

МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ТА ОБЧИСЛЮВАЛЬНІ МЕТОДИ УДК 004.624

В. В. Войтко, С. В. Бевз, С. М. Бурбело, П. В. Ставицький

МОДЕЛІ СИСТЕМИ АНАЛІЗУ ТА РОЗПІЗНАВАННЯ МУЗИЧНИХ КОМПОЗИЦІЙ

Вінницький національний технічний університет, Вінниця

Анотація. У статті розглядаються моделі системи розпізнавання музичних композицій у системі синтезу та аналізу музичних звуків, спрямовані на підвищення ідентифікаційних можливостей автоматизованої системи. Модуль розпізнавання музичних композицій орієнтований на серверну частину системи, яка, незалежно від клієнта, містить базу даних з відбитками музичних композицій. За допомогою алгоритмів розпізнавання мелодій за заданим аргументом у вигляді відбитку сервер повертає список музичних композицій, які найбільше задовольняють умовам пошуку. Клієнтська частина взаємодіє з серверною за допомогою розробленого прикладного програмного інтерфейсу, який, крім відомого функціоналу підходу до архітектури мережесхемних протоколів REST, що базується на протоколі HTTP, де клієнт використовує запити лише в форматі, визначеному специфікацією серверної частини, також передбачає реалізацію можливостей підходу до архітектури мережесхемної взаємодії з використанням мови запитів GraphQL, що дозволяє будувати параметри запиту зі сторони клієнта. Локальна база даних містить набір відбитків та метаданих про музичні композиції для прискорення процесу розпізнавання, оскільки дозволяє покрити більшість сценаріїв використання додатку з найпопулярнішими музичними композиціями з можливістю швидкого повернення результату після локального співставлення даних без необхідності затримок клієнт-серверної взаємодії. Модуль синхронізації бази даних відповідає за своєчасне оновлення локальної бази новими відбитками з серверної частини та за загальну синхронізацію клієнтської і серверної частин системи. Планувальник синхронізації забезпечує формування розкладу синхронізації локальної та серверної баз даних, а також реалізує стратегії оптимізації використання акумулятора та забезпечує роботу з низьким рівнем інтернет-з'єднання. Розглянуто особливості зберігання бази відбитків композицій та стратегії роботи з пристроями на базі мобільних платформ, зокрема, під операційну систему Android з використанням режиму Doze, який забороняє фонову роботу пристрою у стані спокою, окрім коротких проміжків часу, так званих вікон підтримки, які визначаються операційною системою в процесі роботи та дозволяють виконувати короточасні фонові операції. Проведено оптимізацію процесу використання енергії акумулятора мобільного пристрою при синхронізації метаданих музичних композицій між клієнтською та серверною складовими системи.

Ключові слова: мобільний додаток, розпізнавання музики, клієнт-серверна система, зберігання даних.

