

**О. В. Кузьмін, Т. Я. Яцишин**

Національний університет "Львівська політехніка", м. Львів, Україна

ПРОЕКТУВАННЯ ТА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНО-ПОШУКОВОЇ СИСТЕМИ "ВИПУСКНИК"

Одним з основних показників рейтингу українських університетів є якість підготовки майбутніх випускників. Якість підготовки випускника може бути відображена можливістю його працевлаштування. Відстежування кар'єрного росту випускника неможливо без створення централізованого сховища інформації про цей процес і доступу до нього випускника, який міг би заносити інформацію про своє працевлаштування. Основною метою створення інформаційно-пошукової системи "Випускник" є розроблення програмного рішення для кафедр українських університетів, яке зберігало би інформацію про випускників. Для створення програмного рішення та досягнення цілей розроблено веб орієнтовану систему, яка працює за архітектурним шаблоном клієнт-сервер з клієнтською частиною в режимі односторінкового застосування, розгорнутого на сервері Windows Web Server 2016. Система містить: клієнт-серверний застосунок для використання персоналом кафедри та випускниками; функціонал наповнення бази даних (БД) новими випускниками кафедри, та оновлення вже існуючих записів про випускників; функціонал закритого пошуку випускників з інформацією про навчання; статистичну інформацію про випускників на підставі обраних критеріїв; функціонал для випускників для оновлення інформації про професійну кар'єру та досягнення. Для цього програмного рішення спроектовано та розроблено реляційну базу даних MS SQL Server. Систему було розроблено за допомогою мови програмування C#, з використанням технології ASP.NET Core, а дружні користувачські інтерфейси – за допомогою HTML, CSS 3, та мови JavaScript на підставі бібліотеки React та розширенням JSX, з елементами бібліотеки Vootstrap. Така багатофункціональна система матиме велику популярність завдяки зручному користувачському інтерфейсу, високу швидкість опрацювання користувачських запитів, сповіщення користувача про події, що пов'язані з його обліковим записом. Планується удосконалювати систему та додати можливість інтеграції з іншими системами процесу навчання студентів у вищих навчальних закладах.

Ключові слова: архітектура; діаграма; інтерфейс; модуль.

Вступ

Одним з основних показників рейтингу українських університетів є якість підготовки майбутніх випускників. Якість підготовки випускників напряму пов'язана з їх подальшою конкурентоспроможністю на ринку праці. Щороку велика кількість студентів завершує своє навчання в університетах. Основним показником їх конкурентоспроможності є працевлаштування. На жаль, не існує єдиних джерел даних про випускників та їх подальші професійні досягнення. Тому для відстежування подальшого кар'єрного росту випускників пропонують інформаційно-пошукову систему, яка дасть змогу проводити аналіз стану випускників, підтримувати з ними зв'язок, відстежувати динаміку працевлаштування, робити статистичний аналіз.

Основною метою роботи є автоматизація процесів обліку випускників кафедр вищих навчальних закладів (ВНЗ). Об'єктом дослідження виступає інформаційно-пошукова система "Випускник", яка зберігала б інформацію про випускників і охоплювала б такі основні компоненти:

- захищену та нормалізовану базу даних випускників кафедри з інформацією про навчання, дипломні та магістерські роботи, їхні місця працевлаштування, кар'єрний ріст;
- клієнт-серверний застосунок для використання персоналом кафедри та випускниками;
- функціонал наповнення бази даних (БД) новими випускниками кафедри та оновлення вже існуючих записів про випускників;

- функціонал закритого пошуку випускників з інформацією про навчання;
- статистичну інформацію про випускників на підставі обраних критеріїв;
- функціонал для випускників з метою оновлення інформації про професійну кар'єру та досягнення.

Предметом дослідження є автоматизоване збереження даних про випускників, обрахунок статистичних даних та створення звітів про випускників.

