DOI: https://doi.org/10.26642/ten-2025-1(95)-280-288
UDC 004

V.V. Kochura, Master Student
N.O. Kushnir, Senior Lecturer
T.M. Loktikova, Senior Lecturer
A.V. Morozov, Ph.D. in Engineering, Assoc. Prof.
Zhytomyr Polytechnic State University

Hotel and accommodation booking management service for traveling in Ukraine

Even in such a difficult time as today, traveling around the country remains relevant and necessary for many people. They are necessary for work, inspire and cheer people up, make life full, bright, and add meaning. This prompted the creation of a booking management service for hotel and accommodation for traveling in Ukraine. The system allows to choose the desired place of accommodation, according to certain criteria, pay for it, view your booking history in personal account, and leave a review. It also provides tools for cooperation with the owners of such places of accommodation, who can place them on the website, manage them and process bookings. A monolithic architecture and MVC and Client-Server patterns were used to build the service. The server side is implemented in the PHP programming language using the Symfony framework, and the client side is implemented using the JavaScript library React. Nginx was chosen as a web server. The Docker tool was used to deploy the application. The system uses MySQL as the main database, consisting of 18 tables, and is supplemented by MongoDB, which has 2 collections. During the development process, special attention was paid to testing and error handling, which made it possible to increase the reliability, security and intuitiveness of the system, ensuring stable operation and minimizing the possibility of errors. The developed service meets modern requirements and has the following features: modern and adaptive design, user-friendly interface with the ability to create a personal account, and support for signing in via Google.

***Keywords**: web service; booking; accommodation; Symfony; React.*

Relevance of the topic. Modern life is impossible to imagine without traveling, where comfort during a business trip or vacation is ensured by pre-booked accommodation. Unfortunately, due to the current situation in the country, traveling has become less relevant. However, every citizen has the right to temporarily change their place of accommodation, has the right to rest and escape from everyday worries. So how do you book accommodation? The solution to this problem is a system that will allow customers to book accommodation that meets their needs, and owners to post their places of accommodation. It will provide a quick search for accommodation, compare prices and view reviews, greatly simplifying and speeding up the selection process.

The service will not only save time, but also provide a wide range of accommodation, allowing the user to find the option that best suits his requirements.

Analysis of the latest research and publications on which the author relies. The topic under consideration is quite common, so similar developments can be found on the market. In order to form a clear vision of the final product, it is important to analyze existing analogs, identifying their advantages and disadvantages. Consider several similar products for comparison. One of them is Hotels24.ua [1]. This is a free accommodation booking service in Ukraine that contains a large selection of hotels and is popular, although it looks quite simple and outdated. Among the advantages are a convenient search with filtering and sorting, the ability to view the location of the accommodation on the map, and the availability of a review system. The disadvantages include the lack of the ability to create a user account (this feature is available only to hotel owners), outdated design, and the lack of a Google login. GoHotels is another free online accommodation booking service in Ukraine [2]. Although the website also has a somewhat outdated design, it is more interactive and functional. Its main advantages are the availability of a personal account (which the previous analog does not have), a page with announcements and events with interesting articles about places of interest in Ukraine, and an interactive map. Among the disadvantages, it is important to note the outdated design, as well as the lack of adaptability for mobile devices and the ability to log in via Google. Another example is HOTELS-OF-UKRAINE, a website for choosing accommodation that differs from its previous analogs primarily in its pleasant and responsive design [3]. It offers a large number of places to book, as well as an interactive map that allows to view nearby attractions and restaurants. The main disadvantage of this service is the lack of authorization on the site, which is why hotel owners cannot create accounts for cooperation.

The purpose of the article is to research the development of a service for managing hotel and accommodation reservations for traveling in Ukraine. The proposed system is distinguished by a modern and adaptive design, a user-friendly interface with the ability to create a personal account, and support for logging in via Google.

Presentation of the main material. The primary stage in the design of any software product is the choice of architecture, which determines its performance, scalability, and ease of development. The correctness of the architectural solution determines how easy it will be to make changes to the system, add new features and support it. The optimal solution turned out to be a monolithic architecture. Within its implementation, the architectural patterns MVC [4] and Client-Server [5] were applied.

MVC (Model-View-Controller) is an architectural pattern that structures an application into three main components: a model (for working with data), a view (for displaying data to the user), and a controller (for managing the interaction logic). This allows to isolate business logic from the user interface, simplifying system support and development. Today, the MVC pattern is actively used in many web applications, precisely because it provides scalability, ease of support, and ease of extending functionality.

Client-Server architecture is also the basis for most modern websites and Internet services. It involves a client part that requests resources and a server part that is responsible for providing them.

Interaction between the server and the client occurs through HTTP requests using the RESTful API [6]. RESTful API is an architectural approach to an application program interface (API) that uses HTTP requests to work with resources. GET, PUT, POST, and DELETE requests provide the ability to get, update, create, and delete resources, respectively.

This choice of system architecture led to ease of development, scalability, and ultimately perfection.

The stage of choosing the tools to implement the application is also quite important, as it determines the success of the project as a whole.

For the server-side implementation, the popular programming language PHP was chosen, which has established itself as a reliable tool with a large developer community, making it easy to learn and use.

Among the numerous PHP frameworks Symfony [7–8] was chosen, known for its flexibility and scalability in projects. The Symfony framework offers a modular approach that allows to quickly develop solutions using ready-made components. This not only reduced development time but also avoided writing repetitive code. An important aspect is also the ability to implement the MVC architectural pattern, which provides a clear division of responsibilities in the code, as mentioned above.

To optimize the processing of requests, the API Platform framework [9] was chosen, which offers a comprehensive set of tools for efficient and fast work with APIs, in accordance with modern standards.

The client side is implemented using the React JavaScript library [10]. It allows to quickly create interactive interfaces and web applications by writing less code than when using regular JavaScript. React provides the ability to divide interfaces into reusable components, which greatly simplifies development.

Nginx was chosen as the web server due to its high performance, reliability, and ability to efficiently handle numerous simultaneous connections.

To deploy the application, Docker [11] is used, which allowed to represent system components in the form of containers, easily manage and scale them. The use of Docker has significantly increased the efficiency and speed of software product development.

This set of technologies ensured convenient and easy development, which resulted in an advanced system that met the requirements.

