

ВОРОЧЕК ОЛЬГА

Харківський Національний Університет Радіоелектроніки

<https://orcid.org/0000-0002-9054-9894>e-mail: olga.vorochek@nure.ua

СОЛОВЕЙ ІЛІЯ

Харківський Національний Університет Радіоелектроніки

<https://orcid.org/0009-0005-5715-2755>email: illia.solovei@nure.ua

РОЗРОБКА МОБІЛЬНОГО ДОДАТКУ ПРОГРАМНОЇ СИСТЕМИ ДЛЯ ВЕДЕННЯ ЦИФРОФІ КАРТКИ ДОМАШНЬОЇ ТВАРИНИ

В роботі представлено аналіз конкурентів та етапи розробки мобільного додатку, який включає функції паспорту, медичної картки, відстеження активності та дошки оголошень про втрати тварин. Розглянуто недоліки та переваги існуючих додатків, детально описаний їх функціонал. Програмна система пропонує багатofункціональне рішення, що дозволяє користувачам отримати повноцінний електронний документ про своїх тварин з можливістю реєстрації різних видів та порід. Було створено діаграму розгортання, на якій показано як мобільний додаток може бути інтегрований у наявну програмну систему. Також наведено діаграму прецедентів, яка детально описує функціонал мобільного додатку для подальшої розробки. В роботі було розглянуто наявні технології створення мобільних додатків - Swift, Kotlin, React Native і Flutter. Після аналізу у якості технології для розробки мобільного додатку обрано мову програмування Dart і фреймворк Flutter, що забезпечує швидкість виконання та високу оптимізацію. Було розроблено користувацький інтерфейс для операційної системи iOS, у якості бібліотеки для елементів інтерфейсу було використано Cupertino UI. Ця технологія дозволяє створити високопродуктивний та естетично привабливий користувацький інтерфейс, який гармонічно виглядає на пристроях від Apple. В системі реалізовано розширені можливості взаємодії з серверною частиною через Django REST Framework, зокрема, функції відслідковування втрачених тварин з геолокаційними даними. Програмна система поєднує у собі функціональність кількох додатків у одне цілісне рішення, що робить її зручною та універсальною для широкого кола користувачів. Основні переваги системи включають підтримку мультимедійних даних, взаємодію через QR-коди та інтеграцію з глобальними картографічними сервісами. У результаті наведено елементи інтерфейсу розробленого мобільного додатку.

Ключові слова: програмна система, мобільний додаток, паспорт, цифрова картка, домашні тварини, Flutter, iOS, Dart.

VOROCHEK OLGA, SOLOVEI ILLIA
Kharkiv National University of Radio Electronics

DEVELOPMENT OF A MOBILE APPLICATION FOR A SOFTWARE SYSTEM TO MANAGE A DIGITAL PET ID CARD

The work presents an analysis of competitors and the stages of development for a mobile application that includes features such as a passport, medical card, activity tracking, and a bulletin board for lost animals. The advantages and disadvantages of existing applications are discussed, with a detailed description of their functionalities. The software system offers a multifunctional solution that allows users to obtain a comprehensive electronic document about their animals, with the ability to register various species and breeds. A deployment diagram has been created to show how the mobile application can be integrated into the existing software system. Additionally, a use case diagram is presented that describes the functionalities of the mobile application for further development. The work considered the current technologies for creating mobile applications - Swift, Kotlin, React Native, and Flutter. After the analysis, the programming language Dart along with the Flutter framework were chosen for developing the mobile application, which ensures execution speed and high optimization. The user interface for the iOS operating system was developed using the Cupertino UI library, which allows to create a high-performance and aesthetically pleasing user interface that looks harmonious on Apple devices. The system implements advanced features for interacting with the server side via Django REST Framework, including the ability to track lost animals with geolocation data. The software system combines the functionality of several applications into one integrated solution, making it convenient and universal for a wide range of users. Key advantages of the system include support for multimedia data, interaction through QR codes, and integration with global mapping services. The paper presents elements of the interface of the developed mobile application.

Keywords: software system, mobile application, passport, digital id card, pets, Flutter, iOS, Dart.

