

УДК 004.9: 004.4

Л. В. Крилик, А. А. Яровий, І. В. Безсмертна

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ АВТОМАТИЗАЦІЇ ПОДАВАННЯ ЗАЯВОК ДО КОМУНАЛЬНИХ ПІДПРИЄМСТВ

Вінницький національний технічний університет, Вінниця

Анотація. В роботі розглянуто проблематику частини комунальної галузі, яка відповідає за взаємодію комунального підприємства та осіб, що воно обслуговує. Після проведеного аналізу доведено доцільність діджиталізації та реалізації WEB-орієнтованого програмного продукту, що задовольнив би усі потреби, значно спростив та прискорив процес взаємодії, потребував би мінімальних зусиль на навчання, а також був доступний користувачам різних комп'ютерних та мобільних пристроїв за наявності інтернету. На базі цього, з метою автоматизації та удосконалення процесів отримання актуальної інформації та проектування прозорої системи, було описано особливості реалізації інформаційної системи автоматизації подання заявок до комунальних підприємств, її архітектуру, структуру та програмну реалізацію такими інструментами як NodeJS, ReactJS, GraphQL, MongoDB. Як результат розроблено інформаційну систему автоматизації подання заявок до комунальних підприємств, що повністю задовольняє базові потреби користувачів послуг комунальних підприємств та спрощує процес отримання послуг та інформації.

Ключові слова: інформаційна система, WEB-технології, методології проектування програмного забезпечення, автоматизація подання заявок, комунальні підприємства, моніторинг.

Аннотация. В работе рассмотрена проблематика части коммунальной отрасли, которая отвечает за взаимодействие коммунального предприятия и лиц, которых оно обслуживает. После проведенного анализа доказана целесообразность диджитализации и реализации WEB-ориентированного программного продукта, который удовлетворил бы все потребности, значительно упростил и ускорил процесс взаимодействия, требовал минимальных усилий на обучение, а также был доступен пользователям различных компьютерных и мобильных устройств при наличии интернета. На базе этого, с целью автоматизации и усовершенствования процессов получения актуальной информации и проектирование прозрачной системы, было описано особенности реализации информационной системы автоматизации подачи заявок в коммунальные предприятия, ее архитектуру, структуру и программную реализацию такими инструментами как NodeJS, ReactJS, GraphQL, MongoDB. В результате разработана информационная система автоматизации подачи заявок в коммунальные предприятия, которая полностью удовлетворяет базовые потребности пользователей услуг коммунальных предприятий и упрощает процесс получения услуг и информации.

Ключевые слова: информационная система, WEB-технологии, методологии проектирования программного обеспечения, автоматизация подачи заявок, коммунальные предприятия, мониторинг.

Abstract. The work considers the issue of the part of the utility industry that is responsible for the interaction of the utility company and the people it serves. After the analysis, the feasibility of digitalization and implementation of a WEB-oriented software product was proved, which would satisfy all the needs and significantly simplify and accelerate the interaction process, required minimal training efforts, and was also available to users of various computer and mobile devices with the presence of the Internet. Based on this, in order to automate and improve the processes of obtaining relevant information and designing a transparent system, the features of the implementation of information technology for automating applications to utilities, its architecture, structure and program implementation of tools such as NodeJS, ReactJS, GraphQL, MongoDB were described. As a result, an information system for automating the submission of applications to utilities has been developed, which fully satisfies the basic needs of users of utility services and simplifies the process of obtaining services and information.

Keywords: information system, WEB-technologies, software design methodologies, automation of application submission, utilities, monitoring.

DOI: <https://doi.org/10.31649/1999-9941-2021-51-2-23-29>.