Figures 1–3 shows a use case diagram that visualizes the main scenarios of interaction between users of different types with the proposed application.

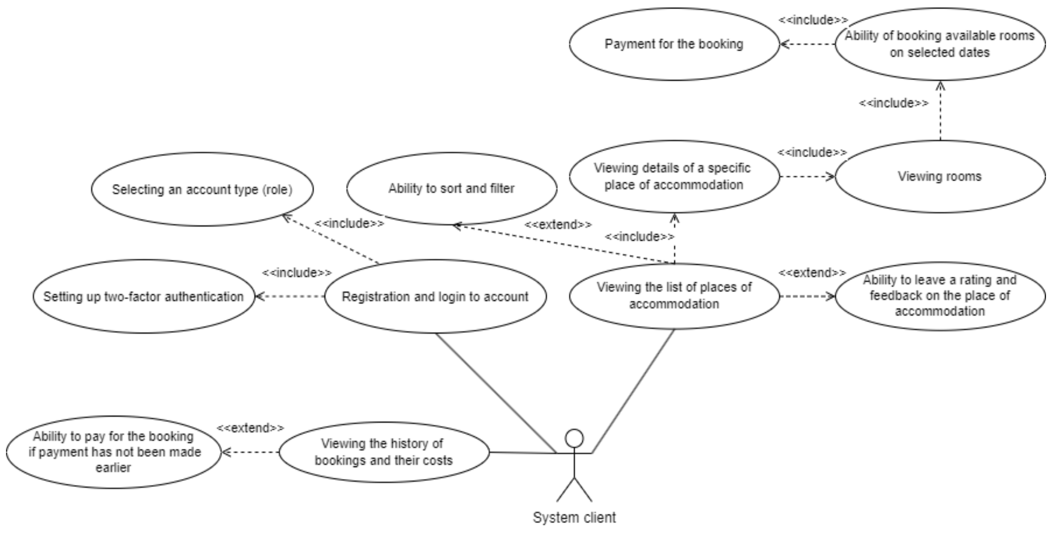

Payment for the booking
<<include>>
Ability of booking available rooms on selected dates
<<include>>
<<include>>
Viewing details of a specific place of accommodation
Viewing rooms
Selecting an account type (role)
Ability to sort and filter
<<include>>
<<extend>>
<<include>>
<<include>>
Setting up two-factor authentication
Registration and login to account
Viewing the list of places of accommodation
<<extend>>
Ability to leave a rating and feedback on the place of accommodation
<<extend>>
Ability to pay for the booking if payment has not been made earlier
Viewing the history of bookings and their costs
System client

Figure 1. System client use case diagram

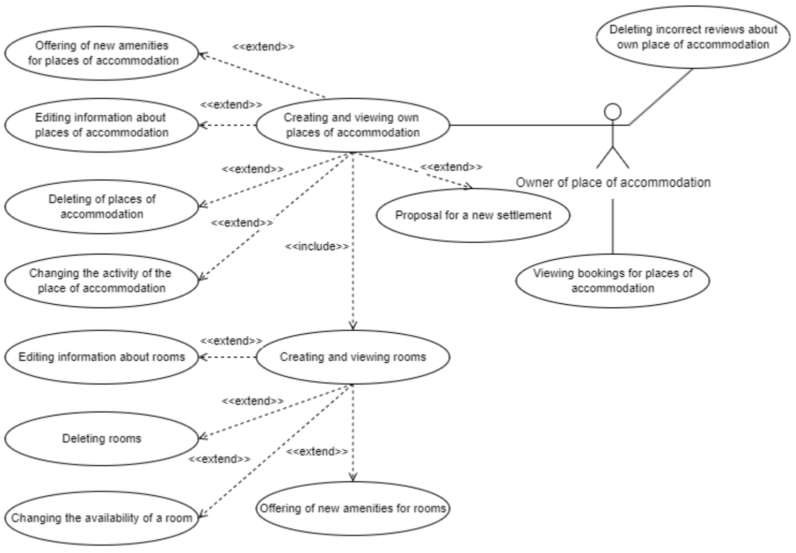

Offering of new amenities for places of accommodation
<<extend>>
Deleting incorrect reviews about own place of accommodation
Editing information about places of accommodation
<<extend>>
Creating and viewing own places of accommodation
Owner of place of accommodation
<<extend>>
<<extend>>
Deleting of places of accommodation
Proposal for a new settlement
<<extend>>
<<include>>
Changing the activity of the place of accommodation
Viewing bookings for places of accommodation
Editing information about rooms
<<extend>>
Creating and viewing rooms
<<extend>>
Deleting rooms
<<extend>>
<<extend>>
Changing the availability of a room
Offering of new amenities for rooms

