

УДК 004.8:02

doi:10.20998/2413-4295.2021.02.12

ПРОЕКТУВАННЯ АРХІТЕКТУРИ АВТОМАТИЗОВАНОЇ БІБЛІОГРАФІЧНОЇ СИСТЕМИ

С. О. ЦИБУЛЬНИК*, Д. С. БІДНИК

кафедра приладів і систем орієнтації і навігації, Національний технічний університет України «Київський політехнічний інститут ім. Ігоря Сікорського», Київ, УКРАЇНА
*e-mail: tsybulnik.s.a@gmail.com

АНОТАЦІЯ Розвиток інформаційних та комп'ютерних технологій призвів до необхідності еволюції концепції універсального бібліографічного контролю. Поява Інтернету та веб-технологій дозволила даній концепції вийти на новий рівень за рахунок створення ряду спільних міжнародних стандартів. Окрім цього, для забезпечення контролю та обміну бібліографічною інформацією були створені загальнодоступні бібліографічні та наукометричні бази даних. Сьогодні програмне забезпечення для управління бібліографічними записами користується попитом у різних країнах Європи та Америки. Розробка подібної автоматизованої бібліографічної системи та адаптація її функціоналу до стандартів та вимог у межах України є актуальною з ряду причин. Серед основних причин можна зазначити необхідність кожного науково-педагогічного співробітника закладів вищої освіти підтверджувати свої наукові досягнення при прийнятті на роботу, поданні наукової роботи на різноманітні конкурси, для отримання наукового звання або вченого ступеня, тощо. Бурхливий розвиток інформаційних та комп'ютерних технологій дозволяє сьогодні відмовитися від ведення переліку наукових праць в ручному режимі та перейти до використання спеціалізованого програмного забезпечення на смартфонах. Саме тому спроектовано архітектуру автоматизованої бібліографічної системи, яка розроблена у вигляді мобільного додатку на базі операційної системи Android. Показано, що більше третини населення планети має мобільний телефон і найбільш популярною мобільною операційною системою є Android. Мовою програмування, на якій буде написано програмне забезпечення, обрано Java, оскільки переважна більшість операційної системи Android написана на цій мові. Для обраної операційної системи визначено ряд технологій, які дозволять спростити процес розробки мобільного додатку. На базі багаторівневої моделі архітектури та шаблону MVVM спроектовано тришарову архітектуру автоматизованої бібліографічної системи. Дана архітектура дозволяє забезпечити основні нефункціональні характеристики якості розроблюваного програмного забезпечення, а також ефективно реалізувати правила бізнес-логіки в межах об'єктно-орієнтованої парадигми програмування.

Ключові слова: програмне забезпечення; Android; MVVM; програмна архітектура; бібліографічна база даних; Java

DESIGN OF THE ARCHITECTURE OF AN AUTOMATED BIBLIOGRAPHIC SYSTEM

S. TSYBULNYK, D. BIDNYK

Department of Orientation and Navigation Instruments and Systems, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, UKRAINE

ABSTRACT The development of information and computer technologies has led to the need to evolve the concept of universal bibliographic control. The creation of the Internet and web technologies has allowed this concept to reach a new level by creating a number of common international standards. In addition, to ensure control and exchange of bibliographic information public bibliographic and scientometric databases were created. Today, software for managing bibliographic records is in demand in various countries in Europe and America. The most popular software in these countries is EndNote, RefWorks, BibTeX and Zotero. The development of such automated bibliographic system and the adaptation of its functionality to standards and requirements within Ukraine is relevant for a number of reasons. The main reasons are the need for every scientist and lecturer of higher education institutions to confirm their scientific achievements when hiring, submitting scientific work to various competitions, to obtain a scientific degree and so on. Today the rapid development of information and computer technology allows us to abandon the list of scientific papers in manual mode and move to the use of specialized software on smartphones. That is why the architecture of an automated bibliographic system, which is developed as a mobile application based on the Android operating system, was designed. Java is chosen as the programming language in which the software will be written, as the vast majority of the Android operating system is written in this language. A number of technologies were chosen for the selected operating system. They will simplify the process of developing a mobile application. The three-layer architecture of the automated bibliographic system is designed on the basis of the multilevel model of architecture and the MVVM template. This architecture allows to provide the main non-functional characteristics of the quality of the developed software, as well as to effectively implement the rules of business logic within the object-oriented programming paradigm.

Keywords: Software; Android; MVVM; software architecture; bibliographic database; Java

Вступ

У 1960-х роках відбулося зародження концепції універсального бібліографічного контролю [1]. Дана концепція була заснована на міжнародно узгоджених принципах каталогізації, що характеризувалася

новими технологічними вимогами до бібліотечного співтовариства. Метою концепції універсального бібліографічного контролю [1] стало просування всесвітньої системи контролю та обміну бібліографічною інформацією, що має бути загальнодоступною в уніфікованій міжнародно

прийнятій формі та містити основні бібліографічні дані про всі видання у всіх країнах.

