Designing a cross-platform user-friendly transport company application

Maksym Y. Salohub, Olena H. Rybalchenko and Svitlana V. Bilashenko

Kryvyi Rih National University, 11 Vitalii Matusevych Str., Kryvyi Rih, 50027, Ukraine

Abstract

This paper delves into the pivotal role of cross-platform development and the significance of user experience (UX) and user interface (UI) research in the creation of an application for a transportation company. User experience involves studying user behaviors, preferences and interactions, while user interface focuses on the design elements that facilitate these interactions. The research methodology includes a user experience research, including a user survey and analysis of analogous applications. The findings from this study are instrumental in researching the visual aspect of the application, aiding in the identification of design solutions and the creation of interface layouts. The paper analyzes technologies that facilitate cross-platform development. By drawing conclusions from the user experience research, interface design development and an examination of cross-platform development tools, the study culminates in the development of a convenient and scalable application tailored for both users and drivers of the transportation service.

Keywords

cross-platform application, software, React Native, mobile, transport, interface, UI/UX, nativity

1. Introduction

In the era of information technology and digitalization of all spheres of life, an important aspect of business is the availability of proprietary software. Achieving cross-platform applications can be done in two ways: using native approaches for each platform or employing technologies for a unified code base. A single code base reduces development costs and enables smaller companies to afford cross-platform applications, increasing demand.

For transportation companies, the ability to offer apps is usually limited to industry giants, leaving smaller regional providers with fewer options. Therefore, exploring the creation of affordable cross-platform applications is crucial for the modern market.

Given the active interaction between transportation services, users, and drivers, conducting user experience research and user interface analysis is essential. This ensures the development of a modern, high-quality design that is minimalistic and non-distracting for drivers. Addressing

CS&SE@SW 2023: 6th Workshop for Young Scientists in Computer Science & Software Engineering, February 2, 2024, Kryvyi Rih, Ukraine

taranmak21@gmail.com (M. Y. Salohub); rybalchenko@knu.edu.ua (O. H. Rybalchenko); bilashenko.s@knu.edu.ua (S. V. Bilashenko)
http://mpz.knu.edu.ua/vikladachi/olena-rybalchenko (O. H. Rybalchenko);
http://mpz.knu.edu.ua/vikladachi/Svetlana_Bilashenko (S. V. Bilashenko)
0009-0007-9696-3766 (M. Y. Salohub); 0000-0001-8691-5401 (O. H. Rybalchenko); 0000-0002-4331-7425 (S. V. Bilashenko)

UI/UX aspects before full development is crucial to optimize resources and enhance the overall application experience.

Another problem with cross-platform development can be that it is difficult to adhere to a single style, because some tools can cause a conflict while displaying interface elements on different platforms. Therefore, after creating a full-fledged design system, it is necessary to select elements that may cause a display conflict and make them custom so that the systems generate them instead of taking them from the ready-made tools. Thus, the developed application will be ready to be deployed to the AppStore and PlayMarket without any problems with platform compatibility [1].

2. Researching approaches to cross-platform development

The principle of a cross-platform application is generally to use a single code base that can be executed on any platform without significant modifications or without any modifications at all. However, cross-platform is a complex thing to implement and has several options for creating it [2].

Some applications use compilation to native code, which allows the application to run as native on a particular platform. Native elements are those that are unique to a particular development platform, even though they have the same name. So, a button on two platforms is still a button, but it may look different. But applications that create the so-called "bridge" can interact with both their own program code and the program code of the platform environment on which they were launched [3].

Some applications use web technologies to display the application. Thus, the use of frameworks in software systems whose work is based on creating mappings in the web environment allows the application to avoid cases of different interface appearance when running on different platforms. The advantage of such systems is that only one developer who has created a responsive website is enough. But for the average user, such applications have a huge drawback: they must always load a web page and therefore are limited to working only on devices that have stable Internet access and a built-in browser.

