

5.1.5. Add a new location

By the sub-clause of adding new locations of the section of substantiation of problem-solving methods, a form was developed using the KoBo Toolbox service. An example of the expanded state, and precisely the element for adding a location on the map is shown in Fig. 15.



*Позначте на карті точний вхід до місця. Якщо входів кілька, позначте головний вхід.

Ви також можете визначити свою поточну посаду

Т
+
−
Рецептка
Свободи проспект
Економічний факультет ЛНУ імені Івана Франка
DS
Львівський обласний клінічний наркологічний диспансер
"Аптека-музей"
Ратуша
Школа №62
ЦНАП
пл. Ринок
Львівський державний університет фізичної культури
Алтея
Регіональне відділення фонду державна України
Сімейна
Верес
© OpenStreetMap & Yohan Boniface & Humanitarian OpenStreetMap Team | Terms
Наступне питання

Figure 15: Indication of the location of the new place.

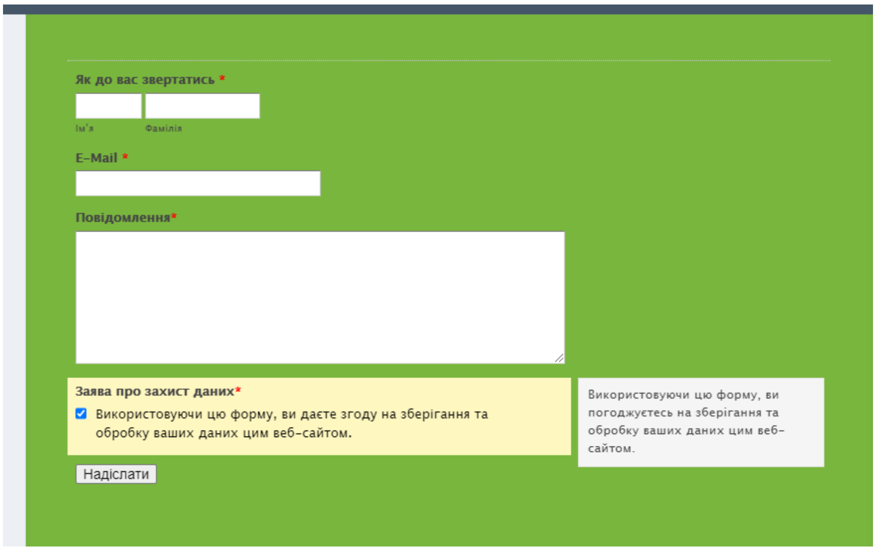
The KoBo Toolbox Internet resource was used to create this form. It is an open-source project and is designed to create conditions for user surveys and information gathering. The structure itself was created using the tools that this resource provides in its work. Making the form made it possible to choose it from the existing options, but their inconsistency with the task led to creating a new constitution. The following elements of the KoBo Toolbox were used when creating the form:

- Title for the form;
- The text box for large and small text;
- File download unit;
- Map with the ability to set the location point;
- Check box to select multiple items;
- Radio button to choose a single option among several;
- Navigation elements for switching questions;
- Navigation element of the transition to the next and previous question;
- Part of closing the survey and sending the answer;
- A drop-down list to select items, in this case, language.

After creating this form, its code was added to the header files of the main page and described a script which through the API will send data with user responses to the information system.

5.1.6. Contact the support team

A communication form has been developed by the sub-section of the communication with the support team of the problem-solving method. Due to the simplicity of the structure, this page was written in Html in combination with CSS. For its use in the project, the page was modified for the functional reaction of the SentFeedback component. For the response page to work correctly and communicate with the support team, a handler react was added to send a message using the SmtpJS service. By connecting its library via import, using the function described in Listing 7, sending a letter was implemented. As a result of the work performed, the resulting page is shown in Figure 16.



Як до вас звертатись *

Ім'я Фамілія

E-Mail *

Повідомлення*

Заява про захист даних*

☑ Використовуючи цю форму, ви даєте згоду на зберігання та обробку ваших даних цим веб-сайтом.

Використовуючи цю форму, ви погоджуєтесь на зберігання та обробку ваших даних цим веб-сайтом.