Бібліотеки в 1990-х роках були змушені переглянути свої стандарти та практики щодо організації бібліографічної інформації. З одного боку вони мали використовувати прийняті на міжнародному рівні стандарти у повсякденній практиці, а з іншого – враховувати розвиток інформаційних технологій та ввести опис нових форм матеріалів – електронних ресурсів (цифрових або оцифрованих) і, згодом, веб-документів – до своїх бібліотечних каталогів, щоб задовольнити потреби користувачів [1]. З того часу було недостатньо розглядати бібліотечний фонд (об'єкти бібліографічного опису) зі статичної точки зору, стало необхідним описати його в контексті життєвого циклу. Міжнародні комунікаційні технології загалом, а також Інтернет та веб-технології зокрема, почали відігравати вирішальну роль [1] на тлі цієї вимоги. Це також викликало необхідність для бібліотек більш ефективно взаємодіяти зі своїм середовищем – видавцями та користувачами. Тому для ефективного просування концепції універсального бібліографічного контролю бібліотеки все частіше використовують прогресивні комп'ютерні технології для формування бібліографічних та наукометричних баз даних.

Використання програмного забезпечення для управління бібліографічними записами, такого як EndNote, RefWorks, BibTeX та Zotero, сьогодні широко зарекомендувало себе серед дослідників та студентів, як інструмент економії часу для написання навчальних робіт. Як наочний приклад, дванадцять з двадцяти п'яти університетів США та чотири з п'ятнадцяти канадських університетів надають своїм клієнтам ліцензії на сайт EndNote, як стандартну частину своїх послуг, а вісім веб-сайтів бібліотек цих університетів пропонують EndNote для придбання, підтримуючи інші програми управління бібліографічними даними. Цей широко розповсюджений доступ у відомих великих академічних закладах характеризує ефективність програмного забезпечення, як інструменту управління бібліографією [2].

У травні 2011 р. з метою вимірювання використання програмного забезпечення для управління посиланнями в академічному середовищі в Талліннському університеті (Естонія) було проведено онлайн-опитування [3] студентів та викладачів. Пізніше подібне опитування [4] було також проведено в одному із найбільших закладів вищої освіти Італії – Торінському університеті. Обидва дослідження показали низьку обізнаність студентів та викладачів щодо функціональних можливостей подібного програмного забезпечення та низьку активність бібліотек у проведенні навчання.

Від моменту зародження концепції універсального бібліографічного контролю та до сьогоднішнього дня завдяки розвитку комп'ютерних

технологій було багато спроб створити програмне забезпечення управління бібліографічними записами [5–8]. Усі вони мали свої переваги та недоліки, але не всі з них пройшли перевірку часом і використовуються сьогодні.

У реаліях закладів вищої освіти України кожен науково-педагогічний співробітник постійно стикається з необхідністю формування та коректного оформлення списку своїх наукових робіт у відповідності до певного стандарту. Серед основних джерел, пов'язаних із необхідністю формування бібліографічного списку наукових праць, можна назвати наступні: конкурс на заміщення посади, захист докторської дисертації або дисертації кандидата наук, різноманітні конкурси, зокрема державні, отримання наукового звання, щорічні звіти структурних підрозділів з науки, тощо.

На сьогоднішній день більшість науково-педагогічних співробітників для формування та ведення списку наукових праць використовує можливість програмних засобів Word та Excel. Проте ці програмні засоби мають ряд недоліків, які призводять до збільшення витрат часу на створення та пошук записів, відслідковування записів-дублів, сортування, підведення статистики. Враховуючи ці недоліки, пропонується створення автоматизованої бібліографічної системи, яка спростить процес роботи (створення, редагування, пошук, сортування, статистика, видалення, тощо) з бібліографічним списком наукових праць у межах структурного підрозділу закладу вищої освіти.

Мета роботи

Метою роботи є проектування архітектури автоматизованої бібліографічної системи, що розробляється у вигляді мобільного додатку на базі операційної системи Android.

Середовище розробки автоматизованої системи

В останні десятиліття у всьому світі спостерігається бурхливий розвиток інформаційних технологій. Завдяки цьому в магазинах з'явилося багато нових пристроїв таких, як планшети, смартфони, ноутбуки та інші. Сьогодні більшість людей не уявляє свого життя без них.

