Development of the information system for navigation in modern university campus

Liudmyla E. Gryzun[1], Oleksandr V. Shcherbakov[1] and Bogdan O. Bida[1]

[1]*Simon Kuznets Kharkiv National University of Economics, 9-a Nauky Ave., 61166, Kharkiv, Ukraine*

Abstract

According to the aims, the paper highlights algorithmic, interface and technological solutions for the development of the information system for navigation in university campus. In the process of searching for algorithmic and interface solutions, the peculiarities of the scientific and applied problem of indoor navigation were analyzed, as well as the capabilities of selected analogues of navigation systems implementing similar functions of this subject domain were evaluated. It was concluded that the analyzed analogues, despite their significant functionality, have the set of limitations. The specification of functional and non-functional requirements for the university navigation system was carried out, its architecture was defined as a set of interconnected modules, for the implementation of which appropriate interface and algorithmic solutions were elaborated and covered. The main stages of design and development of the university navigation system in the context of the elaborated solutions are highlighted. The functionality of the implemented system is characterized. It was established that during the design it was possible to overcome the main limitations inherent in similar systems implementing indoor navigation. The results of the system implementation in the educational practice of a national university are highlighted. Feedback from users received during the approbation testified the feasibility of developing and using the information system for navigation in university campus. The prospects for further work are formulated.

Keywords

indoor navigation, information system for navigation, university campus, web technologies, problem of the shortest path search

1. Introduction

An effective solution to the scientific and applied problem of navigation and orientation in closed spaces is gaining more and more relevance today. Most of the known technological solutions to this problem are aimed at providing informational support for users of different statuses in navigating in buildings with a complex, multi-level and branched structure and topology (office and cultural premises, commercial centers, cultural objects, parking lots, etc.).

Nowadays, easy, quick and reliable navigation in the university campus and its premises requires special attention, because the buildings and spots of a modern university usually have

AREdu 2022: 5th International Workshop on Augmented Reality in Education, May 23, 2022, Kryvyi Rih, Ukraine

lgr2007@ukr.net (L. E. Gryzun); oleksandr.shcherbakov@hneu.net (O. V. Shcherbakov); bida1bohdan@hneu.net (B. O. Bida)
https://kafis.hneu.net/grizun-lyudmila-eduardivna/ (L. E. Gryzun);
https://kafis.hneu.net/shherbakov-oleksandr-vsevolodovich/ (O. V. Shcherbakov);
https://kafis.hneu.net/bida-bogdan-oleksiyovych/ (B. O. Bida)
0000-0002-5274-5624 (L. E. Gryzun); 0000-0002-6473-7629 (O. V. Shcherbakov); 0000-0003-0648-5507 (B. O. Bida)

CEUR Workshop Proceedings (CEUR-WS.org)

a complex structure with a large number of stores, studies, offices and locations of various purposes. Modern universities are important educational, scientific and cultural hubs that are daily attended by a large number of people on different purposes. Applicants, students, participants of competitions and tournaments, new tutoring staff and visitors who are unfamiliar with the location of its buildings and their topology have a need for convenient navigation in the premises of the university, which makes urgent the design and development of the appropriate information system for navigation.

According to Idrees et al. [1], Savchuk and Pasichnyk [2], information and navigation system is an electronic system that provides navigation solving the problem of determining the location of the user, laying a route from one point to another, and also supplementing the above functionality with certain useful information to facilitate users' orientation.

A fairly wide range of information and communication technologies are used to build systems that use information about the user's location inside the premises. In the context of orientation in the university premises, the involvement of web technologies for solving navigation tasks has definite advantages, as web applications do not require pre-installed software (other than browsers, which are usually available in every modern device), do not need to be installed on the device, and enable to use browser cache and PWA technology to provide work offline.

The analysis of scientific and applied sources testifies that there are various technological approaches to solving the problems of orientation [3, 1, 2]. In the context of our research, the focus should be on the analysis of applications providing indoor navigation. Such applications cannot use the capabilities of GPS navigation (like the systems such as Google Maps, 2Gis, OpenStreetMaps, etc.), due to the fact that navigation takes place on a small scale and usually indoor. Therefore, the involvement of GPS navigation technologies will lead to significant errors. Thus, most navigation systems for premises use indoor navigation.

According to the definition, indoor navigation means detecting objects (devices or people) and orientation inside the building using radio waves, acoustic signals, wireless networks (Wi-Fi), infrared sensors, magnetic field, Bluetooth technology or other means [3, 4].

However, the challenge of such navigation is that each building where it is necessary to orient, must have special devices configured to work with the navigation system. According to recent studies, it makes the development of such a system expensive and long-term. It is also pointed out that it causes certain difficulties to its testing, maintenance and expansion opportunities [3, 4, 5, 6].

In addition, researchers note that indoor navigation technologies based on the processing of visual images, as well as QR codes, are underused and unjustifiably forgotten, although they require less investments and have significant prospects for their using for the development of mobile and web applications [1, 6].

Thus, the analysis of the functionality of existing systems for indoor navigation testify their shortcomings and limitations (highlighted below) which cause the necessity of searching new efficient approaches to the design of information system for indoor navigation.

The aim of the paper is to elaborate and cover algorithmic, interface and technological solutions for the development of the information system for navigation in university campus.

2. Theoretical framework

Theoretical background of the work is made by the analysis of the (1) the applications which realize similar functions of the subject domain and (2) approaches to solving the mathematical problem of the shortest path search.

To analyze the possibilities of the applications which realize similar functions of the subject domain, there were selected several services that provide orientation in the premises using indoor navigation and implement the following general functions: visualization of the building/floor layout, virtual movement on the floor (viewing parts of the room), providing the user with visual information to simplify perception of the layout and intuitive navigation.

In the progress of the analysis, we focused on the evaluation of the analogues of indoor navigation systems, according to the following criteria: (1) route building functionality; (2) adaptability of the interface; (3) convenience and ergonomics of the interface; (4) quality of visualization of graphic content; (5) design aesthetics; (6) price; (7) technological possibilities for use in devices of various types; (8) possibilities of functionality extension and refining.

According to the said criteria and based on the resources [7, 8, 9, 10, 11], there were discussed the following applications:

- AAU Map (software for navigation in buildings and area of Aalborg University (Denmark));
- Mapsindoors (extension of the MapsPeople platform built on Google Maps technology which ensures the transition from external to internal navigation and its rapid implementation);
- AnyPlace (a free open indoor navigation service that provides location determination using smartphones);
- Situm Mapping Tool (indoor navigation application that accompanies visitors in a specific building to find their way in real time mode);
- BSB Navigator (a smartphone application that navigates through the library on Ludwigstrasse in Munich and uses beacon technology based on smartphone Bluetooth).