Figure 2. Owner of place of accommodation use case diagram

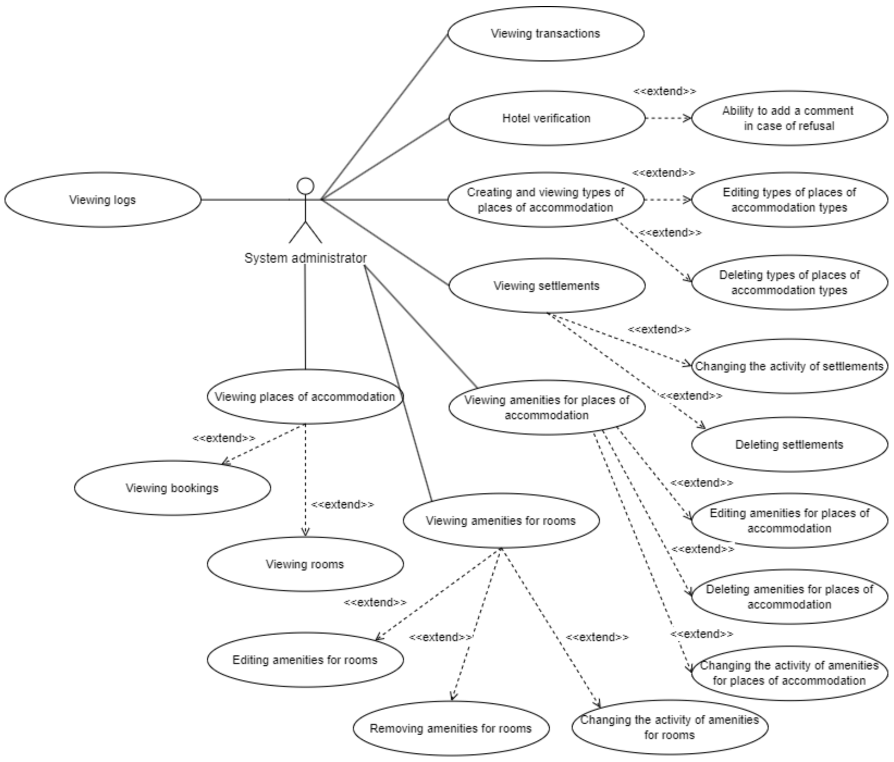

Viewing transactions
Hotel verification
<<extend>>
Ability to add a comment in case of refusal
Viewing logs
Creating and viewing types of places of accommodation
<<extend>>
Editing types of places of accommodation types
<<extend>>
System administrator
Deleting types of places of accommodation types
Viewing settlements
<<extend>>
Changing the activity of settlements
Viewing places of accommodation
Viewing amenities for places of accommodation
<<extend>>
Deleting settlements
<<extend>>
Viewing bookings
<<extend>>
Editing amenities for places of accommodation
Viewing amenities for rooms
Viewing rooms
<<extend>>
Deleting amenities for places of accommodation
<<extend>>
Editing amenities for rooms
<<extend>>
<<extend>>
<<extend>>
Changing the activity of amenities for places of accommodation
Removing amenities for rooms
Changing the activity of amenities for rooms

Figure 3. System administrator use case diagram

This diagram provides a clear idea of what functions the developed system has to meet all the needs of users, among which the main ones are:

- **system client:**
  - registration and login to the account, along with setting up two-factor authorization and choosing a role;
  - viewing places of accommodation (with the use of appropriate filters), along with viewing the list of rooms in them, with their subsequent booking on the selected dates and payment for this reservation;
  - viewing the history of bookings and their costs, with the ability to pay for the reservation within 20 minutes, if payment has not been made earlier;
- **owner of place of accommodation:**
  - creating places of accommodation, along with proposals (subject to the administrator’s approval) of new amenities for them, adding settlements and editing, deleting and changing the activity of places of accommodation;

- adding rooms, along with proposals (subject to the administrator's approval) for new amenities for them, changing the availability of rooms, and editing and deleting them;
- viewing reservations by place of accommodation;
- removal of incorrect reviews about places of accommodation;

- **system administrator:**
  - verification and placement of hotels, with the ability to add a comment in case of refusal to the owner;
  - viewing all accommodations, including their rooms and reservations;
  - creation and viewing accommodation types, edit and delete them;
  - viewing settlements, with the ability to change their activity, edit and delete them;
  - viewing the amenities for accommodation, with the ability to approve, edit and delete them;
  - viewing room amenities with the ability to approve, edit, and delete them;
  - viewing logs;
  - viewing transactions.

The next stage of system design was the creation of a database. The main database in the system is MySQL [12]. Its structure is shown in figure 4.

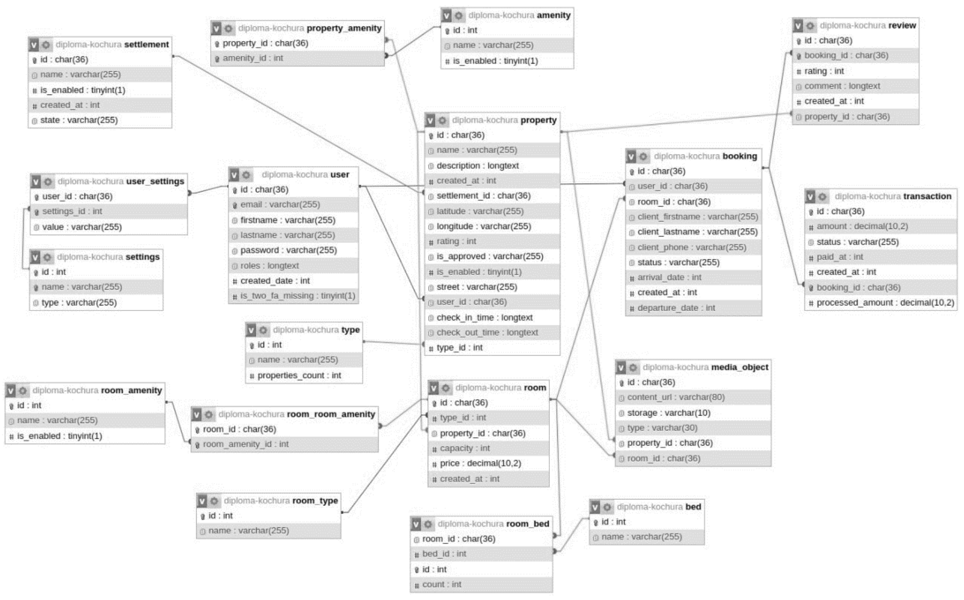


Figure 4. Structure of the MySQL system database

The database contains 18 tables, the main ones are:

- The **«user»** table contains data on system users (email, first name, last name, password, role, availability, two-factor authentication status, etc.)
- The **«property»** table describes the essence of the places of accommodation that users can book (name, description, locality, latitude, longitude, rating, verification status, etc.)
- The **«room»** table describes the essence of the rooms that are located in the places of accommodation (room type, number of guests, price per night, etc.).
- The **«booking»** table contains information about the reservation (data about the person for whom the reservation was made, status, date of arrival, date of departure, etc.).
- The **«transaction»** table contains data on transactions created during booking (amount of the transaction, its status, date of payment, etc.).

The system also uses the MongoDB database [13]. It is the best choice for storing temporary or large data. MongoDB is known for its flexibility and scalability, which makes it possible to efficiently process large amounts of information and quickly perform write and read operations. This made it an ideal complement to the main MySQL relational database in the proposed system. MongoDB is used to store logs and create a temporary user entity at the time of registration. It consists of 2 corresponding collections. Logs are used to track events and actions of users or the system, and a temporary user entity, which is intended to confirm the creation of an account, is used to work with the code that users receive by mail. Let's consider the process of booking accommodation based on the built sequence diagram (fig. 5). The diagram shows all stages of user interaction with the system, from the beginning to the end of this process.