Аннотация. В статье рассматриваются модели системы распознавания музыкальных композиций в системе синтеза и анализа музыкальных звуков, направленные на повышение идентификационных возможностей автоматизированной системы. Модуль распознавания музыкальных композиций ориентирован на серверную часть системы, которая, независимо от клиента, содержит базу данных с отпечатками музыкальных композиций. С помощью алгоритмов распознавания мелодий с заданным аргументом в виде отпечатка сервер возвращает список музыкальных композиций, наиболее удовлетворяющих условиям поиска. Клиентская часть взаимодействует с серверной с помощью разработанного прикладного программного интерфейса, который, кроме известного функционала подхода к архитектуре сетевых протоколов REST, основанном на протоколе HTTP, где клиент использует запросы только в формате, определенном спецификацией серверной части, также предусматривает реализацию возможностей подхода к архитектуре сетевого взаимодействия с использованием языка запросов GraphQL, что позволяет строить параметры запроса со стороны клиента. Локальная база данных содержит набор отпечатков и метаданных о музыкальных композициях для ускорения процесса распознавания, поскольку позволяет покрыть большинство сценариев использования приложения с самыми популярными музыкальными композициями с возможностью быстрого возврата результата после локального сопоставления данных без необходимости задержек клиент-серверного взаимодействия. Модуль синхронизации базы данных отвечает за своевременное обновление локальной базы новыми отпечатками с серверной части и за общую синхронизацию клиентской и серверной частей системы. Планировщик синхронизации обеспечивает формирование расписания синхронизации локальной и серверной баз данных, а также реализует стратегии оптимизации использования аккумулятора и обеспечивает работу с низким уровнем интернет-соединения. Рассмотрены особенности хранения базы отпечатков композиций и стратегии работы с устройствами на базе мобильных платформ, в частности, под операционную систему Android в режиме Doze, запрещающем фоновую работу устройства в состоянии покоя, кроме коротких промежутков времени, так называемых окон поддержки, которые определяются операционной системой в процессе работы и позволяют выполнять кратковременные фоновые операции. Проведена оптимизация процесса использования энергии аккумулятора мобильного устройства при синхронизации метаданных музыкальных композиций между клиентской и серверной составляющими системы.

Ключевые слова: мобильное приложение, распознавание музыки, клиент-серверная система, хранение данных.

Abstract. The article discusses the models of the recognition system for musical compositions in the system of synthesis and analysis of musical sounds, aimed at increasing the identification capabilities of an automated system. The recognition module for musical compositions is oriented to the server part of the system, which, regardless of the client, contains a database with fingerprints of musical compositions. With the help of melody recognition algorithms with a given argument in the form of a fingerprint, the server returns a list of musical compositions that most satisfy the search conditions. The client part interacts with the server part using the developed application programming interface, which, in addition to the well-known functional approach to the REST network protocol architecture based on the HTTP protocol, where the client uses requests only in the format defined by the server part specification, also provides for the implementation of the capabilities of the network architecture approach interactions using the GraphQL query language, which allows to build query parameters on the client side. The local database contains a set of fingerprints and metadata about musical compositions to speed up the recognition process, since it allows to cover most application scenarios with the most popular musical compositions with the ability to quickly return results after local data matching without the need for client-server interaction delays. The database synchronization module is responsible for the timely updating of the local database with new fingerprints from the server side and for the general synchronization of the client and server parts of the system. The synchronization scheduler provides a synchronization schedule for the local and server databases, as well as implements strategies for optimizing battery usage and ensures work with a low level of Internet connection. The features of storing the fingerprint database of compositions and strategies for working with devices based on mobile platforms, in particular, for the Android operating system in Doze mode, which prohibits the background operation of the device at rest, except for short periods of time, the so-called support windows, which are determined by the operating system work process and allow to perform short-term background operations. Performed the optimi-

zation of the process of using the battery energy of a mobile device while synchronizing the metadata of musical compositions between the client and server components of the system.

Keywords: mobile application, music recognition, client-server system, data persistence.

DOI: <https://doi.org/10.31649/1999-9941-2020-47-1-32-38>.

Вступ

Мобільні пристрої швидко набувають популярності і поширення серед пересічних користувачів, а тому можуть слугувати інструментом для розв'язання великої кількості повсякденних задач. Більше того, такі пристрої не є прив'язаними до конкретної фізичної локації, тому додатки на базі їхніх платформ мають широкий спектр сценаріїв застосування. Одним з таких сценаріїв є функціонал обробки аудіо контенту, синтезу та розпізнавання музичних композицій [1-2].