It is also possible to distinguish the way the program works when it does not have access to the native elements of the platform. Such applications use their own graphics tools, thus gaining access to the device's GPU. They use it to display the same interface elements regardless of the platform. But, like every other type of cross-platform application, they have their drawbacks, which in certain situations can be quite significant for the user. For example, they put more strain on weaker devices.

The larger the city and the more places where tourists visit – the more the application must support a large number of requests to the network. Therefore, it is extremely important to design high-quality work with network requests for both iOS and Android systems [4].

3. UX research

The UX research was divided into two parts: competitor analysis and target audience analysis using a questionnaire [5]. To analyze analogs, was selected the applications that also use cross-platform technologies. These are applications from Bolt, Grab and Didi Rider. The analysis results were obtained according to visual analysis and analysis of users reviews (table 1).

Table 1
Results of analogs analyze.

Parameter	Bolt	Grab	DiDi Rider
Popularity	Widely popular	Some popular	Not so popular as analogs
Multiple services in one application	Scooters and passenger transportation	Passenger transportation and food delivery	Only passenger transportation
Cash and non-cash payments support	Widely support	Widely support	No support
Geographical availability	Limited	Limited	Limited
User Experience quality	High quality of UX	Some UX issues	A lot of UX issues
Service issues	Rare issues	Rare issues	Often issues
Ride request edit availability	Can't edit	Can't edit	Can't edit

The next step of UX research is to survey users. The most convenient way to do this is to use Google Forms with a set of questions [6]. At this stage, 30 people were interviewed and the results of the survey were compiled. For a simpler understanding of the results, separate charts and tables were created for the most important points of the survey. Thus, the most voluminous, but important chart was the chart of the type of people who use the app.

Next, the most important part of the survey is which app features are important to users. Thus, for further work, it becomes possible to easily identify the main functionality that is really relevant for the modern market of transport companies. The results from the chart were transformed into a table for easier data processing (table 2).

Table 2
Important functions for users.

Question	Count	Percentage
Driver choosing availiability	16	53.3%
Showing driver photo while choosing driver	19	63.3%
Automatic drivers filtration by geolocation	23	76.7%
Chat with driver only when active trip	10	33.3%
Ending of trip when user and driver agreement both	13	43.3%
Panic button	23	76.7%
Trips archive	17	56.7%
Trip review	20	66.7%
The ability for drivers to take orders at any free time	1	3.3%
The function of confirming the arrival of a taxi only with my permission	1	3.3%

According to this chart, the two most important features for users are the presence of a panic button during the trip to ensure their safety and automatic filtering of drivers by trip details (for example, by selecting the “with child” tag for a trip, drivers without a child seat will not even see this order). As slightly less important features, respondents noted the ability to rate a ride, see a driver’s photo when choosing a driver, the ability to choose a driver and save a ride history. Also, almost 43.3% of users consider it necessary to be able to end the ride only upon their confirmation, so the driver cannot disable the transportation on his/her own. The least important features are the activity of chatting with the driver only during the trip itself, the ability for drivers to take orders at any free time and one answer was added by the user personally, which sounds like “the function of confirming the arrival of a taxi only with my permission”. Thus, the most basic needs of application users in terms of functionality that need to be implemented first was identified.

Equally important is not only the implementation of the functionality, but also what exactly will be displayed on the screen with a higher priority. To determine this, the question “what is paid attention to when choosing a transport” was separately highlighted and a diagram was created based on the answers. The results of diagram were transformed into a table (table 3).

Table 3
Focus of the attention of users when choosing a transport.

Question	Count	Percentage
Cost	24	85.7%
Car brand	5	17.9%
Driving experience	7	25%
Company reputation	15	53.6%
Quality of service	20	71.4%
Payment convenience	19	67.9%
Driver rating	1	3.6%

Let’s summarize the results of the UX research. The app’s users are active citizens who value their time and comfort. They use smartphones with internet access, which makes them ideal users. They also have different socio-economic statuses, including students, office workers, entrepreneurs, tourists and drivers looking for additional income.

