

ПРОГРАМНИЙ ЗАСІБ ДЛЯ ОБРОБКИ ТА ПЕРЕДАЧІ АНАЛІТИЧНИХ ДАНИХ НА ПЛАТФОРМІ ANDROID

Вінницький національний технічний університет

Анотація

Запропоновано додаток для обробки та передачі аналітичних даних на системі Android. Даний додаток може використовуватися в коді клієнтського додатку та проводити збір аналітичних даних з мінімальною кількістю необхідного коду. Вдосконалено метод збору та обробки аналітичних даних з використанням підходів аспектно-орієнтованого програмування, а також інтеграцією з компонентами системи Android. Створений додаток відповідає стандартам регламенту GDPR та не порушує права користувача на обробку персональних даних.

Ключові слова: Android, Firebase Analytics, аналітика, аспектно-орієнтоване програмування, відслідковування.

Abstract

An application for processing and transmitting analytical data on the Android system has been proposed. This application can be used in the code of the client application and collect analytical data with a minimum amount of required code. The method of collecting and processing analytical data was improved using aspects of aspect-oriented programming, as well as integration with components of the Android system. The created application complies with the standards of the GDPR regulations and does not violate the user's right to personal data processing.

Keywords: Android, Firebase Analytics, analytics, aspect-oriented programming, tracking.

Вступ

По мірі того, як світ вдосконалюється, дані стають ключем до конкурентних переваг. Таким чином здатність компанії конкурувати все більше буде залежати від того, наскільки добре вона може використовувати дані, застосовувати аналітику та впроваджувати нові технології. За даними Міжнародного інституту аналітики, до 2022 року підприємства, які використовують аналітичні дані, отримають 430 млрд. доларів переваг у продуктивності порівняно з конкурентами, які відмовились від цього. Не менш актуально постає дана ситуація і у світі програмного забезпечення. За допомогою інформації про поведінку користувача в додатку, можна розвивати відповідні галузі програми, пріоритезувати тестування, а також відповідним чином керувати рекламною інтеграцією. Операційна система Android є лідером на ринку мобільних операційних систем, що робить дослідження в даній галузі цілком доцільним. Більшість компаній використовує для збору аналітичних даних безкоштовні рішення, що надаються іншими підприємствами. Існує і кілька платних рішень, що надають більший функціонал та полегшують налаштування. Незважаючи на це, існуючі рішення не є ідеальними та мають ряд недоліків, найголовнішими з яких є відсутність інтеграції програмної бібліотеки з компонентами операційної системи та відсутність архітектурних рішень щодо автоматизації формування та збереження аналітичних даних.

Результати дослідження

Головною ідеєю методу, що розробляється, є спрощення реалізації функціоналу збору та обробки аналітичних даних у коді клієнтського додатку. Ці дані формуються у вигляді події з певним набором параметрів. Відповідно, для формування події необхідні назва події, ключі для кожного з параметрів та самі дані. Оскільки в об'єктно-орієнтованому кодї за виконання будь-яких дій відповідають методи, можна зробити висновок, що саме вони будуть джерелом аналітичних подій [1]. У цьому випадку можна автоматизувати даний процес, певним чином позначаючи необхідні методи, використовуючи сигнатуру метода, а саме його назву та назви параметрів, як вихідні дані для генерування події.

Оскільки у системі Android у кожного додатку наявне лише одне активне вікно на якому знаходиться користувач, воно може виступати у вигляді контексту, в рамках якого виконуються дії.

Тому доцільним є збереження назви екрану та його подальше використання у згенерованих подіях, а також відправка відповідної події про зміну екрану.

Контекст екрану може розширятись не лише його назвою, а й певним набором параметрів, який є актуальним для нього. Через це постає необхідність у створенні додаткового сховища параметрів, що зберігало б дані, які відносяться до поточного екрану. Завдяки відслідковуванню зміни екранів, надається можливість очищати це сховище на кожний перехід.

Так як усі події повинні бути унікальними, немає сенсу у обробці даних, що не є релевантними для конкретної події. Для вирішення даного питання можна скористатись динамічною конфігурацією додатку, що завантажуватиметься віддалено. Таким чином, файл конфігурації вказуватиме набір параметрів, який вимагається для події, що дозволить відфільтрувати дані в кодї та, цим самим, економити оперативну пам'ять пристрою та кількість інтернет трафіку, що надсилається.

Дані підходи дозволяють автоматизувати процес збору та обробки аналітичних даних наскільки це можливо, враховуючи необхідність абстрагування від клієнтського коду, в рамках якого буде використовуватись бібліотека. Візуалізація методу зображена на рисунку 1.1.