The analysis carried out according to the specified criteria proved that despite the significant functionality of these indoor navigation services, they have the following set of drawbacks and limitations. Most of the analyzed analogues are able to provide navigation inside a specific building, but they are not suitable for use in other premises. At the same time, they do not have a sufficiently developed functionality of extension and refinement. They do not provide users with language localization. They are either web services or mobile applications.

These limitations cause the search for appropriate algorithmic, interface, and technological approaches to design of the information system for university navigation. It is also relevant to take them into account when determining the functional and non-functional requirements for the said system.

Apart from this, it is essential to discuss approaches to solving the mathematical problem of the shortest path search, which is going to be solved by the module of navigation system as one of its functions.

The mathematical formulation of the problem can be presented as follows. Let the graph is given:

$$G = (V, E)$$

Here V is a set of vertices, E is a set of edges with given values c_{ij} on each edge (i, j). In terms of our subject area, the vertices of the graph are the various locations of the room and the QR code labels; the edges of the graph are the physical connections between the vertices that allow to get to them. It is necessary to find the shortest path between the selected vertices. This means that it is necessary to find such a path P between the vertices v_1 and v_2 $P(v_1, v_2, ..., v_n)$, that

$$\sum_{i=1}^{n} C_{ij} \to \min,$$

This problem in the scientific sources is also called "The Shortest Path Problem" (SPP) [12, 13, 14].

There are several different but related problems in which it is necessary to find out the shortest paths in a graph:

1) find out the shortest path from *s* to *t* (one selected pair of vertices);
2) find out the shortest path from *s* to all vertices;
3) search for all pairs of vertices of the shortest paths.

In addition, it is also distinguished the tasks of finding the shortest paths depending on the type of graphs that are received as input data.

To solve the specific practical problem of finding the shortest path for indoor navigation, an appropriate analysis was conducted to select an adequate search algorithm among such well-known algorithms as Dijkstra, Bellman-Ford, Floyd-Warshall, Johnson's and others [12]. The main selection criteria for this problem are the properties of the graph, as well as the execution time of the algorithm. As a result, the two most popular algorithms were chosen: the Dijkstra algorithm and the Floyd-Warshall algorithm.

When selecting between them, the following characteristics were taken into account. The biggest difference between the algorithms is that Floyd's algorithm finds the shortest path between all vertices, while Dijkstra's algorithm finds the shortest path between a single vertex and all other vertices. At the same time, the costs for Dijkstra's algorithm are much higher than for Floyd's algorithm. If Dijkstra's algorithm is run *n* times, on n different vertices, the theoretical time complexity is $O(n * n^2) = O(n^3)$. In other words, when using Dijkstra's algorithm to find paths from each vertex to any other vertex, we get the same efficiency and result as using Floyd's algorithm. Of course, a specific problem can be solved by repeatedly applying Dijkstra's algorithm with successive selection of each vertex of the graph as an initial vertex. However, Floyd's algorithm is more efficient than multiple iterations of Dijkstra's algorithm. At the same time, Floyd's algorithm is much simpler to implement. In this regard, the choice was made in favor of the Floyd-Warshall algorithm.

Let us characterize its characteristics and features. Although this algorithm has the name of Floyd-Warshall, and its appearance in the information space dates back to the early 1960s, it essentially includes the works of previously published work by Roy [15] and is closely related to the Kleene's algorithm, published in 1956 [16, p. 37-40], for converting a deterministic finite automaton into a regular expression. The modern formulation of the algorithm as three nested for-loops was first described by Ingerman [17] in 1962. The Floyd-Warshall algorithm [13] is an example of dynamic programming, and was published in its currently recognized form by Floyd [18] in 1962. This allows to apply it to solve many problems (optimal routing; search for the shortest paths in graphs; transitive closure of graphs; calculation of similarity between graphs and others).

As mentioned above, the Floyd-Warshall algorithm is based on the use of the dynamic programming method, which is an alternative to solving the problem by the brute force method or greedy algorithms. In a general interpretation, this method consists of the following stages:

1. Splitting the task into smaller subtasks.
2. Finding the optimal solution of subproblems by recursive method.
3. Use of the obtained solution for problems to construct a solution to the original problem.

The step-by-step implementation of the Floyd-Warshall algorithm based on the dynamic programming method is shown in figure 1 [12].

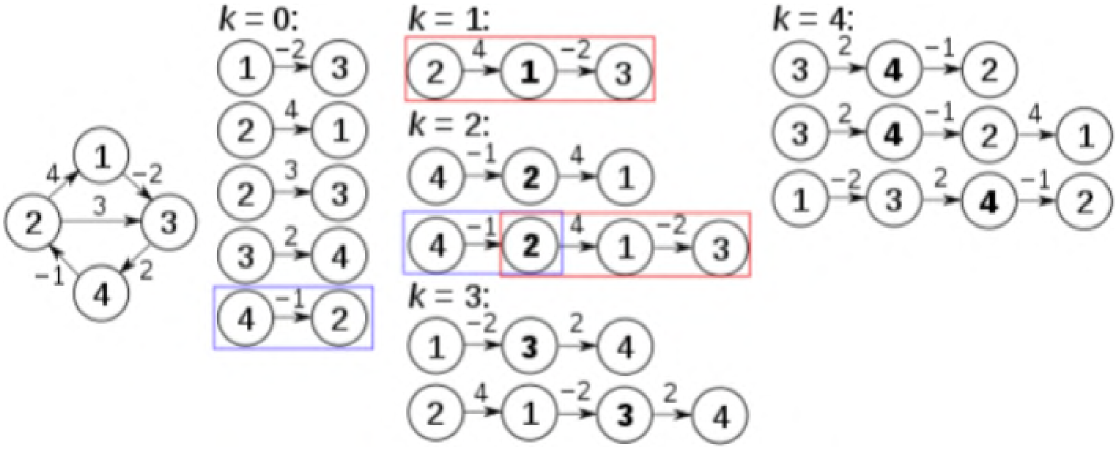

Figure 1: The step-by-step implementation of the Floyd-Warshall algorithm based on the dynamic programming method.

Characterizing this algorithmic solution, it is essential to point out that to find out the shortest paths between all vertices of the graph, it is not used brute force (which in fact is an enumeration of all possibilities that will lead to a significant time of the algorithm's operation and takes up more memory), but bottom-up dynamic programming. Therefore, all subproblems that will later be needed to solve the original problem are solved in advance, and then their solutions are used.

The presented theoretical framework is used below for elaboration of proper solutions at different stages of the navigation system design.