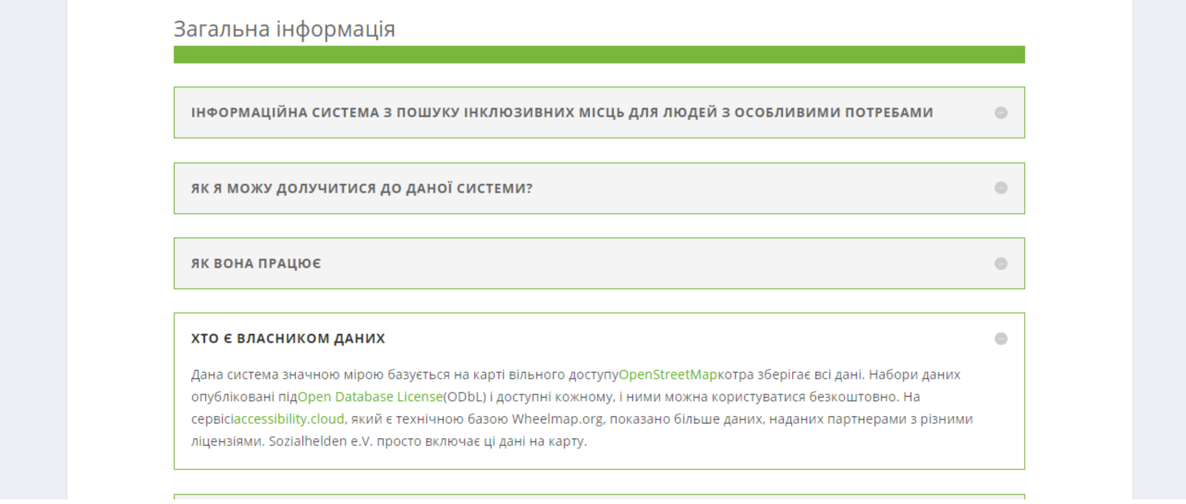
Figure 16: Developer contact page.

Listing 7. Event handler for letter feed

```
function sendEmailHandler() {
Email.send({
    Host : "smtp.mailtrap.io",     Username : this.props.userName,
    Password : this.props.password,     To : this.props.to,
    From : this.props.from,d     Subject : this.props.subject,
    Body : this.props.body });
```

5.1.7. Help and FAQ page

By the sub-item on the help page and the breadth of questions, the section justifying methods of solving the problem, a form was developed to view the system's questions. Similarly to the question page, the functional component of the reaction was implemented. A handler has been added to download questions and answers from the database and renders them for the user to view. An example of the developed window is shown in Figure 17.



Загальна інформація

ІНФОРМАЦІЙНА СИСТЕМА З ПОШУКУ ІНКЛЮЗИВНИХ МІСЦЬ ДЛЯ ЛЮДЕЙ З ОСОБЛИВИМИ ПОТРЕБАМИ

ЯК Я МОЖУ ДОЛУЧИТИСЯ ДО ДАНОЇ СИСТЕМИ?

ЯК ВОНА ПРАЦЮЄ

ХТО Є ВЛАСНИКОМ ДАНИХ

Дана система значною мірою базується на карті вільного доступуOpenStreetMapкотра зберігає всі дані. Набори даних опубліковані підOpen Database License(ODbL) і доступні кожному, і ними можна користуватися безкоштовно. На сервісіaccessibility.cloud, який є технічною базою Wheelmap.org, показано більше даних, наданих партнерами з різними ліцензіями. Sozialhelden e.V. просто включає ці дані на карту.

Figure 17: Frequently Asked Questions Page.

The JQuery library was used to collapse and expand the answers to the questions, namely the code shown in Listing 8.

Listing 8. Function to collapse and expand the response

```
$(document).ready(function(){
    $(".slideT").hide();
    $(".pointer").click(function () {
        $(this).children().slideToggle("slow")
       .siblings(".pointer > .slideT").slideUp("slow");
    });
})
```

5.1.8. Testing the functionality of the user interface

To test the functionality of the user interface, it was decided to use a functional test plan. The practical test plan is based on case studies, which are artefacts described by a set of steps and specific conditions and parameters to verify the implemented information system. For each case, there can be only the following test results:

- The result "passed" is noted if the test case was completed successfully, the expected result was obtained.
- The result "failed" is noted if the expected outcome is not obtained due to the test case.
- The result "blocked" is noted if the system hangs, does not respond to user actions as a result or during the test case execution.