Практично кожен існуючий гаджет має операційну систему, яка дозволяє йому виконувати своє функціональне призначення. Лідером серед операційних систем (платформ) подібних приладів на сьогоднішній день є Android від компанії Google (рис. 1). Ця система використовується на самих різних пристроях, а саме: смартфонах, планшетах, телевізорах, смарт-годинниках та інших гаджетах і навіть автомобілях. Відповідно до [9] за останній рік операційною системою Android користуються близько 72% власників смартфонів, а загальна кількість користувачів смартфонів на операційній системі Android оцінюється в більш ніж 2,5 млрд. людей по всьому світу [10].

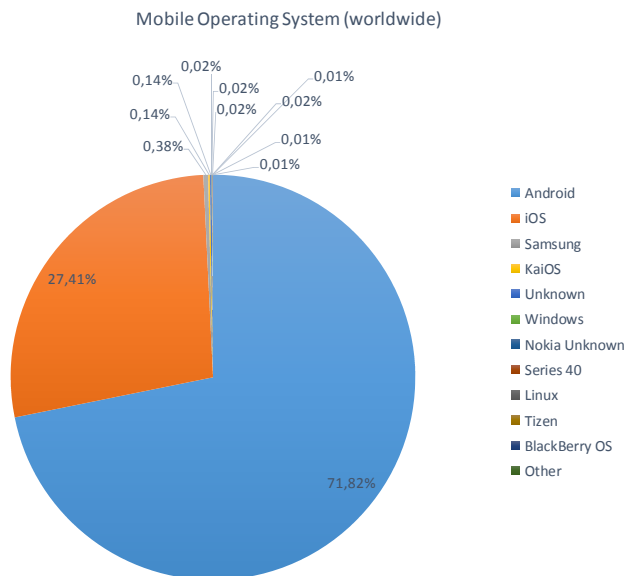


Рис. 1 – Діаграма розподілу користувачів за операційними системами мобільних телефонів

Програмне забезпечення на базі операційної системи Android розробляється за допомогою різних мов програмування та відповідних технологій. Найбільш поширена та використовувана мова серед усіх – Java, оскільки переважна більшість операційної системи Android написана на цій мові. Саме тому розробка програмного забезпечення (додатків) для платформи Android ведеться переважно на мові Java.

Для створення програм на мові Java необхідне спеціальне програмне забезпечення. До спеціального програмного забезпечення відносяться такі інструменти, як JRE (Java Runtime Environment) і JDK (Java Development Kit) [11]. JRE – являє собою набір програмних інструментів для розробки додатків Java. Він об'єднує віртуальну машину Java, основні класи платформи і підтримуючі бібліотеки. JRE є частиною Java Development Kit (JDK), але може бути завантажений окремо. JDK – це цілий набір інструментів, комплект розробника додатків на мові Java, до якого входять стандартні бібліотеки класів Java, компілятор javac, документація, приклади коду і різноманітні службові утиліти. Проте в JDK не входить інтегроване середовище розробки, його необхідно інстальювати окремо.

Для роботи з операційною системою Android існує багато інтегрованих середовищ розробки, але найбільш популярним та зручним є Android Studio, яке створене спеціально для розробки під операційну систему Android. Android Studio [12] – універсальне середовище розробки, яке дозволяє оптимізувати роботу додатків не тільки на смартфонах, але і на інших пристроях, які працюють на основі даної операційної системи. Головною особливістю цього середовища є те, що в ньому вже є вбудований емулятор, який дозволяє перевірити коректну роботу програми на пристроях з різним співвідношенням сторін робочого екрану.

Для збереження структурованих даних зазвичай використовують бази даних. Існують різні системи керування базами даних (СКБД), наприклад, Oracle, SQLite, MySQL та інші. Для створення автоматизованої системи було обрано базу даних SQLite, тому що операційна система Android вже має вбудований інструментарій для керування цією базою.

SQLite [13] – це автономна реляційна база даних без сервера SQL, вона популярна і проста в освоєнні та займає декілька мегабайт в порівнянні з іншими. SQLite підтримує типи даних TEXT (аналог String в Java), INTEGER (аналог long в Java) та REAL (аналог double в Java). Решта типів даних слід конвертувати, перш ніж зберігати в базі даних. Програмне забезпечення SQLite являє собою бібліотеку, з якої компонується програма. Таким чином база даних стає складовою частиною програми. Уся база даних зберігається в єдиному стандартному файлі на пристрої, на якому виконується програма.

Для того щоб можна було запустити розроблений додаток автоматизованої системи на іншому пристрої, який не містить відповідного середовища розробки, необхідно використати одну з існуючих систем збірки. Система автоматизованої збірки [14] – це програмне забезпечення, що забезпечує автоматизацію збірки проекту. Така система дозволяє завантажити бібліотеки до проекту з мережі Інтернет, скомпілювати класи всього проекту, згенерувати допоміжні файли (наприклад, SQL-скрипти, XML-конфіги) та займатися упакуванням скомпільованих класів проекту в архіви різних форматів: zip, rar, grm, jar та інші. Подібні системи часто також дозволяють генерувати програмну документацію та різні звіти.

