Study on Approaches of Building an Interactive Map in Web Application

Larysa Katerynych[1], Nikita Sazonov [2]

[1] *Taras Shevchenko National University of Kyiv, Faculty of Computer Science and Cybernetics, proskekt Akademika Hlushkova, 4D, Kyiv, 03022, Ukraine*
[2] *Taras Shevchenko National University of Kyiv, Faculty of Computer Science and Cybernetics, proskekt Akademika Hlushkova, 4D, Kyiv, 03022, Ukraine*

Abstract

It is quite a common request for a student or a teacher to visit university buildings, especially when looking for a specific classroom or department. Maps and navigation signs are meant to assist people with this certain task, although they are either inconvenient to use or absent altogether.

Information technologies can help with such navigation problem, specifically online maps, nevertheless even they lack floor schemas with classroom numbering and a detailed information about them. This is the reason for creating useful and convenient web-app for vicinity and building navigation is actual.

The purpose of this article is to formulate a list of rules for creating an efficient and convenient informational system for vicinity and building navigation and to create such a web-app. To accomplish this, a web-app user experience, navigation data representation options, their user perception and influence on their emotions were analyzed. Also, a list of recommendations for data representation and storing inside the app alongside with ways of fetching them was composed.

General navigational application development practices were described in this work, a test version of an application was created to harvest users' reviews and feedback and a "University interactive map" web-application for classroom, teacher rooms, departments, laboratories information displaying and navigating buildings and vicinities was created. Development kit used in this project consists of a free and widely distributed Integrated Development Environment (IDE) called Microsoft Visual Studio Code, TypeScript programming language, React JS library and CSS Tailwind library.

"University interactive map" web-application can be used during first-year students meeting, while travelling to previously unseen university locations, dormitory, and campus registration, when searching for a certain classroom or a building.

Keywords

web-app, map objects selection, interactive map, interactive interface, navigation.

1. Preliminaries

A correctly constructed navigation system, which is convenient to use and easy to find the necessary objects with, is the key to emotional confidence and a positive mood among travelers [1]. Computers and smartphones play an important role in the navigation process – most people plan their route on a computer before leaving home and check their progress on their phone while traveling.

In most cases, travelers also need to get the information about the object of their interest, or even about the surrounding objects. Moreover, quite often situations arise when people do not clearly know the location of the desired object and begin to physically look for it (for example, inside a building,

when searching for the classroom, a person starts to look at all the signs searching for the one they need).

Students and teachers needing to visit different buildings of their educational institution is a common situation, most often with the aim of finding a specific classroom. A navigation map and signs exist to simplify movement around the university territory and inside the buildings, but, according to user comments, they are either inconvenient to use, or are completely absent.

Information technologies, namely online maps, can help with navigation, but even they lack floor schemas with numbered classrooms and enough information for visitors about them. Moreover, there are a lot of extraneous information not related to the university that can distract the user, such as advertisements or reviews. It is worth noting that the necessary data can be found on the websites of departments, which have data on audiences, but there are no building maps. That is why the issue of creating an application that could combine high-quality floor plans of university buildings with information about classrooms is relevant.

To create a web application for displaying enough data, creating routes for the necessary classrooms or teachers' offices, departments, convenient and simple navigation through the territories and buildings of the university, as well as their floors, the following tasks were set and completed:

1. Describe best practices of creating information systems for navigating territories and buildings
2. Choose technologies for creating a web application
3. Periodically test and discuss with users the implemented functionality
4. Develop a map of the university territory, floor plans
5. Develop application page layouts
6. Implement application pages without functionality
7. Implement the functionality of the application pages

The chosen programming language was TypeScript, which extends the most popular web development language, JavaScript, by adding static typing, which makes it much easier for developers to track errors when ones occur [2].

The web application "University Interactive Map" can be used during admission campaigns to the university, settling in dormitories, searching for an office in an unfamiliar university building, etc.

2. Navigation data

2.1. Ways of presenting navigational data

To develop a more convenient and effective web application, it is necessary to outline the features of perception and presentation of navigation data.

A map is a schematic image of the earth's surface, on which some type of data (political, geological, topological, etc.) is displayed. Navigation maps most often depict the area near the place where this map is installed: streets, landmarks, famous buildings, tourist routes, parks, places of public recreation.