The choice of the format of the tested actions should be based on displaying as many details of testing as possible. When trying the system's main menu, all menu items correspond to the purpose and are available. Actions are performed that correspond to the aim of each menu item. The test result obtained has been passed. When testing the system filter buttons, it was found that all the buttons are valid; their activation leads to filtering corresponding to the name of the controls. For other regulators, the actions corresponding to the purpose of each button are performed. The test result obtained has been passed. When testing switches and other controls, all switches lead to their stated actions. Actions are taken to prevent sending letters to the post office without signing a personal data processing agreement. The test result obtained has been passed. When testing the operation of dialogue boxes, the dialogue box has the appropriate name, provided with the necessary buttons, corresponds to its purpose. When testing the logic and sequence of actions, the program performs activities in the specified line. The test result has been passed. As a result of testing, we can safely say that the program has passed all the tests presented to it and meets all the requirements that formed the basis for the development of this program.

5.1.9. Filling pages

After completing the task of creating the main functionality of the information system, the question arose of filling it with information. The pages of the questions were supplemented with broad and deep answers that would not lead to the emergence of new questions. There is a review of existing fonts for compliance with readability during actual use for the developed service. All references that were omitted when typing static text were added. All menu items were re-checked, and headings were added and rewritten in places to highlight the depth of the issue they were revealing. As a result, all the developed graphic materials were added to the web pages in their areas and viewed in the positioning places to match the chosen locations. There was also a small amount of data on locations that people with inclusive needs can visit in Lviv. Graphic material is added to the description. Users will use existing place descriptions to follow their example to describe new ones or expand them with new notes.

5.2. Description of the implementation of the server part

5.2.1. Creating a server kernel, setting endpoints

By the subsection on the methods laid on the server-side, the implementation of the server-side was performed using the Express library. Using this library, write scripts to process requests from the server. The code from the file is shown in the example in Listing 9.

Code Listing 9. Main file code

```
const express = require('express');
const config = require('config');
const mySQLconnection = require("./local_modules/mySql/mysql");
const app = express();
const PORT = config.get("port") || 5000;
async function start() {
    app.listen(PORT, () =>console.log(`server was started on port ${PORT} ...`));
}
start();
```

To install modules, use the command npm install module, in which module ¬– the module's name being installed. The following modules were used to write this project:

- Config – a module that allows you to make individual server settings in a separate config.json file. It speeds up refactoring and code changes in the future for the different development of the same project;
- Express;
- Mysql – ORM module for working with MySQL databases. Easy to use, to establish a connection, the configuration of access to the database was included in the file config.json;
- React – for writing the client part, described in chapter 3;
- Webpack – used to configure automatic assembly of a project from individual files to one. It is used on the client-side when many js scripts written on a reaction are combined into one mimicked script. Because Javascript on the client-side is accessible and bulky in size after coding, and the mimicry process removes different names from the script and replaces them with simple and abbreviated ones, all spaces and hyphens are removed from the file. It makes the file simpler and smaller in size, which speeds up its data transfer. Changing the names protects the script from unauthorized modification, which is challenging to perform even for experienced programmers after mimicry.

Other libraries and modules were also used in the development process. That ensured the full functioning of the project.

The main file code listing describes how to connect the main modules via the required command.

Below is the file code app = express (); An instance of the express class was created from the express library, responsible for routing and creating the daemon server, which is a process that runs in a constant loop, waiting for requests from software-installed ports. After that, the configuration port of the server on which the requests from the client-side of the information system will be listened to with the code PORT = config.get ("port") || 5000 ;. Also, in this script, the possibility of missing or deleting the configuration file and replacing the port with the default value of 5000 was calculated.

The app. listen method started the server to listen on the port. This method takes two parameters, the listening port number and the handler function, which will be executed in case of a successful start of the server. The next step is to create a database and connect it to an existing project.