З іншого боку ключовою особливістю професійного програмного забезпечення на базі мобільних платформ є обмеженість ресурсів мобільних пристроїв у порівнянні з настільними або веб-рішеннями. Зокрема, енергія акумулятора є вичерпною і потребує додаткових стратегій економії, об'єм оперативної та фізичної пам'яті є досить обмеженим [2-3]. Крім того, зв'язок клієнтських мобільних додатків з серверною частиною є не завжди можливим, а його якість не є ідеальною і залежить від сили сигналу стільникового зв'язку. Враховуючи наведені особливості функціонування програмного забезпечення на базі мобільних платформ, необхідним є використання додаткових стратегій та алгоритмів для оптимізації розроблюваної системи синтезу та аналізу музичних звуків.

Актуальність теми

Автоматизація процесу обробки аудіо контенту набуває популярності в процесі розвитку і поширення як професійних, так і любительських мобільних програм, орієнтованих на розпізнавання музичних композицій та синтез звукових сигналів у процесі створення власних мелодій. Це обумовлює актуальність розробки та дослідження сучасних засобів обробки аудіо сигналів з використанням мобільних платформ та технологій у процесі аналізу й розпізнавання аудіо контенту, що, в свою чергу, відкриває новий спектр сценаріїв використання розроблюваної системи.

Мета

Метою дослідження є підвищення ідентифікаційних можливостей процесу аналізу музичних звуків з використанням сучасних програмних технологій, які, в тому числі, базуються на мобільних платформах, що дозволяє автоматизувати процес обробки аудіо контенту.

Задачі

1. Провести аналіз сучасних технологій і підходів у реалізації швидкої синхронізації клієнтської складової мобільної системи з серверною.
2. Розробити структуру та архітектуру процесу розпізнавання музичних звуків та модель взаємодії клієнтської та серверної складових.
3. Проаналізувати особливості використання мобільних платформ, зокрема, оптимізацію ресурсів для ефективного енергозбереження, а також обмеженість об'єму доступної пам'яті.
4. Дослідити процес і послідовність розпізнавання музичних композицій.

Розробка моделей автоматизованої системи розпізнавання музичних композицій

Однією з базових компонентів розроблюваної системи обробки аудіо контенту є модуль розпізнавання музичних композицій. Для його реалізації необхідна наявність серверної частини, яка, незалежно від клієнта, міститиме базу даних з відбитками музичних композицій [1, 2, 3]. За допомогою алгоритмів розпізнавання мелодій за заданим аргументом у вигляді відбитку сервер повертатиме список музичних композицій, які найбільше задовольняють умовам пошуку. Клієнтська частина матиме змогу взаємодіяти з серверною за допомогою розробленого прикладного програмного інтерфейсу (англ. API). Існує декілька шляхів для реалізації такого інтерфейсу. Одним з них є підхід REST, що базується на протоколі HTTP [4]. Іншим підходом є використання мови запитів GraphQL, що дозволяє будувати параметри запиту зі сторони клієнта [5, 6], на відміну від REST, де клієнт використовує запити лише в форматі, визначеному специфікацією серверної частини.

Узагальнену модель автоматизованої системи розпізнавання музичних звуків можна подати у формалізованому вигляді виразу 1:

$$M = \{I, N, D\}, \quad (1)$$

де I – вхідний аудіо потік, що записується користувачем;

N – рівень шумів у вхідному аудіо записі;

D – обсяг даних у базі відбитків існуючих композицій.

Модель клієнт-серверної мобільної системи розпізнавання музичних композицій, що презентує структуру автоматизованої системи, наведена на рис. 1.

