

Д. Л. КИРИЙЧУК

кандидат технічних наук, доцент,
доцент кафедри програмних засобів і технологій
Херсонський національний технічний університет
ORCID: 0000-0002-4905-6932

А. В. ЯЦКЕВИЧ

студент кафедри програмних засобів і технологій
Херсонський національний технічний університет
ORCID: 0000-0002-5823-7334

О. М. ЛЯШЕНКО

кандидат технічних наук, доцент,
доцент кафедри програмних засобів і технологій
Херсонський національний технічний університет
ORCID: 0000-0002-5429-8389

ПРОЄКТУВАННЯ ПРОГРАМНОГО СЕРВІСУ СКЛАДАННЯ ПОЗОВНИХ ЗАЯВ «PROSHUSUD»

Метою роботи є проєктування програмного сервісу складання позовних заяв «ProshuSud».

Програмний сервіс дозволяє користувачеві грамотно скласти позовну заяву, висвітлити основні нюанси, головні обставини юридичної справи, скласти перелік відповідних документів, визначити суд, до якого подається позов, розрахувати розмір судового збору для відповідної категорії справи.

За допомогою програмного сервісу «ProshuSud» користувач може отримати консультації з будь-яких юридичних питань в онлайн режимі, замовити складання позову та супровід в суді по всій території України.

При розробці програмного сервісу було використано методологію об'єктно-орієнтованого програмування, технології крос-платформної розробки Node.js та React, перевагою яких є можливість асинхронного виконання коду, тобто сервер може обробляти безліч запитів одночасно, не очікуючи на завершення кожного з них, що призводить до більш ефективного використання ресурсів та швидкого відгуку сервера.

В роботі розроблено архітектуру програмного сервісу із застосуванням патерну MVC. Описано функціональні можливості програмного сервісу «ProshuSud». Побудовано діаграму варіантів використання в UML, на якій відображено взаємодію між варіантами використання, що представляють функції системи, та дійовими особами, які представляють людей або системи, які отримують або передають інформацію до програмного сервісу. Також описано процес завантаження проєкту до хмарного сервісу GitHub та роботу із системою контролю версій Git, що дозволяє захищати вихідний код від втрати, забезпечує командну та розподілену роботу та допомагає скасувати зміни, що зроблені у проєкті.

Також в роботі розроблено БД програмного сервісу, описано файлову структуру проєкту, розроблено інтерфейс користувача, наведено програмні методи роботи з формами та описано процес керування функціональними компонентами, що були використані при розробці проєкту.

Ключові слова: програмний сервіс, позовні заяви, технології крос-платформної розробки, методологія об'єктно-орієнтованого програмування.

D. L. KYRYICHUK

Candidate of Technical Sciences, Associate Professor,
Associate Professor at the Department of Software Tools and Technologies
Kherson National Technical University
ORCID: 0000-0002-4905-6932

A. V. YATSKEVYCH

Student at the Department of Software Tools and Technologies
Kherson National Technical University
ORCID: 0000-0002-5823-7334

O. M. LIASHENKO

Candidate of Technical Sciences, Associate Professor,
Associate Professor at the Department of Software Tools and Technologies
Kherson National Technical University
ORCID: 0000-0002-5429-8389

DESIGN A SOFTWARE SERVICE FOR DRAFTING CLAIMS “PROSHUSUD”

The purpose of the work is to design a software service for drafting claims “ProshuSud”. The software service allows the user to competently draw up a statement of claim, highlight the main nuances, the main circumstances of the legal case, make a list of relevant documents, determine the court to which the lawsuit is filed, and calculate the amount of court fee for the corresponding category of the case. With the help of the software service “ProshuSud”, the user can get advice on any legal issues online, order a lawsuit and support in court throughout Ukraine. When developing the software service, the methodology of object-oriented programming, cross-platform development technologies Node.js and React were used, the advantage of which is the possibility of asynchronous code execution, that is, the server can process many requests simultaneously without waiting for the completion of each of them, which leads to more efficient use of resources and fast server response.

The architecture of the software service using the MVC pattern was developed. The functionality of the “ProshuSud” software service is described. A UML use case diagram is constructed, which shows the interaction between use cases representing system functions and actors representing people or systems that receive or transmit information to a software service. The process of loading the project into the GitHub cloud service and working with the Git version control system is described, which allows protecting the source code from loss, provides teamwork and distributed work and helps to cancel changes made in the project.