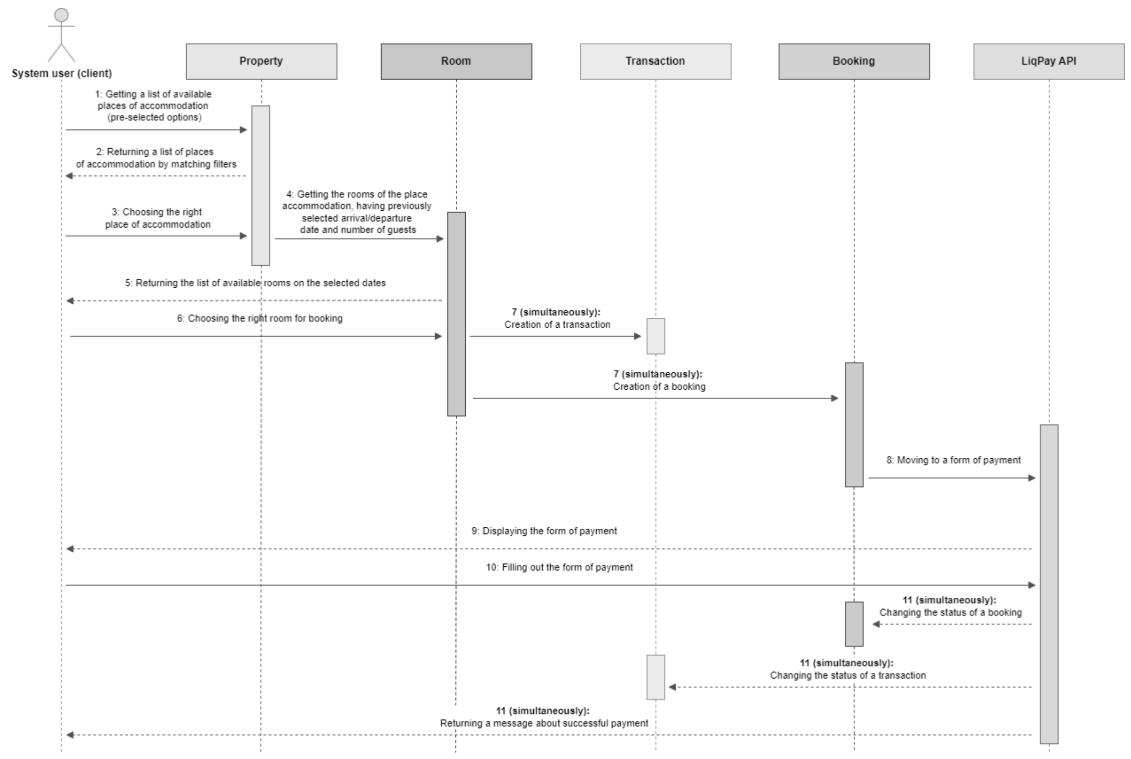
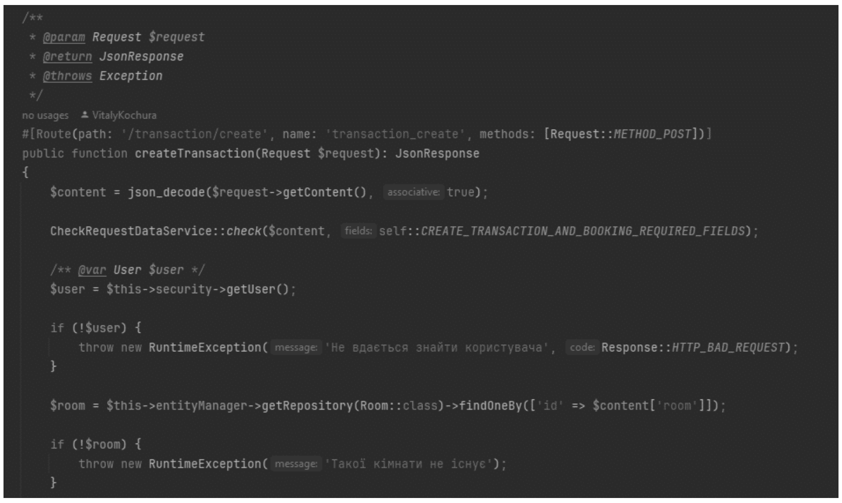

System user (client)
Property
Room
Transaction
Booking
LiqPay API
1: Getting a list of available places of accommodation (pre-selected options)
2: Returning a list of places of accommodation by matching filters
3: Choosing the right place of accommodation
4: Getting the rooms of the place accommodation, having previously selected arrival/departure date and number of guests
5: Returning the list of available rooms on the selected dates
6: Choosing the right room for booking
7 (simultaneously): Creation of a transaction
7 (simultaneously): Creation of a booking
8: Moving to a form of payment
9: Displaying the form of payment
10: Filling out the form of payment
11 (simultaneously): Changing the status of a booking
11 (simultaneously): Changing the status of a transaction
11 (simultaneously): Returning a message about successful payment

Figure 5. Sequence diagram of accommodation booking

First, the system user (client) selects the desired place of accommodation from the list, setting the necessary search parameters. Having read the detailed information about the selected place, indicates the dates of arrival and departure, as well as the number of guests. From the list of available options, which displays only available rooms, the user selects the desired one and goes to the form for entering the booking data. After checking all the details, the user goes to the payment page using the external service LiqPay. At this stage, the system creates a transaction and a reservation with the status «Pending».