Most objects on the map should be marked with icons, hatches, lines, areas of distinct colors so that a person can easily perceive what is depicted. It is impossible to mark absolutely all objects - then the map will turn into text on top of the image, which will be exceedingly difficult to understand.

A scheme (schematic image) is a simplified depiction of an object that conveys its distinctions. Usually, the shape of the object is simplified, the ratio of its dimensions is not respected, some of its parts are not indicated, but it is necessary to depict unusual elements - they will be recognized and point to the main object.

2.2. Features of navigation data

Navigational data can carry not only informational but also emotional load.

A person feels joy due to success after achieving a goal, when he can easily understand the direction of movement to the desired object, and then successfully traverse this route. Moreover,

convenient navigation can save a lot of human time in searching for a route. Considering the above arguments, we can conclude that high-quality navigation is an especially important aspect of spatial organization.

The university space is quite specific: university buildings (faculties, institutes, etc.) play a significant role for students and teachers, and therefore they should be highlighted. Once inside, a person begins to look for a route to the necessary audience, that is, internal navigation plays no less a role than external. However, it is worth noting that the surrounding area is also important, because to reach your destination, a person will necessarily pass by objects that are said to be points of attraction.

Therefore, a high-quality navigation system should include a map of the territory indicating points of interest and university buildings, floor plans and pointers to classrooms.

3. Architecture of a web application for navigating areas and buildings

To create a convenient and effective web application, it is necessary to clearly describe and justify its architecture. In fact, it will be a set of rules, according to which information systems for navigation among territories or buildings can be qualitatively designed. This is possible because these rules are research-based and include best practices in the field.

A navigation application should display a wealth of data about surrounding objects, the building of interest, its floor plans, etc. However, it will not be effective to put all the data available on physical maps or signs on a computer monitor or smartphone screen - the user will not effectively perceive the information due to its large amount. It is worth choosing a different approach - to use the interactive capabilities of the web page and to provide the user with the opportunity to specify the stage of his journey or search for an object, and to provide only those data related to this stage [3].

A stage of the journey is a separate web page or window that displays related information: departments, floors, or classrooms. Transitions between stages should be done when the user interacts with data that indicates the next stage, and during this, information about the stage that the user needs appear on the screen. For example, if the user needs to find out data about a certain classroom of a faculty, he must first find and select this faculty (stage 1), then find the required audience (stage 2) and click on it to view the information.

It is worth highlighting interactive elements, clicking which changes the user's stage. The main purpose of such emphasis is for the user to quickly and easily understand that from this stage it is possible to go to the next one, or even to jump over a certain number of stages. If the navigation elements are not visible or they are poorly highlighted, the user will feel insecure, as if he is lost in the city without maps, street, and navigation signs [3].

There is also a "rule of three clicks", which states that an effective interface needs no more than three clicks to satisfy a user's request [4]. To follow this rule, it is necessary either to divide all data into no more than 3 stages, or to add the possibility of going through several stages at once, for example, a global search at the very first stage.

For the effective operation of the designed information system, it is necessary to store the following information:

1. The name of the current stage of the user, for example, in the form of a unique identifier (floor-1, faculty-computer-science, etc.).
2. The history or path the user has undertaken. An analogy to this data would be a list of vertices of a binary tree that were traversed on the way from the root to a particular leaf. Such a data structure will allow the user to navigate among the available stages easily and quickly in the application and even jump through them.
3. Description of the current stage - a schema of classrooms on a certain floor of the faculty, a list of floors of the building, etc. An algorithm for fetching this data should also be created when moving to the specified stage.
4. The data of all stages is the source from which state descriptions are taken. Saving data on all stages can be organized in the form of a database, if there is a large amount of information, or in the form of a certain markup language (json, yaml, etc.). Then the shared data object can be loaded synchronously with the web application loading.

So, the rules of an effective and easy-to-use application for navigating territories and buildings are:

1. Separation of all navigation data into groups and displaying them in the form of separate stages of the journey to the user.
2. Clear selection of interactive elements, necessarily those that redirect the user to another stage.
3. For the convenience of the user and for the efficiency of the application, store information about the name of the current stage, the history of parent stages, the description of the current stage and the data of all stages.