The work also developed a database of the software service, describes the file structure of the project, developed a user interface, provides software methods for working with forms and describes the process of managing the functional components that were used in the development of the project.

Key words: *software service, claims, cross-platform development technologies, object-oriented programming methodology.*

Постановка проблеми

Нині все більш поширеними стають програмні сервіси, які надають консультаційні послуги з будь-яких юридичних питань в онлайн режимі. Такі програмні рішення також дозволяють суб'єктам юридичної діяльності зберігати, керувати та обмінюватися інформацією із захищених баз даних.

Таким чином, розроблення програмного сервісу складання позовних заяв є актуальною науково-прикладною задачею, а його побудова потребує виявлення та аналізу сучасних тенденцій розвитку та застосування таких програм, насамперед, на основі провідних наукових публікацій.

Аналіз останніх досліджень і публікацій

Близькою за підходом до проектування сервісів для суб'єктів юридичної діяльності є робота [1], в якій описано процес розроблення інтегрованої інформаційної системи для комплексного забезпечення адвокатської діяльності. Описано практику застосування інформаційної системи та проаналізовано проблеми, що виникли під час її використання.

Проблемам інформаційної безпеки адвокатської діяльності присвячено роботу [2]. Автором виділено класи загроз інформаційної безпеки: внутрішні загрози адвокатської діяльності (незаконна діяльність інсайдерів), зовнішні загрози кіберпростору (віруси, хакерські атаки тощо), змішані форми загроз, тобто поєднання зусиль зовнішніх і внутрішніх порушників інформаційної безпеки.

В роботі [3] проведено дослідження, що дозволяє оцінити перспективи впровадження інформаційних систем в роботу суб'єктів юридичної діяльності. Розглянуто виклики, що виникають внаслідок значного та радикального впливу інформаційних технологій на закон та юридичну практику.

В роботі [4] описано процес розроблення та впровадження інформаційних технологій у діяльність юридичних компаній. В роботі також розглянуто питання впливу інформаційних технологій на сучасну юридичну практику.

В роботі [5] описано процес впровадження інформаційних технологій в юридичних фірмах. Розглянуто питання державної підтримки процесів діджиталізації юридичних послуг.

Формулювання мети дослідження

Метою статті є проектування програмного сервісу складання позовних заяв «ProshuSud» із застосуванням методології об'єктно-орієнтованого програмування, технології крос-платформної розробки Node.js та React.

Викладення основного матеріалу дослідження

За допомогою програмного сервісу «ProshuSud» користувач може отримати консультації з будь-яких юридичних питань в онлайн режимі, замовити складання позову та супровід в суді по всій території України.

Таким чином, до основних функціональних особливостей програмного сервісу можна віднести такі:

1. Розірвання шлюбу.
2. Стягнення аліментів на утримання неповнолітніх дітей.
3. Позовна заява про оспорування батьківства.
4. Позбавлення батьківських прав.
5. Розділ майна подружжя.
6. Стягнення аліментів на утримання повнолітніх дітей.
7. Встановлення режиму спілкування з дитиною.
8. Наказ про стягнення аліментів.
9. Стягнення додаткових витрат на утримання дитини.
10. Визначення місця проживання дитини з батьком.

Для підготовки позову треба зареєструватися в програмному сервісі «ProshuSud» та заповнити анкету.

Діаграму варіантів використання в UML подано на рис. 1. На діаграмі відображається взаємодія між варіантами використання, що представляють функції системи, та дійовими особами, які представляють людей або системи, які отримують або передають інформацію до програмного сервісу. Діаграма на рис. 1 визначає загальну функціональність сервісу.

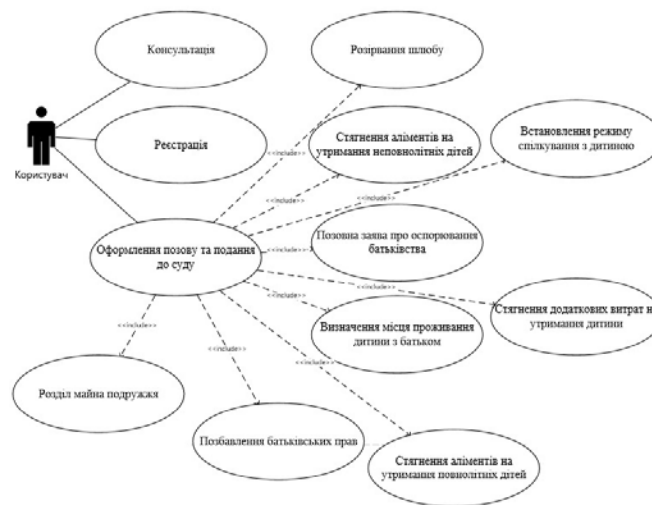


Рис. 1. Діаграма варіантів використання програмного сервісу

Архітектуру програмного сервісу розроблено із застосуванням патерну MVC (рис. 2).