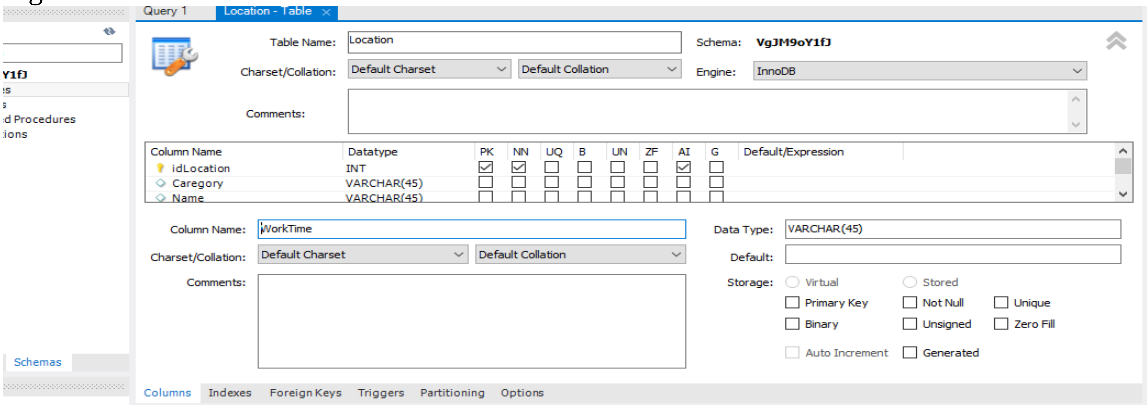
5.2.2. Creating a server database

The MySQL Workbench database management and development environment were used to create the MySQL database. The database itself was hosted by remote MySQL [25]. In this environment, you can create a free database, which will be provided without interruption 99.99 per cent of the time per year. After registering in this web service, I will receive an access file to the created database with the content shown in Listing 10.

Code Listing 10. Database access file

```
"mysql": {
    "host": "remotemysql.com",
    "user": "USER",
    "password": "PASSWORD",
    "database": "DATABASEMAP"  }
```

The result was a file in which configurations are stored for access to the database in remote systems or remote software access, which was used in this information system. In the file, the user field is responsible for the user name to connect to the database. The password field is responsible for the connection password. The database is the name of the database. Finally, the host field corresponds to the URL of the host link to which the connection will take place. Now let's start the process of creating a database using MySQL Workbench tools. An example of creating a table with location data is shown in Fig. 18.



Query 1 | Location - Table

Y1fJ
:s
s
:d Procedures
:ions
Schemas

Table Name: Location
Schema: VgJM9oY1fJ
Charset/Collation: Default Charset | Default Collation
Engine: InnoDB
Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
idLocation	INT	☑	☑	□	□	□	□	☑	□	
Caregory	VARCHAR(45)	□	□	□	□	□	□	□	□	
Name	VARCHAR(45)	□	□	□	□	□	□	□	□	

Column Name: WorkTime
Data Type: VARCHAR(45)
Charset/Collation: Default Charset | Default Collation
Default:
Comments:
Storage: ○ Virtual ○ Stored
□ Primary Key □ Not Null □ Unique
□ Binary □ Unsigned □ Zero Fill
□ Auto Increment □ Generated

Columns | Indexes | Foreign Keys | Triggers | Partitioning | Options

Figure 18: Creating a spreadsheet in MySQL Workbench.

Similarly, tables have been created such as Staff, responsible for the number of descriptions of employees and their suitability for serving people with inclusive needs in the facility, and WC, accountable for the size of toilet seats and their accessibility to people. The scheme of the created doubles is given in Figure 19. In general, such tables were created as Location; Staff; WC; Enterance; Stairs; Elevator; Ramps.