4. UI analysis, creating interface prototypes and design system

The development of the UI part of the application can also be divided into several parts – creating low-detail mockups (wireframing), graphic design and designing full-fledged design mockups. Wireframing helps to visualize the concept of the application and provides an opportunity to quickly check and improve the structure of the product before moving on to more detailed design and development. After low-detail prototypes are created, they are refined into highly detailed mockups, where colors, textures, graphics and other elements are added to increase the level of detail and prepare the mockup for use by the developer when building the app [7, 8].

First, was created a prototype of the login screen, because this is the first thing the user sees, but it is more important to consider the next screen after the login. This is the home page screen. It is divided into two versions, because the user can be either a passenger or a driver. Since they are still quite similar, let’s take a look at the passenger screen (figure 1).

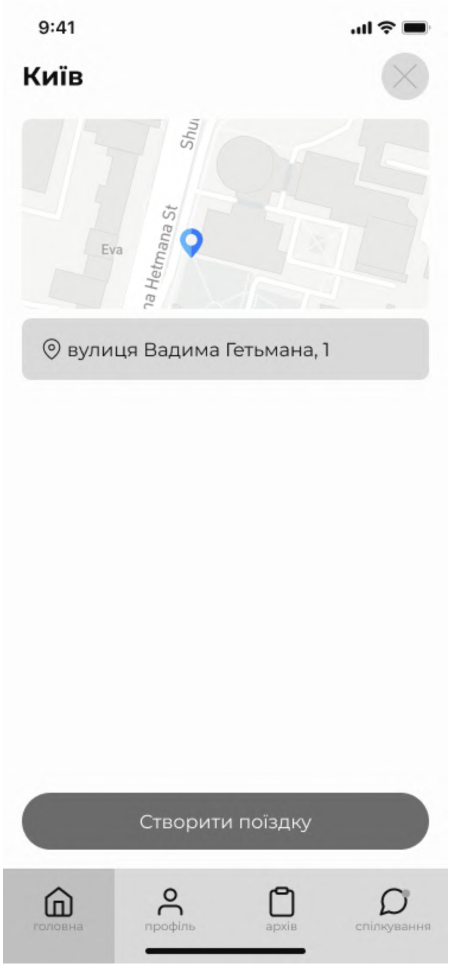


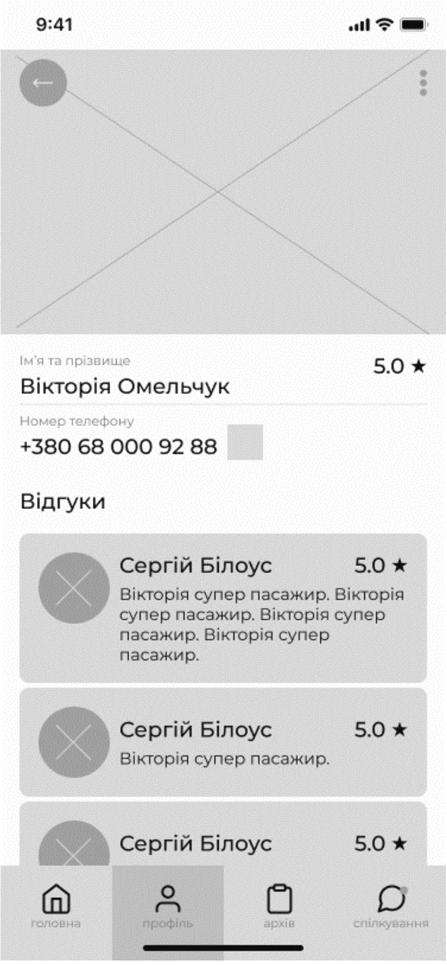
Figure 1: Prototype of the main screen of passenger.

It was also useful to create a prototype of the profile screen to understand the information architecture of the application (figure 2).

These were all examples of creating low-detail interface layouts. Their task is to structure the data on navigation in the application and the general arrangement of elements.

The next stage of UI modeling is graphic design. When the information architecture of the application is already known, it becomes important to select the right icons that will fully express the semantic meaning that was embedded in them. That’s why status icons were identified as part of the design system (figure 3).