Аналіз останніх досліджень та публікацій. Існують роботи, в яких висвітлено питання щодо побудови інформаційних систем управління ВНЗ. В роботі [6] наведено огляд автоматизованих систем управління вищим навчальним закладом (АСУ ВНЗ), поданих на українському ринку програмного забезпечення. Наведені результати порівняння переваг та недоліків АСУ ВНЗ, а також рекомендації щодо вибору найбільш ефективної системи автоматизації всіх сфер діяльності освітньої установи. У роботі [1] розглянуто заходи вдосконалення інформаційного забезпечення вищих навчальних закладів України, проаналізовано недоліки вже існуючих інформаційних систем та переваг впровадження нових інформаційних систем, які допоможуть якомога краще вирішити проблему управління даними. У роботі [8] висвітлено питання щодо розробки інформаційної системи обліку та моніторингу працевлаштування випускників ВНЗ. Розглянуто методологію проектування та розробки інформаційної системи на підставі уніфікованої моделі.

Огляд літературних джерел показав, що існуючі системи управління ВНЗ не містять ресурс, який би відповідав за працевлаштування випусників. У роботі [8] розглянуто це питання в загальних рисах і не наведено практичної реалізації та результатів експлуатації. Тому розробка інформаційної системи, яка б відстежувала кар'єрний ріст випусників ВНЗ, є актуальною задачею.

Викладення основного матеріалу

Концептуальна модель. Для створення програмного рішення та досягнення цілей необхідно створити веб-орієнтовану систему, яка буде працювати за архітектурним шаблоном клієнт-сервер з клієнтською частиною в режимі односторінкового застосунку і буде розташована на сервері Windows Web Server 2016 (рис. 1) [6].

Підвищення продуктивності можна досягти, якщо кожен рівень системи буде мати свою функціональність. Рівень Controller відповідає за обробку користувачських запитів та повернення результатів його оброблення. Рівень сервісу міститиме бізнес-логіку маніпуляцій з даними і буде викликати процедури від рівня доступу до даних DAL (Data Access Layer), які за допомогою технології ADO.Net будуть спілкуватись з джерелом даних – БД MS SQL Server – за допомогою збережених процедур та їхніх параметрів. Також для прямої вибірки даних на DAL рівні було розроблено доступ до даних за допомогою ORM фреймворку Entity Framework Core [13].

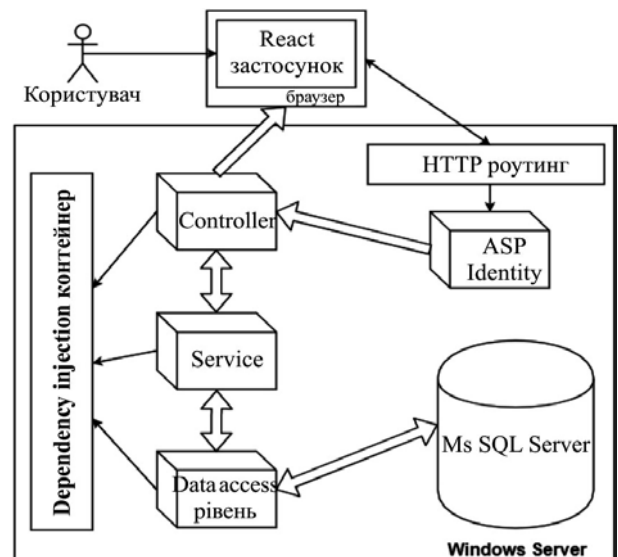


Рис. 1. Архітектура інформаційно-пошукової системи "Випусник"

Розроблення та відображення алгоритмів роботи компонент системи. Діаграма варіантів використання (рис. 2) відображає можливі функції системи, користувачів системи та зв'язки між ними, тобто який користувач може виконувати які функції. Також у цій діаграмі відображаються умови виконання певної послідовності функціональностей (прецедентів), тобто який прецедент повинен виконуватись перед іншим [7].