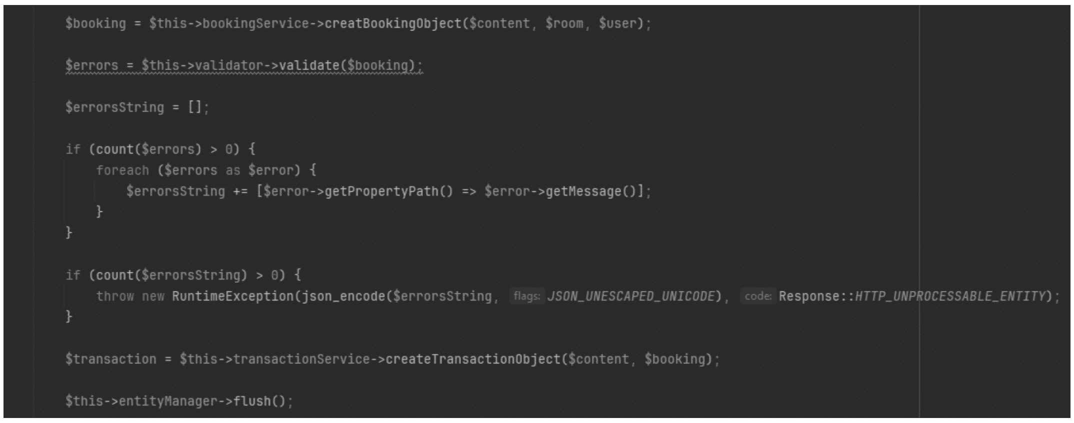
After entering the payment details, the customer completes the payment, after which the status of the reservation and transaction changes to «Paid» and the selected room becomes booked. Since working with the LiqPay service is a rather important stage of booking, let’s consider one of the methods of the Transaction Controller controller. In the createTransaction method, where the Transaction and Booking entities are created, the data from the request is first received, then the availability of the required data is checked using the CheckRequestDataService, after which the existence of the user and the selected room is checked (fig. 6).



```
/**
 * @param Request $request
 * @return JsonResponse
 * @throws Exception
 */
no usages  VitalyKochura
#[Route(path: '/transaction/create', name: 'transaction_create', methods: [Request::METHOD_POST])]
public function createTransaction(Request $request): JsonResponse
{
    $content = json_decode($request->getContent(), associative: true);

    CheckRequestDataService::check($content, fields: self::CREATE_TRANSACTION_AND_BOOKING_REQUIRED_FIELDS);

    /** @var User $user */
    $user = $this->security->getUser();

    if (!$user) {
        throw new RuntimeException( message: 'Не вдається знайти користувача', code: Response::HTTP_BAD_REQUEST);
    }

    $room = $this->entityManager->getRepository(Room::class)->findOneBy(['id' => $content['room']]);

    if (!$room) {
        throw new RuntimeException( message: 'Такої кімнати не існує');
    }
```

Figure. 6. Checking the data from the request, the existence of the user and the selected room

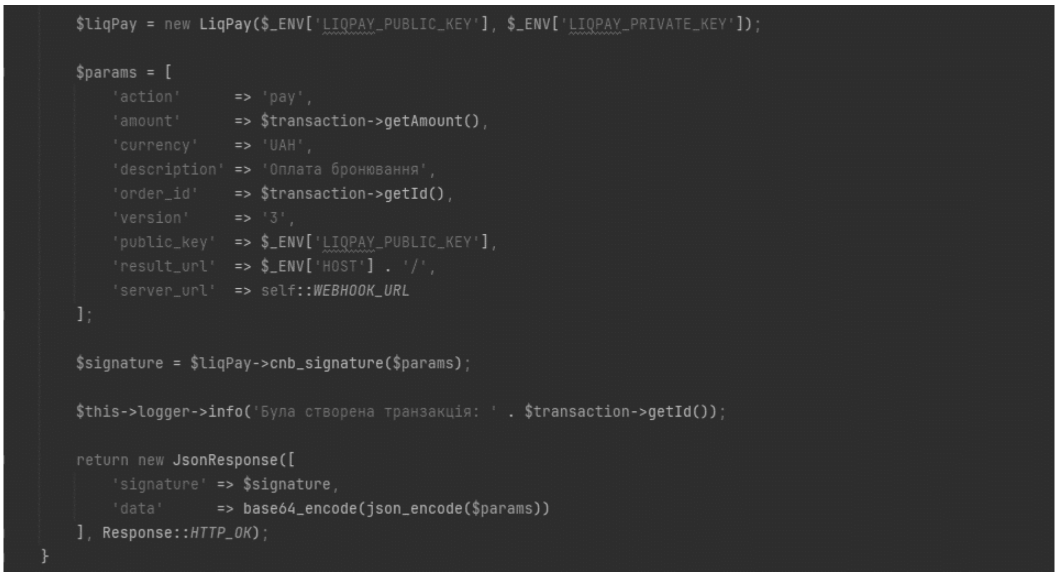
Next, a reservation is created and validated, and a transaction is created using the appropriate services (fig. 7). Both entities receive the status «Pending». After that, they are saved into the database.



```
$booking = $this->bookingService->creatBookingObject($content, $room, $user);

$errors = $this->validator->validate($booking);

$errorsString = [];

if (count($errors) > 0) {
    foreach ($errors as $error) {
        $errorsString += [$error->getPropertyPath() => $error->getMessage()];
    }
}

if (count($errorsString) > 0) {
    throw new RuntimeException(json_encode($errorsString, flags: JSON_UNESCAPED_UNICODE), code: Response::HTTP_UNPROCESSABLE_ENTITY);
}

$transaction = $this->transactionService->createTransactionObject($content, $booking);

$this->entityManager->flush();
```

Figure 7. Creating a booking entity and a transaction and writing them in the database

Subsequently, direct work with the LiqPay service takes place (fig. 8). A signature is generated with predefined parameters that are filled in according to the documentation. In addition, the online service Webhook site is also used, which allows testing and debugging the receipt of webhooks. It is used to receive a payment-callback from the LiqPay service, after which the data is redirected to the website for further processing. The response returns the actual signature with its parameters for further use on the front-end of the application.