As you can see, some of these icons are also colored – this is necessary to further emphasize to the user what exactly has happened in the application at the moment. Thus, additional



9:41

Ім'я та прізвище

Вікторія Омельчук 5.0 ★

Номер телефону

+380 68 000 92 88

Відгуки

Сергій Білоус 5.0 ★

Вікторія супер пасажир. Вікторія супер пасажир. Вікторія супер пасажир. Вікторія супер пасажир.

Сергій Білоус 5.0 ★

Вікторія супер пасажир.

Сергій Білоус 5.0 ★

Figure 2: Prototype of main profile screen.

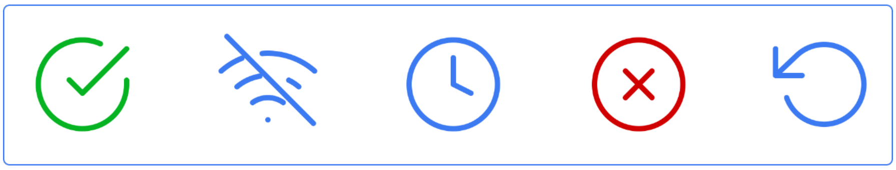
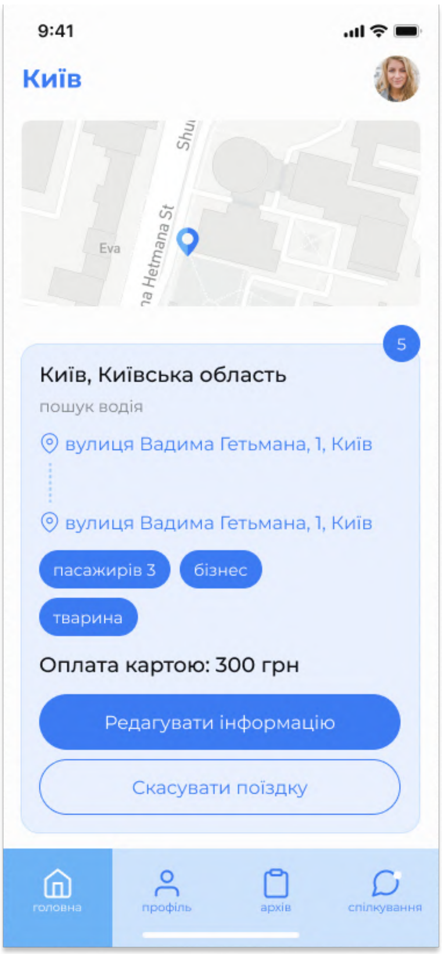


Figure 3: Icons for informing about processes in the application.

semantic coloring is provided for the images.

So, when the low-detail mockups and the graphic part of the design are ready, you can move on to developing high-detail mockups. Since, according to the results of the UX analysis, it is clear that the users of the application are a wide range of people, a blue was chosen as the main color. It is associated with calmness and reliability, which is essential for a transportation company (figure 4, 5).



9:41

Київ

Київ, Київська область

пошук водія

вулиця Вадима Гетьмана, 1, Київ

вулиця Вадима Гетьмана, 1, Київ

пасажирів 3 бізнес

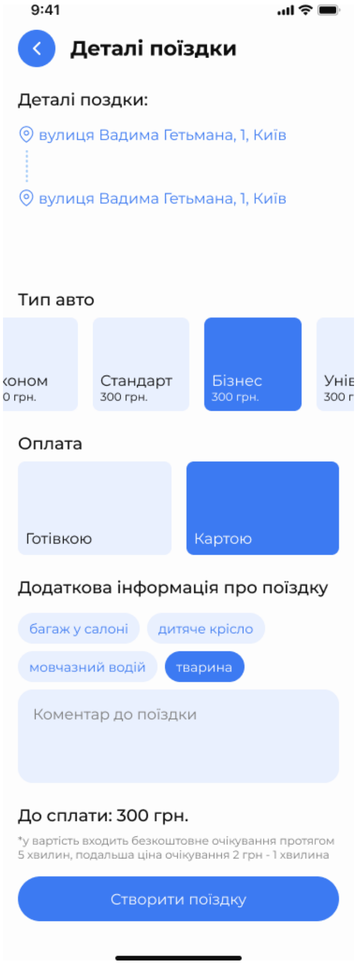
тварина

Оплата картою: 300 грн

Редагувати інформацію

Скасувати поїздку

Figure 4: Main screen of the user.