Therefore, we can define the technical specification of effective and convenient web-app for navigating areas and buildings, which includes webpages and specific functionality available at them:

1. University map page
   a. View the map of the university
   b. Viewpoints of interest (shops, pharmacies, and other objects useful for students and teachers)
   c. Search by address or building name
   d. Selecting a building to view information about it (description, link to the site, link to the online map, etc.)
   e. Selecting a building to go to its page
2. Building page
   a. View a list of floors with a brief information about each of them
   b. Selecting the floor of the building to go to its page
3. Floor plan of the structural unit page
   a. Viewing the floor plan of the structural subdivision with the possibility of selecting an audience and viewing information about it
   b. The ability to copy and send the link to the selected classroom

4. Application development process

4.1. Development kit

Selected technologies are one of the most important aspects of the development process, because they will affect the speed and structure of the project implementation.

The Microsoft Visual Studio Code development environment was chosen to create the work. It is a free cross-platform editor with open-source code that supports many programming languages, which is possible thanks to a huge number of extensions that can be quickly installed for free [5].

The following technologies were chosen for the first iteration of development:

- JavaScript is an interpreted, high-level language that is dynamically typed, oriented toward prototyped objects, and class functions. The most widely used for web development.
- React is an open-source JavaScript library for building component-based user interfaces. React deals only with state management and its reproduction in the DOM, so creating React applications usually requires the use of additional libraries for routing, as well as certain functionality on the client side. It is React that will allow us to store stage data in the state of the application.
- Bootstrap is a free and open-source CSS framework aimed at responsive web development, including for mobile devices. It contains HTML, CSS, and JavaScript-based design patterns for typography, interaction forms, buttons, navigation, and other interface components.

4.2. Development methodology

For an effective development process, it is worth choosing a development model - that is, a set of rules and practices that determine when each stage of development should be dealt with, and how much time and resources to spend on it.

Among the most famous models, the following can be distinguished:

1. One of the oldest is **cascade** or **waterfall** methodology, which involves successive stages, each must be implemented completely before the beginning of the next. In the Waterfall model, project management is easy. Thanks to its rigidity, the development is fast, the cost and terms are determined in advance. But not everything is so simple. The cascade model will give an excellent result only in projects with clearly defined requirements and methods of their implementation (namely technical specifications). There is no possibility to take a step back, testing starts only after the development is completed or almost completed [6].

2. An **iterative model** does not require knowing the complete requirements specification in advance. Instead, creation begins with the implementation of a piece of functionality, which becomes the basis for defining further requirements. This process is repeated. The version may not be perfect, the main thing is that it works. The final goal is approached so each step is effective, and each version is working [6].

3. In an **agile** development methodology, after each iteration, the customer can observe the result and understand whether it satisfies them or not. This is one of the advantages of this model. Its disadvantages include the fact that due to the lack of specific wording of the results, it is difficult to estimate the labor intensity and cost required for development. The methodology is suitable for large or long-life projects that are constantly adapting to market conditions. Accordingly, in the process of implementation, the requirements change. Companies develop startups precisely according to Agile [6].

For the implementation of the navigating web application, developers should bear in mind all the specialties of project and choose a development methodology accordingly. For the work in this article, we will choose an iterative development model, since at the beginning of creation the criteria will already be known, but the functionality of the project may change over time by considering the wishes of users and making changes after testing the project.
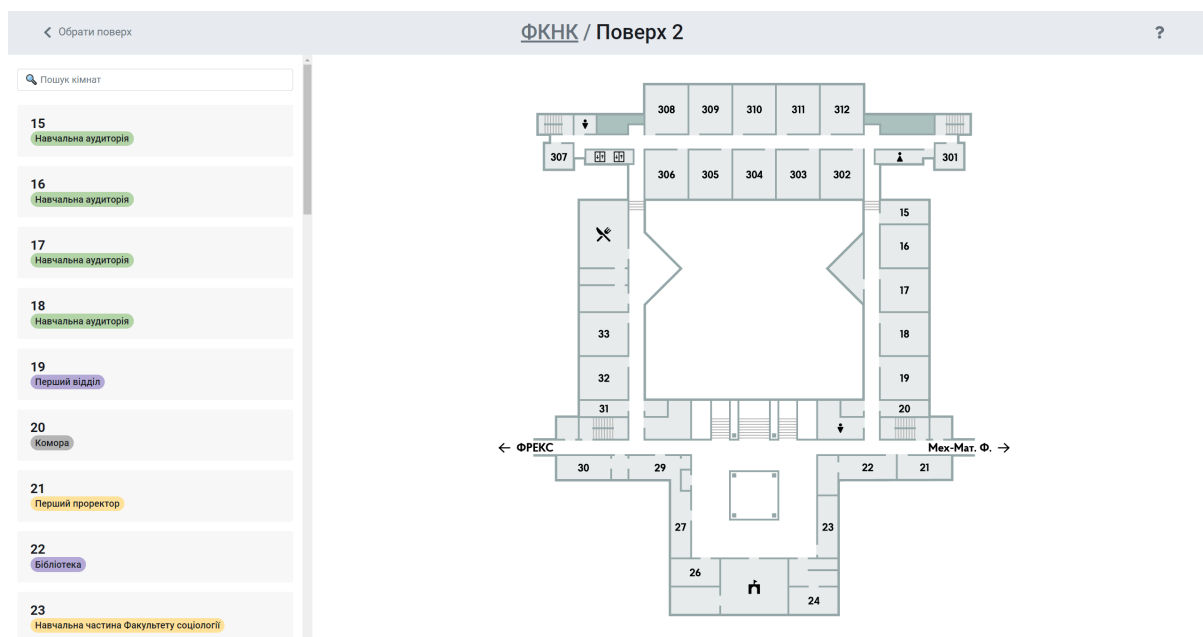
4.3. Test prototype