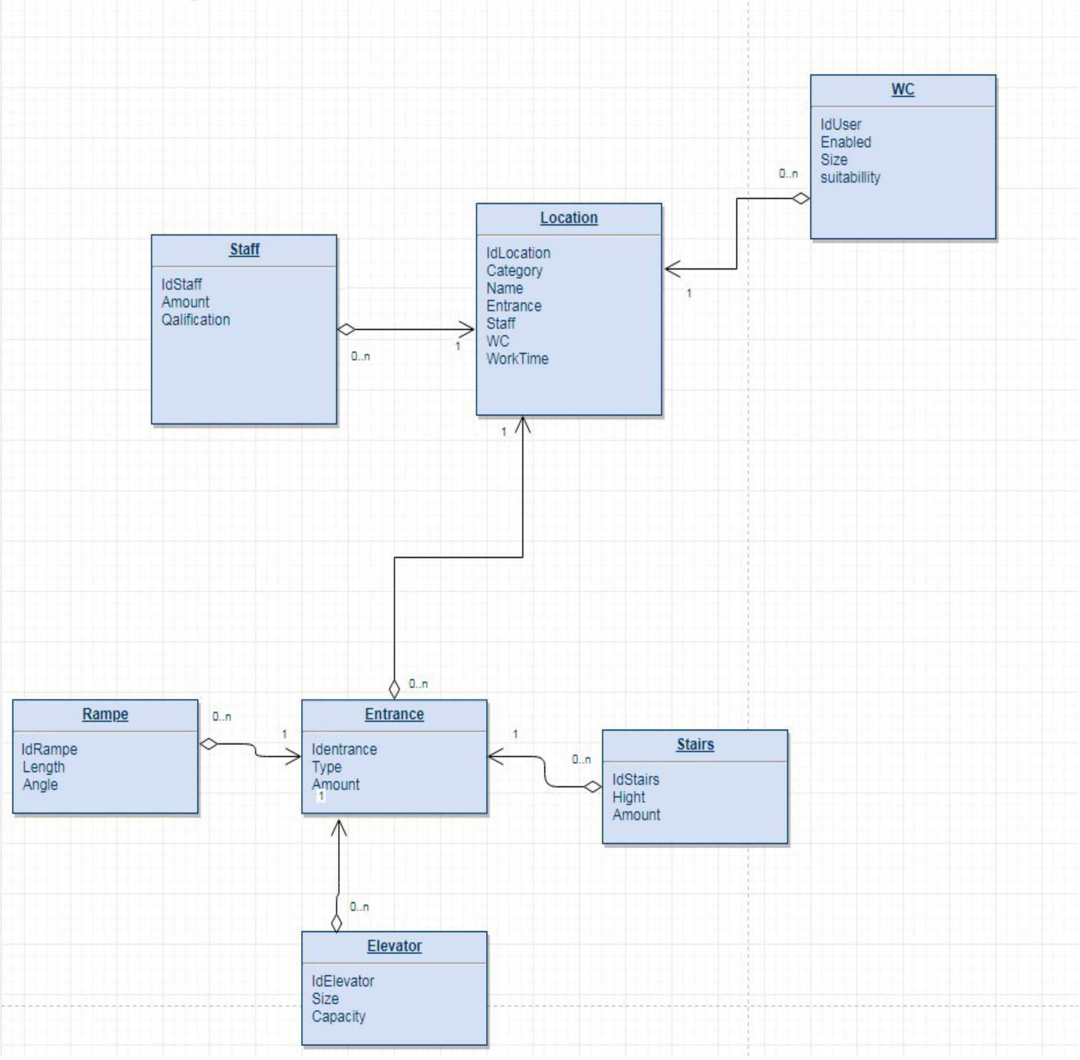


Figure 19: Scheme of created tables - Location, WC, Staff

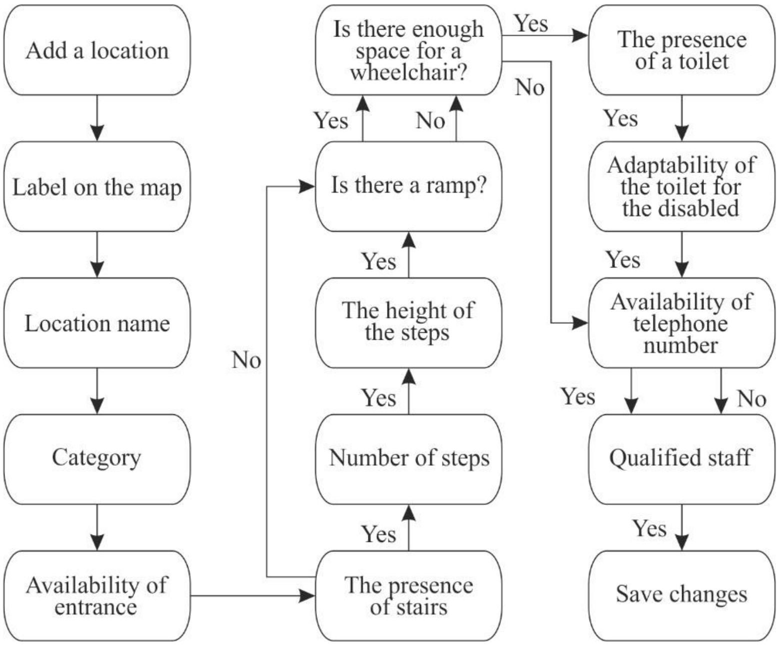


Figure 20: Example figure

5.2.3. Database module

To work with the MSSQL database from the Nodejs server programming environment, the Mysql library was used. For convenient operation, the connection module was used. The code was written in a separate connection file, which allows you to connect to the database and perform SQL queries in the places where it was connected. This logical connection structure distances the programmer from the database module and provides only the writing of SQL scripts by the data processing processes in the database. The connection module code is given in Listing 11. Also, to the main file of the server during start, the script checking the availability of a DB for remote connection was added.