9:41

Деталі поїздки

Деталі поздки:

вулиця Вадима Гетьмана, 1, Київ

вулиця Вадима Гетьмана, 1, Київ

Тип авто

Оплата

Додаткова інформація про поїздку

Коментар до поїздки

До сплати: 300 грн.

*у вартість входить безкоштовне очікування протягом 5 хвилин, подальша ціна очікування 2 грн - 1 хвилина

Створити поїздку

Figure 5: Trip confirmation screen.

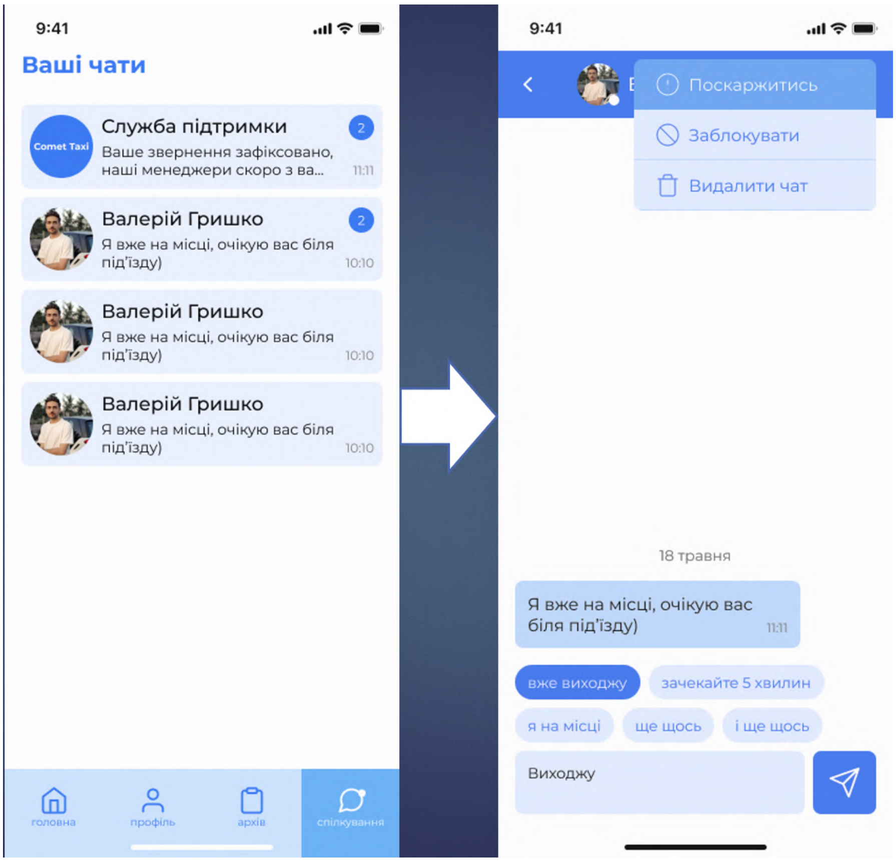
A grouping of screens based on a specific set of functionality is called a flow. For example, this is how a chat flow looks like (figure 6).

As a result of the work, we have a completely ready-made design system that can be used for software development, knowing that it will be useful and convenient for each user.

5. System development

To implement the application, JavaScript was chosen to avoid switching between languages when creating different parts of the project. Thus, the Express.js framework is used for the backend part and React Native is used for the application itself.

React Native is used to develop mobile applications. The main advantage of React Native is that this technology allows you to write code once and then use it to create applications for both iOS and Android. This significantly reduces the development time and eliminates the need to write two separate programs for different platforms [9].