Модель визначає структуру та логіку даних, що використовуються у програмі. В межах програмного сервісу Модель визначає структуру бази даних та модулі програми, які взаємодіють з базою даних.

Подання визначає візуальну частину програмного сервісу.

Контролери обробляють вхідні http-запити, використовуючи для обробки Модель та Подання, і відправляють у відповідь клієнту деякий результат обробки (html-код). Контролери дозволяють зв'язувати Подання і Модель та виконувати певну логіку для обробки запиту. Контролер взаємодіє з Моделлю та обслуговує реакцію та функціональність перегляду. Коли кінцевий користувач робить запит, він надсилається до контролера, який взаємодіє з базою даних.

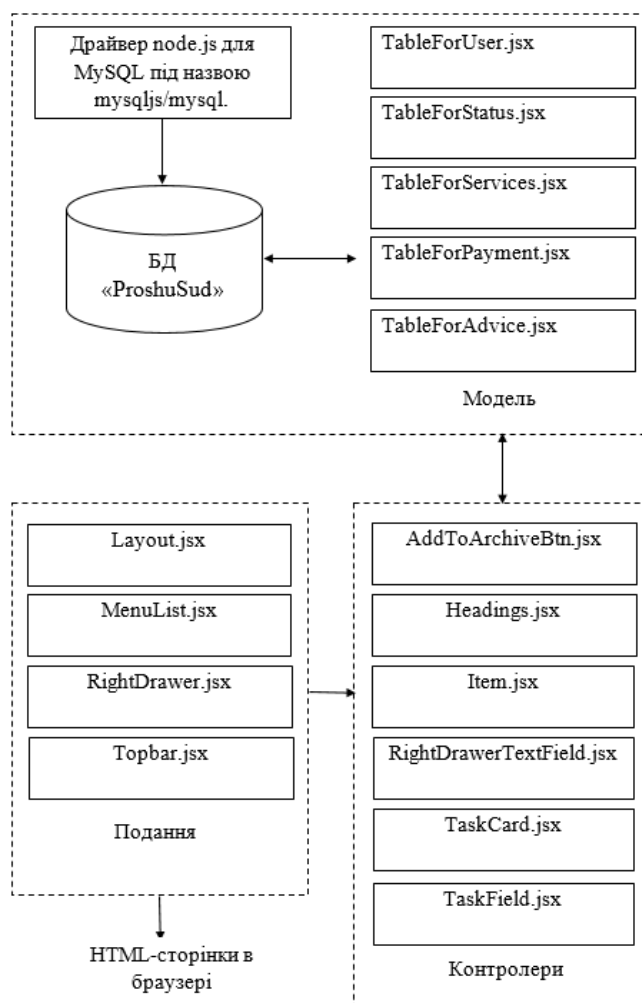


Рис. 2. Архітектура програмного сервісу із застосуванням патерну MVC

Для розроблення БД програмного сервісу було використано MySQL Workbench. Розроблювана БД включає такі об'єкти: Користувач, Замовлення, СтатусЗамовлення, Оплата, ОплатаЗамовлення, Послуга, ТипПослуги.

Зв'язки між об'єктами БД програмного сервісу (рис. 3):

1. Користувач – Послуга (М:N), зв'язок забезпечено за допомогою додаткової таблиці «Замовлення».
2. Замовлення – Оплата (М:N), зв'язок забезпечено за допомогою додаткової таблиці «ОплатаЗамовлення».
3. Замовлення – СтатусЗамовлення (1:N).
4. Послуга – ТипПослуги (1:N).