Постановка проблеми у загальному вигляді та її зв'язок із важливими науковими чи практичними завданнями

Програмна система для ведення цифрової картки домашньої тварини містить у собі паспорт, медичну картку, відстеження активності та дошку оголошень, яка забезпечує захист тварини від втрати. У програмній системі потребують як власники тварин, так і медичні заклади. Метою проекту є надати користувачам зручний доступ до даних про домашніх тварин, щоб ці дані могли використовуватися не просто у вигляді щоденника, а як справжній документ. Сервіс допоможе полегшити життя власникам тварин, поліпшити сервіс обслуговування у ветеринарних установах.

Метою дослідження є пошук необхідних технологій та проектування мобільного додатку для вже існуючої програмної системи. Функціонал веб-частини має бути доступний у мобільному додатку, але з додатковими можливостями, які доступні тільки на мобільному пристрої. Застосунок має бути інтернаціоналізованим, підтримувати девайси з різними діагоналями екрану.

Мобільний додаток має бути розроблений таким чином, щоб у майбутньому його функціонал міг бути розширений, та адаптований під нові пристрої. Він має взаємодіяти з серверною частиною програмної системи для ведення цифрової картки домашньої тварини завдяки REST API, коректно отримувати, надсилати, оновлювати та видаляти дані. Також застосунок має правильно обробляти файли, які надсилаються на сервер з пристрою, конвертувати їх у формат, який буде підтримуватись на інших пристроях, що зробить систему універсальною.

Дані користувача мають відправлятися по захищеному протоколу, додаток повинен мати можливість реєстрації нового кабінету користувача та авторизації у існуючий. Основна мета мобільного додатку – бути окремою від веб-застосунку частиною, щоб користувачам не доводилось використовувати декілька платформ для вирішення комплексних задач.

Аналіз досліджень та публікацій

Створення мобільного додатку для існуючого проєкту може забезпечити значні переваги, включаючи підвищення ефективності, зручності користування та розширення досяжності на ринку. Нижче наведені декілька ключових аргументів:

1. покращення користувацького досвіду. Мобільні додатки забезпечують зручнішу і швидшу взаємодію з користувачами, ніж традиційні сайти. Використання мобільних додатків дозволяє забезпечити користувачам неперервний доступ до послуг або продуктів, що значно підвищує їх задоволеність та лояльність [1];

2. оптимізація бізнес-процесів. Використання мобільних додатків може значно оптимізувати бізнес-процеси, автоматизувати збір даних, та інтегрувати різноманітні бізнес-функції, що знижує загальні операційні витрати та підвищує продуктивність. Зокрема, мобільні додатки допомагають скоротити час на обробку запитів та покращити обслуговування клієнтів, сприяючи ефективному вирішенню їхніх проблем [2];

3. комерціалізація та монетизація. Для бізнес-орієнтованих проєктів, мобільні додатки відкривають широкі можливості для монетизації через вбудовані покупки, рекламу та підписки. Це не тільки сприяє зростанню доходів, але й забезпечує постійний зв'язок з користувачами [3].

Розглянемо популярні технології для розробки мобільних додатків:

1. Swift — мова програмування, розроблена Apple, використовується для створення додатків для iOS, macOS, watchOS, і tvOS. Swift забезпечує безпечне управління пам'яттю, швидкість виконання та включає сучасні функції програмування [4];

2. Kotlin — мова програмування, що використовується для розробки Android-додатків, славиться своєю сумісністю з Java і більш безпечним кодом порівняно з Java [5];

3. React Native — фреймворк для розробки крос-платформних мобільних додатків, який дозволяє використовувати JavaScript і React для створення застосунків, які ефективно працюють на iOS та Android [6];

4. Flutter — UI інструментарій від Google для створення вражаючих нативних інтерфейсів на iOS і Android з однієї кодової бази за допомогою мови Dart, що сприяє швидкій розробці додатків [7].

Ці технології допомагають оптимізувати процес розробки, зменшити час виведення продукту на ринок і забезпечити високу якість кінцевих мобільних додатків. Вибір між ними залежить від конкретних вимог проєкту, бюджету та цілей.

Також, при розробці мобільних додатків особливу увагу слід приділяти оптимізації інтерфейсу та взаємодії з користувачем. За даними, опублікованими у "International Journal of Human-Computer Interaction", правильно розроблений користувацький інтерфейс може значно підвищити задоволеність користувачів, знизити час на навігацію по додатку, і, таким чином, збільшити загальну ефективність використання програми. В статті аналізуються різні стратегії покращення інтерфейсів, від адаптації потреб користувачів до інтеграції новітніх технологій для персоналізації їх досвіду [8].