Вступ

Ми живемо в добу, де панують цифрові технології, у час, коли усі процеси підлягають автоматизації для подальшого їх спрощення, у час, коли більшість користувачів користуються веб-платформами з мобільних пристроїв, оскільки це зручно і доступно у будь-який момент. Галузь комунальних підприємств залишається однією з невід'ємних частин щоденного життя пересічних громадян і у теперішній реалізації не є сучасним рішенням, тому і потребує автоматизації, що дозволило б спростити процеси взаємодії та обміну інформації, прискорити їх та структурувати. Проте, сучасна реалізація потребує сучасних рішень з використанням перевірених технологій обраних під потреби та задачі системи, що вона вирішує. Досягнення поставленої цілі неможливо і без правильних підходів вибору архітектури та користувацького інтерфейсу, адже саме сукупність усіх влучно підібраних інструментів є запорукою створення повноцінної інформаційної технології.

Актуальність

У сучасному світі фактор часу є надзвичайно важливим, тому жодна людина не хоче марнувати його в довгезелних чергах для простого отримання інформації, і тим більше повторювати цю операцію ще декілька разів або витримувати довгезелне очікування на гарячій лінії, яке може не принести жодного результату. Адже це все забирає багато часу, нервів та сил, а також псує настрої і подекуди може навіть нашкодити здоров'ю. Саме тому, у більшості прогресивних країн установи намагаються спростити та автоматизувати такі процеси, щоб їх громадяни були не прив'язані до місця і не марнували так багато часу, а наприклад сидючи вдома чи перебуваючи на роботі за лічені хвилини могли отримати чітку та актуальну інформацію щодо роботи комунального підприємства [1].

Оскільки держава розуміє необхідність діджиталізації комунальної галузі, в результаті низки реформ було створено законопроект, що підтверджує актуальність такого рішення [2].

Варто зазначити, що в ході проведених досліджень, прямих аналогів не знайдено, проте є ряд систем, що вирішують схожі проблеми [3]:

Гаряча лінія – система, що функціонує вже багато років, де можна отримати інформацію (або надати її) про певні поломки. Проте цей спосіб організації має ряд таких недоліків: довге очікування, що подекуди змушує відмовитись від використання; ускладнений, а подекуди неможливий доступ до інформації; неактуальна інформація; відсутність прозорості; присутній значний людський фактор.

Система «Прозорий офіс» – система, що поєднує в собі багато можливостей, але для взаємодії, саме з комунальною галуззю є відносно складною, орієнтованою на більш глобальний рівень та не завжди відповідає специфіці саме комунальної галузі [3].

Ситуаційний центр – система, що в цілодобовому режимі відстежує всі аварії на лініях. Але такий центр орієнтований на більш глобальний рівень та не включає безліч речей, які стосуються комунальних підприємств таких як «ЖЕК» або «ОСББ».

Є також досить зручні веб-системи, але вони орієнтовані у переважній більшості на оплату послуг і розподіл бюджетів. Оскільки мають іншу спеціалізацію, тому не були детально розглянуті у даному дослідженні.

Ряд недоліків у схожих системах: непрозорість та відсутність функціоналу, що б спростив та автоматизував процес подачі заявок та отримання інформації. Все вищевказане стало передумовою для розробки системи автоматизації подання заявок до комунальних підприємств.

Мета

Метою дослідження є розробка інформаційної системи автоматизації подання заявок до комунальних підприємств, огляд структури, інструментів реалізації їх переваг та недоліків, як наслідок виділення особливостей організації цієї системи.

Задачі

1. Дослідити особливості системних вимог та архітектури інформаційної системи автоматизації подання заявок до комунальних підприємств.

2. Розробити структурну організацію та програмну реалізацію інформаційної системи автоматизації подання заявок до комунальних підприємств.