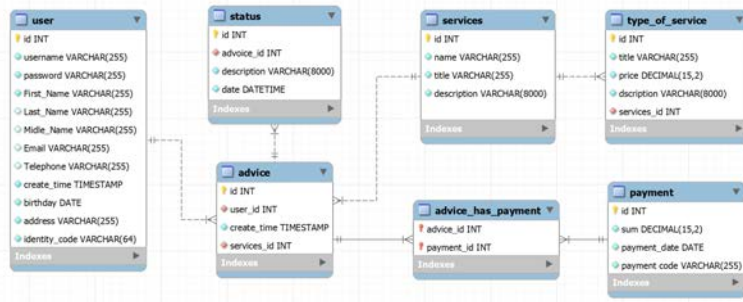


Рис. 3. Зв'язки між об'єктами БД

Для завантаження проєкту до хмарного сервісу GitHub та роботи із системою контролю версії Git було використано GitHub Desktop.

Програма GitHub Desktop синхронізується з віддаленим репозиторієм Git і надає можливість розробнику додавати нові файли у віддалений репозиторій. Структуру каталогів проєкту в GitHub подано на рис. 4.

Каталог public/ містить усі статичні файли програми, такі як index.html, favicon.ico.

Каталог src/ містить усі вихідні файли програми.

Каталог components/ містить всі компоненти React-програми, які використовуються для створення інтерфейсу користувача.

Каталог containers/ містить компоненти, які є «контейнерами» для інших компонентів і пов'язують Redux-стан з компонентами. При використанні Redux, основні дані для всієї програми представлені єдиним JavaScript об'єктом з посиланням на стан або дерево станів.

Каталог actions/ містить усі файли дій, які визначають, які типи дій можна виконувати у програмі.

Каталог reducers/ містить усі файли редьюсерів, які визначають, як змінюється стан програми у відповідь на дії.

Каталог services/ містить усі сервіси, які використовуються в програмі, такі як запити до API або маніпуляції з даними.

Файл index.js є головним файлом програми, який є точкою входу у програму.

Файл App.js є основним компонентом програми і містить маршрутизацію та загальну структуру програми.

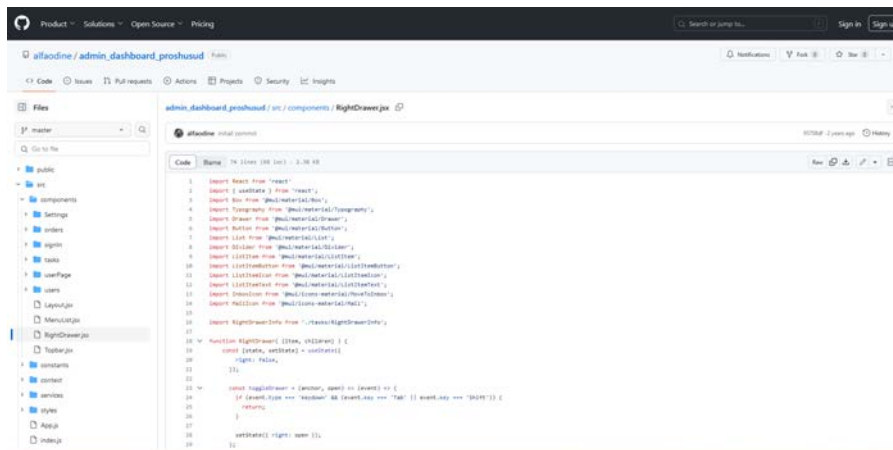


Рис. 4. Структура каталогів проєкту в GitHub

Для розроблення та візуалізації інтерфейсу користувача було використано React – бібліотеку JavaScript. В React розробнику доступні для використання всі стандартні елементи форм, які є в html.

Головним у React є стан. Будь-яку зміну у формі має бути перенесено до стану. Елементи форм, дані яких зберігаються в стані React, називаються керованими компонентами (controlled components) [6; 7].

Реалізація валідації форм в React передбачає перевірку їх введених значень, і якщо ці значення відповідають вимогам, тоді відбувається зміна стану компонента.

На рис 5 форма містить поля для введення даних позивача: дата народження, адреса реєстрації, ідентифікаційний код, телефон і т.д.