Для розробки Android додатку зазвичай застосовують автоматичну систему збірки Gradle [15–17], крім того ця система вже встановлена в середовищі розробки Android Studio. Gradle в якості переваг має відкритий вихідний код, є універсальною та може використовувати різні плагіни.

Враховуючи усі вищезазначені переваги, для розробки автоматизованої бібліографічної системи у вигляді мобільного додатку на базі операційної системи Android було обрано середовище розробки Android Studio та СКБД SQLite.

Проектування архітектури

Область комп'ютерних наук з моменту свого утворення зіткнулася з проблемами, пов'язаними зі складністю програмних систем. Раніше подібні проблеми вирішувалися розробниками шляхом правильного вибору структур даних, розробки алгоритмів і застосування концепції розмежування повноважень. Основною ідеєю створення архітектури програмного забезпечення є ідея зниження складності системи завдяки введенню абстракцій та розмежування повноважень. Під програмною

архітектурою розуміється [18] структура компонентів, їх взаємозв'язки, принципи і напрямки розвитку, що визначають їх розробку і еволюцію.

Архітектурним проектуванням системи називається перший етап процесу проектування, на якому розробляється базова структура системи, тобто визначаються основні компоненти (підсистеми), а також їх структура, елементи керування і шляхи взаємодії між собою. Модель системної архітектури часто є відправною точкою для створення специфікації для різних частин програмного забезпечення чи системи [19].

Підсистема – це окрема система, функції (методи) якої не залежать від сервісів, що надаються іншими підсистемами. Підсистеми складаються з модулів і мають певні інтерфейси, за допомогою яких взаємодіють з іншими підсистемами [19].

Модуль – це зазвичай компонент підсистеми, який надає один або декілька сервісів для інших модулів. Модуль може використовувати сервіси, які підтримуються іншими модулями. Як правило, модуль не розглядається як незалежна система. Модулі зазвичай складаються з ряду інших, більш простих компонентів [19].

Для надання автоматизованій системі, яка проектується, певних атрибутів якості часто застосовуються різні архітектурні шаблони. Кожен шаблон має свої завдання, переваги і недоліки.

Для реалізації автоматизованої бібліографічної системи було обрано архітектурний шаблон MVVM. Шаблон MVVM (Model-View-ViewModel) є однією з можливих реалізацій моделі абстрактної машини [19,20] (іноді також називається багаторівневою або шаруватою моделлю архітектури). MVVM (рис. 2) дозволяє відокремити логіку додатку від візуальної частини та складається з трьох компонентів: моделі (Model), моделі подання (ViewModel) і уявлення (View).

Model – модель описує використовувані в додатку дані, їх перетворення та збереження. Моделі можуть містити логіку, безпосередньо пов'язану цими даними, наприклад, логіку валідації властивостей моделі. У той же час модель не повинна містити ніякої логіки, пов'язаної з відображенням даних і взаємодією з візуальними елементами керування (наприклад, графічним інтерфейсом користувача).

View – визначає візуальний інтерфейс, через який користувач взаємодіє з додатком (наприклад, код в html, який реалізує графічний інтерфейс у вигляді кнопок, текстових полів та інших візуальних елементів).

ViewModel – пов'язує Model і View за допомогою механізму прив'язки даних. Якщо в моделі змінюються значення властивостей, при реалізації інтерфейсу автоматично йде зміна відображуваних даних, хоча Model і View не мають безпосереднього зв'язку між собою. ViewModel також містить логіку по отриманню даних з моделі, які потім передаються на рівень вище у View. Також ViewModel визначає логіку по оновленню даних в моделі.

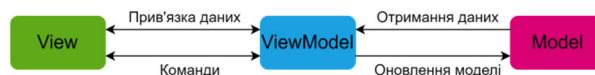


Рис. 2 – Шаблон MVVM

На основі шаблону MVVM побудовано архітектуру мобільного додатку автоматизованої бібліографічної системи (рис. 3).

На рис. 3 можна побачити що View утворює зв'язок лише з ViewModel, а ViewModel – лише з Model. Завдяки обраному шаблону MVVM вдалося відділити логіку роботи з даними від логіки їх відображення. У даному випадку кожен рівень відповідає за свою конкретну функцію та поєднаний функціональними зв'язками лише з сусіднім рівнем, що відповідає визначенню багаторівневої архітектури. Таким чином додаток на базі операційної системи Android можна буде легко підтримувати у подальшому.