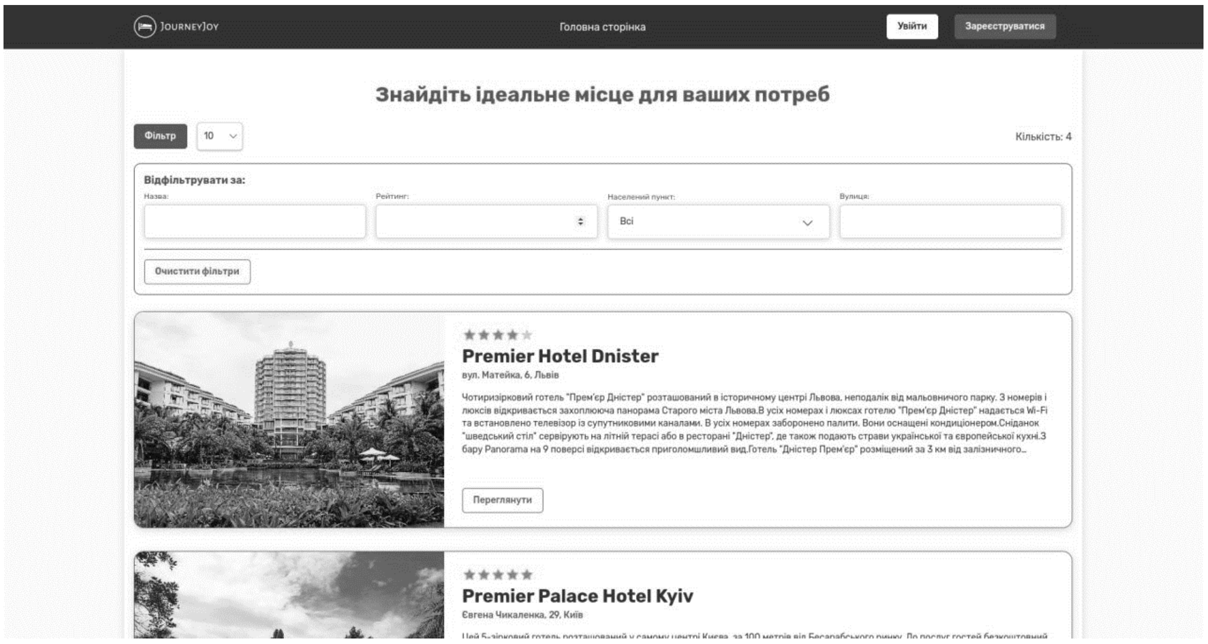
```
    $liqPay = new LiqPay($_ENV['LIQPAY_PUBLIC_KEY'], $_ENV['LIQPAY_PRIVATE_KEY']);

    $params = [
        'action'      => 'pay',
        'amount'      => $transaction->getAmount(),
        'currency'    => 'UAH',
        'description' => 'Оплата бронювання',
        'order_id'    => $transaction->getId(),
        'version'     => '3',
        'public_key'  => $_ENV['LIQPAY_PUBLIC_KEY'],
        'result_url'  => $_ENV['HOST'] . '/',
        'server_url'  => self::WEBHOOK_URL
    ];

    $signature = $liqPay->cnb_signature($params);

    $this->logger->info('Була створена транзакція: ' . $transaction->getId());

    return new JsonResponse([
        'signature' => $signature,
        'data'      => base64_encode(json_encode($params))
    ], Response::HTTP_OK);
}
```

Figure 8. Working with the LiqPay service and returning data to the client side of the system

Below is a demonstration of the application interface and how it works.

At the beginning of the website, the user is greeted by the main page, which displays all the available accommodation (fig. 9).

The client has the opportunity to select the number of places displayed on the page, and the total number of available places is displayed on the right side. There is also a filter that helps to quickly find accommodation that meets specific needs.



Знайдіть ідеальне місце для ваших потреб

Кількість: 4

Відфільтрувати за:

Назва:

Рейтинг:

Населений пункт: Всі

Вулиця:

Очистити фільтри

Premier Hotel Dnister

вул. Матейка, 6, Львів

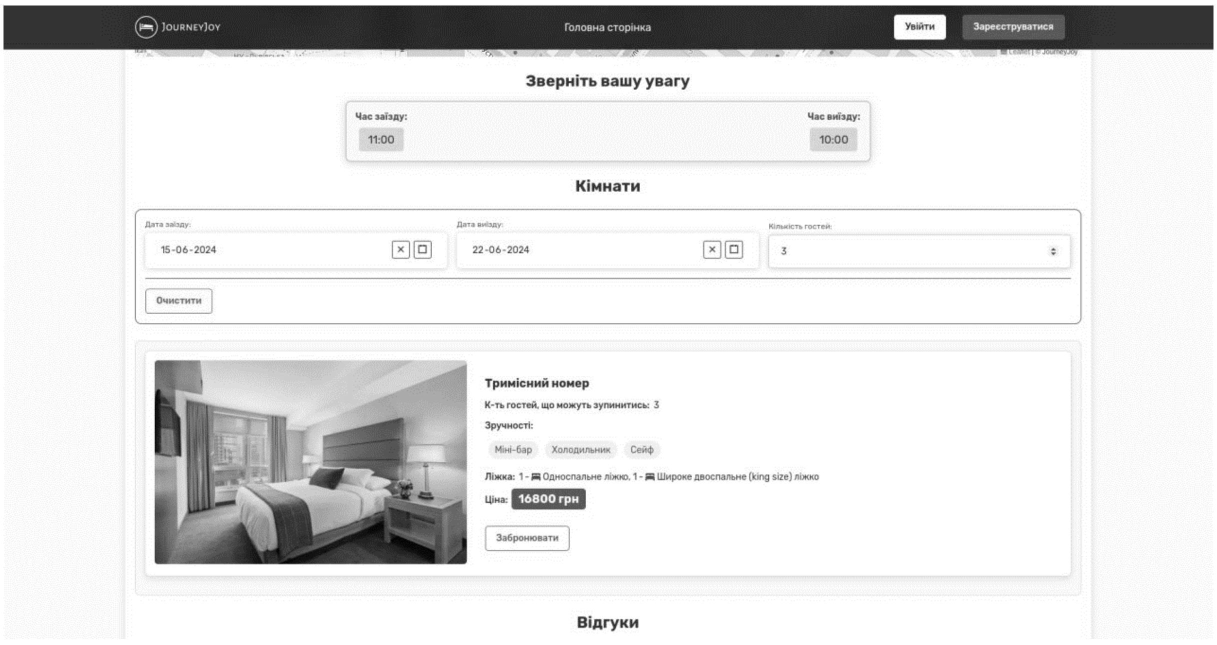
Чотиризірковий готель "Прем'єр Дністер" розташований в історичному центрі Львова, неподалік від мальовничого парку. З номерів і люксів відкривається захоплююча панорама Старого міста Львова.В усіх номерах і люксах готелю "Прем'єр Дністер" надається Wi-Fi та встановлено телевізор із супутниковими каналами. В усіх номерах заборонено палити. Вони оснащені кондиціонером.Сніданок "шведський стіл" сервірують на літній терасі або в ресторані "Дністер", де також подають страви української та європейської кухні.З бару Panorama на 9 поверсі відкривається приголомшливий вид.Готель "Дністер Прем'єр" розміщений за 3 км від залізничного...