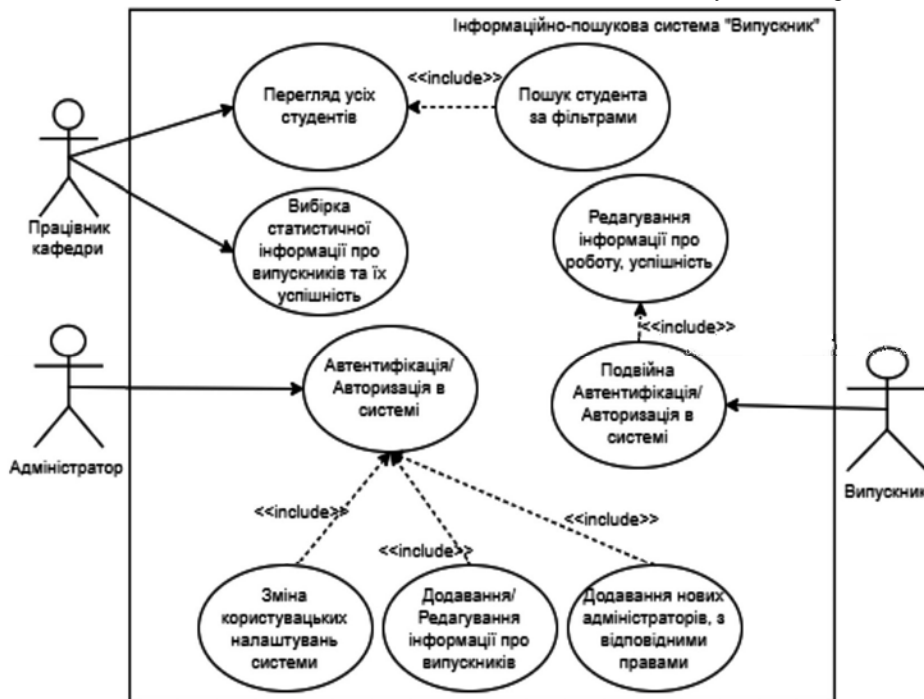


Рис. 2. Діаграма варіантів використання системи

Система передбачає три типи користувачів: адміністратор, випусник, працівник кафедри [12], [9].

Для адміністратора передбачено таку функціональність [3], [11]:

- зміна користувачських налаштувань системи;
- додавання/редагування/видалення інформації про випусників кафедри;
- додавання нових адміністраторів, з відповідним правами;
- автентифікація та авторизація в системі;
- додавання нових викладачів;

- автоматизоване додавання студентів за допомогою спеціалізованих скриптів виконання.

Для випусника системи передбачено таку функціональність [3], [11]:

- подвійна автентифікація та авторизація в системі та підтвердження інформації про особистість;
- редагування персональної інформації про успішність та нові місця роботи.

Для працівника кафедри передбачено такі основні функції [1], [9]:

- пошук студента за фільтрами;
- перегляд інформації про студента;
- вибір статистичної інформації про випускників та їхню успішність;
- перегляд статистики за фільтрами;
- перегляд інформаційних форм про працюючих випускників з інформацією про місця роботи;
- перегляд інформаційної форми про випускників, їхню працевлаштованість та форму навчання, згруповану за роками випусків студентів;
- перегляд інформаційної форми про почесні звання, набуті випускником за останні 10 років;
- друк інформаційних форм про випускників.

Однією з найбільш зручних мов для моделювання різноманітних бізнес-процесів є діаграма IDEF0, на якій система, яка проектується, відображається як сукупність функцій, джерел та робіт, що взаємодіють між собою. Така чисто функціональна орієнтація є принциповою – функції системи аналізуються незалежно від об'єктів, якими вони оперують. Це дає змогу чіткіше змодельовати логіку і взаємодію процесів [2].

На рис. 3 наведено IDEF0-діаграму 1-го рівня, на якій видно, що на вхід до всієї діяльності, що здійснює система, надходять такі запити на: пошук випускників;

додавання випускників; отримання статистики та різні форми.

Усі запити повинні відповідати правилам системи та її обмеженням. Запити, які відправляють користувачі, комунікують з базою даних та зовнішніми пристроями (принтер). На виході роботи системи отримуємо результати виконання користувацьких запитів у вигляді візуального подання на веб-сторінці, збереженого файлу або роздрукованого документа принтером [2], [3], [6].

Внаслідок декомпозиції системи отримуємо IDEF0 діаграму 2-го рівня, в якій зображено потоки запитів, правил, зовнішніх комунікацій та результатів оброблення функцій системи (рис. 4). Діаграма містить не узагальнені зв'язки потоків даних та правил, а вже направлені потоки на виконання блоків певних функцій. За подальшої декомпозиції системи, відображається докладніша інформація про виконання функціональних блоків, аж до елементарних функціональних одиниць [8], [9], [12].

Одним з основних алгоритмів, який повинен бути присутній в системі, – це алгоритм швидкого пошуку студентів за змінною кількістю параметрів. Цей алгоритм базується на підставі предикатів.