Before implementing the definitive version of the application, it is worth implementing a test version (prototype) to check the convenience of use and feasibility of certain design solutions. The most dubious among them is the decision to direct the user first to the page of the structural unit with the number of floors and their description (Fig. 1), and only then give users the opportunity to choose a floor.

Figure 1: Faculty page example in implemented test prototype

As a prototype, it was decided to implement the structure described above. Thus, the application "Interactive map of the faculty" was created with the following functionality:

- View the floor map of the Faculty of Computer Science and Cybernetics, which is a profile image of the faculty building
- Highlight brief information about each of the floors, which includes the main classrooms or offices located on it
- Click on a floor to go to its page
- View the floor plan (Fig. 2)
- Hover over an audience or cabinet to display information about it
- Click on an audience or cabinet to highlight
- Search for an audience or office in the list on the left (Fig. 2)
- If the teachers are assigned to a certain room, then clicking on the audience in the list to view the list of teachers with links to them

Figure 2: Floor schema page example in implemented test prototype

After the implementation of the prototype, the end users were asked to assess the convenience of using it. Most of the reviews were positive, but there were also negative ones, among which we can highlight:

- Inconvenience of using floor plans, because to switch between them, you need to return to the faculty page.
- Lack of global audience search, as the user may not know on which floor the desired audience is located.

Also, during development, it was noticed that it is inconvenient to use dynamically typed JavaScript, because typecasting caused errors that were difficult to find. It should be noted that the use of Bootstrap and React-Bootstrap introduced certain inconveniences due to the peculiarities of these libraries.

4.4. Correction of technical task

Considering the received feedback, it is necessary to amend the technical task so that the end user receives a product of a higher quality. We will add the following to the previously defined features:

- Viewing the list of floors with the possibility of filtering it by the audience number or information about the certain one
- Ability to change floors without leaving the floor schema page

Also, considering the experience gained in the development of the prototype, we will change the technologies used:
- Let's add TypeScript – a strongly typed language for web development, which is an extension of JavaScript, which helps to significantly reduce the number of errors due to the use of distinct types [2].
- And we will also replace Bootstrap with Tailwind CSS - this is a first-class CSS framework for quickly creating user interfaces. It is a low-level, highly configurable CSS framework that provides all the so-called building blocks needed to create designs to suit any taste without the need to override styles.

5. "University interactive map" web application

To implement the application, several libraries were used that were not described earlier, namely:
- React-Router – a library also for React, with which you can turn a single-page web application into a multi-page one. It adds the ability to manage the url address, including the creation of dynamic links. Thus, this library effectively replaces creating many html files to navigate to them [7] (Code Snippet 1).

```
function App() {
  return (
    <Router basename="/interactive-univ-map">
      <Switch>
        <Route exact path="/" component={ UniversityPage } />
        <Route path="/:facultyId/:floorId" component={ FloorPage } />
        <Route path="/:facultyId" component={ FacultyPage } />
        <Route exact path="/404-page-not-found" component={ PageNotFound } />
      </Switch>
    </Router>
  );
}
```

Code Snippet 1: basic App component, which describes the rules for redirecting the user to specific application page depending on the URL being manipulated by the application.

```
<FaShareAlt
    size={16}
     className="inline-block  text-slate-400  hover:cursor-pointer  hover:text-slate-600"
    onClick={handleShareClick}
    title="Copy link to clipboard"
/>
```

Code Snippet 2: Usage of a vector icon with size, styles, and a click handler.