Listing 11. Database connection code

```
const mysql = require("mysql");
const config = require("config");
const conectionObj= config.get('mysql');
module.exports = ()=>connection = mysql.createConnection(conectionObj) ;
```

The config.get code ('mysql') retrieves access keys from the configuration file and creates a local variable with them. With the module.exports code, we export the connection function for further use.

5.2.4. Module for adding reviews and receiving them

All actions with client requests to process the addition and receipt of feedback are transmitted to the server through the API on the client-side via the HTTP protocol with different requests according to a particular CRUD operation. Accordingly, for each request, a separate URL path is called remotely from the client page, processed on the server, and the result is sent back to the user.

The express library was used to write script processing scripts. With this library, you can quickly write client-side request processing scripts. The feedback.routes.js module has been created to work with queries from the client that will add reviews to the database. A line of app.use code ("/ API / feedback", require ("./routes/feedback.routes")); has been added to the main project file, indicating that the localhost / path must be used when requesting feedback. api / feedback.

The survey is performed according to the algorithm shown in the block diagram given in the appendices to add a response to the database. When adding a location, the user is consistently asked for such data as a label on the map, the name of the area, the name of the category, the presence of an entrance to this location, the company of stairs. In the absence of stairs, the survey takes the step of having a ramp. In the case of the presence of stairs, there is a transition to the question of the number of steps and the height of the steps and, after that, the presence of a ramp. Next is whether there is enough space for the cart. If not, the algorithm moves on to the question of the availability of a telephone number. If there is enough space, then additional questions are asked about the availability of the toilet, its accessibility for the disabled and even then the question about the phone number.

```
const { Router } = require("express");
const bodyParser = require("body-parser");
const mySQLconnection = require("../local_modules/mySql/mysql");
const { checkBody } = require("../middleware/reqDataChecker");
const config = require("config");
const connection = mySQLconnection();
const router = Router();
const tableName = config.get("tableName");
router.use(bodyParser.json());
router.use(checkBody);
// /api/feedback/
router.delete("/:id/delete", (req, res) => {
  let id = req.params["id"];
  try {
    connection.query(`DELETE FROM ${tableName} WHERE  id=${id} ;`, function (
      error,      results,      fields
    ) {
      if (error) throw error;
      res.sendStatus(204);
    });
  } catch (e) {
    console.log("error", e.message);
  }});
```

The beginning of the file describes the libraries and modules for which you need to work, such as the previously created module for working with the MySQL database. Also, for each request, the received data is parsed into objects using the written reqDataChecker module. The code router.delete ("/: id / delete") indicates that in the application, for a query of type delete on a path with a variable value id with the string/delete, a processing function will be executed, which will generate a question to the database by connection.query from SQL command (`DELETE FROM $ {tableName} WHERE id = $ {id};`, whose values tableName and id are taken from the client request.

Similarly, data is taken from the client's request to execute queries to the database, which implements a module for working with feedback on locations for people with inclusive needs.

5.2.5. Module for working with filters

Similar to the previous question, a module for working with filters were implemented. Its essence lies in the fact that when adding a location, the user indicated the type of accessibility to a public place. For this type, the areas that were created together with the specified ones were saved. When the user wants to find places with a given level of accessibility, the program executes a request to obtain locations from the database and give them to the front page interface, which will already provide them to the user. To work with filters, a line of code app.use ("/ api / filters", require ("./ routes / filters.routes") has been added to the main server processing file); which points to the filter request processing file will be called by the API / filters path. Similar to the created previous module, in this one, all scripts of scenarios of data processing from the client concerning filtering of the displayed data on a map were written. An example of requesting locations in the database according to the filter type is given in Listing 13. In this script, the user receives the filter's id, for which you need to get a list of locations. The next step in this id is an SQL query to the database, and the resulting result set is sent to the client that executed this query.

Code Listing 13. Request location code for the filter.

```
// /api/ filters /
router.get("/:id", (req, res) => {  let id = req.params["id"];
     try {
       connection.query(`SELECT * FROM ${tableName} WHERE  id=${id} ;`, function (
error,      results,      fields
       ) {      if (error) throw error;

         res.status(200).send({ results });
       });
     } catch (e) {
       console.log("error", e.message);
     }
   });
```

Also, in this module, scripts of processing and locations' locations from tables of areas of the corresponding filters are written.