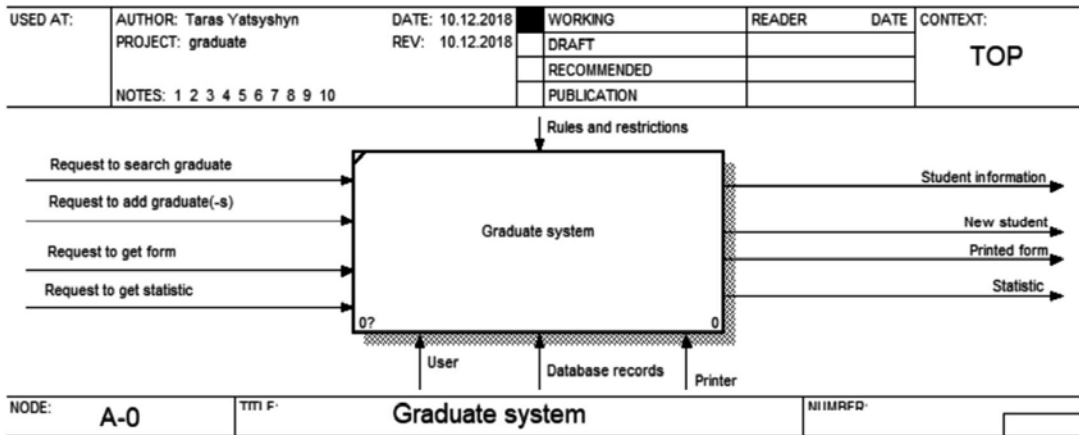


Рис. 3. Контекстна діаграма системи

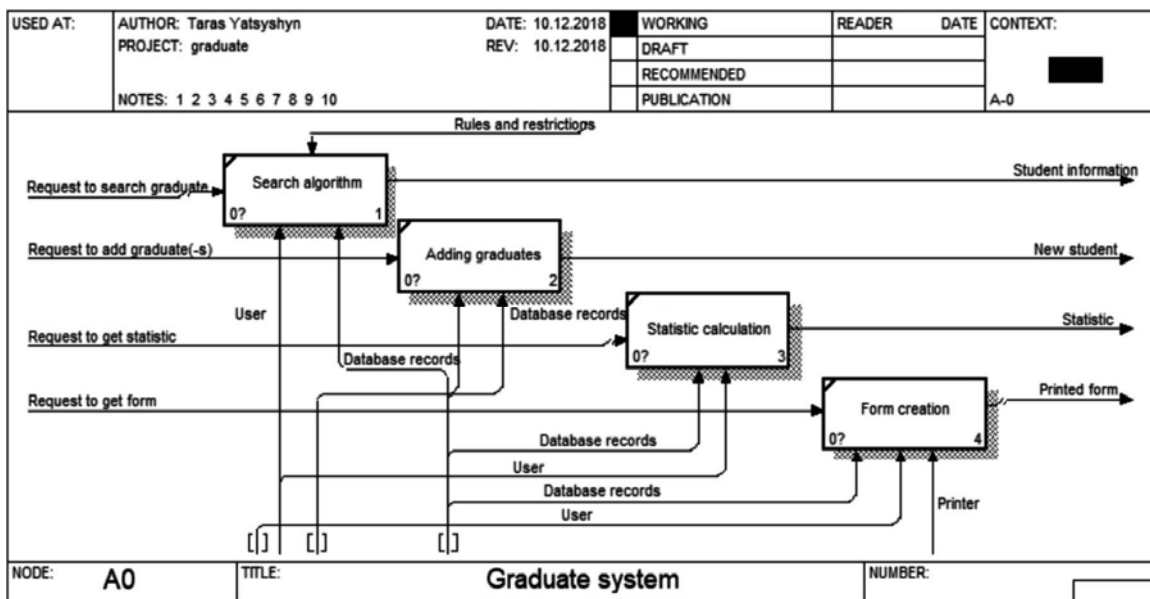


Рис. 4. Декомпозиція інформаційної системи першого рівня

З веб-частини застосунку приходять об'єкт, який містить масив всіх можливих умов пошуку студента. Після цього в масиві, що прийшов з веб-частини, відбу-

вається перевірка чи присутній критерій пошуку. Якщо такий критерій присутній, то цей критерій додається в об'єкт предикату. Під час додавання нового предикату

відбувається динамічне формування критеріїв вибірки даних з БД. Це відбувається у такий спосіб:

- кожен критерій пошуку, що прийшов з веб-частини, має свій еквівалент поля в БД;
- на підставі критерію обирається оператор;
- разом із значенням критерію формується стрічка (string).