Розв'язання задач

Системні вимоги та архітектура інформаційної системи автоматизації подання заявок до комунальних підприємств

В основу розробки запропонованої архітектури системи автоматизації покладено такі особливості:

- система має забезпечувати взаємодію користувачів та комунальних підприємств;
- наявність модуля інформування, що спростить процедуру отримання інформації для користувачів послуг комунальних підприємств;
- введення модуля локалізації, що надасть змогу користувачу бачити актуальну інформацію, що стосується саме його району проживання або локалізації;
- забезпечення захищеного доступу, тому важливим елементом системи є модуль входу та реєстрації на етапі якого мають бути розділені звичайні користувачі та користувачі-адміністратори. Також на цьому етапі має відбуватись певна фільтрація для уникнення відсотку неправдивої інформації, що може лише змарнувати час на перевірку змісту заявки та самого випадку;
- система має забезпечувати можливості розподілення ролей, оскільки система має однаково задовольняти потреби користувачів; робітників, що працюють в бригадах та користувачів-адміністраторів, що контролюють процес робіт та заявок;
- користувачі з різним рівнем підготовки можуть повноцінно користуватись системою, час на навчання є мінімальним.

Ключовим елементом інформаційної системи автоматизації подання заявок до комунальних підприємств є підсистема, що реалізує саме модуль автоматизації процесу подання заявок, можливість створення і подання заявки в один клік, що значно спрощує цю процедуру та заощаджує час. Також цей модуль включає відображення актуальної інформації про них та реалізації взаємодій між користувачами, користувачами та комунальним підприємством.

Найбільш вдалим користувацьким інтерфейсом з точки зору UX-дизайну для відображення прозорого стану заявок, взаємодії та робіт, які проводились чи ні, оцінки проведених робіт та відгуків, на базі яких можливе формування заохочень до продуктивнішої роботи бригад, що виконують роботи, було обрано артефакт «дошка» з таких гнучких сучасних методологій, як Agile, Scrum, Kanban [4], що ідеально задовольняє потреби системи.

Потреби, які задовольняє реалізація модуля взаємодії із заявками у вигляді дошки:

- користувач з легкістю може орієнтуватись у статусах заявок, оскільки межі колонок є чіткими, а статуси інтуїтивно зрозумілими;

- користувач з привілеями може змінювати статус заявок через інтерфейс максимально швидко та інформативно, що буде відображено в реальному часі та не потребує знань бази даних;

- робочі бригади можуть здійснювати моніторинг кількості робіт та їх статуси, що дає змогу структурувати роботу та зробити її більш продуктивною, а також побачити повний «score» роботи.

Ця дошка повинна включати в себе такі речі як: колонки з чітко визначеними статусами, де кожна колонка демонструє актуальний статус задачі, картки задач з інформацією про них, можливість переходу з одного статусу в інший, виділення відповідальної людини шляхом призначення на неї певної картки задачі [5].

Основними функціональними можливостями, що повинні бути реалізовані в цій системі, є:

- можливість віддаленого доступу через мережу інтернет;
- кросплатформність та кросбраузерність;
- наявність можливості кешування даних для забезпечення швидкодії відображення інформації;
- можливість реєстрації та підтримка різних видів користувачів відокремлених певними можливостями;
- можливість створити та подати заявку;
- відображення списку усіх заявок та надання актуальної інформації;
- відображення актуального статусу робіт по заявці;
- підтримка реєстрації з «розподіленням ролей», що буде надавати певній ролі дещо розширені можливості взаємодії з системою;
- можливість зміни статусу заявки для ролі користувач-адміністратор або користувач-робітник;
- можливість залишити відгук про пророблену роботу;
- підтримка оновлень у реальному часі.

Нижче наведений приклад діаграми активності на верхньому рівні деталізації роботи системи:

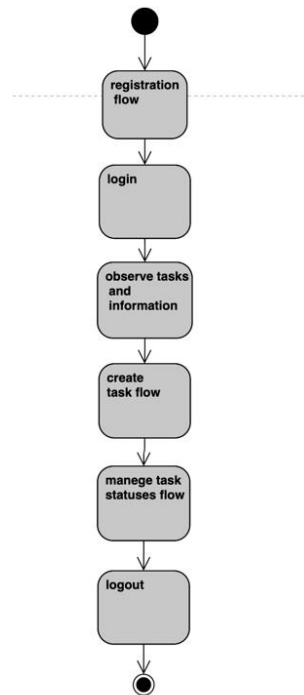


Рисунок 1 – Діаграми активності на верхньому рівні деталізації роботи системи

Структурна організація інформаційної системи автоматизації подання заявок до комунальних підприємств