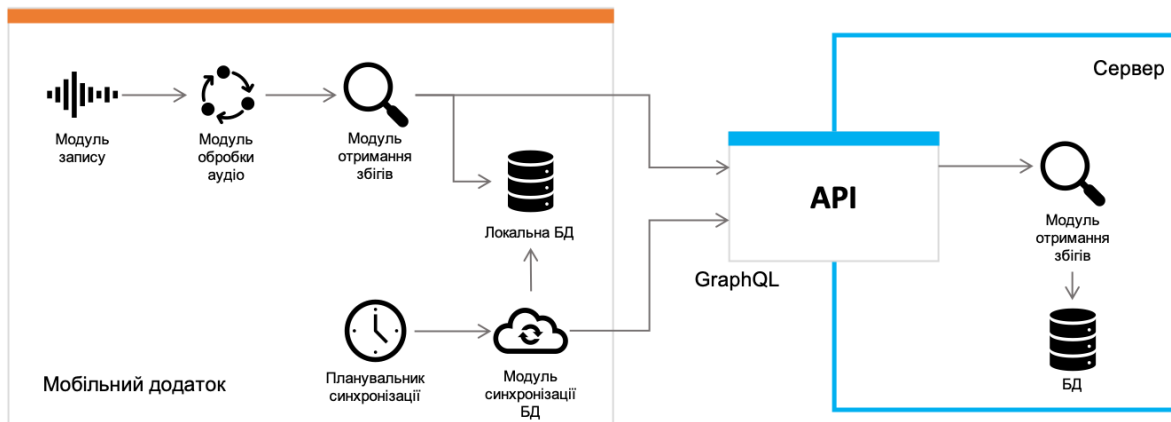


Рисунок 1 – Модель клієнт-серверної мобільної системи розпізнавання музичних композицій

Серед основних компонентів клієнтського мобільного додатку виділимо:

- Модуль запису – модуль запису вхідного аудіо потоку.
- Модуль обробки аудіо – компонента, що відповідає за дискретизацію вхідного аудіо потоку та створення його відбитку [1, 2, 3].
- Модуль отримання збігів – модуль, що реалізує логіку пошуку збігів у базах даних, а також оркеструє взаємодію локальної та серверної баз даних. В основу його реалізації покладено шаблон репозиторій (repository) [7].
- Локальна база даних (БД) містить набір відбитків та метаданих про музичні композиції для прискорення процесу розпізнавання. Наявність такої компоненти є важливою, оскільки вона дозволяє покрити більшість сценаріїв використання додатку з найпопулярнішими музичними композиціями з можливістю швидкого повернення результату після локального співставлення даних без необхідності затримок клієнт-серверної взаємодії.
- Модуль синхронізації БД – компонента, що відповідає за своєчасне оновлення локальної бази даних новими відбитками з серверної частини та за загальну синхронізацію клієнтської і серверної частин.
- Планувальник синхронізації – планувальник, що відповідає за розклад синхронізації локальної та серверної баз даних, а також реалізує стратегії оптимізації використання акумулятора та забезпечує роботу з низьким рівнем інтернет-з'єднання.

Основна задача планування оновлень – реалізація синхронізації локальної та серверної баз даних за мінімальних обчислювальних витрат і економії енергії акумулятора мобільного пристрою. Необхідно враховувати особливості виконання фонових задач на мобільних платформах, де є досить жорстке їх обмеження при роботі пристрою в стані спокою. Зокрема, операційна система Android використовує режим Doze, який забороняє фонову роботу пристрою у стані спокою, окрім коротких проміжків часу, так званих вікон підтримки, які визначаються операційною системою в процесі роботи та дозволяють виконувати короткочасні фонові операції [8].

Використання повної синхронізації в такому випадку є недоцільним, адже під час роботи в режимі Doze розроблювана система не має достатньо часу на завантаження оновлень у базу даних. Проте, такий підхід може використовуватися для коротких статусних оновлень. Крім того, необхідно враховувати силу сигналу інтернет-з'єднання і забороняти завантаження великого об'єму даних при слабкому рівні сигналу.

Таким чином, доцільно режим оновлення здійснювати шляхом отримання коротких статусних оновлень у фоновому режимі, що є можливим також при низькій якості інтернет-зв'язку. Повне оновлення та синхронізацію бази даних необхідно виконувати на старті додатку за наявним з'єднанням WiFi. Крім того, для зменшення об'єму даних, що необхідно завантажити мережею, виконуваний файл додатку міститиме встановлений початковий набір відбитків у базі даних.