Після формування всіх предикатів вони об'єднуються у спільний блок пошуку та загальним запитом, який є обов'язковим блоком вибірки. Після цього сформований запит передається в сервіс доступу до даних. Відбувається виконання сформованого запиту, оброблення його результатів у список студентів та повертається на користувацьку частину додатку.

База даних. Спроектвана реляційна база даних складається з 8 таблиць (рис. 5):

- **AspNetUsers** – ця таблиця містить інформацію про користувачів системи, яких безпосередньо реєструють в системі інші користувачі, облікові записи яких вже присутні в цій таблиці. Унікальний ідентифікатор цієї таблиці (первинний ключ) має тип GUID, що є 128-бітним

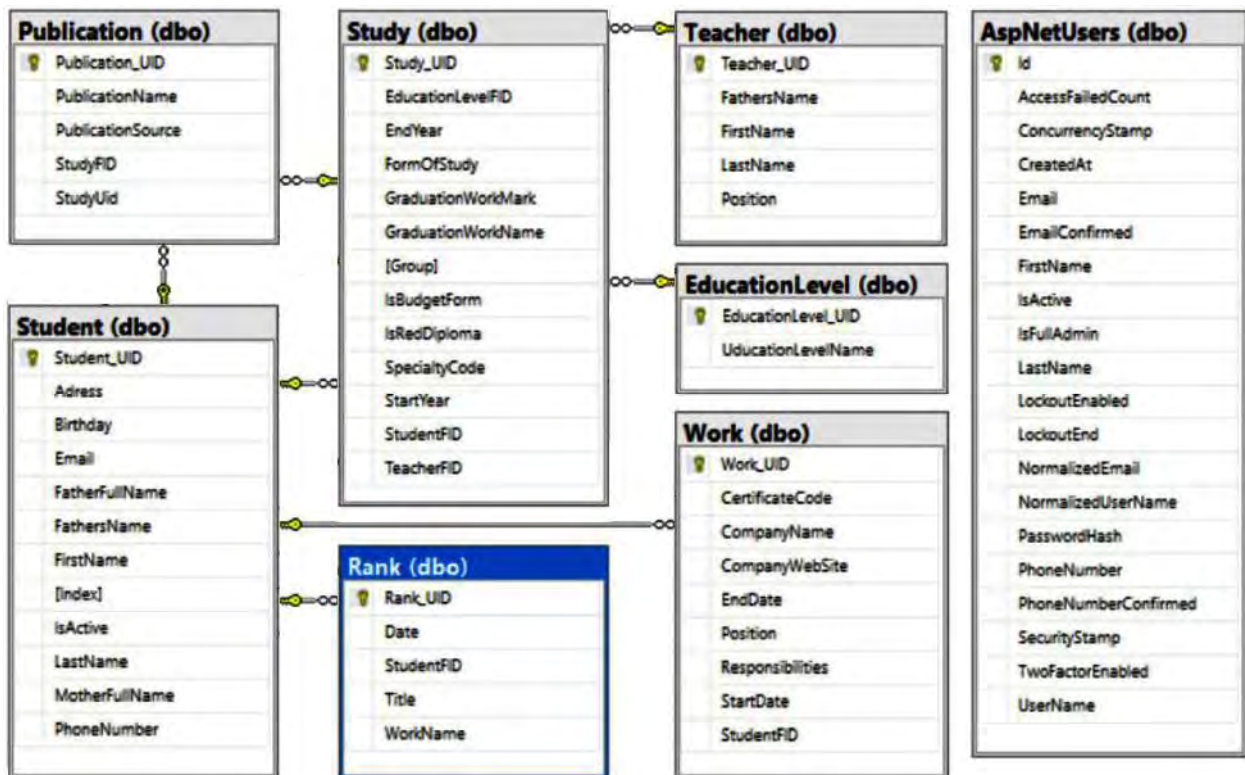


Рис. 5. Схема бази даних

Усі таблиці бази даних, за винятком AspNetUsers, містять кластеризовані індекси для швидкого виконання інформаційних запитів. Оскільки код, який виконується ORM EntityFramework Core, транлюється в код мови T-SQL, кластеризовані індекси враховуються під час оптимізації виконання запиту.