У системі потрібно реалізувати основні модулі, які б забезпечували виконання основних вимог до системи, а саме:

- взаємодію між об'єктами системи;
- взаємодію між системою та користувачем;
- виконання функцій кожного об'єкту;
- відображення елементів на екрані.

Основними модулями програми є такі:

- модуль реєстрації;
- модуль відображення інформації;
- модуль створення заявок;
- модуль взаємодії з заявкою як сутністю;
- модуль кастомізації особистих даних – профайл.

Користувачами цієї системи є: комунальні підприємства, громадяни (власники приміщень або спостерігачі), ОСББ, виконавці робіт (бригади робітників), органи контролю виконаних робіт та оператори. Діаграму прецедентів системи представлено на рис. 2.

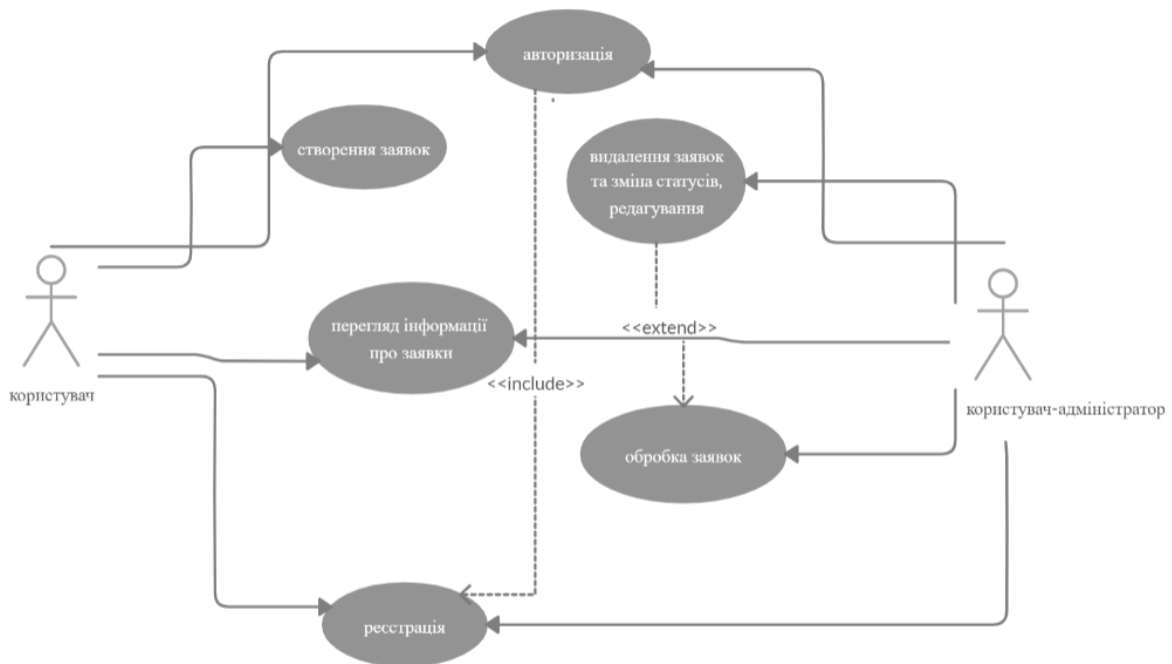


Рисунок 2 – Діаграма прецедентів системи

Деталізовану структурну схему ключового модуля взаємодії з заявками до комунальних підприємств подано на рис. 3.