3. Results and discussion

In the progress of solving algorithmic, interface and technological tasks, the specification of the requirements for the whole information system was carried out by building use case diagram, minding that the system user is a university campus visitor who has the system installed on their smartphone as a web application or a web link to it. There were defined the following intended options for using the system by the user:

- reading of user instructions;
- viewing settings (including changing language localization);
- determining user's location;
- selection of the initial spot (the point that corresponds to the coordinates of location of the physical QR code relative to the building scheme);
- selection of the final spot (any location in the university premises marked on the interactive map of the campus for which the user wants to find out a route);
- receiving the built route between the initial and final spots;
- obtaining an animated visualization of the built route;
- use of a QR code scanner;
- viewing of the current floor plan;
- changing the scale of the interactive map;
- changing the current floor in the map.

The analysis of the use cases enabled to provide detailed specification of the functional requirements for the system being developed. In particular, the set system for navigation has to supply for a user:

- language localization settings;
- selection of existing locations on the interactive map;
- determining the location of the user;
- finding the optimal route from the start spot to the final spot;
- visualization of the obtained route;
- review of the current floor plan;
- free change of floors on the interactive map of university premises;
- changing the scale of the interactive map;
- providing prompts on types of locations;
- adaptability of the interface to different sizes of devices and their graphic settings.

Based on the defined functional requirements, the technological approaches to the design and development of the navigation system were substantiated. In addition, it was taken into account the need to minimize the limitations inherent in the existing analogues of navigation systems emphasized above (lack of the possibility of extension the functionality of the system, absence of adaptation to navigation in different premises and language localization; strict dependence on the platform of operation).

Thus, it was substantiated the involvement of a combination of technologies:

1) Angular as a platform for the development of complex single-page applications, which uses the TypeScript dialect of the JavaScript language to describe interface components using the principles of object-oriented programming [19];

2) SVG.js library which applies the HTML and JavaScript facilities for dynamic generation, processing and animation of vector documents, which is helpful in creating an interactive map [20];
3) reactive programming based on RxJs library [21].

Then, the specification of non-functional requirements for navigation system was provided, which were followed in the progress of development of algorithmic and interface solutions.

Based on the determined functional and non-functional requirements, the general architecture of the system was built as a set of interconnected modules: graphic module, module of detecting user's location, shortest path search module, module of the searched rout visualization.

In the context of interface solutions implemented in the navigation system, a key place in the architecture is taken by its graphic module that plays the role of an intermediary between the user and the entire logic of the application including data processing and execution of complicated algorithms of detecting the user's location and visualization of built optimal route from the start spot to the final spot. Its role is to provide the user with a convenient and intuitively clear interface for interaction with the functionality of the system, as well as visualization of the interactive map and navigation information (names of specific locations, additional marks, etc.). However, analyzing the place of the graphic module in the architecture of the navigation system, it should be emphasized that it provides not only the appearance of the application together with the means of interaction with the user, but also the logic of interaction of the data related to the visual part of the system (map scale factor, position relative to the screen, selected user setting, etc.).

In general, the graphic module can be characterized as a set of interface elements and program code that allows the user to interact with the system, influence its state, and also obtain certain information. In addition, the graphic module implements the visualization of the premises map and the user's route for his navigation. A core component of the graphic module is the interactive map of the current floor for which the said module provides the ability to change the position of the screen regarding the map as well as its scale using the touchpad of the user's device.

When designing the graphic module, its connections with other modules of the system and features of data exchange between them were considered. Therefore, the main task was the realization of data management responsible for the interface and state of the interactive map (scale factor, position of the map regarding the screen, current floor, etc.), which would ensure the correct functioning of the entire system and its further extension. Accordingly, the following approach was implemented for development of the graphic module of the university navigation system:

- implementation of interface control components in the form of a template, styles and controller, html, css and ts files for each component;
- implementation of an adaptive user interface as a grouping of all control elements with additional display logic;
- organization of the business logic of the interface in the form of services that store data and means of their processing, using reactive programming based on RxJs library;
- the organization of utilitarian classes and methods that enable to remove repetitive parts of processing that are not directly related to the logic of the whole graphic module.

In accordance with the offered approach and the above-defined functional requirements for the system, the development of graphic module was implemented as an element of the general architecture of the university navigation system which is responsible for the interface, and generation and processing of interactive map data.

Mockups and prototypes of the system interface were created, along with the design of premises schemes for each floor. Mockups were designed for maximum ergonomic use. On this purpose, the screen was divided into three interactive zones which can adapt to ease of use and accessibility according to the screen size of the user's device, scale, etc. These zones are upper area, right and left ones.

The upper zone is the search zone, where there are interface elements for establishment of the start and final spots for the searching rout. One of the most important functions of navigation system is to determine the user's location and destination spot to enable finding the optimal route or indoor orientation. It is crucial that the system does not track the user's location in real time which is not user's physical location, but close initial spot with a physical QR code. Therefore, there is a need for the user to independently choose the start and final spots for searching and visualizing the route (or only the destination spot without rout). On this purpose, the system interface provides the upper area of the screen, which contains two forms for searching. Each form consists of the poper control elements including (1) an input box where the user can input the name of a start location where the physical QR code is; (2) a drop-down list of hints based on the text entered by the user; (3) search buttons with a magnifying glass icon, which sets the entered point as the start or final spot for the current one, if such exists, and starts the route search algorithm if both points are defined (figure 2).

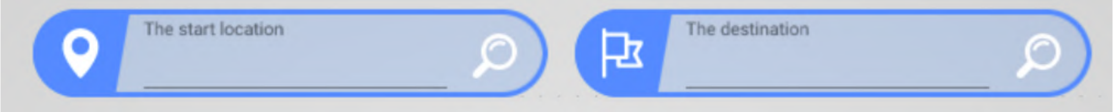

Figure 2: Two forms of the upper (search) zone of the screen.

Control elements of graphic module for changing the position on the map, its scale, etc. are placed in the left area of the screen in order not to interfere with the user's view of the map, to ensure an ergonomic arrangement without accidentally pressing on unnecessary elements with the user's finger when holding a mobile device. This zone of the screen contains map scale controls, map centering button, arrows for changing the floors and the number of the current floor (figure 3).

Graphic module also enables additional functions that are not mandatory and refer to the general control elements of the navigation system. Such control elements are placed into the right zone of the screen and include reference getting button (marked with letter "I"), settings button with a gear icon that opens a window with selection of the interface language localization, a button with a scanner icon that allows to open a QR code scanner and determine the point to which a successfully scanned code corresponds.