Елемент «Графічний інтерфейс користувача», який знаходиться у блоці View відповідає за відображення усіх візуальних компонентів мобільного додатку. Він містить графічні компоненти інтерфейсу, що дозволяють вводити та відображати первні дані. Цей рівень має зв'язок з сусіднім рівнем ViewModel через модуль «Інтерфейс вибору користувача».

Даний інтерфейс виконує декілька основних функцій: 1) формує запит на перевірку наявності користувачів у базі даних й передає його в «Підсистему прийняття рішень»; 2) залежно від вибору на початковому екрані додатку, виконує перехід до модулів «Існуючий користувач» або «Новий користувач».

Модуль «Новий користувач» дозволяє записати ключові дані нового користувача: ID, логін та пароль, прізвище, ім'я та по-батькові, а також інші, визначені в технічному завданні. Після чого ця інформація передається в підсистему «Введення даних», де перевіряється на коректність. Результат перевірки направляється у «Підсистему прийняття рішень».

Модуль «Існуючий користувач» дозволяє отримати доступ до трьох підсистем: «Введення даних», «Редагування даних», «Видалення даних». Відповідно до назв, дані підсистеми мають змогу створювати нові дані (бібліографічні записи), редагувати чи видаляти існуючі (бібліографічні записи, профіль поточного користувача). Також через підсистему «Введення даних» визначається інформація для пошуку та сортування. Результати роботи зазначених вище підсистем передаються в «Підсистему прийняття рішень».

На основі отриманих даних «Підсистема прийняття рішень» може виконувати наступні функції: 1) передавати дані для їх подальшого запису (редагування чи видалення) у базу даних; 2) передавати інформацію щодо запитів (наявність користувача в базі даних, пошук, сортування) у модуль «Інтерфейс формування запитів»; 3) передавати повідомлення в підсистему «Виведення даних» щодо некоректного вводу даних.

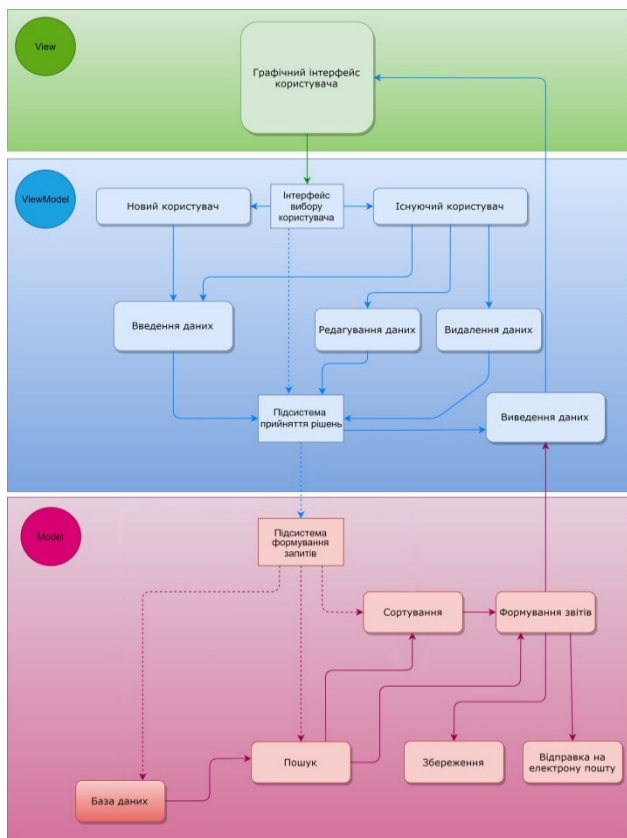


Рис. 3 – Архітектура мобільного додатку

Підсистема «Виведення даних» відповідає за перетворення отриманих даних (звіти з підсистеми «Формування звітів», повідомлення про помилки з «Підсистеми прийняття рішень», тощо) для їх адекватного відображення у графічному інтерфейсі користувача.

Загалом архітектурний рівень ViewModel реалізує бізнес-логіку розроблюваного додатку. Програмна ж логіка (алгоритми пошуку, сортування, збереження, тощо) знаходиться на рівні Model.

«Підсистема прийняття рішень» передає матрицю з даними в «Підсистему формування запитів», яка знаходиться на сусідньому рівні Model. Ця підсистема виконує наступні функції: 1) визначає який саме набір даних отримано; 2) розподіляє отримані дані за відповідними категоріями (запис, редагування, видалення з бази, ключові слова пошуку, умови сортування, тощо); 3) направляє запит на зчитування даних з «Бази даних»; 4) направляє умови пошуку та сортування у підсистему обробки даних, яка представлена двома модулями, а саме: «Пошук» та «Сортування».