Дослідження процесу розпізнавання звукового контенту

Після початкового запису вхідного аудіо сигналу, його дискретизації та створення відбитку [1,2,3] система надсилає запит до локальної бази даних на предмет пошуку збігів. При їх наявності відбувається

сортування за відсотковим показником збігу в порядку його спадання та обирається перший запис, що буде відображений користувачеві у вигляді метаданих бажаної музичної композиції.

Процес розпізнавання музичних композицій у вигляді блок-схеми алгоритму зображено на рис. 2.

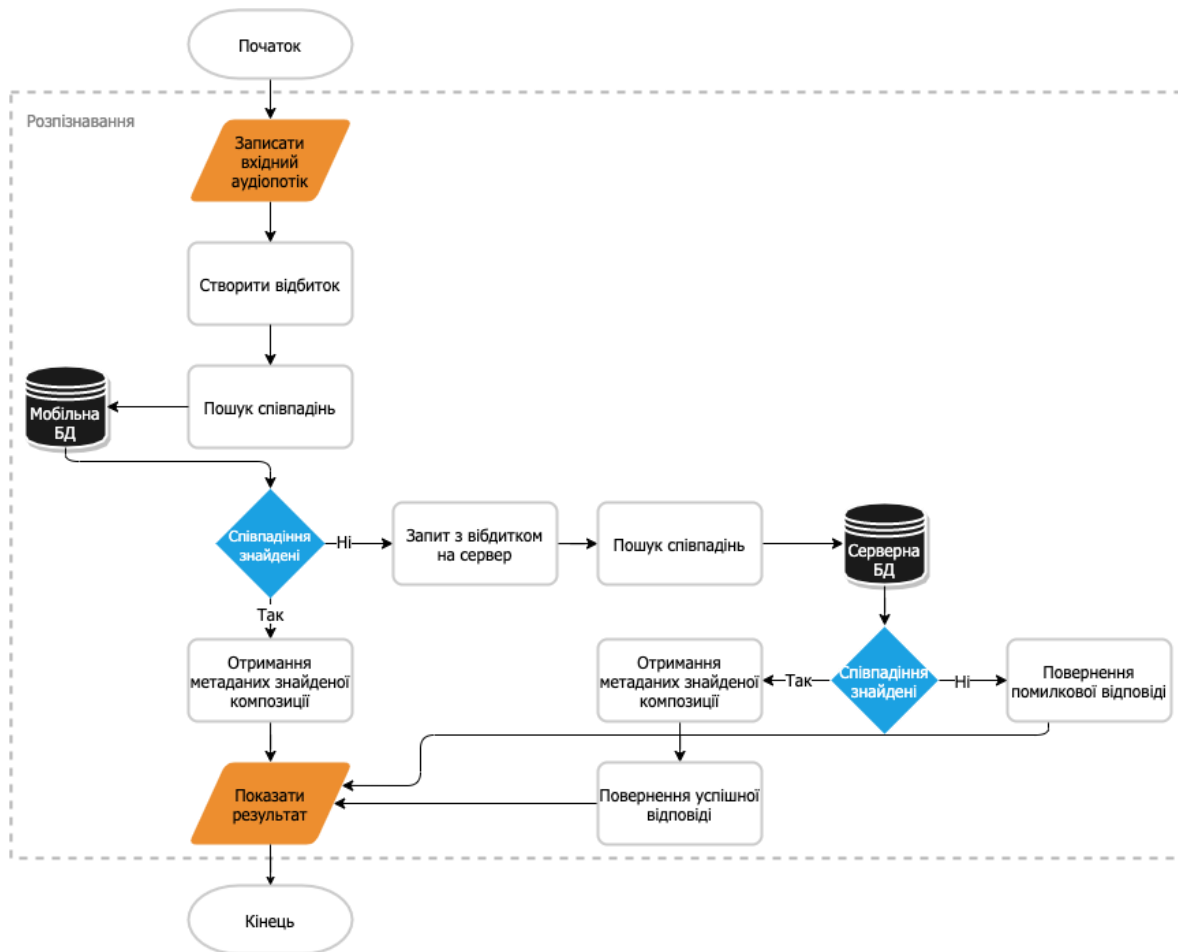


Рисунок 2 – Блок-схема алгоритму процесу розпізнавання музичних композицій

У разі відсутності локальних збігів, мобільний додаток створює запит до серверної компоненти і передає відбиток бажаної композиції у якості аргументу. Сервер, у свою чергу, робить запит до власної бази даних на предмет збігу відбитків. При позитивному результаті сервер формує успішну відповідь для клієнтського додатку з поверненням списку збігів. Мобільний додаток зберігає отримані дані у локальну базу, визначає найбільш релевантний елемент, що характеризує бажану музичну композицію, та відображає його користувачеві як результат.

У разі відсутності збігів з елементами серверної бази даних, клієнтський додаток отримуватиме відмову в шуканому результаті. Відповідне повідомлення про відмову буде відображене користувачеві.

Описаний процес розпізнавання аудіо контенту відображено на рис. 3 у вигляді діаграми послідовності з ідентифікацією архітектури процесу обробки звукових даних.

Описаний процес аналізу та розпізнавання музичних мелодій дозволяє оптимізувати ідентифікацію аудіо контенту за несприятливих комунікаційних умов клієнт-серверної взаємодії компонентів мобільної системи.