Premier Palace Hotel Kyiv

Євгена Чикаленка, 29, Київ

Figure 9. Home page with accommodation search

Figure 10 shows a list of rooms for a given accommodation. In order to view their availability, the user must specify the arrival and departure dates and the number of guests.



Зверніть вашу увагу

Час заїзду: 11:00

Час виїзду: 10:00

Кімнати

Дата заїзду: 15-06-2024

Дата виїзду: 22-06-2024

Кількість гостей: 3

Тримісний номер

К-ть гостей, що можуть зупинитись: 3

Зручності:

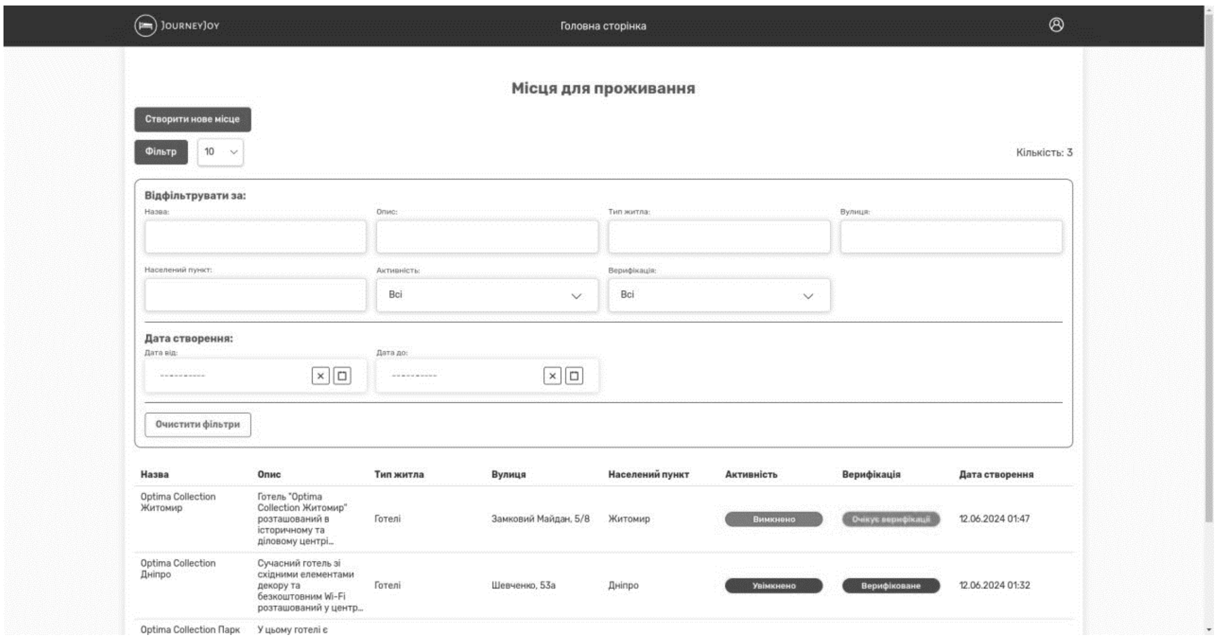
Ліжка: 1 - Односпальне ліжко, 1 - Широке двоспальне (king size) ліжко

Ціна: 16800 грн

Відгуки

Figure 10. Accommodation details (check-in / check-out time, rooms)

When logging into the system as the owner, all the places you own are displayed. It is easy to find what is needed by filtering (fig. 11).



Місця для проживання

Кількість: 3

Відфільтрувати за:

Назва:
Опис:
Тип житла:
Вулиця:
Населений пункт:
Активність: Всі
Верифікація: Всі

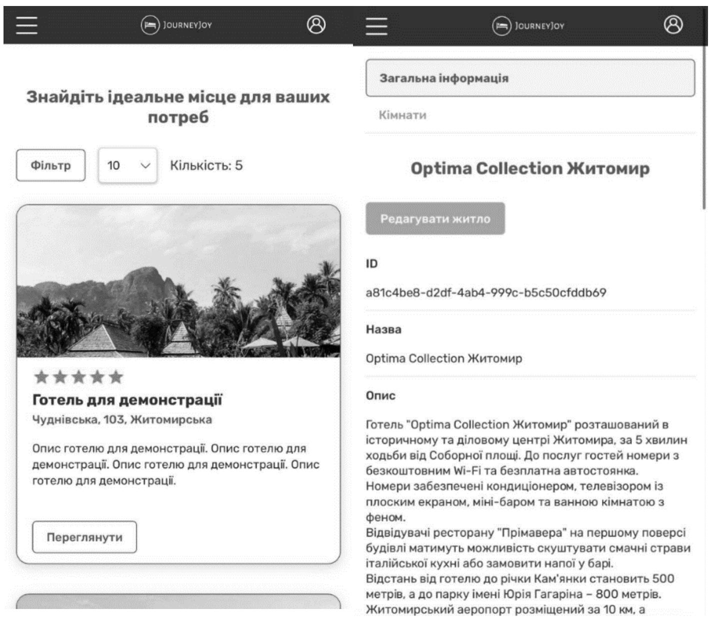
Дата створення:

Дата від: ----------
Дата до: ----------

Назва	Опис	Тип житла	Вулиця	Населений пункт	Активність	Верифікація	Дата створення
Optima Collection Житомир	Готель "Optima Collection Житомир" розташований в історичному та діловому центрі...	Готелі	Замковий Майдан, 5/8	Житомир	Вимкнено	Очікує верифікації	12.06.2024 01:47
Optima Collection Дніпро	Сучасний готель зі східними елементами декору та безкоштовним Wi-Fi розташований у центр...	Готелі	Шевченко, 53а	Дніпро	Увімкнено	Верифіковане	12.06.2024 01:32
Optima Collection Парк	У цьому готелі є						

Figure 11. List of owner's places of accommodation

The application is fully responsive, and each page can be easily viewed on mobile devices. This provides users with convenient access to all functions and services of the application, regardless of the type or size of their device screen, which significantly improves the overall user experience (fig. 12).