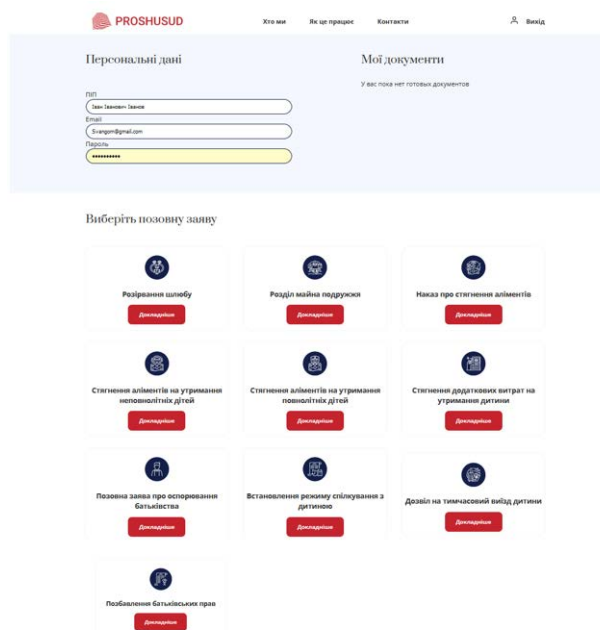


Рис. 7. Форма «Персональний кабінет»

Висновки

Описано функціональні можливості програмного сервісу «ProshuSud». Побудовано діаграму варіантів використання програмного сервісу. Описано процес завантаження проекту до хмарного сервісу GitHub. Описано файлову структуру проекту. Розроблено інтерфейс користувача за допомогою React – бібліотеки JavaScript. Наведено програмні методи роботи з формами. Побудовано діаграму об'єктів бази даних «ProshuSud». Описано зв'язки об'єктів БД. Наведено приклади роботи програмного сервісу.

Список використаної літератури

1. Lawlor R.C. Information Technology and the Law. *Advances in Computers*. 3(1).2021. Pp. 299–352.
2. Резнікова Г.І. Інформаційна безпека адвокатської діяльності: Криміналістичний погляд. *Науковий вісник Міжнародного гуманітарного університету. Серія Юриспруденція*. 29(2). 2017. С. 117–122.
3. Julian Webb. Information technology and the future of legal education: a provocation. *Griffith journal of law & human dignity*. 1(1).2019. Pp. 1–33.
4. Tanya Du Plessis. Competitive Legal Professionals' use of Technology in Legal Practice and Legal Research. *Potchefstroom Electronic Law Journal/Potchefstroomse Elektroniese Regsblad*. 11(4).2008. pp. 32–60.
5. Owoeye J.E. Information Communication Technology (ICT) Use as a Predictor of Lawyers' Productivity. *Library Philosophy and Practice*. 11(1). 2011. Pp. 662–773.
6. Alex Banks, Eve Porcello. Learning React: Modern Patterns for Developing React Apps. *O'Reilly Media*. 2020. P. 307.
7. David Griffiths, Dawn Griffiths. React Cookbook: Recipes for Mastering the React Framework. *O'Reilly Media*. 2021. P. 510.

References

1. Lawlor R.C.(2020) Information Technology and the Law. *Advances in Computers*, 3(1), pp. 299–352 [in English].
2. Reznikova H.I (2017) Informatsiina bezpeka advokatskoj diialnosti: Kryminalistychnyi pohliad [Information security of advocacy: a criminalistic view]. *Naukovyi visnyk Mizhnarodnoho humanitarnoho universytetu. Seriya Yurysprudentsiia [Scientific Bulletin of the International Humanities University. Series: Jurisprudence]*, 29(2), P. 117–122 [in Ukrainian].
3. Julian Webb. (2019) Information technology and the future of legal education: a provocation. *Griffith journal of law & human dignity*, 1(1), pp. 1–33 [in English].
4. Tanya Du Plessis. (2008) Competitive Legal Professionals' use of Technology in Legal Practice and Legal Research. *Potchefstroom Electronic Law Journal/Potchefstroomse Elektroniese Regsblad*, 11(4), pp. 32–60 [in English].
5. Owoeye J.E. (2011) Information Communication Technology (ICT) Use as a Predictor of Lawyers' Productivity. *Library Philosophy and Practice*, 11(1), pp. 662–773 [in English].
6. Alex Banks, Eve Porcello. (2020) Learning React: Modern Patterns for Developing React Apps. *O'Reilly Media*, 307 p. [in English].
7. David Griffiths, Dawn Griffiths. (2021) React Cookbook: Recipes for Mastering the React Framework. *O'Reilly Media*, 510 p. [in English].