Після запиту з «Бази даних» інформація у вигляді матриці передається в підсистему обробки даних, де проходить пошук і сортування отриманої інформації. Далі відсортована інформація потрапляє у модуль «Формування звітів», де за відповідними правилами формується матриця з даними для відображення. Ця інформація далі передається у трьох напрямках: у модулі «Збереження» та «Відправка на електронну пошту», а також у підсистему «Виведення даних».

Підсистема «Виведення даних» призначена для перетворення отриманої інформації у відповідні повідомлення для відображення. Серед повідомлень можуть бути наступні: помилки введення, редагування, видалення чи пошуку, результати пошуку та сортування, персональні дані користувача, тощо. Результати роботи даної підсистеми передаються на більш високий архітектурний рівень View, де виводяться у відповідних елементах графічного інтерфейсу користувача.

Використовуючи описану архітектуру мобільного додатку на базі операційної системи Android, можна оптимізувати процес розробки та значно зменшити час на розробку. Також розташування елементів в окремих підсистемах дозволяє зменшити ймовірність виникнення помилок, оскільки працездатність кожної підсистеми можна протестувати незалежно від інших підсистем.

Висновки

У роботі було показано, що розробка автоматизованої бібліографічної системи є актуальною з ряду причин, серед яких можна зазначити необхідність кожного викладача закладів вищої освіти підтверджувати свої наукові досягнення при прийнятті на роботу, поданні наукової роботи на різноманітні конкурси, для отримання наукового звання або вченого ступеня, тощо.

Бурхливий розвиток інформаційних та комп'ютерних технологій дозволяє сьогодні відмовитися від ведення переліку наукових праць в ручному режимі та перейти до використання спеціалізованого програмного забезпечення на смартфонах. Показано, що більше третини населення планети має мобільний телефон і найбільш популярною мобільною операційною системою є Android.

Для обраної операційної системи визначено ряд технологій, що дозволять спростити процес розробки мобільного додатку.

На базі багаторівневої моделі архітектури та шаблону MVVM спроектовано тришарову архітектуру автоматизованої системи.

У подальшій роботі розроблена архітектура у поєднанні з обраними технологіями дозволить реалізувати якісний та надійний мобільний додаток автоматизованої бібліографічної системи.

Список літератури

1. Willer M., Dunsire G. *Bibliographic Information Organization in the Semantic Web*. Chandos Publishing. 2013. 318 p. doi: 10.1016/B978-1-84334-731-6.50001-6.
2. Fitzgibbons M., Meert D. Are Bibliographic Management Software Search Interfaces Reliable?: A Comparison between Search Results Obtained Using Database Interfaces and the EndNote Online Search Function. *The Journal of Academic Librarianship*. 2010. Vol. 36. Issue 2. P. 144–150. doi: 10.1016/j.acalib.2010.01.005.
3. Francese E. The Usage of Reference Management Software (RMS) in an Academic Environment: A Survey at Tallinn University. *International Conference on Integrated*