Формулювання цілей статті

Метою роботи є: вибір технології розробки та проєктування мобільного додатку для ведення цифрової картки домашньої тварини, створення макету застосунку та виведення результатів дослідження.

Виклад основного матеріалу

Перед початком розробки мобільного додатку варто проаналізувати функціонал вже існуючих на ринку застосунків. Конкурентами програмної системи є такі сервіси як «Doglogbook» та «Paw Tracks». Вони працюють тільки в одному напрямку – є електронним щоденником та не мають додаткового функціоналу.

Розглянемо сервіс «Doglogbook» більш детально. В результаті дослідження було виявлено наступні недоліки:

– користувач може додати дані про тварину у систему, але вважати їх офіційним документом не можна, система не має можливості підтвердження «паспорту»;

– активність можна додати у систему, але у користувача немає можливості фільтрації за датою – відображається лише остання активність;

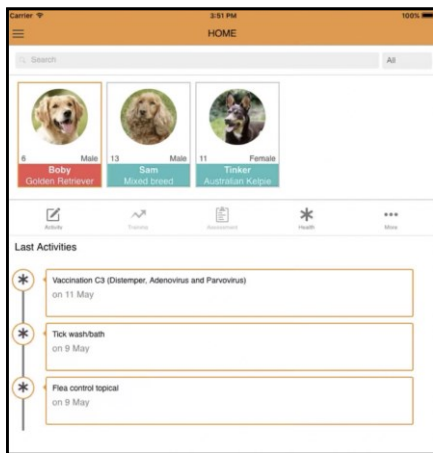


Рис. 1. Головний екран мобільного додатку «Doglogbook»

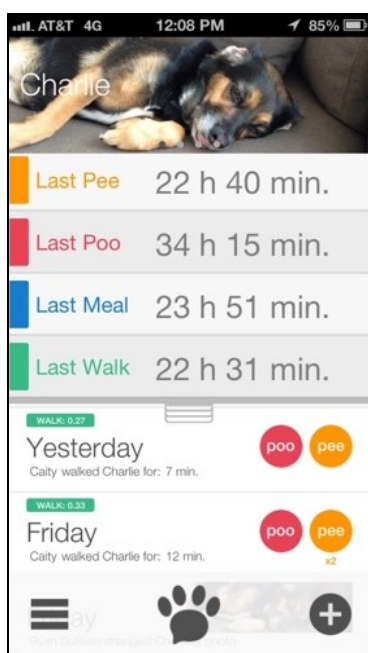


Рис. 2. Головний екран мобільного додатку «Paw Tracks»

– підтримується тільки додавання собак (іншу тварину додати неможливо);

– підтримується тільки англійська локалізація.

На рисунку 1 наведено зображення головного екрану мобільного додатку «Petlogbook», на якому одразу можна побачити більшість описаних недоліків.

Також було проаналізовано додаток «Paw Tracks» (див. рис. 2), який має наступні недоліки:

– відсутній веб-клієнт;

– відсутня підтримка для планшета (iPad);

– підтримується лише відслідковування активності;

– підтримується тільки додавання собак;

– дані зберігаються локально, а не на сервері, існує ризик їх втрати;

– доступна лише англійська локалізація.

Розробляемий мобільний додаток не містить наведених вище недоліків. На відміну від перелічених сервісів він являє собою 4-в-1: паспорт, медична картка, відстеження активності і дошка оголошень про втрати тварин з прив'язкою до географічної позиції.

Для розробки мобільного застосунку у якості мови програмування було обрано Dart та фреймворк Flutter з підключеною бібліотекою Cupertino [9], яка дозволяє створити iOS-стилізований додаток.

Flutter вирізняється серед інших технологій розробки завдяки своїй кросплатформенності та високій продуктивності. Цей фреймворк дозволяє розробникам використовувати одну і ту ж кодову базу для створення додатків як для iOS, так і для Android, що значно спрощує процес адаптації продукту під різні платформи. Код Flutter компілюється в нативний байт-код (Objective-C, Swift, Java, або Kotlin), що забезпечує високу швидкість виконання додатків та плавність інтерфейсу без втрати продуктивності на різних пристроях [10].