The color palette of the system interface was chosen based on the general physical conditions of its using (in well-lit university premises), so the image on the screen should be quite bright and contrasty. The main color of the interface light blue, white, and hues of gray with smooth

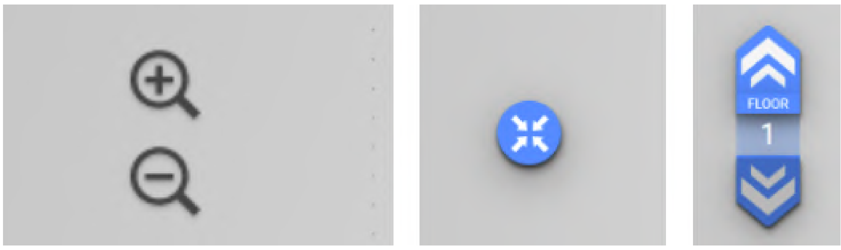

Figure 3: Interface elements of the left zone of the screen for interactive map control.

and light gradients. Soft shapes of the interface elements and various icons are used to simplify user perception.

The interface solutions implemented by the graphic module of the university navigation system also include the interactive map generation by digitizing floor layout of university premises using the vector image editor Adobe Illustrator. The results of the digitization of the floors layout generated by the graphic module of the system are used for interactive maps building. Thanks to the application of vector graphics, a map was obtained for each floor in the form of a document with its own syntax and rules, which describes the corresponding graphic elements of the image using attributes and formulas. This provides dynamic interaction with such a document using program code, which will allow further adaptation of the input data of the graphic module to extend the capabilities of the system for its application to navigation in different premises. In addition, the use of digitized maps prepared by means of vector graphics gives an advantage over raster analogues in terms of loading speed and image quality.

The interface solutions used in the development of the graphic module ensure the adaptability and flexibility of the whole navigation system, thanks to which it can be used both on computers (laptops) and on mobile devices with the user interface displayed correctly on the device screen. Adaptive design was obtained by taking into account the geometric dimensions of the device, as well as the density of virtual pixels with respect to physical ones, according to [22]. Possible browser and device display scale settings were also regarded.

As a result, the system interface is adapted correctly for different devices (figure 4): on relatively large screen sizes, the application has a fairly convenient interface with intuitive division of the screen zones, preserving the same properties on the screens of a mobile device. On mobile devices with small screen sizes, low pixel density or in portrait mode, the indentation from the edges of the screen and between the control elements, as well as the orientation of the map scale controls are changed to provide enough space for the upper zone elements, map center buttons, and floor switcher buttons to be displayed.

One of the important interface facilities of the system for navigation in a modern university is its implemented language localization, which allows you to select the language (Ukrainian or English) for all of the interface elements and user reference.

Thus, the interface solutions implemented by the graphic module of the navigation system (1) enable interaction with all subsystems (modules); (2) allow to visualize an interactive map of the premises with the user's route built by the proper module of the system; (3) provide the user with a flexible and adaptive interface.

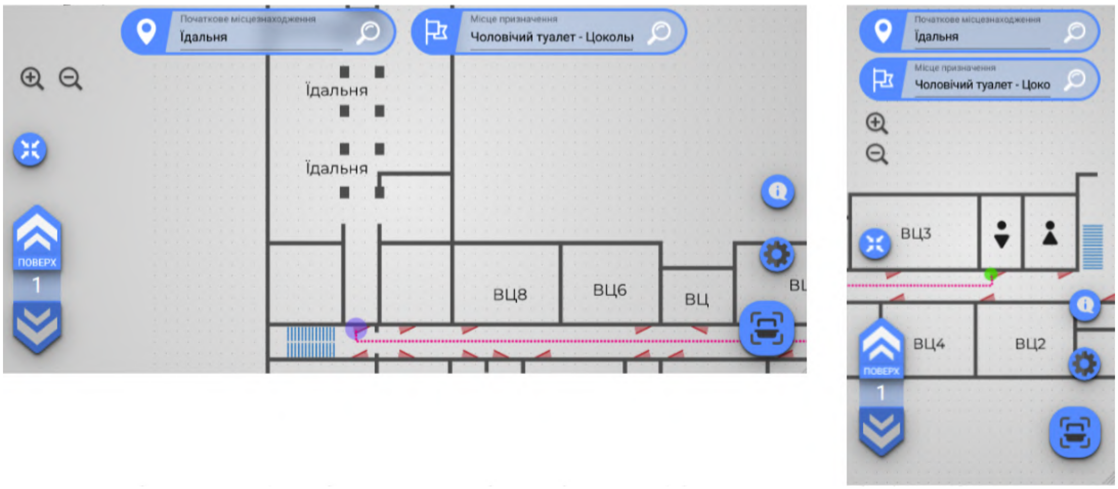

Figure 4: The system interface presented on a laptop with a screen resolution of 1366x768 and on an iPhone 5 SE in portrait mode.

The algorithmic solutions for the system design were shaping in the lines of data storing and algorithms realization for user's optimal rout finding. At the stage of design of the database structure, conceptual infological design, logical and physical database design was carried out. They are characterized below, as data structure has a paramount significance and allows to determine basic classes and objects for the system program realization, as well as to opt for proper search algorithms.

It is important to point out that the whole logic of the university navigation system does not require regular and dynamic creation, updating or deletion of data. However, it needs for the storage certain static data that describe the necessary information for the operation of the navigation system (to determine the path, indicate spots, etc.). The use of full-fledged DBMS and a server that requires a regular Internet connection is impractical due to the fact that the system must fully work offline in PWA mode.

In this regard, it was decided to use a local file with serialized data in JSON format. The text data exchange format JSON is built on two universal data structures [19]: a collection of name-value pairs and ordered list of values which in most languages is implemented as an array, list or sequence, vector, and so on. This format was also chosen because it is a fairly compact and valid description of the essence of the JavaScript language, which makes it easy to read such a file that is convenient for human perception. This solution allows to store all the necessary data for the operation of search algorithms locally on the server hosting the system. When the browser accesses the application address, this data will be automatically downloaded in parallel with the web application itself, and when using a PWA, such data will be fully stored on the user's device. In addition, the flexible and hierarchical nature of the database document enables to be developed in accordance to the extension of the navigation system.

Input and output information for the navigation system is represented by documents provided for processing and received as a result, respectively. Input information is data on user location (start spot) and/or final (destination) spot; output information is visualized location of the start spot and/or built route to the final (destination) spot.

When designing the logical data model, it was taken into account that the data are not updated by users, are neither to be regular requested nor to be parallelly requested, and they are stored on the application server itself, so the weight of the data directly affects the speed of initial application loading. Therefore, the data should be stored in a compact format that will be easily processed by the navigation system. It was also taken into account that when dividing the main entity of points by categories, in the case of corridors and stairs (physically existing in the university premises), there will be connections between entities of the same category (each point of the corridor has connections with other points of the corridor, each point of the stairs has connections with other stairs points).