- Information 29th September to 3rd October 2011*. Kos, Greece, 2011. P. 293–296.
4. Francese E. Usage of Reference Management Software at the University of Torino. *Journal of Library and Information Science*. 2013. Vol. 4. № 2. P. 145–174. doi: 10.4403/jlis.it-8679.
 5. Wipke W. T. REFFORM: An automatic reference and bibliography formatting system. *Tetrahedron Computer Methodology*. 1988. Vol. 1. Iss. 1. P. 87–92. doi: 10.1016/0898-5529(88)90011-5.
 6. Johnson S. B., Bales M. E., Dine D., [et al.] Automatic generation of investigator bibliographies for institutional research networking systems. *Journal of Biomedical Informatics*. 2014. Vol. 51. P. 8–14. doi: 10.1016/j.jbi.2014.03.013.
 7. Zhu Y., Yan E., Song I.-Y. A natural language interface to a graph-based bibliographic information retrieval system. *Data & Knowledge Engineering*. 2017. Vol. 111. P. 73–89. doi: 10.1016/j.datak.2017.06.006.
 8. Helman W. P., Ross A. B. Radiation chemistry data center: Information services produced from the bibliographic data base. *Radiation Physics and Chemistry*. 1980. Vol. 16. Iss. 6. P. 425–430. doi: 10.1016/0146-5724(80)90186-7.
 9. Mobile Operating System Market Share Worldwide. URL: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (дата звернення: 26.04.2021).
 10. Android. URL: <https://developer.android.com/about> (дата звернення: 26.04.2021).
 11. Oracle Java. URL: <https://www.oracle.com/java/> (дата звернення: 26.04.2021).
 12. Android Studio. URL: <https://developer.android.com/studio> (дата звернення: 26.04.2021).
 13. What Is SQLite? URL: <https://www.sqlite.org/index.html> (дата звернення: 26.04.2021).
 14. Top 10 Best Build Automation Tools Speed Up Deployment Process. URL: <https://uk.myservername.com/top-10-best-build-automation-tools-speed-up-deployment-process> (дата звернення: 26.04.2021).
 15. Accelerate developer productivity. URL: <https://gradle.org/> (дата звернення: 26.04.2021).
 16. Muschko B. *Gradle in Action*. Manning Publications. 2014, 480 p.
 17. Dabir K., *Abhinandan Gradle Essentials: Master the fundamentals of Gradle using real-world projects with this quick and easy-to-read guide* (Community Experience Distilled). Packt Publishing, 2015. 176 p.
 18. Обзор терминологии SOA: Часть 1. Сервис, архитектура, управление и бизнес-термины. URL: <https://www.ibm.com/developerworks/ru/library/ws-soa-term1/?lnk=hm> (дата звернення: 26.04.2021).
 19. Гудов А. М., Завозкин С. Ю., Трофимов С. Н. *Технология разработки программного обеспечения: Учебное пособие*. Кемерово. 2009. 138 с.
 20. Паттерн MVVM. Определение паттерна MVVM. URL: <https://metanit.com/sharp/wpf/22.1.php> (дата звернення: 26.04.2021).
 21. *Academic Librarianship*, 2010, Vol. 36, Iss. 2, pp. 144–150, doi: 10.1016/j.acalib.2010.01.005.
 22. Francese E. The Usage of Reference Management Software (RMS) in an Academic Environment: A Survey at Tallinn University. *International Conference on Integrated Information 29th September to 3rd October 2011*, Kos, Greece, 2011, pp. 293–296.
 23. Francese E. Usage of Reference Management Software at the University of Torino. *Journal of Library and Information Science*, 2013, Vol. 4, no. 2, pp. 145–174, doi: 10.4403/jlis.it-8679.
 24. Wipke W. T. REFFORM: An automatic reference and bibliography formatting system. *Tetrahedron Computer Methodology*, 1988, Vol. 1, Iss. 1, pp. 87–92, doi: 10.1016/0898-5529(88)90011-5.
 25. Johnson S. B., Bales M. E., Dine D., [et al.] Automatic generation of investigator bibliographies for institutional research networking systems. *Journal of Biomedical Informatics*, 2014, Vol. 51, pp. 8–14, doi: 10.1016/j.jbi.2014.03.013.
 26. Zhu Y., Yan E., Song I.-Y. A natural language interface to a graph-based bibliographic information retrieval system. *Data & Knowledge Engineering*, 2017, Vol. 111, pp. 73–89, doi: 10.1016/j.datak.2017.06.006.
 27. Helman W. P., Ross A. B. Radiation chemistry data center: Information services produced from the bibliographic data base. *Radiation Physics and Chemistry*, 1980, Vol. 16, Iss. 6, pp. 425–430, doi: 10.1016/0146-5724(80)90186-7.
 28. Mobile Operating System Market Share Worldwide. Available at: <https://gs.statcounter.com/os-market-share/mobile/worldwide> (accessed: 26.04.2021).
 29. Android. Available at: <https://developer.android.com/about> (accessed: 26.04.2021).
 30. Oracle Java. Available at: <https://www.oracle.com/java/> (accessed: 26.04.2021).
 31. Android Studio. Available at: <https://developer.android.com/studio> (accessed: 26.04.2021).
 32. What Is SQLite? Available at: <https://www.sqlite.org/index.html> (accessed: 26.04.2021).
 33. Top 10 Best Build Automation Tools Speed Up Deployment Process. Available at: <https://uk.myservername.com/top-10-best-build-automation-tools-speed-up-deployment-process> (accessed: 26.04.2021).
 34. Accelerate developer productivity. Available at: <https://gradle.org/> (accessed: 26.04.2021).
 35. Muschko B. *Gradle in Action*. Manning Publications, 2014, 480 p.
 36. Dabir K., *Abhinandan Gradle Essentials: Master the fundamentals of Gradle using real-world projects with this quick and easy-to-read guide* (Community Experience Distilled). Packt Publishing, 2015, 176 p.
 37. Obzor terminologii SOA: Chast' 1. Servis, arhitektura, upravlenie i biznes-terminy [Overview of SOA Terminology: Part 1. Service, Architecture, Governance, and Business Terms]. Available at: <https://www.ibm.com/developerworks/ru/library/ws-soa-term1/?lnk=hm> (accessed: 26.04.2021).
 38. Gudov A. M., Zavozkin S. Ju., Trofimov S. N. *Tehnologija razrabotki programmnogo obespechenija: Uchebnoe posobie* [Software Development Technology: A Tutorial]. Kemerovo, 2009, 138.
 39. Pattern MVVM. Opredelenie patterna MVVM. Available at: <https://metanit.com/sharp/wpf/22.1.php> (accessed: 26.04.2021).