Розроблення програмного рішення. За своєю структурою програмне рішення є монолітним проектом, оскільки клієнтський застосунок та серверна логіка знаходяться в одному проекті та його розгортання здійснюватиметься одним потоком.

Програмне рішення складається з низки папок, що містять логічно згруповані компоненти (рис. 6):

- **Wwwroot** – папка, яка містить готові до виконання статичні файли, такі як main.js, vendor.js та vendor.css, які

ідентифікатором, який відображається 32 цифрами у шістнадцятковій системі числення;

- **Study** – ця таблиця репрезентує інформацію про навчання студента за різними освітніми рівнями, інформацію про вступ та випуск за освітнім рівнем, інформацію про успішність студента під час навчання, інформацію про дипломну кваліфікаційну роботу;
- **Student** – у цій таблиці зберігається особиста інформація про випускника, який навчався на кафедрі;
- **Publication** – ця таблиця зберігає інформацію про наукові публікації студента під час навчання та джерела їх публікації;
- **Teacher** – у цій таблиці зберігається інформація про викладачів кафедри та їхні посади;
- **Rank** – ця таблиця репрезентує почесні звання випускника, якщо такі призначались;
- **Work** – ця таблиця містить інформацію про місця роботи колишнього студента, а саме: інформацію про компанію чи організацію, посаду, початок роботи та її завершення;
- **EducationLevel** – ця таблиця зберігає інформацію про освітні рівні, які наявні в системі.

створюються під час процесу компілювання програмного рішення. Усі скомпільовані файли автоматично мінімізуються, для швидшого завантаження на клієнтський браузер;

- **ClientApp** – містить ресурси клієнтського застосунку, такі як файли з розширенням *.jsx, написані на мові React, що містять розмітки сторінок із програмною логікою та маршрутизацією, та файли з розширенням *.css для відображення сторінок;
- **Controllers** – зберігає усі контролери, які є вхідними точками виконання серверної логіки, що викликаються http (-s) запитами;
- **Infrastructure** – зберігає інфраструктурні класи;
- **Interfaces** – містить усі інтерфейси системи;
- **Migrations** – зберігає класи міграції, які створюються під час зміни структури БД;

- Models – зберігає моделі таблиць БД;
- Services – містить класи, які зберігають відокремлені логічні одиниці оброблення даних, та доступ до даних у БД;
- ViewModels – зберігає класи моделей, які повертаються до користувачів;
- Views – зберігає завантажувальні сторінки системи.

Окремо необхідно виділити такі файли:

- Appsettings.json – цей файл зберігає дані налаштувань;
- Startup.cs – це файл налаштувань усієї системи від даних про БД до реєстрації сервісів у контейнерах;
- Program.cs – це вхідна точка виконання системи;
- Webpack.config.js – файл налаштувань компілятора клієнтських файлів.