Мова програмування Dart, розроблена компанією Google, є оптимізованою для створення високопродуктивних мобільних, настільних, серверних і веб-додатків. Вона була спеціально створена для використання у фреймворку Flutter, що дозволяє розробникам компілювати код у нативний, забезпечуючи високу швидкість виконання та плавність інтерфейсу.

Dart підтримує об'єктно-орієнтований та функціональний стилі програмування, що робить його

достатньо гнучким і зручним для розробників. Однією з ключових особливостей Dart є його система управління пам'яттю зі збирачем сміття, що оптимізує використання ресурсів і знижує ризик витоків пам'яті.

Мобільний застосунок взаємодіє з серверною частиною завдяки REST API [11].

Діаграма розгортання використовується для візуалізації апаратних процесорів / вузлів / пристроїв системи, каналів зв'язку між ними та розміщення програмних файлів на цьому обладнанні. Наприклад, для опису веб-сайту діаграма розгортання покаже які апаратні компоненти існують, які програмні компоненти працюють на кожному вузлі і як підключені різні частини. На рисунку 3 було зображено діаграму розгортання програмної системи для ведення цифрової картки домашньої тварини з інтегрованим у неї мобільним додатком.

Для подальшої розробки мобільного додатку потрібно детально позначити його функціонал. Опишемо взаємодію користувачів із мобільним додатком за допомогою UML діаграми прецедентів (див. рис. 4).

Отже, після того як ми визначилися з технологіями та цілями розробки, можна перейти безпосередньо до створення компонентів додатку. Застосунок запускається завдяки файлу main.dart та поділяється на два основних модулі – Authentication та Tabs.

Модулі поєднані наступними компонентами:

– api.dart, відповідає за взаємодію мобільного застосунку з сервером;

– api_localizations.dart, відповідає за локалізацію додатку;

– app_state.dart, компонент стану, який дозволяє обмінюватися даними між вкладками та оновлювати їх незалежно від відкритої вкладки.

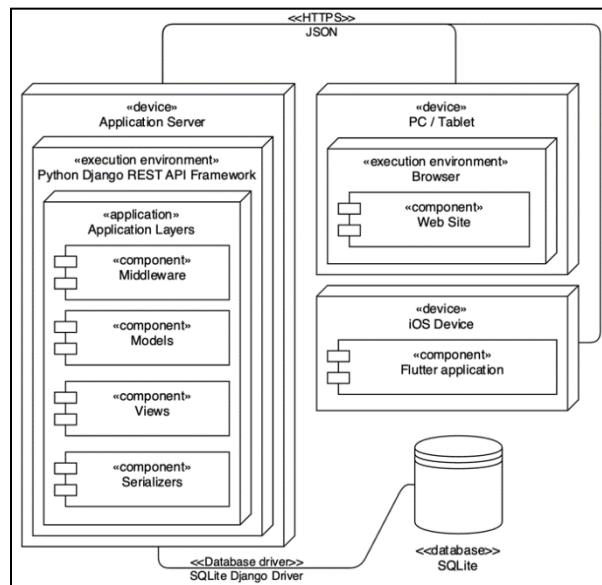


Рис. 3. Діаграма розгортання

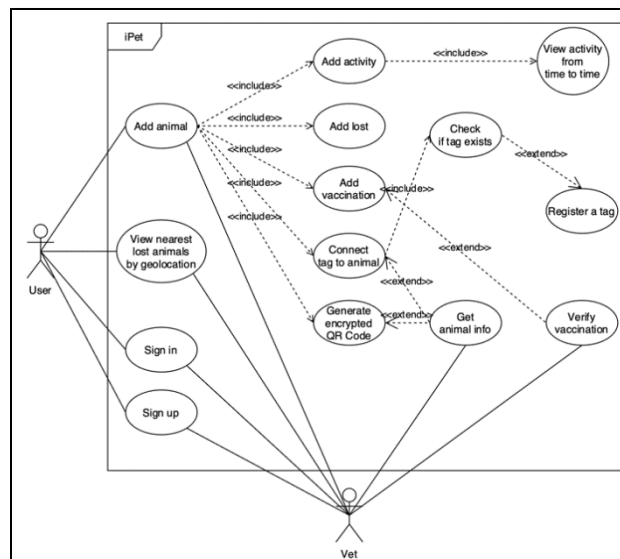


Рис. 4. Діаграма прецедентів

Модулі використовують наступні бібліотеки:

- barcode_scan.dart, використовується для зчитування QR коду камерою смартфона;
- image_picker.dart, використовується для зчитування фотографії з пам'яті смартфона та конвертації пропріетарного формату HEIC у JPEG;
- location.dart, дозволяє зчитувати дані місцезнаходження (широту та довготу) з GPS модуля мобільного пристрою;

Модуль Authentication містить компоненти авторизації за реєстрації sign-in.dart та sign-up.dart.