References (transliterated)

1. Willer M., Dunsire G. *Bibliographic Information Organization in the Semantic Web*. Chandos Publishing, 2013, 318 p., doi: 10.1016/B978-1-84334-731-6.50001-6.
2. Fitzgibbons M., Meert D. Are Bibliographic Management Software Search Interfaces Reliable?: A Comparison between Search Results Obtained Using Database Interfaces and the EndNote Online Search Function. *The Journal of*

Відомості про авторів (About authors)

Цибульник Сергій Олексійович – кандидат технічних наук, доцент, КПІ ім. Ігоря Сікорського, доцент кафедри приладів і систем орієнтації і навігації; м. Київ, Україна; ORCID: 0000-0002-4462-0936; e-mail: tsubulnik.s.a@gmail.com.

Serhii Tsubulnyk – Candidate of Technical Sciences (Ph. D.), Docent, Associate Professor, Department of Orientation and Navigation Instruments and Systems, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine; ORCID: 0000-0002-4462-0936; e-mail: tsubulnik.s.a@gmail.com.

Бідник Данило Сергійович – студент кафедри приладів і систем орієнтації і навігації; м. Київ, Україна.

Danylo Bidnyk – student, Department of Orientation and Navigation Instruments and Systems, Igor Sikorsky Kyiv Polytechnic Institute, Kyiv, Ukraine.

Будь ласка, посилайтесь на цю статтю наступним чином:

Цибульник С. О., Бідник Д. С. Проектування архітектури автоматизованої бібліографічної системи. *Вісник Національного технічного університету «ХПІ»*. Серія: Нові рішення в сучасних технологіях. – Харків: НТУ «ХПІ». 2021. № 2 (8). С. 83-89. doi:10.20998/2413-4295.2021.02.12.

Please cite this article as:

Tsubulnyk S., Bidnyk D. Design of the architecture of an automated bibliographic system. *Bulletin of the National Technical University "KhPI". Series: New solutions in modern technology*. – Kharkiv: NTU "KhPI", 2021, no. 2 (8), pp. 83-89, doi:10.20998/2413-4295.2021.02.12.

Пожалуйста, ссылайтесь на эту статью следующим образом:

Цибульник С. А., Биднык Д. С. Проектирование архитектуры автоматизированной библиографической системы. *Вестник Национального технического университета «ХПИ»*. Серия: Новые решения в современных технологиях. – Харьков: НТУ «ХПИ». 2021. № 2 (8). С. 83-89. doi:10.20998/2413-4295.2021.02.12.

АННОТАЦІЯ Развитие информационных и компьютерных технологий привело к необходимости эволюции концепции универсального библиографического контроля. Появление Интернета и веб-технологий позволило данной концепции выйти на новый уровень за счет создания ряда совместных международных стандартов. Кроме этого, для обеспечения контроля и обмена библиографической информацией были созданы общедоступные библиографические и наукометрические базы данных. Сегодня программное обеспечение для управления библиографическими записями пользуется спросом в разных странах Европы и Америки. Наиболее популярным программным обеспечением в данной сфере является EndNote, RefWorks, BibTeX и Zotero. Разработка подобной автоматизированной библиографической системы и адаптация ее функционала к стандартам и требованиям в пределах Украины актуальна по ряду причин. Среди основных причин можно отметить необходимость каждого научно-педагогического сотрудника высших учебных заведений подтверждать свои научные достижения при приеме на работу, представлении научной работы на различные конкурсы, для получения научного звания или ученой степени, и тому подобное. Бурное развитие информационных и компьютерных технологий позволяет сегодня отказаться от ведения перечня научных работ в ручном режиме и перейти к использованию специализированного программного обеспечения на смартфонах. Спроектирована архитектура автоматизированной библиографической системы, разработанная в виде мобильного приложения на базе операционной системы Android. Показано, что более трети населения планеты имеет мобильный телефон и наиболее популярной мобильной операционной системой является Android. Языком программирования, на котором будет написано программное обеспечение, избран Java, поскольку подавляющее большинство операционной системы Android написано на этом языке. Для выбранной операционной системы определен ряд технологий, которые позволят упростить процесс разработки мобильного приложения. На базе многоуровневой модели архитектуры и шаблона MVVM спроектирована трехслойная архитектура автоматизированной библиографической системы. Данная архитектура позволяет обеспечить основные нефункциональные характеристики качества разрабатываемого программного обеспечения, а также эффективно реализовать правила бизнес-логики в пределах объектно-ориентированной парадигмы программирования.

Ключевые слова: программное обеспечение; Android; MVVM; программная архитектура; библиографическая база данных; Java

Надійшла (received) 02.05.2021