Рисунок 1.1 – Візуалізація роботи вдосконаленого методу збору й обробки аналітичних даних

Аспектно-орієнтований підхід розглядає програмну систему як набір модулів, кожен із яких відображає певний аспект – мета, особливість функціонування системи. Набір модулів, що утворюють програму, залежить від вимог до програми, особливостей її предметної галузі. При проектуванні програмної системи розробник вибирає модулі так, щоб кожен з них реалізовував певну функціональну вимогу до системи. Однак реалізація деяких вимог до програми часто не може бути локалізована в окремому модулі в рамках процедурного або об'єктно-орієнтованого підходу. В результаті код, що відображає такі аспекти функціонування системи, буде зустрічатися в різних модулях. Традиційні парадигми програмування використовують при проектуванні програми функціональну декомпозицію і дозволяють локалізувати наскрізну функціональність в окремих модулях. Необхідність реалізації наскрізної функціональності наявними засобами веде до того, що деякий компонент містить код, що відображає безліч ортогональних вимог до системи. Це робить такий модуль вузькоспеціалізованим, погіршує можливості його повторного використання та в деяких випадках призводить до дублювання коду. У свою чергу, це викликає підвищення ймовірності внесення помилок, збільшення часу налагодження, знижує якість програми та значною мірою ускладнює її супровід. Аспектно-

орієнтований підхід дозволяє уникнути описаних проблем та покращити загальний дизайн системи, забезпечуючи можливість локалізації наскрізної функціональності у спеціальних модулях – аспектах [2].

Таким чином, за допомогою аспектно-орієнтованого програмування, зручно виділити в окремий модуль логіку щодо збору даних, щоб даний код міг бути застосований в будь-якому місці додатку без його повторного написання. Оскільки операційна система Android працює на віртуальній машині JVM, практично єдиним варіантом реалізації АОП є бібліотека AspectJ, що розроблена для мови Java.

Більшість програм на Android містять кілька екранів, по яких користувач переміщається за допомогою жестів, натискання кнопок і вибору меню. До впровадження компонентів Android Jetpack реалізація навігації в програмі в основному була ручним процесом кодування без легкого способу перегляду та організації потенційно складних шляхів навігації. Ця умова була значно покращена завдяки введенню Android Jetpack Navigation Component.

В основному, навігація відноситься до взаємодій, які дозволяють користувачам переміщатися між різними частинами вмісту у додатку. Іншими словами, навігація по програмі, переміщення з одного екрана на інший є абсолютно фундаментальною частиною розробки Android. Раніше це виконувалось за допомогою намірів (Intent) або фрагментних транзакцій, але на сьогодні дані підходи є застарілими та не використовуються. Навігаційний компонент Android Jetpack допомагає реалізувати навігацію, від простих натискань кнопок до більш складних шаблонів, таких як панелі програм і панель навігації. Точніше, компонент навігації – це набір бібліотек, плагінів та інструментів, які спрощують навігацію Android. Крім спрощення налаштування звичайних шаблонів, таких як нижня навігація, компонент обробляє стек, транзакції фрагментів, передачу аргументів, анімацію на основі навігації та глибоке посилення. Найголовніше, він може зібрати всю цю навігаційну інформацію та помістити її в одне візуалізоване розташування додатка за допомогою графіка навігації. Фактично, компонент Navigation включає три основні частини: графік навігації, NavHost і NavController [3].

NavController дозволяє зареєструвати власний лістнер, який викликатиметься на зміну екрану, при цьому передаючи до нього такі дані, як його назва та ідентифікатор. Це дозволить успішно відслідковувати зміну екранів та визначати які екрани мають ігноруватись за отриманим ідентифікатором.

Висновки

Таким чином, запропонований додаток для обробки та передачі аналітичних даних на системі Android може використовуватися в коді клієнтського додатку та проводити збір аналітичних даних з мінімальною кількістю необхідного коду. При розробці було вдосконалено метод збору та обробки аналітичних даних з використанням підходів аспектно-орієнтованого програмування, а також інтеграцією з компонентами системи Android. Створений додаток відповідає стандартам регламенту GDPR та не порушує права користувача на обробку персональних даних.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Эккель Б. Философия Java. Спб. : Питер, 2018. 1168 с.
2. Aspect Oriented Programming with Spring. URL: <https://docs.spring.io/spring-framework/docs/2.5.x/reference/aop.html> (дата звернення: 09.11.2021).
3. Get started with the Navigation component. URL: <https://developer.android.com/guide/navigation/navigation-getting-started> (дата звернення: 15.11.2021).

Чурчун Владислав Вікторович — студент групи ІКІ-20М, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, email: entermannn@gmail.com

Войцеховська Олена Валеріївна — кандидат технічних наук, доцент кафедри обчислювальної техніки, Вінницький національний технічний університет, Вінниця

Churchun Vladyslav V., — student of group ІКІ-20М, Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: entermannn@gmail.com

Voytsekhovska Olena V. — PhD, Assistant Professor of the Computer Techniques Department, Vinnytsia National Technical University, Vinnytsia.