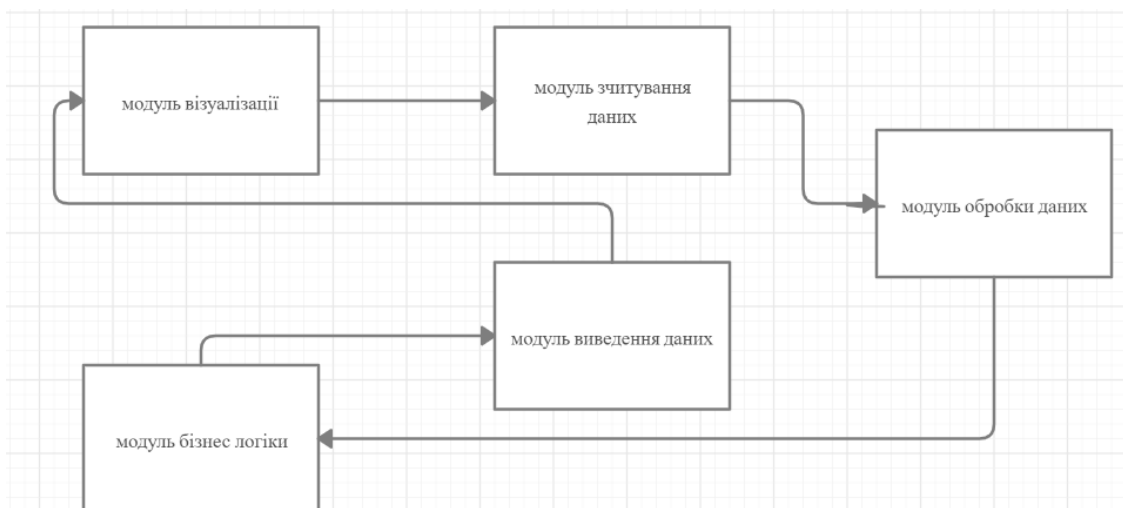


Рисунок 3 – Структурна схема програмного модуля автоматизації подання заявок до комунальних підприємств

Модуль візуалізації – це модуль користувацького інтерфейсу, який є візуальним представленням системи, напряму пов’язаний з модулями зчитування та виведення інформації. Головними функціями

якого є: відображення, можливість введення інформації та надання зв'язків, що будуть в подальшому використовувати інші модулі.

Модуль зчитування даних – модуль, що відповідає за зчитування введених даних, структурує їх, адаптує під певний формат та передає їх на подальшу обробку.

Модуль обробки даних – модуль, що безпосередньо відповідає за обробку отриманих даних та проводить всі відповідні маніпуляції по даним до того, як вмикається апарат бізнес логіки. Так званий про-шарок, що відповідає за з'єднання клієнта і серверу та БД.

Модуль виведення даних – модуль, що відповідає за збір даних та формування їх для подальшого візуального представлення у модулі візуалізації.

Модуль бізнес-логіки – модуль, що вміщає в себе всю бізнес-логіку, а саме функції, що вирішують основні задачі, наприклад: створення заявки, також модуль включає взаємодію між об'єктами системи та їх поведінку, що включає в себе модуль авторизації, модуль створення та подання заявки, модуль допоміжної бізнес-логіки.

Програмна реалізація інформаційної системи автоматизації подання заявок до комунальних підприємств

Критерії вибору програмних засобів:

- можливість створення шаблонів коду;
- наявність об'єктно-орієнтованого проектування;
- сучасний підхід;
- високий рівень оптимізації;
- наявність функціоналу для зберігання локального стану елементів;
- можливість розробки, як фронт так і бек частини сайту з використанням лише однієї мови;
- підтримка бібліотек для роботи з базою даних;
- можливість кешування даних;
- підтримка реалізації кросплатформного веб-додатку з акцентом на користувачів мобільних пристроїв.

Одним із найбільш вдалих варіантів, що надає можливість реалізувати всі вище описані вимоги є JavaScript, але для кращої реалізації доцільно буде використовувати фреймворк. Після проведеного аналізу варіантів реалізації front-end частини відповідно до вимог системи обрано ReactJS [5, 6], який має високий рівень оптимізації та покриває усі потреби.