- React Icons – a library for React that adds many vector icons that can be easily added to a page (Code Snippet 2).
- React-svg-map - a library that allows you to add svg objects, select them and add a description to them.

Let's review the functionality of the application:

On the map, you can select objects that will be highlighted there and in the side menu, where you can also select objects and they will be highlighted on the map. This is done to emphasize that the list

and the map are linked structures (Fig. 3). The list contains all structural buildings of the university, their abbreviated names, addresses, links to google maps and links to the website. Certain information is hidden from the user so as not to disturb him, and it can be viewed by clicking anywhere in the area about the structural unit.
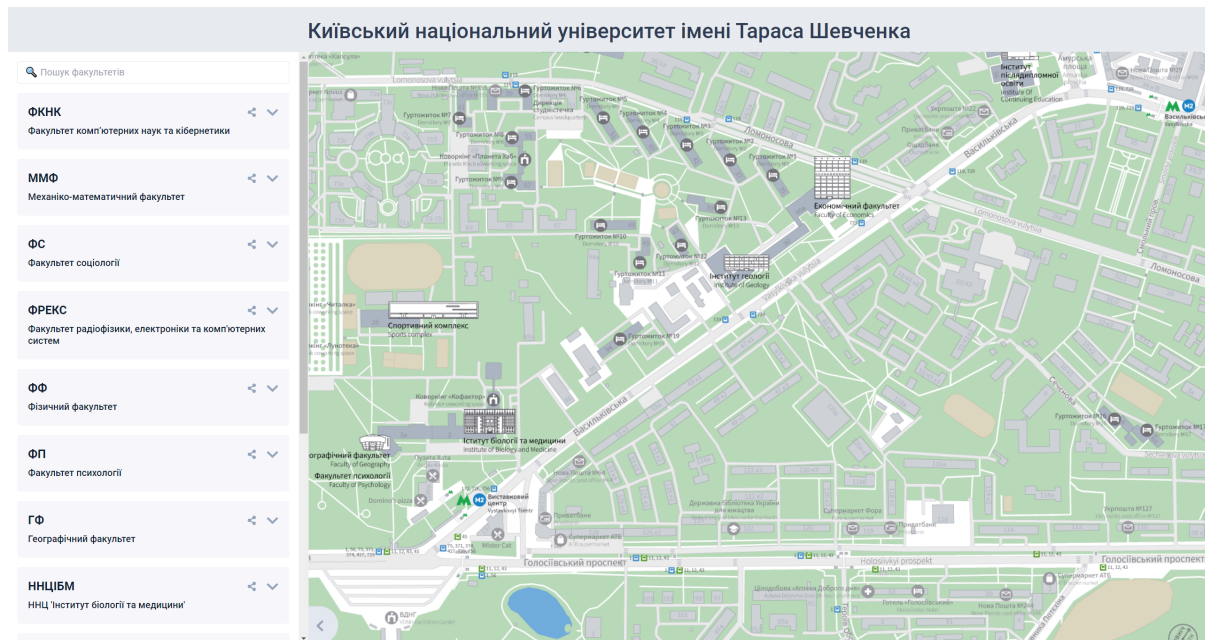
Figure 3: Web application home page

Each element of the list of faculty buildings is a separate React component (Code snipped 3), which is styled and has handlers for click and hover events. In this way, the main paradigm of the React library is followed: the separation of the logical elements of the web application into separate components.

Also, faculties in the list can be filtered using the "Faculty Search" form located above.

```
interface IFacultyItemProps {
  faculty: IFaculty,
  facultyLink: string,
  forwardToFaculty: MouseEventHandler,
  onFocus?: FocusEventHandler,
  onBlur?: FocusEventHandler,
  onMouseOver?: MouseEventHandler,
  onMouseOut?: MouseEventHandler,
  className?: string,
  tabIndex?: number,
}


const FacultyItem = ({faculty, facultyLink, forwardToFaculty, onFocus,
onBlur, onMouseOver, onMouseOut, className, tabIndex}: IFacultyItemProps)
```

Code snippet 3: Definition of FacultyItem React component and its props.