9:41

Ваші чати

Служба підтримки 2
Ваше звернення зафіксовано, наші менеджери скоро з ва... 11:11

Валерій Гришко 2
Я вже на місці, очікую вас біля під'їзду) 10:10

Валерій Гришко
Я вже на місці, очікую вас біля під'їзду) 10:10

Валерій Гришко
Я вже на місці, очікую вас біля під'їзду) 10:10

9:41

18 травня

Я вже на місці, очікую вас біля під'їзду) 11:11

Виходжу

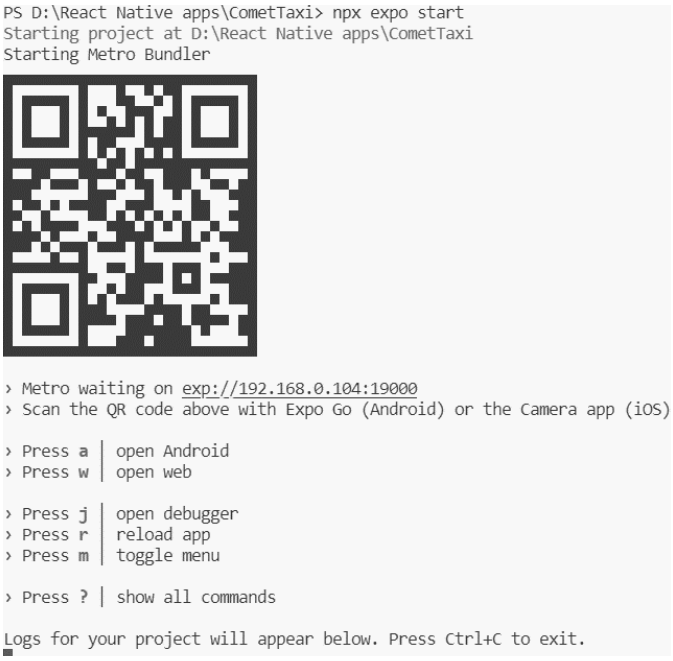
Figure 6: Chats flow.

Express.js is used to develop the server side of the system. It is a flexible and powerful framework that allows you to easily create web applications and APIs. Express.js integrates well with MongoDB and other technologies used in the project [10].

Another feature of the project is the use of Expo CLI. One of the most important features of Expo is the ability to develop mobile applications on the React Native platform. With Expo, you can easily build, test and publish your React Native apps without having to set up complex development tool configurations. Testing is done by scanning a QR code from the IDE terminal (figure 7) [11].

This project uses MongoDB to store data. It is one of the most popular NoSQL database management systems. MongoDB is a document-oriented database that uses a JSON-like structure to store data rather than working with tables in the usual sense of databases. Features of MongoDB include data structure flexibility, high performance, horizontal scaling and the ability to work with large amounts of data. The database is created using MongoDB Atlas, a MongoDB cloud service [12].

A special collection of icons called Feather Icon was used in the application. With the help of this service, a collection of icons was created that are used to improve the semantic interaction



```
PS D:\React Native apps\CometTaxi> npx expo start
Starting project at D:\React Native apps\CometTaxi
Starting Metro Bundler

› Metro waiting on exp://192.168.0.104:19000
› Scan the QR code above with Expo Go (Android) or the Camera app (iOS)

› Press a │ open Android
› Press w │ open web

› Press j │ open debugger
› Press r │ reload app
› Press m │ toggle menu

› Press ? │ show all commands

Logs for your project will appear below. Press Ctrl+C to exit.
```

Figure 7: Terminal interface for testing.

of users with the application. React Native technology does not support easy interaction with SVG icons. But this is necessary, because only this image format scales to any screen size without changing the quality. Therefore, to display such icons in the application, additional libraries react-native-svg and react-native-feather are used. This way, a contact is established between the application and the cloud service, which is the repository of the necessary icons in SVG format.

There is a separate algorithm for saving images on the server. Due to the use of MongoDb, it was decided to encode the images transferred to the server in base64 using the multer auxiliary library.