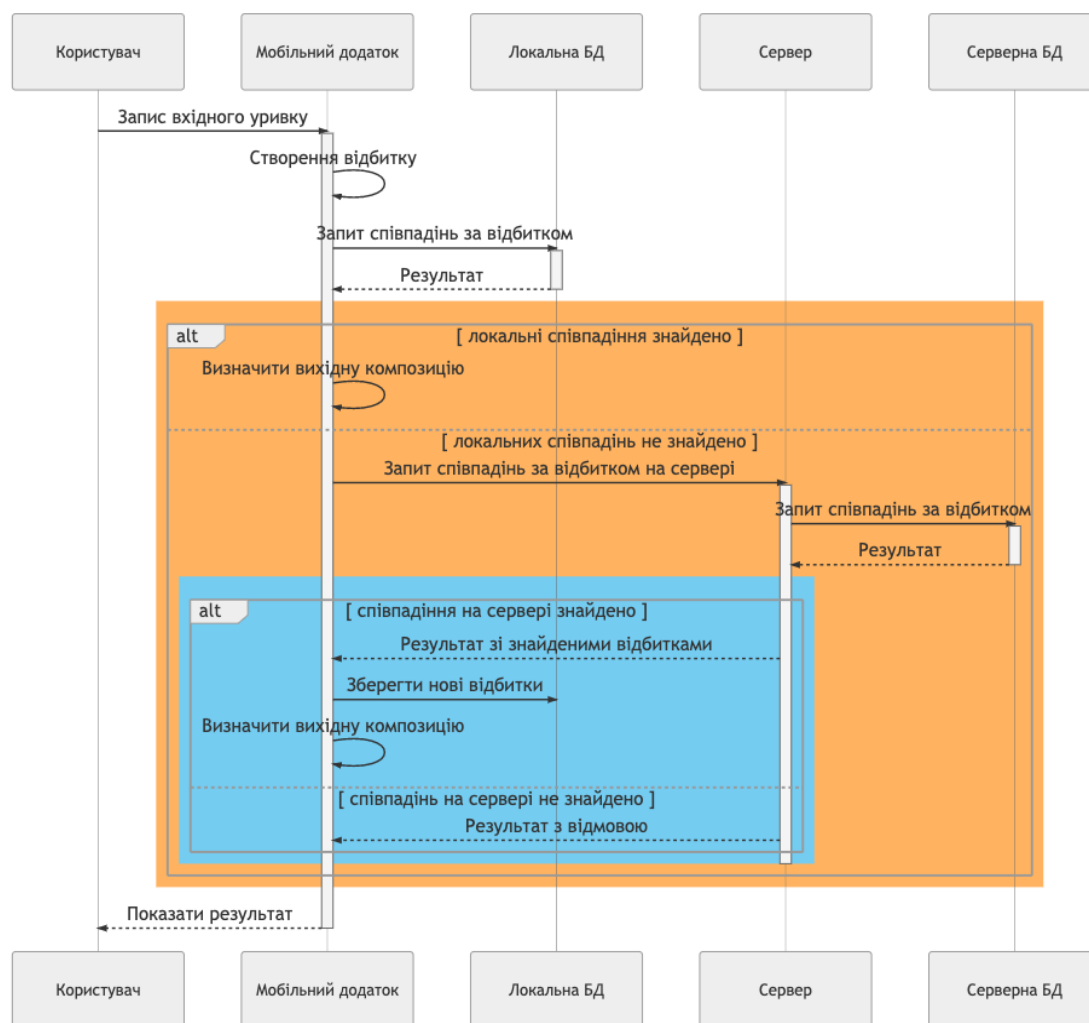


Рисунок 3 – Діаграма послідовності процесу розпізнавання музичних композицій

Висновки

Запропоновано моделі клієнт-серверної архітектури мобільної системи розпізнавання музичних композицій, орієнтовані на підвищення ідентифікаційних можливостей автоматизованої системи. Процес аналізу звукових потоків використовує закладені спеціалізовані особливості зберігання бази відбитків музичних композицій для прискорення процесу знаходження співпадань досліджуваного аудіо контенту. Проведено оптимізацію роботи мобільного додатку шляхом ефективного оновлення локальної бази музичних композицій з використанням режиму Doze на базі операційної системи Android, що дозволяє контролювати якість інтернет-зв'язку та враховувати можливість економії енергії акумулятора шляхом отримання коротких статусних оновлень бази даних у фоновому режимі.

Список літератури

- [1] Ставицький П.В. Використання технологій аналізу та синтезу музичних звуків для розробки музичного синтезатора / П.В. Ставицький, А.В. Денисюк, В.В. Войтко. НТКП ВНТУ. Факультет інформаційних технологій та комп'ютерної інженерії : XLVI Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії, 2017. С. 3 – URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2017/paper/view/2793/2521>