If you click on the abbreviated or full name of a structural subdivision, the user is redirected to the page of the first floor of this subdivision (next stage) (Fig. 4). The redirection happens straight to the floor plan and not to the subdivision page, as this avoids the unnecessary click for the user, which was mentioned earlier. It is worth noting that redirection to another page occurs by adding text to the URL (Code Snippet 4).
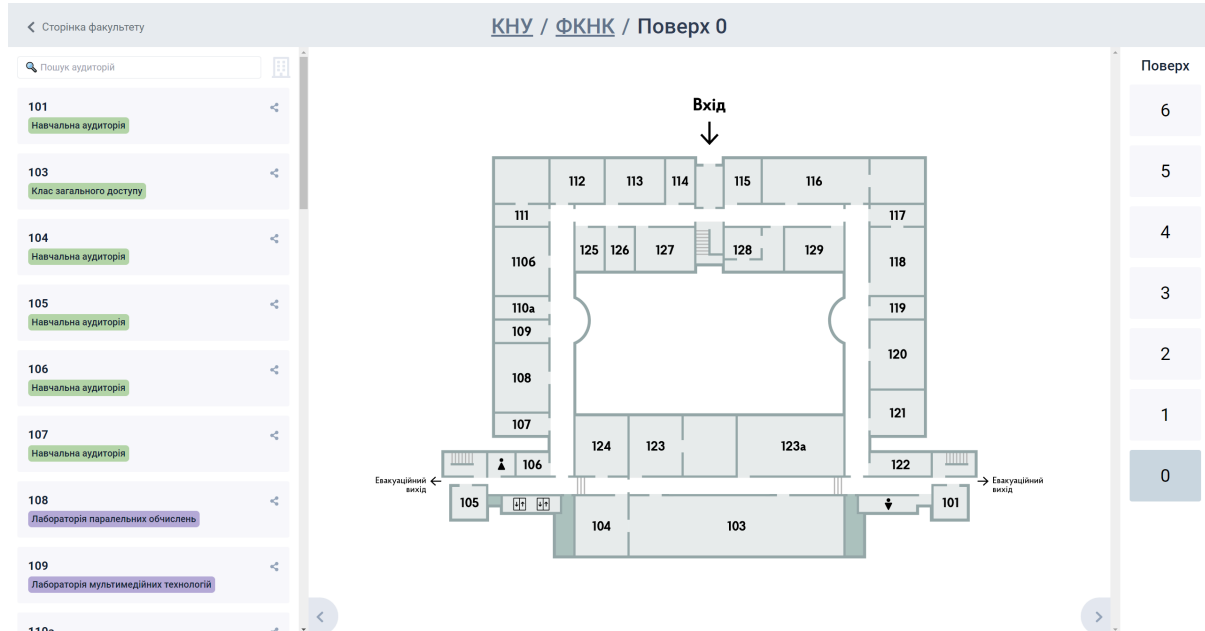
**Figure 4:** Floor plan page

```
forwardToFaculty={() => forwardToFloorPage(faculty.id + (faculty.floors[0]
? '/' + 'floor-' + faculty.floors[0].num : ''))}
```

Code snippet 4: Click event handler that adds some text to the URL of the given page.

At the top of the page there is an element called "breadcrumbs", which helps to understand which section of the application is opened. By clicking on the name of the structural division or the name of the university, the user can return to the corresponding pages. There is an inscription "Faculty page" with an arrow, which, when clicked, opens the faculty page. Thus, all the navigation between the stages in the application is placed in the header, which is quite noticeable and understandable for the user.

There is a panel of floors on the right for quick transition between them. This panel can be hidden, as indicated by a special graphic element with an arrow at the bottom.