For a more specific consideration of the data structure in the form of entities and relationships between them, in accordance with the method of their preservation, a physical data model was designed, shown in figure 5. At the next stage, the general structure of the system class hierarchy and their semantics was modeled using the UML class diagram, minding the following basic principles of the university navigation system architecture: (1) all business logic of the system, including that related to user interface components, is located in services; (2) utilitarian classes and methods are created to avoid duplication of code that does not generally relate to the business logic of services; (3) all data of a non-primitive type must have interfaces describing them.

In the context of the elaboration of algorithmic solutions in the design of the navigation system, the implementation of the shortest path search module as a component of the integrated architecture of the entire system, whose input data is the location (start spot) and the destination (final spot), is of significant importance.

For the software development of the shortest path search module by the Floyd-Warshall algorithm, an appropriate data structure was created that reproduces the graph on which the search is performed and which can be easily expanded when new vertices and edges are added. On this purpose, the Graph class, its methods and necessary properties were built, which allows adding new vertices, removing and editing them. The common structure of the Graph class is presented in figure 6.

After creating a graph and executing the Floyd-Warshall algorithm (function floydWarshall.ts), a third-party program, code, or service can request the shortest route between the specified vertices of the graph by passing two arguments, "from" and "to", which are the names of the start and end points search accordingly.

The necessary vertices and their indices are obtained from the Graph class. Next, the presence of a route in the matrix, obtained thanks to the implementation of the Floyd-Warshall algorithm, is checked. If the route exists, it is checked whether the next vertex is the end vertex of the route (then the recursion should stop). If the matrix still has vertices, the recursion continues. During recursion, the proper array is filled with the optimal route. If the route in the matrix does not exist, an empty array is returned, that is, the route between the specified vertices was not found.

In figures below there are selected examples of the results of searching for the shortest route. The result of the search and display of the optimal path from the initial location to the searched location, which are on the same floor, is shown in figure 7. The dashed line shows the found optimal path from one vertex (university dining room) to another vertex (computer laboratory).

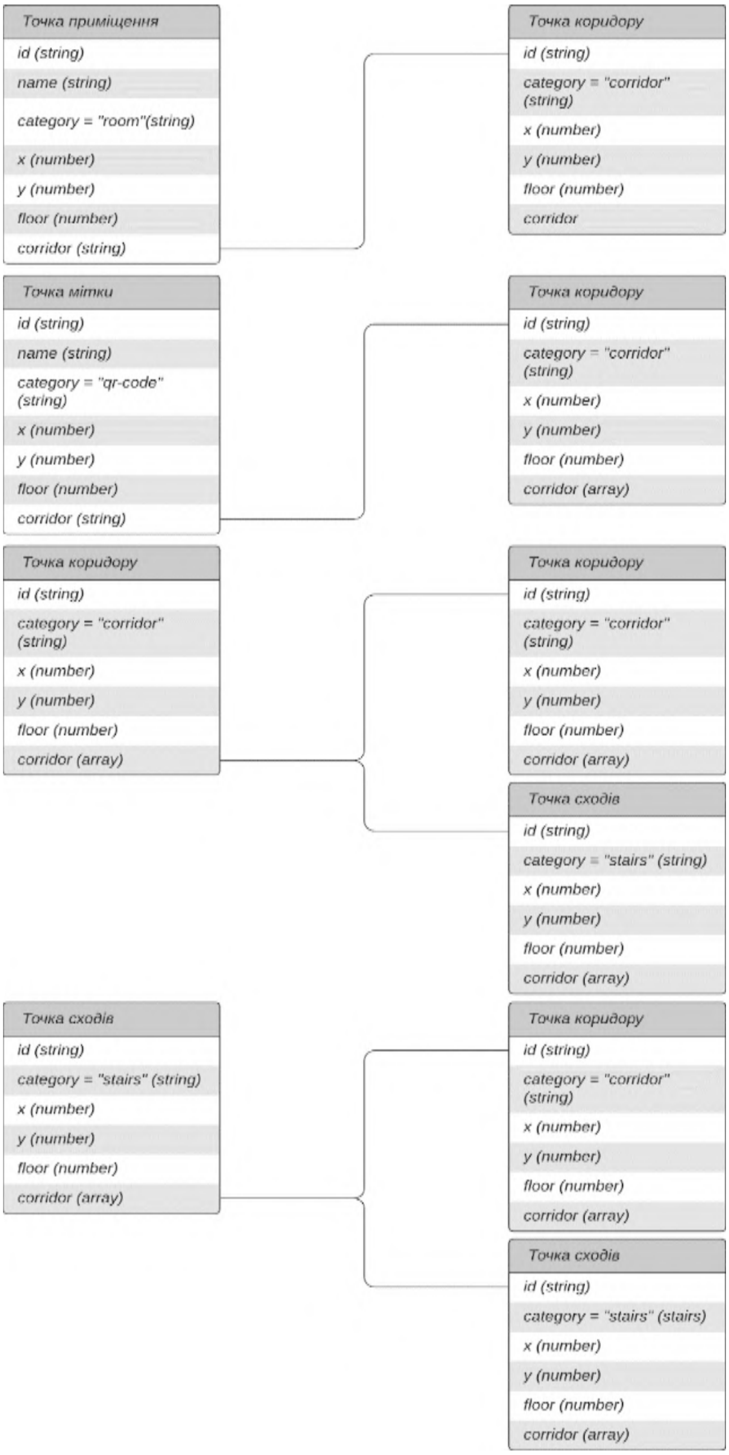

Figure 5: Entity-relationship diagram for physical model entities.

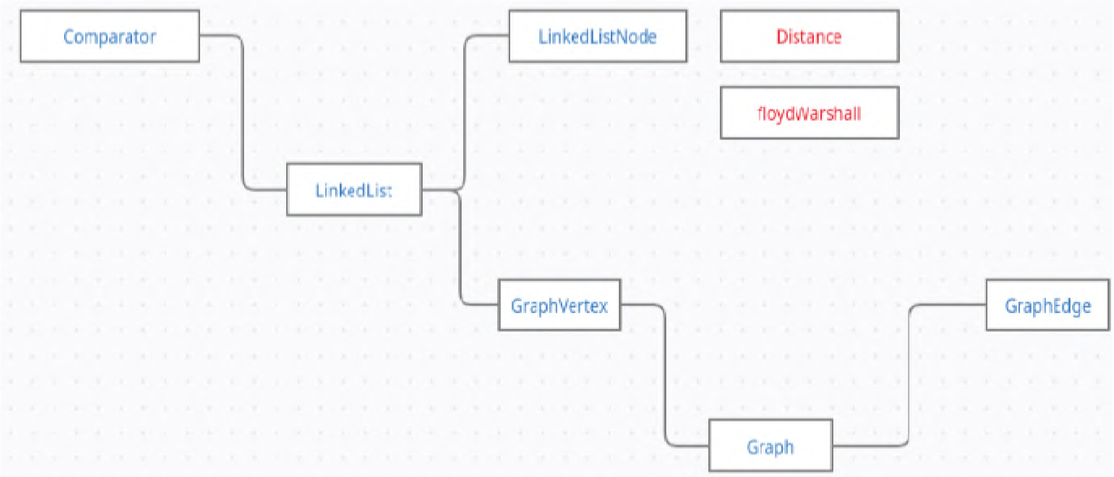

Figure 6: The common structure of the Graph class.