Модуль Tabs містить наступні компоненти:

- Profile, містить модуль tab.dart, який відповідає за відображення та обробку даних користувача;
- My Animals, містить tab.dart (вивод тварин користувача), row.dart (рядок списку), view_animal.dart (сторінка перегляду), new_animal.dart (створення) та edit_animal.dart (редагування тварини), взаємодіє з row.dart завдяки інтерфейсу MyAnimal;
- Lost Animals, містить tab.dart (вивод втрачених тварин, взаємодіє з row.dart завдяки інтерфейсу LostAnimal, який залежить від Owner), row.dart (рядок списку) та view_animal.dart (сторінка перегляду).

Після того, як було розроблено структуру програмного коду мобільного застосунку перейдемо до створення його інтерфейсу. Для мобільного додатку має бути використана бібліотека Cupertino UI. Це означає, що у застосунку використовуватимуться елементи інтерфейсу, розроблені компанією Apple.

Інтерфейс мобільного застосунку має кольорову гаму, яка складається з білого, сірого та синього кольорів. На рисунку 5 наведено інтерфейс сторінки «Мої тварини». На ній відображаються тварини, власником яких є користувач. Користувач має можливість додати нову тварину, відредагувати її дані, видалити запис. Завдяки місцезнаходженню, тварина може бути відмічена як втрачена для відображення в інших користувачів.

На сторінці «Втрачені тварини» відображені втрачені тварини інших користувачів (див. рис. 6).

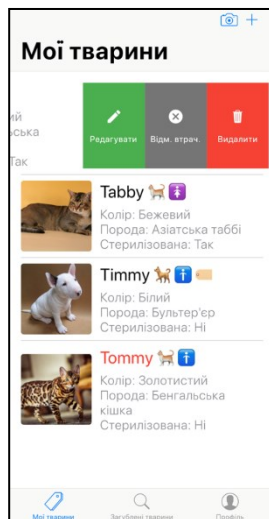


Рис. 5. Інтерфейс сторінки «Мої тварини»

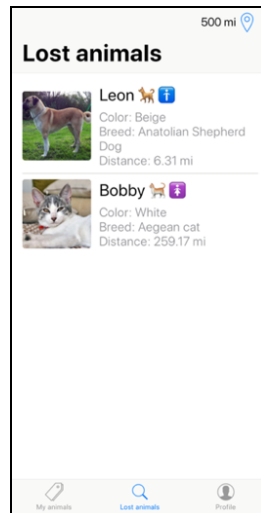


Рис. 6. Інтерфейс сторінки «Втрачені тварини»

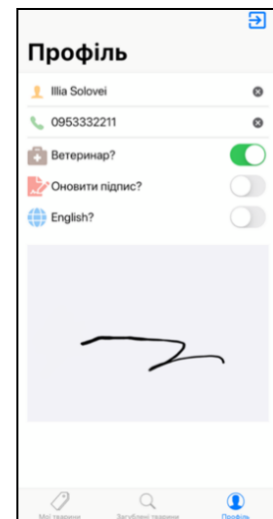


Рис. 7. Інтерфейс сторінки «Профіль»

Останньою вкладкою мобільного додатку є сторінка «Профіль» (див. рис. 7). На ній відображаються дані користувача, які він може редагувати. Також користувач має можливість оновити підпис завдяки сенсорному екрану смартфона.

Як ми бачимо із зображень вище, додаток було інтернаціоналізовано, тобто він підтримує декілька мов і створений як глобальний застосунок, призначений не тільки для внутрішнього використання.

Мобільний додаток є незалежним від веб-клієнту частиною програмної системи, тому дозволяє користувачу авторизуватися та зареєструватися без участі настільного клієнта.

Висновки з даного дослідження

і перспективи подальших розвідок у даному напрямі

У ході дослідження було обрано технології для створення мобільного додатку програмної системи для ведення цифрової картки домашньої тварини. У якості технології розробки було обрано мову програмування Dart, яка використовується для фреймворку Flutter. Розписано структуру коду мобільного додатку. Розроблено інтерфейс з використанням бібліотеки Cupertino UI.