Знайдіть ідеальне місце для ваших потреб

Кількість: 5

★★★★★

Готель для демонстрації

Чуднівська, 103, Житомирська

Опис готелю для демонстрації. Опис готелю для демонстрації. Опис готелю для демонстрації. Опис готелю для демонстрації.

Optima Collection Житомир

ID

a81c4be8-d2df-4ab4-999c-b5c50cfddb69

Назва

Optima Collection Житомир

Опис

Готель "Optima Collection Житомир" розташований в історичному та діловому центрі Житомира, за 5 хвилин ходьби від Соборної площі. До послуг гостей номери з безкоштовним Wi-Fi та безплатна автостоянка.
Номери забезпечені кондиціонером, телевізором із плоским екраном, міні-баром та ванною кімнатою з феном.
Відвідувачі ресторану "Прімавера" на першому поверсі будівлі матимуть можливість скуштувати смачні страви італійської кухні або замовити напої у барі.
Відстань від готелю до річки Кам'янки становить 500 метрів, а до парку імені Юрія Гагаріна – 800 метрів.
Житомирський аеропорт розміщений за 10 км, а

Figure 12. Application adaptability

During the development of the software product, special attention was paid to testing and error handling, which significantly improved the reliability and security of the system. Thorough validation of forms, the use of loaders for buttons and filters, and blocking of inaccessible actions helped to ensure the intuitiveness and stability of the application. This comprehensive approach to handling errors and preventing unwanted user actions improved the overall user experience and minimized the possibility of errors in the process of interacting with the system.

Conclusions and prospects for further research. As a result of the development, a perfect software product was obtained that meets the formulated requirements and expectations of users. The introduction of such a hotel and accommodation booking management service for traveling in Ukraine will provide a convenient booking process, increase the level of customer service and simplify the process of managing accommodation.

References:

1. *Broniuvannia hoteliv, kvartyr, khosteliv*, [Online], available at: https://hotels24.ua/
2. *GoHotels – bezkoshtovne onlain broniuvannia hoteliv Ukrainy*, [Online], available at: https://gohotels.com.ua/
3. *Hoteli ta apartamenty Ukrainy*, [Online], available at: https://www.hotels-of-ukraine.com/
4. «Pobudova dynamichnykh vebzastosunkiv za dopomohoiu MVC», [Online], available at: https://blog.ithillel.ua/articles/building-dynamic-web-applications-using-mvc
5. «Rozuminnia Kliient-Servernoi Arkhitektury na prykladakh», [Online], available at: https://dou.ua/forums/topic/44636/
6. «Vstup do REST API – RESTful vebservisy», [Online], available at: https://robotdreams.cc/uk/blog/466-vstup-do-rest-api-restful-vebservisi
7. *Symfony documentation*, [Online], available at: https://symfony.com/doc/current/index.html
8. «Kurs z Symfony», [Online], available at: https://spacelab.ua/course/symfony/
9. «API Platform documentation», [Online], available at: https://api-platform.com/docs/distribution/
10. *React documentation*, [Online], available at: https://react.dev/
11. *Docker documentation*, [Online], available at: https://docs.docker.com/
12. *MySQL documentation*, [Online], available at: https://dev.mysql.com/doc/
13. *Build PHP Symfony Apps with MongoDB Atlas*, [Online], available at: https://mongodb-developer.github.io/symfony-mongodb-rental-workshop/

Kochura Vitaly Viktorovych – Master student at the Zhytomyr Polytechnic State University.
https://orcid.org/0009-0002-9598-7380.
Scientific interests:
– information systems and technologies.
E-mail: kochuravitaly@gmail.com.

Kushnir Nadia Oleksandrivna – Senior Lecturer at the department of software engineering at Zhytomyr Polytechnic State University.
https://orcid.org/0000-0002-0797-3687.
Scientific interests:
– combinatorial optimization; information technologies.
E-mail: kipz_kno@ztu.edu.ua.

Loktikova Tamara Mykolaivna – Senior Lecturer at the department of software engineering at Zhytomyr Polytechnic State University.
https://orcid.org/0000-0002-3525-0179.
Scientific interests:
– digital image processing; information systems and technologies.
E-mail: dfikt_ltn@ztu.edu.ua.

Morozov Andriy Vasyliovych – Ph.D. in Engineering, Associate Professor, vice-rector for scientific and pedagogical work of Zhytomyr Polytechnic State University.
https://orcid.org/0000-0003-3167-0683.
Scientific interests:
– combinatorial optimization; information technologies.
E-mail: morozov@ztu.edu.ua.

Кочура В.В., Кушнір Н.О., Локтікова Т.М., Морозов А.В.

Сервіс керування бронюванням готельного та житлового фонду для подорожей Україною

Навіть у такий складний як сьогоднішній час подорожі країною залишаються актуальними та необхідними для багатьох людей. Вони потрібні по роботі, надихають та покращують настрій, роблять життя наповненим, яскравим, додають сенсу. Це спонукало до створення сервісу керування бронюванням готельного та житлового фонду для подорожей Україною. Система дозволяє обрати бажане місце проживання за певними критеріями, оплатити його, переглянути історію бронювань в особистому кабінеті й залишити відгук. А також надає інструменти для співпраці з власниками таких місць проживань, що можуть розміщувати їх на вебсайті, керувати ними та оброблювати бронювання. Для побудови сервісу було використано монолітну архітектуру та патерни MVC і Client-Server. Серверна частина реалізована на мові програмування PHP з використанням фреймворку Symfony, а клієнтська частина – з використанням JavaScript бібліотеки React. Як вебсервер обрано Nginx. Для розгортання застосунку використаний інструмент Docker. У системі застосована MySQL як основна база даних, що складається з 18 таблиць, та доповнена MongoDB, що має 2 колекції. В процесі розробки особлива увага приділялася тестуванню та обробці помилок, що дозволило підвищити надійність, безпеку та інтуїтивність системи, забезпечуючи стабільну роботу і мінімізуючи можливість виникнення помилок. Розроблений сервіс відповідає сучасним вимогам та має такі особливості: сучасний та адаптивний дизайн, зручний інтерфейс з можливістю створення особистого кабінету, підтримка входу через Google.

Ключові слова: *вебсервіс; бронювання; житло; Symfony; React.*

The article was sent to the editorial board on 18.04.2025.