Для серверної частини єдиним можливим варіантом, що покрив би усі вказані потреби є NodeJS. NodeJS – серверна платформа, що призначена для створення масштабованих розподілених серверних додатків, яка використовує подієво-орієнтовану архітектуру та неблокуючу асинхронну взаємодію [7]. Головна ідея Node.js: використання неблокуючого подієво-орієнтованого введення/виведення, щоб залишатися «легким» та ефективним під час роботи з додатками, що обробляють великі обсяги даних в реальному часі і функціонують на розподілених пристроях.

Щодо реалізації зв'язку клієнт-серверу, за результатами дослідження системи подачі заявок до комунальних підприємств можна стверджувати: при використанні Rest API ми маємо архітектуру з великою кількістю «endpoints», що призводить до величезної кількості запитів у різні місця, проте найбільшим мінусом такої архітектури є складність побудови запитів, а також «over fetching». А це означає, що з виконанням кожного запиту майже завжди тягнеться безліч «непотрібних» даних, що сповільнює роботу системи. Крім того, кешування і офлайн організація є досить складними у реалізації з використанням Rest API. Стосовно GraphQL, використання цієї технології дозволяє вирішити проблему зайвих даних, оскільки має структуру, яка дозволяє в одному запиті отримати усі потрібні запити роблячи запит на один endpoint. Така поведінка забезпечується при організації GraphQL серверу, що має одну точку входу, проте об'єднує множини інформації. Також GraphQL використовується у поєднанні з Apollo як на клієнті так і на сервері, що в свою чергу надає ряд переваг: чітка типізація, кешування, оптимістичний користувацький інтерфейс, що дозволяє оновлювати інтерфейс в процесі відправлення запиту, також таке поєднання зменшує навантаження на мобільному пристрої, що суттєво прискорює роботу системи [8, 9].

Отже, оскільки система подання заявок та їх моніторинг передбачає такі процеси як: створення заявок, корегування даних, відображення інформації про заявки, обробка заявок, коментарі, профайли користувачів, розподіл ролей, а також модуль живого спілкування – це означає, що в системі буде велике навантаження і велика кількість запитів [1]. Алгоритм кешування допомагає вирішити певні проблеми. Apollo кеш передбачає кешування деяких запитів з можливістю оновлення лише тих полів, які змінюються за допомогою функцій запису до кешу, а зчитування даних з кешу допоможе уникнути великої кількості однакових запитів, що в свою чергу зменшує навантаження і додає можливість відображення певної інформації в офлайн режимі. Також така оптимізація позитивно впливає на користувачів з мобільних пристроїв. Функція оптимістичного інтерфейсу добре підходить для реалізації чату, оскільки допомагає відображати повідомлення вже в процесі відправлення.

Отже, реалізація системи потребує впровадження GraphQL на зміну Rest API.

Оскільки програмний модуль буде розроблено у вигляді веб-сайту на технології ReactJS для front-end частини та NodeJS для back-end, то база даних має бути максимально простою та через суміжні технології максимально підходити. Оскільки для запитів було обрано застосування GraphQL, через використання GraphQL база даних має підтримувати зберігання записів в форматі json для більшої зручності.

Основними вимогами до бази даних є:

- здатність до масштабованості;
- можливість розширення;
- швидкість розробки;
- можливість зберігання даних в форматі json;
- гнучкість і легка зміна схем;
- сумісність з GraphQL та Apollo.