It is realized the possibility to search and visualize routes between two university locations on different floors. In figure 8a and figure 8b the dashed line visualizes the optimal route from the start location (the men's toilet, which is located on the basement floor) to the stairs to the second floor (figure 8a) and from the stairs to the final location (lecture room 222 located on the second floor of the main building of the university) (figure 8b).

Implementation of both the interface and algorithmic solutions highlighted above is facilitated by the module of detecting user's location, which works as an intermediary between the core modules of the navigation system in accordance with its architecture.

According to the use case diagram, with the help of the system, a user can determine his location in two ways: choose it from a list of available ones or get it contextually in the university premises by scanning the QR code from the label placed near certain locations.

In order to choose a location from the list, the user has to get access to the navigation system (as a web-application) in one of the ways and find the field "Start spot" on its main screen. After clicking on this field, the user will be able to select a location by performing a search by name in this field or by finding the required item with the desired location in the drop-down list. Determining the location by selecting from a list is used in cases when the user wants to plan his movements in advance and is outside the campus; or the user knows exactly his location and does not need additional help in positioning. This method is implemented through user interaction with the graphical interface of the application.

In order to determine the location using the QR code from the label, it is also necessary to access the application in one of the ways and press the corresponding button on the main screen. After clicking on this button, the QR code scanner window will open. At this stage, the navigation system may send a request for permission to access the camera of the user's electronic device, if it has not been given before. After obtaining permission, the user will see the image from the camera of his device in the scanner window and will have to point it at the QR code from the label. The scanner will decode the code and process the data. Positioning takes place relative to the label with the QR code, so, the user's location is considered to be the

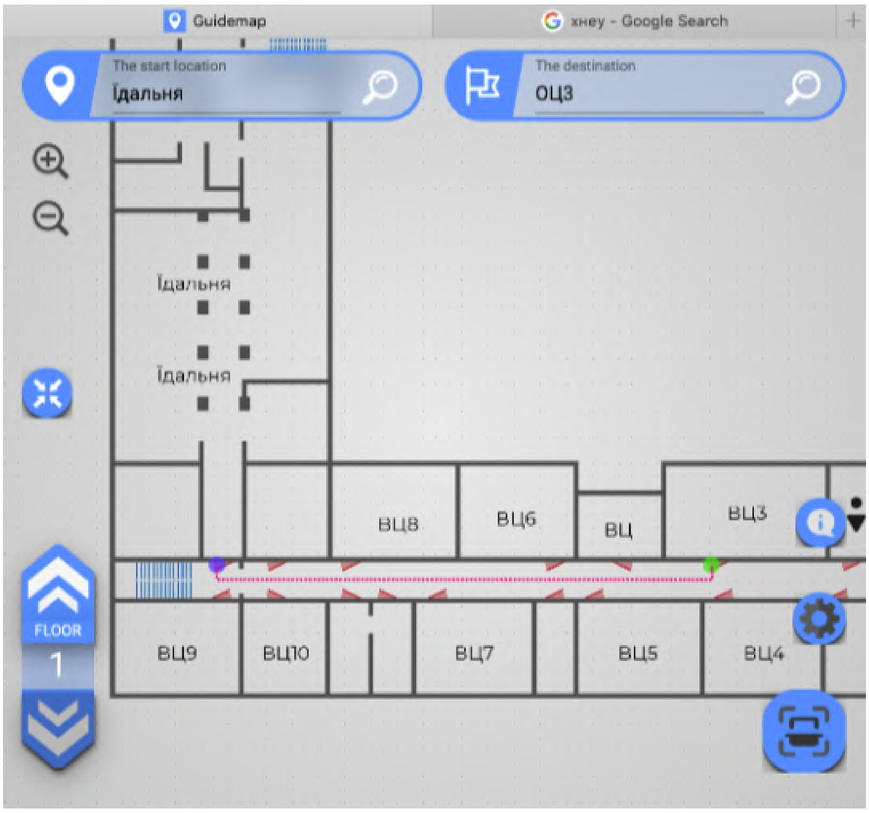

Figure 7: Visualized optimal path from the initial location to the searched spot located on the same floor.

physical location of the label itself.

Positioning by scanning the QR code is used in cases when the user is in university campus, but does not know his exact location and needs help in determining it. The expediency of using this method is explained by the fact that among the existing positioning methods (global positioning system, positioning relative to radio or Bluetooth tags), location determination by QR codes of labels has the highest accuracy in combination with the simplest implementation and the lowest cost of implementation [19].

Determining the location is implemented through the processing of data received from the QR code. Each label contains a link to the application with a specific code of the saved point. Thus, a third-party user who does not have the link can gain access to the application by scanning the code with any scanner that is often supplied with electronic devices with operating systems [1, 14]. If the QR code is scanned in the application itself, the stored point code will be obtained from its content and the search will be performed among the stored point objects.

The result of positioning the user with the help of this system module in any of these ways is a point object found in the storage and stored, containing the coordinates by which the location is visualized on the floor plan of the room and which will be used later as the start spot of the route by other modules of the system for navigation in university campus.