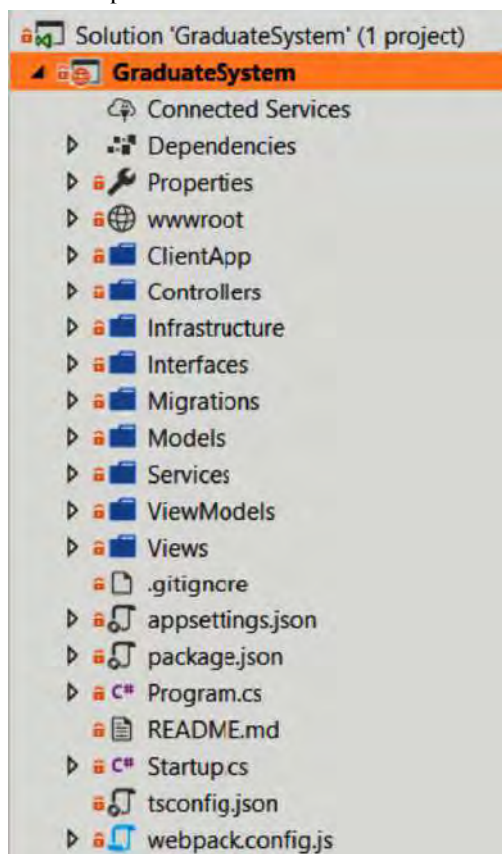


Рис. 6. Структура програмного рішення

Одним із важливих компонент системи є графічний інтерфейс, який є множиною засобів для оброблення та відображення інформації під час роботи зі системою користувача. Успішно спроектований графічний інтерфейс повинен задовольняти такі критерії: естетичність, ясність, продуктивність, виразність, легкість у навчанні, відповідність, послідовність, запобігання помилок. Під час розроблення інформаційно-пошукової системи "Випускник" було дотримано і впроваджено в дію ці критерії, що стало наслідком успішно розроблених графічних інтерфейсів системи. Графічний інтерфейс охоплює такі сторінки, як: вхід у систему, пошук випускника (за прізвищем, ім'ям, по-батькові за певні роки), додавання випускника (містить не тільки особисту інформацію, а й теми виконаних випускних робіт, працевлаштованість), формування статистичної інформації про випускників (за освітнім рівнем, оцінкою випускної роботи, формою навчання), функції налаштування системи, права викладачів та адміністратора, та форми видачі звітної інформації для підрозділів університетів.

Висновки

Наукова новизна роботи полягає у створенні ресурсу клієнт-серверної архітектури за допомогою технології ASP.NET Core, яка дає змогу створювати серверні частини для мобільних додатків, виконувати розгортання у хмарі або локальному середовищі з можливістю віддаленого доступу до ресурсу через мережу Internet. Розглянуто задачі обліку та аналізу інформації про випускників кафедри. Описано архітектуру та програмну реалізацію виконання розглянутих задач – веб-сервіс "Випускник". Для цього програмного рішення було спроектовано та розроблено реляційну базу даних MS SQL Server.

Систему було розроблено за допомогою мови програмування C#, з використанням технології ASP.NET Core, а дружні користувацькі інтерфейси – за допомогою HTML, CSS 3, та мови JavaScript на підставі бібліотеки React та розширенням JSX, з елементами бібліотеки Bootstrap [3], [9], [12], [11]. Така багатofункціональна система матиме велику популярність завдяки зручному користувацькому інтерфейсу, високу швидкість опрацювання користувацьких запитів, сповіщення користувача про події, що пов'язані з його обліковим записом.

Плануємо удосконалювати систему та додати можливість інтеграції з іншими системами процесу навчання студентів у вищих навчальних закладах.

References

- [1] Bilyk, L. O., & Pashenko, G. V. (2013). *Improvement of information support of management of higher educational institutions of Ukraine*. Retrieved from: [jvestnik-sss.donnu.edu.ua >article> download](http://jvestnik-sss.donnu.edu.ua/article/download).
- [2] Fowler, M. (2006). *The basics*. (3rd ed.). Moscow: Symbol-Plus, 192 p. [In Russian].
- [3] Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2001). *Techniques for object-oriented design. Design patterns*. St. Petersburg: Peter, 368 p. [In Russian].
- [4] Grytsyuk, M. Yu., & Hrytsiuk, Yu. I. (2018). Nature and Sustainable Development of Tourism in the Carpathian Region Ukraine. *Scientific Bulletin of UNFU*, 28(2), 99–110. <https://doi.org/10.15421/40280219>.
- [5] Hrytsiuk, Yu. I., & Dalyavskyy, V. S. (2018). Using Petal Diagram for Visualizing the Results of Expert Evaluation of Software Quality. *Scientific Bulletin of UNFU*, 28(9), 95–104. <https://doi.org/10.15421/40280919>
- [6] Kuzma, K. T. (2008). *Analysis of automated systems of higher education institution management*. Dnipropetrovsk National University. O. Gonchara, 134–144.
- [7] Kuzmin, O., & Yatsyshyn, T. (2018). Architecture of the graduate search engine. *Proceedings 23th International Modeling School of AMSE-UAPL, Shatsk-Ukraine*, July 2-6, 2018. Technical News, 1(47), 2(48), 101–103. Lviv. [In Ukrainian].
- [8] Osadchy, V., Osadchy, K., & Burdey, V. (2015). Experience in developing an information system for accounting and monitoring employment of university graduates. (Ser. Pedagogy). *Scientific Bulletin of MSRU B. Khmelnytskyi*, 1, 264–268. Retrieved from: http://nbuv.gov.ua/UJRN/Nvmdpu_2015_1_46
- [9] Tidwell, J. (2011). *Designing Interfaces: Patterns for Effective Interaction Design* Pages. (2nd ed.). O'Reilly Media, January 9, 2011. (578 p.).
- [10] Hrytsiuk, Yu. I., & Andrushchakevych, O. T. (2018). Means for determining software quality by metric analysis methods. *Scientific Bulletin of UNFU*, 28(6), 159–171. <https://doi.org/10.15421/40280631>