Після проведеного аналізу архітектурних потреб системи для реалізації бази даних обрано MongoDB, що має ряд переваг: відсутність схеми, оскільки бізнес-модель ще не до кінця визначена і з великою ймовірністю проект до виходу на ринок зазнає багато змін, в тому числі на рівні організації даних [10]. MongoDB саме те, що потрібно, адже на відміну від SQL в MongoDB немає необхідності створювати таблиці, змінювати їх схеми, створювати міграції, перейматися типами даних, багатий функціонал агрегації. Дані можна отримувати за допомогою MongoDB набагато простіше і зручніше, а саме через map-reduce: групування за складними умовами, переформатування документів миттєво, отримання випадкових документів, сортування.

Також дані можна або перемістити, тому що відсутність схеми це і має на увазі, або просто продублювати в потрібне місце. Тобто, можливість мати одне і те ж поле з одними і тими ж даними, але в різних колекціях. Або два поля в одній колекції, а плюс до них ще одне поле, яке є композицією перших двох [11].

Висновки

Встановлено, що розробка інформаційної системи автоматизації подання заявок до комунальних підприємств є доцільною та актуальною задачею.

1. Сформовано основні потреби та особливості реалізації системи у вигляді Web-технології, що значно спрощує процес взаємодії між комунальним підприємством та особою, яку воно обслуговує, що в свою чергу, значно заощаджує час, структурує інформацію та саму роботу комунальних бригад. Такий підхід надає змогу усім користувачам мати швидкий доступ до платформи за наявності мережі з мобільних пристроїв, планшетів, ноутбуків, персональних комп'ютерів тощо.

2. Здійснено аналіз вимог до інструментальних засобів, яких потребує реалізація системи, та як наслідок реалізовано її за допомогою мови програмування JS, на базі двох бібліотек: ReactJs, NodeJS для клієнтської та серверної частини відповідно. Для запитів обрано мову запитів GraphQL, що забезпечує вищий рівень оптимізації запитів, ніж RestAPI та спроектовано базу даних MongoDB.

Як результат, розроблено інформаційну систему автоматизації подання заявок до комунальних підприємств.

В подальших дослідженнях планується дослідити доцільність застосування методів штучного інтелекту, для подолання «людського фактору» в обробці задач для зменшення часу витраченого оператором на роботу, а також для збільшення швидкодії роботи самої системи. Також планується реалізувати аналіз потреб користувачів та локалізацію інформаційної системи під певний регіон.

Список літератури

- [1] І. В. Безсмертна, Л. В. Крилик, "Перспективи впровадження автоматизованої системи подання заявок та їх відстеження," *Матеріали XLIX науково-технічної конференції факультету інформаційних технологій та комп'ютерної інженерії*, с. 919 – 920, 2020. [Електронний ресурс]. Режим доступу: https://conferences.vntu.edu.ua/public/files/1/vntu_2020_netpub.pdf. Дата звернення: Черв. 25, 2021.
- [2] Закон про житлово-комунальні послуги – Законодавство України, 2021. [Електронний ресурс]. Режим доступу: <https://zakon.rada.gov.ua/laws/show/1875-15>. Дата звернення: Черв. 25, 2021.
- [3] Прозорий офіс, 2021. [Електронний ресурс]. Режим доступу: <https://transparent.vmr.gov.ua/default.aspx>. Дата звернення: Черв. 25, 2021.
- [4] Kanban, 2021. [Online]. Available: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board>. Accessed on: June 25, 2021.
- [5] ReactJs, 2021. [Online]. Available: <https://ru.reactjs.org/>. Accessed on: June 25, 2021.
- [6] Порівняльна характеристика фреймворків для створення користувацького інтерфейсу, 2021. [Електронний ресурс]. Режим доступу: <https://mkdev.me/posts/sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019>. Дата звернення: Черв. 25, 2021.
- [7] NodeJs, 2021. [Online]. Available: <https://nodejs.org/ru/docs/>. Accessed on: June 25, 2021.
- [8] GraphQL, 2021. [Online]. Available: <https://graphql.org/learn/>. Accessed on: June 25, 2021.