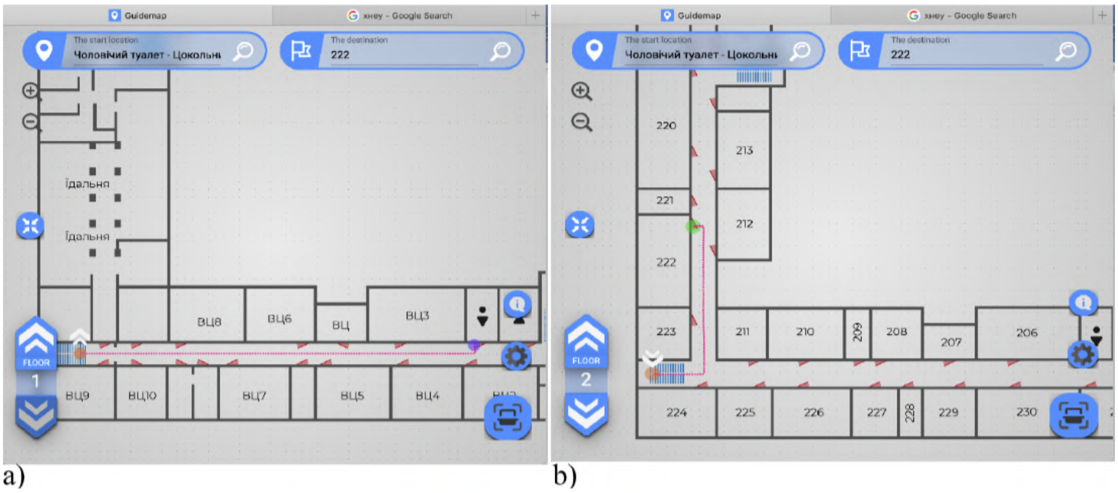

Figure 8: Visualization of the routes between two university locations on different floors.

To ensure the operation of the module for determining the user's location as a part of the navigation system, it is necessary to store certain information on the user's electronic device, namely a detailed plan of the buildings and a description of the university premises. It should contain the coordinates of points of corridors, stairs, labels and locations, as well as concise data about them and their location with respect to each other. Each stored point object will contain a unique code, coordinates on the building's floor plan, floor number, and the point code of the nearest corridor or staircase with which the object is associated.

All saved objects will be divided into several specified categories. Space point objects and QR code labels will have a unique name that can be used by the user to search when selecting a location from the list. All data that will be used in the positioning process must be available at any time for continuous and error-free operation of the module. In order to download the application, the detailed building plan and description of the premises must be stored in some form on the web server and ready to be transmitted over an Internet connection. Once downloaded, all data will be stored in the internal memory of the user's device for use in the positioning process. Thus, for the operation of the module for determining the location of the user, it is necessary to apply the set of all data elements. The module for determining the user's location as a component of the architecture of the navigation system interacts with the graphic module of the system (which provides input data for the module's operation) and the shortest path search module (for which the location module provides input data). So, the module under consideration has a rather complex and specific object-oriented structure.

Characterizing the functionality of the developed information system for navigation in university campus, designed based on the presented technological, interface and algorithmic solutions, it should be emphasized a number of its implemented capabilities. The said system which has both mobile and web versions provides a user with the following basic functions:

- Free viewing of the scheme of the university premises on the interactive map.
- Free change of the current floor of the room.

- Free change of map scale.
- Finding the shortest route from the start spot corresponding to the physically located QR code to any accessible university location.
- Visual prompts about the location of all available locations.
- Change the floor by clicking on the stairs icon, if the path goes through them.
- Scanning of QR codes directly by means of the system.
- Centering the map.
- Localization of the user interface in three languages (Ukrainian, Russian, English).
- Viewing the instructions for the system using and additional information.

The developed navigation system provides a number of additional functions. In particular, the system implements the possibility for users to share a link to an application with an already built route from the nearest point with a QR code to the point of the desired location. This can be useful when organizing mass events, because it provides quick and reliable orientation of a large number of university visitors at the same time.

It is also possible to scan QR codes with third-party software to move to the navigation system. In addition to the location of the QR code itself, a QR code can also include any final spot or an entire route, which significantly expands the areas of use of QR codes for navigation systems users. The developed system can be also installed on a mobile device and used offline if necessary.

Due to the use of vector graphics for the generation and processing of interactive maps of premises, as well as due to the elaborated technological approaches, the developed navigation system has sufficient functionality for extension and refinement, and it is also suitable for adaptation to use in other premises as well.

Thus, thanks to the proposed algorithmic, interface and technological solutions, during the design it was possible to overcome the main limitations inherent in similar systems implementing indoor navigation.

Developed information system for navigation in modern university in its test version has been implemented into the educational practice of the Simon Kuznets Kharkiv National University of Economics. In particular, QR code labels were placed in the relevant locations of the university premises, which enables university visitors to access the system with their smartphones and use the functionality described above (figure 9).

In the progress of the system implementation, feedback from the system users (university visitors of different categories) was collected and analyzed. Users were asked to fill in a survey form evaluating the quality of implementation of both functional and non-functional requirements for the system on a five-point scale. In general, there was received positive feedback regarding the interface quality, the speed of loading the interactive map and the convenience of work with it, the clarity of the visualization of the user's initial location and the destination spot, as well as the correctness of the route proposed by the system. A positive user experience was pointed out by the user-tested ability to share a link to an application with a ready-built route as well as language localization facility. Among the wishes for improving the operation of the system, it was suggested to improve the quality of the animated visualization of the built route between locations on different floors. Feedback from users received in the process

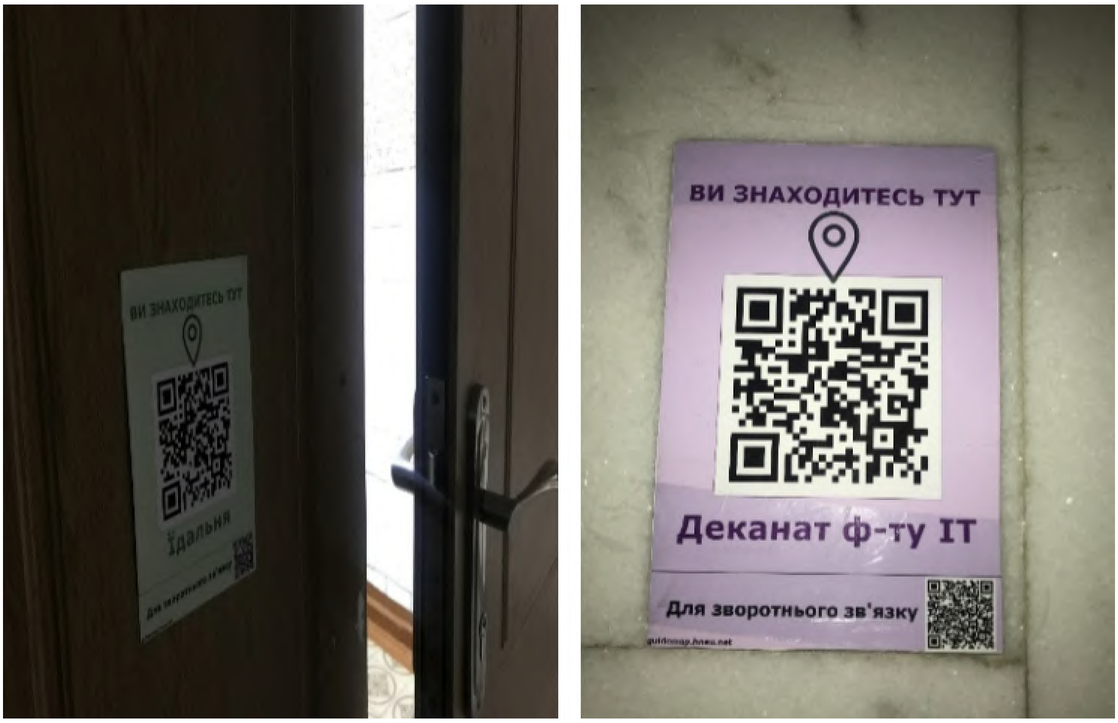

Figure 9: QR code labels placed in the relevant locations of the university premises which enable university visitors to access the navigation system, use it, and give a feedback.

of the system implementation testifies the feasibility of its developing for efficient navigation in the university campus. The users' wishes will be taken into account during the extension and improvement of the navigation university system, which is related to the prospects of our work.