- [11] Wiggers, C. (2003). *Development of software requirements*. The Russian Editorial Board, 487 p. [In Russian].
- [12] Troelsen, A. (2013). *C# 5.0 programming language and .NET 4.5 platform*. (6th ed.). Moscow: LLC "ID. Williams", 1312 p. (Trans. from English). [In Russian].
- [13] Core 2.2. (2019). Download .NET Core 2.2. Retrieved from: <https://dotnet.microsoft.com/download/dotnet-core/2.2>

O. V. Kuzmin, T. Ya. Yatsyshyn

National University "Lviv Polytechnic", Lviv, Ukraine

DESIGN AND IMPLEMENTATION OF THE INFORMATION SEARCH SYSTEM "GRADUATE"

One of the main indicators of the rating of Ukrainian universities is the quality of preparation of future graduates. The quality of the graduate's training may be reflected with a possibility of his employment. It is impossible to track career growth of graduates without creation of centralized informational storage to cover this process as well as possibility for graduate to input information about his employment. Main aim of creation of information search system graduate was to provide different departments of Ukrainian universities with software solution which could store graduates related information. To create the software solution and achieve the goals, a web-based system was developed that runs on a client-server architectural template with client side created as single page application, application deployed on Windows Web Server 2016. The system includes: client-server application for use by department staff and graduates; functionality for filling the database (database) with new graduates of the department, and updating of already existing records about graduates; functionality exposed for departments staff with search for graduates with information about their studying in university; statistical information on graduates based on selected criteria; functionality for graduates to update career information and achievements. For given software solution relational database was designed and exposed with the help of MS SQL Server capabilities. The system was developed with the help of C # programming language, using ASP.NET Core technology in particular for backend, for front-end – user-friendly interfaces using HTML, CSS 3, and JavaScript based on React library and JSX extension, with elements of the Bootstrap library.

One of the important components of the system is the graphical interface, which is a multitude of tools for processing and displaying information for user working with the system. Successfully designed GUI must meet the following criteria: aesthetics, clarity, performance, expressiveness, ease of learning, consistency, consistency, error prevention. During the creation of information search system, "Graduate" these criteria was adhered and implemented which is the result of successfully developed graphical interfaces of the system. Graphical interface includes such pages as: login, search for a graduate (by name, patronymic for certain years), adding a graduate (includes not only personal information, but also topics of completed graduation work, employment), forming of statistical information on graduates (by educational level, assessment of graduation work, form of training), system setup functions, rights of teachers and administrators, and forms for reporting information for university departments. Such multifunctional system is ought to have big popularity due to convenient user interface, high speed of processing of user requests, informing user about event that are associated with his account. It is planned to improve the system and to add the possibility of integration with other systems of the student learning process in higher educational institutions.

Keywords: architecture; diagram; interface; module.

Інформація про авторів:

Кузьмін Олександр Васильович, канд. техн. наук, доцент, кафедра автоматизованих систем управління. **E-mail:** kuzminavua@yahoo.com

Яцишин Тарас Ярославович, магістр, кафедра автоматизованих систем управління. **E-mail:** yatarasis@gmail.com

Цитування за ДСТУ: Кузьмін О. В., Яцишин Т. Я. Проектування та реалізація інформаційно-пошукової системи "Випускник". *Український журнал інформаційних технологій*. 2019, т. 1, № 1. С. 46–51.

Citation APA: Kuzmin, O. V., & Yatsyshyn, T. Y. (2019). Design and implementation of the information search system "Graduate". *Ukrainian Journal of Information Technology*, 1(1), 46–51. <https://doi.org/10.23939/ujit2019.01.046>