The basic encryption code looks like this:

```
const multer = require('multer');
const storage = multer.diskStorage({
    destination: function (req, file, cb) {
        cb(null, './uploads/');
    },
    filename: function (req, file, cb) {
        cb(null, new Date().toISOString() + file.originalname);
    },
});
const upload = multer({ storage: storage });
module.exports = upload;
```

The code of the auxiliary file used for intermediate image encoding looks like this:

```
const multer = require('multer');
const storage = multer.memoryStorage();
const upload = multer({
    storage: storage,
    limits: {
        fileSize: 1024 * 1024 * 2,
    },
});
module.exports = upload.single('image');
```

It is important to note that this file size was not chosen by chance. It was important to save as much space as possible in the database without loading the server with long image processing. Therefore, the image will be successfully saved only if its size is 1024*1024 or less. At the same time, the maximum image size is limited to 2 MB.

6. Conclusions

This project was a fascinating challenge that allowed us to delve deeper into the world of developing and researching the benefits of cross-platform applications for transportation companies. One of the key points of focus was to create an application that not only meets the needs of users, but also has a convenient and efficient interface for both parties – both passengers and drivers.

A number of technologies were used to achieve this goal. React Native allowed us to efficiently implement the visual particle, ensuring its cross-platform compatibility. Express.js was used to create a powerful backend and MongoDb served as a reliable database.

Analysing of the subject area, was identified the main problems of interaction between passengers and drivers. This insight allowed us to clearly define the project's goals and objectives, focusing our efforts on solving specific problems.

Target audience research confirmed that drivers and passengers using modern mobile technologies are the main users of the app. The results of the survey emphasized the high demand for such services, which became the basis for further improvement of the app.

Overall, this project not only demonstrated the effectiveness of using modern development methodologies and tools, but also emphasized the flexibility and adaptability of the application to changes in the passenger transportation market. The result is a high-quality mobile application that is ready to quickly adapt to any conditions and user requirements.

References

[1] N. V. Ichanska, S. I. Ulko, Osnovni aspekty stvorennia mobilnykh dodatkiv ta vybir instrumentiv yikh rozrobky [The main aspects of creating mobile applications and the choice of tools for their development], Systemi upravlinnia, navigatsii ta zv'iazku 59 (2020) 74–78. doi:10.26906/SUNZ.2020.1.074.

[2] M. Martinez, S. Lecomte, Towards the quality improvement of cross-platform mobile applications, 2017. URL: http://arxiv.org/abs/1701.06767.

[3] B. Eisenman, Learning React Native: Building Native Mobile Apps with JavaScript, 2nd ed., O'Reilly Media, Inc., 2017.

[4] A. Noulas, V. Salnikov, D. Hristova, C. Mascolo, R. Lambiotte, Developing and Deploying a Taxi Price Comparison Mobile App in the Wild: Insights and Challenges, 2017. URL: http://arxiv.org/abs/1701.04208.

[5] R. L. Veal, How to Conduct User Experience Research Like a Professional, 2021. URL: https://careerfoundry.com/en/blog/ux-design/how-to-conduct-user-experience-research-like-a-professional/.

[6] M. Y. Salohub, Opytuvannia dlia vivchennia tsilovoi auditorii dlia stvorennia dodatku sluzhby taksi [Survey to study the target audience for creating a taxi service app], 2022. URL: https://docs.google.com/forms/d/e/1FAIpQLSerQrKhXxOJ7s1rqLY5fAyWI_9HHUgLS4HVPGarZb6k9Gc0Ag/viewform.

[7] J. Yablonski, Laws of UX, 1st ed., O'Reilly Media, Inc., 2020.

[8] I. D. Foundation, What is Wireframing: The complete guide, 2023. URL: https://www.interaction-design.org/literature/topics/wireframing.

[9] Meta, React Native: Get Started, 2023. URL: https://reactnative.dev/.

[10] Expressjs.com, Express js backend documentation, 2023. URL: https://expressjs.com/.

[11] Expo.com, Expo developer documentation, 2023. URL: https://expo.dev/.

[12] Mongodb.com, MongoDB documentation, 2023. URL: https://www.mongodb.com/.