5.2.6. Module for working with questions

Also, similarly, in accordance with the task, a module for processing requests was created to work with frequently asked questions by users. In order not to constantly add questions manually, changing them in a static Html file, a panel was created to edit it with the ability to add and delete questions along with the answers. Only site administrators have access to this panel.

In this module, scripts were written that received a request to add text and answer questions and execute it. Similarly, a request to delete the response is completed, which provides the identification number of the question to be deleted. After receiving by executing an SQL query to delete the record from the database, the script removes the required question.

5.2.7. Testing

Similarly to client-side testing, server-side testing was performed. To test the functionality of the server scripts, it was also decided to use a functional test plan. Also, stretching the server side requires client-side functionality because it is possible to perform the test without special tools.

In addition, testing was performed with a set of special requests and queried through API query programs, to view the body of the query and the resulting data set or the response received from the server due to the end of the trusted script. When testing the system startup, the system starts after completion without any problems. In case of an emergency shutdown, these reasons are logged in the appropriate files. Actions are performed that correspond to the described events before the implementation of the server-side. The test result obtained has been passed. When testing the system connection module to the database, it was found that the module performs the connection task or notifies the log file and console about the impossibility of connection, server response, or lack of Internet. The test result obtained has been passed. When testing the location filter module, it is established that all filters are created in the database. Calling filters leads to getting the resulting data set with the corresponding data to the selected location availability filter. The test result obtained has been passed.

When testing the operation of scripts for modification of frequently asked questions from the client, it was found that the script performs the task set before it, namely, adds, modifies or deletes questions from the administrator of the page of frequently asked questions from customers. Test result passed.

As a result of testing the functionality of the server party, it is possible to tell that all functionality works according to the set tasks before its creation. Therefore, the data server crash scripts, reboot scripts and server error logging scripts are prescribed.

The result indicates that the server is ready for full daily use.

This chapter describes and gives examples of creating the core of this information system, which will not leave questions in understanding its basic functionality. The next step was to connect the open-source map to this project and a detailed technical description of this action. Then, a detailed description of the setting is up and writing the client-side functionality to implement the filtering of elements on the map to display only those that the client has selected. Also, a necessary component of the functionality is to display points on the map that are not set by the developers of OpenStreetMap, but will be added by users of this information system. This chapter highlights the main scope of work that needed to be implemented to build a full-fledged client-side as:

- Selection based on the future use of page layout;
- Description and development of the functionality declared for implementation in this project;
- Creation of software components for reuse and joint service in the information system;
- Drawing illustration elements, entering content for client pages.

In this chapter, work was carried out to implement the server part of the information system to find inclusive places for people with special needs. First, creating the main file and the starting point of the server boot is described in detail. Then, creating an essential element of storage and data manipulation of the application, namely the database, is described. Also described is developing a module that provides the ability to manipulate data from the completed application. Finally, by the tasks that should support the client's work, scripts for processing requests from the client are described and implemented, which will provide information for the entire operation of the interface of the browser version of the information system. Testing scripts were written for each developed script, indicating the successful implementation of the tasks and the readiness of the server for operation.

6. Conclusion

This work describes the process and result of developing an information system for finding inclusive locations and obtaining detailed and relevant information about them for people with disabilities. This information system is implemented in the form of a browser version and a mobile application. To confirm the relevance and usefulness of the developed system, a study of the problem, analysis of existing analogues and their comparison are with specific requirements for the service.

Taking into account this study, as well as a detailed analysis of the requirements and comments set in the task of the work, a project was created, and the described software product was implemented. The developed application allows users to obtain information about the locations entered into the system database using the map or search function implemented in the application interface and perform other operations using the tools of the map and search algorithm.

An essential feature of the user interface and server part of the application is the sorting of all available locations into categories according to their profile business activities and the level of suitability of these places for use by people with special needs.

In addition, in the information system, an important role is given to the interaction of users with the service and developers. An example of such interaction is the ability for users to independently add new places to the application database, providing structured data about them using specially designed algorithms and appropriate elements of the user interface. Users also have the opportunity to provide their feedback and comments on the use of the application, suggest improvements, report bugs in the program, and receive answers to questions about using the service. As a result, the developed information system performs the tasks set before it, offers a solution to the current problem, and the implemented interface and functionality provide efficient and convenient use for the target audience.

7. References

[1] L. M. Tsyhanets′, Vytyah z bazy Ministerstva Sotsial′noyi polityky. URL: https://dostup.pravda.com.ua/request/kilkist_liudiei_z_invalidnistiu.