- [9] Порівняння Rest API та GraphQL, 2021. [Електронний ресурс]. Режим доступу: <https://www.howtographql.com/basics/1-graphql-is-the-better-rest/>. Дата звернення: Черв. 25, 2021.
- [10] MongoDB, 2021. [Online]. Available: <https://www.mongodb.com/>. Accessed on: June 25, 2021.
- [11] Переваги MongoDB, 2021. [Електронний ресурс]. Режим доступу: <https://echo.lviv.ua/dev/9693>. Дата звернення: Черв. 25, 2021.

Стаття надійшла: 18.08.2021.

References

- [1] V. Bezsmertna, L. V. Krylyk, "Perspektyvy vprovadzhennia avtomatyzovanoi systemy podannia zaiavok ta yikh vidstezhennia," *Materialy XLIX naukovo-tekhnichnoi konferentsii fakultetu informatsiinykh tekhnolohii ta kompiuternoї inzhenerii*, s. 919 – 920, 2020. [Elektronnyi resurs]. Rezhym dostupu: https://conferences.vntu.edu.ua/public/files/1/vntu_2020_netpub.pdf. Data zvernennia: Cherv. 25, 2021. – [in Ukrainian].
- [2] Law on housing and communal services – Legislation of Ukraine, 2021. [Online]. Available: <https://zakon.rada.gov.ua/laws/show/1875-15>. Accessed on: June. 25, 2021. – [in Ukrainian].
- [3] Transparent office, 2021. [Online]. Available: <https://transparent.vmr.gov.ua/default.aspx>. Accessed on: June. 25, 2021. – [in Ukrainian].
- [4] Kanban, 2021. [Online]. Available: <https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board>. Accessed on: June 25, 2021.
- [5] ReactJs, 2021. [Online]. Available: <https://ru.reactjs.org/>. Accessed on: June 25, 2021.
- [6] Comparative characteristics of frameworks for creating a user interface, 2021. [Online]. Available: <https://mkdev.me/posts/sravnenie-javascript-freymvorkov-vue-js-react-i-angular-2019>. Accessed on: June. 25, 2021. – [in Ukrainian].
- [7] NodetJs, 2021. [Online]. Available: <https://nodejs.org/ru/docs/>. Accessed on: June 25, 2021.
- [8] GraphQL, 2021. [Online]. Available: <https://graphql.org/learn/>. Accessed on: June 25, 2021.
- [9] Comparison of Rest API and GraphQL, 2021. [Online]. Available: <https://www.howtographql.com/basics/1-graphql-is-the-better-rest/>. Accessed on: June. 25, 2021. – [in Ukrainian].
- [10] MongoDB, 2021. [Online]. Available: <https://www.mongodb.com/>. Accessed on: June 25, 2021.
- [11] Advantages of MongoDB, 2021. [Online]. Available: <https://echo.lviv.ua/dev/9693>. Accessed on: June. 25, 2021. – [in Ukrainian].

Відомості про авторів

Крилик Людмила Вікторівна – кандидат технічних наук, доцент, доцент кафедри комп'ютерних наук.

Яровий Андрій Анатолійович – доктор технічних наук, професор, завідувач кафедри комп'ютерних наук.

Безсмертна Інна Василівна – студентка групи 2КН-20м, факультет інформаційних технологій та комп'ютерної інженерії.

Л. В. Крылик, А. А. Яровой, И. В. Бессмертная

ОСОБЕННОСТИ РЕАЛИЗАЦИИ ИНФОРМАЦИОННОЙ СИСТЕМЫ АВТОМАТИЗАЦИИ ПОДАЧИ ЗАЯВОК В КОММУНАЛЬНЫЕ ПРЕДПРИЯТИЯ

Винницкий национальный технический университет, Винница

L. Krylik, A. Yarovyi, I. Bezsmertna

PECULIARITIES OF IMPLEMENTATION OF INFORMATION SYSTEM OF AUTOMATION SUBMISSION OF APPLICATIONS TO UTILITY ENTERPRISES

Vinnitsia National Technical University, Vinnitsia