- [2] Ставицький П.В. Розробка модуля розпізнавання музики для мобільного додатку / П.В.Ставицький, В.В.Войтко. НТКП ВНТУ. Факультет інформаційних технологій та комп'ютерної інженерії : XLVII Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії, 2018. – URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2018/paper/view/5209/4571>

- [3] Voitko Viktoriia Automated system of audio components analysis and synthesis / Viktoriia V. Voitko, Svitlana V. Bevz, Sergii M. Burbelo, Pavlo V. Stavytskyi, Bogdan Pinaiev, Zbigniew Omiotek, Doszhon Baitussupov, Aigul Bazarbayeva. Proc. SPIE 11045, Optical Fibers and Their Applications, 2018, 110450V (15 March 2019); doi: 10.1117/12.2522313.
- [4] Identify songs playing near you: веб-сайт. [Електронний ресурс] – Режим доступу: <https://support.google.com/googleplaymusic/answer/2913276?hl=en>
- [5] GraphQL Specification Versions [Електронний ресурс] – Режим доступу: <https://spec.graphql.org>
- [6] GraphQL: A data query language – Facebook Engineering [Електронний ресурс] – Режим доступу: <https://engineering.fb.com/core-data/graphql-a-data-query-language/>
- [7] Fowler M. Patterns of Enterprise Application Architecture, Addison-Wesley Professional / M. Fowler, D.Rice, M. Foemmel, E. Heatt, R. Mee, R. Stafford, 1 edition, 560, (November 15, 2002) – p. 322.
- [8] Optimize for Doze and App Standby / Android Developers [Електронний ресурс] – Режим доступу: <https://developer.android.com/training/monitoring-device-state/doze-standby>
- Стаття надійшла: 20.04.2020.