4. Conclusions

According to the aims, the paper highlights algorithmic, interface and technological solutions for the development of the information system for navigation in university campus.

In the process of searching for algorithmic and interface solutions, the peculiarities of the scientific and applied problem of indoor navigation were analyzed, as well as the capabilities of selected analogues of navigation systems implementing similar functions of this subject domain were evaluated. It was concluded that the analyzed analogues, despite their significant functionality, have the following limitations. They are able to provide navigation inside a specific building, but they are not suitable for use in other premises. At the same time, they do not have a sufficiently developed functionality of extension and refinement. The said applications do not provide users with language localization.

The specification of functional and non-functional requirements for the university navigation system was carried out, its architecture was defined as a set of interconnected modules, for the implementation of which appropriate interface and algorithmic solutions were elaborated and covered. The main stages of design and development of the university navigation system in the context of the elaborated solutions are highlighted.

The functionality of the implemented system is characterized. It was established that during the design it was possible to overcome the main limitations inherent in similar systems implementing indoor navigation.

The results of the system implementation in the educational practice of the Simon Kuznets Kharkiv National University of Economics. Feedback from users received during the approbation testifies the feasibility of developing and using the information system for navigation in university campus.

The prospects for further work are outlined. It is planned to generalize the results of the users' survey, and elaborate the ways of the system improvement and their estimation.

References

[1] A. Idrees, Z. Iqbal, M. Ishfaq, An efficient indoor navigation technique to find optimal route for blinds using QR codes, in: 2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA), 2015, pp. 690–695. doi:10.1109/ICIEA.2015.7334197.

[2] V. Savchuk, V. Pasichnyk, Intelectual system "Mobile inforamation assistant of a tourist": functional and technological features., Visnyk Nats. Un-tu "Lvivska politechnika" 832: Information systems and networks (2015) 225–241.

[3] A. Basiri, E. S. Lohan, T. Moore, A. Winstanley, P. Peltola, C. Hill, P. Amirian, P. Figueiredo e Silva, Indoor location based services challenges, requirements and usability of current solutions, Computer Science Review 24 (2017) 1–12. doi:10.1016/j.cosrev.2017.03.002.

[4] A. Brunello, A. Montanari, N. Saccomanno, A Framework for Indoor Positioning Including Building Topology, IEEE Access 10 (2022) 114959–114974. doi:10.1109/ACCESS.2022.3218301.

[5] J. Kunhoth, A. Karkar, S. Al-Maadeed, A. Al-Ali, Indoor positioning and wayfinding systems: a survey, Human-centric Computing and Information Sciences 10 (2020) 18. doi:10.1186/s13673-020-00222-0.

[6] Z. Shao, M. A. Cheema, D. Taniar, Trip Planning Queries in Indoor Venues, The Computer Journal 61 (2017) 409–426. doi:10.1093/computer_journal/bxx107.

[7] MapsIndoors, AAU Maps, 2023. URL: https://clients.mapsindoors.com/aau/venues.

[8] dmslucy, Anyplace Indoor Service, 2020. URL: https://play.google.com/store/apps/details?id=com.dmsl.anyplace&hl=en&gl=US.

[9] Bayerische Staatsbibliothek, BSB Navigator, 2023. URL: https://www.bsb-muenchen.de/en/search-and-service/apps/bsb-navigator/.

[10] Situm Technologies, Indoor Navigation and Indoor Wayfinding, 2024. URL: https://situm.com/en/indoor-navigation/.

[11] M. M. Atia, S. Liu, H. Nematallah, T. B. Karamat, A. Noureldin, Integrated indoor navigation system for ground vehicles with automatic 3-d alignment and position initialization, IEEE Transactions on Vehicular Technology 64 (2015) 1279–1292. doi:10.1109/TVT.2015.2397004.

[12] E. Aleshko, M. Yakymenko, L. Polishchuk, Algorytmy ta structury danyh. Student book, Kropyvn., 2019.

[13] Ramadiani, D. Bukhori, Azainil, N. Dengen, Floyd-Warshall algorithm to determine the shortest path based on Android, IOP Conference Series: Earth and Environmental Science 144 (2018) 012019. doi:10.1088/1755-1315/144/1/012019.

[14] Liveli, Technology, 2023. URL: https://www.liveli.com.au/Our-Technology.

[15] B. Roy, Transitivité et connexité, C. R. Acad. Sci. Paris 249 (1959) 216–218. URL: https://gallica.bnf.fr/ark:/12148/bpt6k3201c/f222.image.

[16] S. C. Kleene, Representation of Events in Nerve Nets and Finite Automate, in: C. E. Shannon, J. McCarthy (Eds.), Automata Studies, volume 34 of *Annals of Mathematics Studies*, Princeton Univ. Press, 1956, pp. 1–44. URL: http://www.dlsi.ua.es/~mlf/nnafmc/papers/kleene56representation.pdf.

[17] P. Z. Ingerman, Algorithm 141: Path Matrix, Commun. ACM 5 (1962) 556. doi:10.1145/368996.369016.

[18] R. W. Floyd, Algorithm 97: Shortest Path, Commun. ACM 5 (1962) 345. doi:10.1145/367766.368168.

[19] Y. Fine, A. Moiseev, Angular and TypeScript. Sitebuilding for professionals, 2018.

[20] W. Fierens, SVG.js v3.1, 2021. URL: https://svgjs.dev/docs/3.1/.

[21] B. Troncone, Introduction - Learn RxJS, 2023. URL: https://www.learnrxjs.io/.

[22] Pikseli realnye i virtualnye (“Anatomiia svetodiodnogo ekrana”, chast 11), 2015. URL: https://www.sea.com.ua/ua/led-ekrany/news/pikseli-realnye-i-virtualnye-anatomia-svetodiodnogo-ekrana-cast-11/.