[2] OpenStreetMap. URL: https://en.wikipedia.org/wiki/OpenStreetMap.

[3] GIT. URL: https://en.wikipedia.org/wiki/Git.

[4] J. Hamano, Announce: Git 1.0.0, 2005.

[5] Leading the web to its full potential. URL: https://www.w3.org/.

[6] J. Duckett, HTML and CSS: Design and Build Websites, 2011.

[7] DOM Enlightenment. Cody Lindley, 2013.

[8] E. Burchard, Refactoring JavaScript: Turning Bad Code Into Good Code, 2017.

[9] E. Eric, Programming JavaScript Applications: Robust Web Architecture with Node, HTML5, and Moderns JS Libraries.

[10] jQuery in Action, Third Edition.Bear Bibeault, Yehuda Katz., 2015

[11] jQuery Succinctly. Cody Lindley., 2014

[12] J. S. Blanchard., Applied JQuery: Develop and Design, 2011.

[13] D. Herman, Effective JavaScript: 68 Specific Ways to Harness the Power of JavaScript, 2012.

[14] E. Jarrell, Fetching API Data with React.JS. URL: https://blog.hellojs.org/fetching–api–data–with–react–js–460fe8bbf8f2.

[15] M. Dere, How to integrate create–react–app with all the libraries you need to make a great app, 2018.

[16] M. T. Thomas, React in Action, 2018.

[17] L. Richardson, RESTful Web APIs: Services for a Changing World 1st Edition, 2013.

[18] Javatpoint. URL: https://www.javatpoint.com/pros–and–cons–of–react.

[19] P. Sturgeon, Build APIs You Won't Hate: Everyone and Their Dog Wants an API, So You Should Probably Learn how to Build Them, 2015.

[20] S. Clarke, Measuring API Usability, 2004.

[21] J. Bloch, How to design a good API and why it matters.

[22] Introducing JSON. URL: https://json.org/.

[23] T. Reynolds, A. Mahan, Writing Modern JavaScript with TypeScript. URL: http://www.instrument.com/articles/writing–modern–javascript–with–typescript.

[24] P. Krill, React: Making faster, smoother UIs for data–driven Web apps, 2014.

[25] FREE MYSQL. URL: https://remotemysql.com/.

[26] V. Lytvynenko, N. Savina, J. Krejci, M. Voronenko, M. Yakobchuk, O. Kryvoruchko, Bayesian Networks' Development Based on Noisy-MAX Nodes for Modeling Investment Processes in Transport, volume Vol-2386 of CEUR Workshop Proceedings, 2019, pp. 1-10.

[27] I. Krislata, A. Katrenko, V. Lytvyn, V. Vysotska, Y. Burov, Traffic flows system development for smart city, volume Vol-2565 of CEUR Workshop Proceedings, 2020, pp. 280-294.

[28] A. Katrenko, I. Krislata, O. Veres, O. Oborska, T. Basyuk, A. Vasyliuk, I. Rishnyak, N. Demyanovskyi, O. Meh, Development of Traffic Flows and Smart Parking System for Smart City, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 730-745.

[29] M. Bublyk, V. Vysotska, Y. Matseliukh, Development of intelligent system for visual passenger flows simulation of public transport in Smart City based on neural network, volume Vol-2870 of CEUR Workshop Proceedings, 2021, pp. 1087-1138.

[30] M. Bublyk, Y. Matseliukh, U. Motorniuk, M. Terebukh, Intelligent System of Passenger Transportation by Autopiloted Electric Buses in Smart City, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 1280-1294.

[31] Y. Matseliukh, V. Vysotska, M. Bublyk, Intelligent System of Visual Simulation of Passenger Flows, volume Vol-2604 of CEUR workshop proceedings, 2020, pp. 906-920.

[32] O. Lozynska, V. Savchuk, V. Pasichnyk, Individual Sign Translator Component of Tourist Information System, volume 1080 of Advances in Intelligent Systems and Computing IV, Springer Nature Switzerland AG 2020, Springer, Cham, 2020, pp. 593-601.

[33] O. Y. Boreiko, V. M. Teslyuk, A. Zelinskyy, O. Berezsky, Development of models and means of the server part of the system for passenger traffic registration of public transport in the "smart" city, volume 1(2-85) of Eastern-European Journal of Enterprise Technologies, 2017, pp. 40-47.