У майбутньому було б доцільно дослідити масштабування мобільного застосунку та його оновлення для підтримки операційної системи Android. Фреймворк Flutter є кросплатформним, тобто таким, що підтримує різні операційні системи, тому подальші дослідження завжди будуть актуальними. Також мобільний додаток можна було б доповнити технологіями машинного зору, наприклад, TensorFlow Lite [12] для автоматичного визначення породи тварини при її додаванні.

Література

1. The complete guide to mobile app development in 2023. K&C. URL: <https://kruschecompany.com/mobile-app-development/>.
2. Mobile App Development: Quick Start Guide. Coursera. URL: <https://www.coursera.org/articles/mobile-app-development>.
3. How to Make an App. Buildfire. URL: <https://buildfire.com/how-to-create-a-mobile-app/>.
4. Saleh M. Beginning iOS 12 Programming with Swift / M. Saleh. — Cupertino: AppCoda Publishing, 2019. — 450 p.
5. Jemerov Y., Isakova S. Kotlin in Action / Y. Jemerov, S. Isakova. — Shelter Island, NY: Manning Publications, 2017. — 360 p.
6. Banks A., Porcello E. Learning React Native / A. Banks, E. Porcello. — Sebastopol: O'Reilly Media, 2020. — 300 p.
7. Pugh K. Flutter in Action / K. Pugh. — Manning Publications, 2019. — 375 p.
8. Green L., Johnson D. "Enhancing User Experience Through Effective Mobile Interface Design", International Journal of Human-Computer Interaction, 2022, Vol. 38, No. 11, pp. 1034-1050.
9. Cupertino (iOS-style) widgets. Flutter. URL: <https://docs.flutter.dev/ui/widgets/cupertino>.
10. Flutter Tutorial. GeeksForGeeks. URL: <https://www.geeksforgeeks.org/flutter-tutorial/>.
11. Massé M. REST API Design Rulebook / M. Massé. — CA: O'Reilly, 2012. — 116 p.
12. Титаренко С., Кириченко І. В. Огляд розвитку технології tensorflow lite під платформу Android // Інформаційні системи та технології: матеріали 9-ї Міжнар. наук.-техн. конф., 17-20 листопада 2020 р. — Харків: Друкарня Мадрид, 2020. — С. 170–173.

References

1. The complete guide to mobile app development in 2023. *K&C*. URL: <https://kruschecompany.com/mobile-app-development/>.
2. Mobile App Development: Quick Start Guide. *Coursera*. URL: <https://www.coursera.org/articles/mobile-app-development>.
3. How to Make an App. *Buildfire*. URL: <https://buildfire.com/how-to-create-a-mobile-app/>.
4. Saleh M. *Beginning iOS 12 Programming with Swift* / M. Saleh. — Cupertino: AppCoda Publishing, 2019. — 450 p.
5. Jemerov Y., Isakova S. *Kotlin in Action* / Y. Jemerov, S. Isakova. — Shelter Island, NY: Manning Publications, 2017. — 360 p.
6. Banks A., Porcello E. *Learning React Native* / A. Banks, E. Porcello. — Sebastopol: O'Reilly Media, 2020. — 300 p.
7. Pugh K. *Flutter in Action* / K. Pugh. — Manning Publications, 2019. — 375 p.
8. Green L., Johnson D. "Enhancing User Experience Through Effective Mobile Interface Design", *International Journal of Human-Computer Interaction*, 2022, Vol. 38, No. 11, pp. 1034-1050.
9. Cupertino (iOS-style) widgets. *Flutter*. URL: <https://docs.flutter.dev/ui/widgets/cupertino>.
10. Flutter Tutorial. *GeeksForGeeks*. URL: <https://www.geeksforgeeks.org/flutter-tutorial/>.
11. Massé M. *REST API Design Rulebook* / M. Massé. — CA: O'Reilly, 2012. — 116 p.
12. Titarenko S., Kyrychenko I. V. Ohliad rozvytku tekhnolohii tensorflow lite pid platformu Android // *Informatsiini systemy ta tekhnolohii: materialy 9-yi Mizhnar. nauk.-tekhn. konf., 17-20 lystopada 2020 r.* – Kharkiv: Drukarnia Madryd, 2020. – S. 170–173.