References

- [1] Stavytskyi P.V. Vykorustannia tehnologiy analyzy ta suntezy myzuchnyh zvukiv dlia rozrobku myzuchnogo suntezatora / P.V. Stavytskyi, A.V. Denusiuk, V.V. Voitko. NTKP VNTU. Facultet informatsijnyh tehnology ta kompjuternoj engenerii : XLVI Naukovo-tehnichna konferenzia facultety informatsijnyh tehnology ta kompjuternoj engenerii, 2017. P. 3 – URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2017/paper/view/2793/2521>
- [2] Stavytskyi P.V. Rozrobka modulia rozpoznavannia myzuku dlia myzuchnogo dodatku / P.V. Stavytskyi, V.V. Voitko. NTKP VNTU. Facultet informatsijnyh tehnology ta kompjuternoj engenerii : XLVII Naukovo-tehnichna konferenzia facultety informatsijnyh tehnology ta kompjuternoj engenerii, 2018. – URL: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2018/paper/view/5209/4571>
- [3] Voitko Viktoriia Automated system of audio components analysis and synthesis / Viktoriia V. Voitko, Svitlana V. Bevz, Sergii M. Burbelo, Pavlo V. Stavytskyi, Bogdan Pinaiev, Zbigniew Omiotek, Doszhon Baitussupov, Aigul Bazarbayeva. Proc. SPIE 11045, Optical Fibers and Their Applications, 2018, 110450V (15 March 2019); doi: 10.1117/12.2522313.
- [4] Identify songs playing near you: web-sait. [Electronic resource] – Available: <https://support.google.com/googleplaymusic/answer/2913276?hl=en>
- [5] GraphQL Specification Versions [Electronic resource] – Available: <https://spec.graphql.org>
- [6] GraphQL: A data query language – Facebook Engineering [Electronic resource] – Available: <https://engineering.fb.com/core-data/graphql-a-data-query-language/>
- [7] Fowler M. Patterns of Enterprise Application Architecture, Addison-Wesley Professional / M. Fowler, D.Rice, M. Foemmel, E. Heatt, R. Mee, R. Stafford, 1 edition, 560, (November 15, 2002) – p. 322.
- [8] Optimize for Doze and App Standby / Android Developers [Electronic resource] – Available: <https://developer.android.com/training/monitoring-device-state/doze-standby>

Відомості про авторів

Войтко Вікторія Володимирівна – канд. техн. наук, доцент кафедри програмного забезпечення, Вінницький національний технічний університет.

Бевз Світлана Володимирівна – канд. техн. наук, доцент кафедри електричних станцій і систем, Вінницький національний технічний університет.

Бурбело Сергій Михайлович – канд. техн. наук, старший викладач кафедри програмного забезпечення, Вінницький національний технічний університет.

Ставицький Павло Валерійович – аспірант кафедри програмного забезпечення, Вінницький національний технічний університет, Вінниця

В. В. Войтко, С. В. Бевз, С.М. Бурбело, П. В. Ставицький

МОДЕЛИ СИСТЕМЫ АНАЛИЗА И РАСПОЗНАВАНИЯ МУЗЫКАЛЬНЫХ КОМПОЗИЦИЙ

Винницкий национальный технический университет, Винница

V.V. Voitko, S.V. Bevz, S.M. Burbelo, P.V. Stavytskyi

**MODELS OF ANALYSIS AND RECOGNITION SYSTEM OF
MUSICAL COMPOSITIONS**

Vinnitsia National Technical University, Vinnitsia