On the left there is the list of classrooms of this floor, which can also be filtered, like the list of structural units. Also, if the teacher is assigned to the classroom, his name is listed under the corresponding classroom. For convenience, a link to the teacher's page on the website of a certain department was also added. The list of offices can be hidden in the same way as the list of floors.
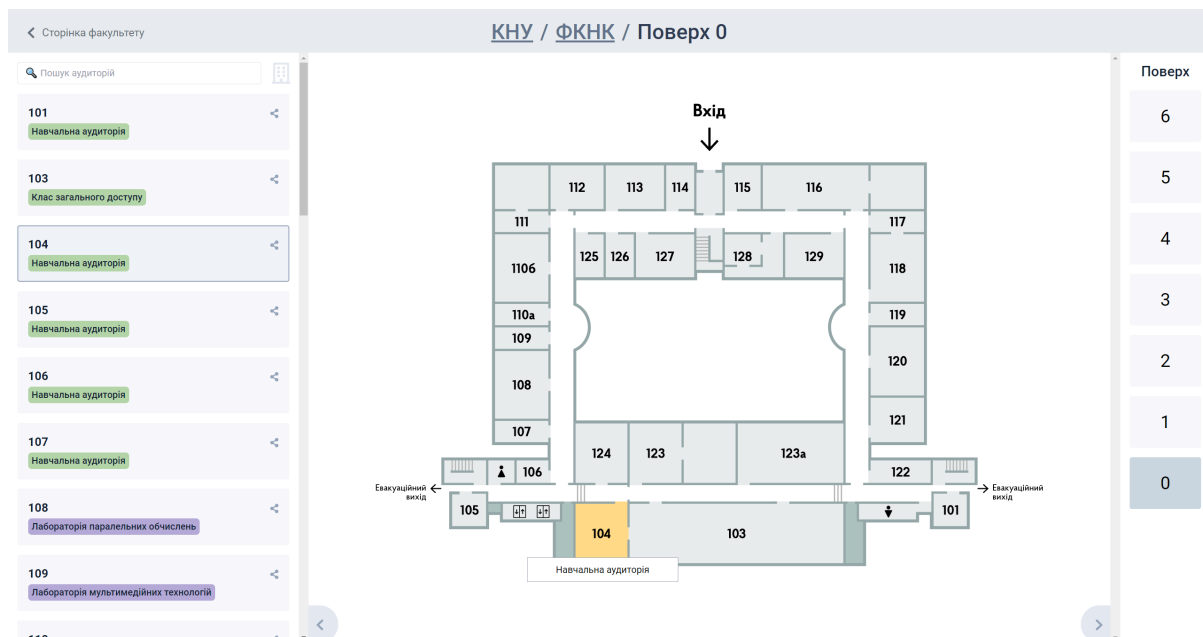
Figure 5: Selection of the selected cabinet on the schema and in the list

When clicking on an office on the schema or in the list, this office is highlighted in both places, and the name of the classroom appears on the schema (Fig. 5). When clicking on an office on the map or in the list, it is selected and highlighted without the need for the user to point to it.

It is also possible to filter / search the office not only of the selected floor, but also of all in the building. To do this, you need to select the building icon located to the right of the search form. In this case, clicking on the room will redirect the user to another floor, if the room is located there, and this audience will be selected.

It is worth noting that it is also possible to copy the link to any classroom or faculty. To do this, click on the "Share" icon located in the upper right part of the classroom or faculty element.

6. Conclusions

Based on the research results, the following conclusions can be drawn:

1. For the development of an information system for navigation among territories and buildings, it is important to divide all navigation data into certain stages, highlighting elements for redirecting the user between these stages, as well as storing certain data about the user's stage in the state of the application.
2. During the development of a technical project, it is important to determine the technical task in advance, as well as development methodology and development kit.
3. It is necessary to be ready for changes in both technologies and the technical task, so it is necessary to implement code that is easy to read, optimize and reuse.
4. Testing not only by the developer of his work, but also by a third party plays a huge role in the development process - that is, when end users use an intermediate version to identify bugs or suggest improvements.
5. The created web application "University Interactive Map" can be used both by people who are just getting to know the university and do not know or have not yet learned where certain objects are located, as well as by experienced students or teachers, if their route runs to structural unit which they are not familiar with.

7. References

[1] Lavrov V. A., Urban environment, its state, transformation, and conditions of perception, 2nd. ed., Minsk, 1974.

[2]   Vlad Riskutia, Programming with Types: Examples in TypeScript, Manning (2019): 11 – 17.

[3]   Tracy Osborn, Hello Web Design, No Starch Press (2021): 40 – 67.

[4]   Jeffrey Zeldman, Taking your talent to the Web: a guide for the transitioning designer, Indianapolis, Ind.: New Riders (2001): 97 – 98.

[5]   code.visualstudio.com, VS Code User guide. Why did we build Visual Studio Code? 2022. URL: https://code.visualstudio.com/docs/editor/whyvscode

[6]   Nancy L. Russo, Information Systems Development – Methods-in-Action, McGraw Hill Education (2001): 61 – 65.

[7]   reactrouter.com, React Router docs, 2021. URL: https://reactrouter.com/docs/en/v6