

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління
(повна назва)

Кафедра Автоматизації проектування обчислювальної техніки
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти другий (магістерський)
(рівень вищої освіти)

Модель збору та аналізу вебметричних даних з сайтів навчальних закладів
(тема)

Виконав: студент 2 курсу, групи СКСм-21-1
Ломага К. А.
(прізвище, ініціали)

Спеціальність 123 Комп'ютерна інженерія
(код і повна назва спеціальності)

Тип програми освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва освітньої програми)

Керівник роботи Литвинова Є.І.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

Чумаченко С.В.
(прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерної інженерії та управління

Кафедра Автоматизації проектування обчислювальної техніки


Рівень вищої освіти другий (магістерський)

Спеціальність 123 Комп'ютерна інженерія
(шифр і назва)

Тип програми Освітньо-професійна
(освітньо-професійна або освітньо-наукова)

Освітня програма Спеціалізовані комп'ютерні системи
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри 
(підпис)

« » 2022 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Ломага Костянтину Альбертовичу
(прізвище, ім'я, по батькові)

Тема роботи (проекту) Модель збору та аналізу вебметричних даних з сайтів навчальних закладів

The model for collecting and analyzing webometric data from the sites of educational institutions

затверджена наказом по університету від «14» 11 2022 р. № 1478 Ст

2. Термін подання студентом роботи 16.12.2022

3. Вихідні дані до роботи (проекту) функція – збирання та аналіз вебметричних даних з сайтів навчальних закладів; діаграми вебметричної аналітики; сервіси для збору вебметричних даних; середовище Android Studio; Мова програмування Kotlin

4. Перелік питань, що потрібно опрацювати у роботі

Вступ; Огляд літератури за темою дослідження; Вибір моделі збору та аналізу вебметричних даних; Розробка архітектури мобільного застосунку; Розробка програмного забезпечення; Тестування мобільного застосунку; Висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних

ілюстрацій (слайдів) 16 слайдів

6. Консультанти розділів роботи (проекту)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання	
		підпис	дата

7. Дата видачі завдання 01.09.2022

КАЛЕНДАРНИЙ ПЛАН

	Назва етапів роботи (проекту)	Термін виконання етапів проекту (роботи)	Примітка
1	Отримання завдання на кваліфікаційну роботу	01.09.2022 - 05.09.2022	виконано
2	Огляд літератури за темою дослідження	05.09.2022 - 30.09.2022	виконано
3	Вибір моделі збору та аналізу вебметричних даних	01.10.2022 - 15.10.2022	виконано
4	Розробка архітектури мобільного застосунку	15.10.2022 - 05.11.2022	виконано
5	Розробка програмного забезпечення	05.11.2022 - 20.11.2022	виконано
6	Тестування мобільного застосунку	20.11.2022 - 01.12.2022	виконано
7	Оформлення роботи	01.12.2022 - 15.12.2022	виконано
8	Представлення роботи до захисту	15.12.2022 - 25.12.2022	виконано

Студент _____



(підпис)

Керівник роботи (проекту) _____



(підпис)

Литвинова Є.І.

(посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка до кваліфікаційної роботи: 66 с., 28 рис., 14 джерел посилання.

GOOGLE ANALYTICS, KOTLIN, ANDROID STUDIO, ANDROID, МОБІЛЬНИЙ ЗАСТОСУНОК, ВЕБ-АНАЛІТИКА, ГРАФІК ДАННИХ, АНАЛІЗ ВЕБ-ДАННИХ, ВЕБОМЕТРИЧНИЙ АНАЛІЗ

Метою кваліфікаційної роботи є підвищення продуктивності засобів вебметричного аналізу веб-сайтів навчальних закладів за рахунок автоматизації отримання вебметричних даних.

Задачі кваліфікаційної роботи: аналіз існуючих засобів отримання вебметричних даних сайтів вищих навчальних закладів; вибір моделі збору та аналізу вебметричних даних; розробка архітектури мобільного застосунку; розробка програмного забезпечення; тестування мобільного застосунку.

Мобільний застосунок реалізований на базі операційної системи Android. Програмна реалізація алгоритму автоматизованого отримання вебметричних даних здійснена мовою програмування Kotlin у середовищі розробки Android Studio.

ABSTRACT

The explanatory note contains: 66 pages, 28 figures, 14 references.

GOOGLE ANALYTICS, KOTLIN, ANDROID STUDIO, ANDROID, MOBILE APP, WEB ANALYTICS, DATA GRAPH, WEB DATA ANALYSIS, WEBOMETRICS ANALYSIS.

The purpose of the qualification work is to increase the productivity of the means of webometric analysis of the websites of educational institutions due to the automation of obtaining webometric data.

Tasks of the qualification work: analysis of existing means of obtaining webometric data of the sites of higher educational institutions; choosing a model for collecting and analyzing webometric data; development of mobile application architecture; software development; mobile application testing.

The mobile application is implemented on the basis of the Android operating system. The software implementation of the algorithm for the automated acquisition of webometric data is carried out using the Kotlin programming language in the Android Studio development environment.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	7
ВСТУП.....	8
1 ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ ДОСЛІДЖЕННЯ.....	10
1.1 Дані до аналізу	21
1.2 ВебOMETричний аналіз.....	23
1.3 Постановка завдання	33
2 ВИБІР МОДЕЛІ ЗБОРУ ТА АНАЛІЗУ ВЕБОМЕТРИЧНИХ ДАНИХ.....	34
3 РОЗРОБКА АРХІТЕКТУРИ МОБІЛЬНОГО ЗАСТОСУНКУ.....	39
3.1 Мова програмування.....	39
3.2 Середовище розробки.....	40
3.3 Організація клієнт-серверної взаємодії зі сторони мобільного додатк.....	42
3.4 Firebase Cloud Storage.....	43
4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	45
4.1 Структура головного екрану	47
4.2 Структура екрану порівняльної аналітики BarChart та PieChart.....	50
4.3 Структура екрану порівняльної аналітики RadarChart.....	56
5 ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКА	59
ВИСНОВКИ.....	64
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	65
ДОДАТОК А. Графічна частина кваліфікаційної роботи.....	67
ДОДАТОК Б. Відомості кваліфікаційної роботи.....	75

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ І ТЕРМІНІВ

API – описання методів за допомогою яких одна програма може здійснювати комунікацію з іншою (анг., Application Programming Interface);

IT – інформаційні технології (анг., Information Technologies);

ПЗ – програмне забезпечення;

GA - Google Analytics

ВСТУП

Інтернет - це одна з небагатьох сфер нашого життя, де фактично все піддається вимірюванню. Сайт - це не формальне визначення для декількох сторінок в мережі. В наш час сайт - це, перш за все, інструмент вирішення задач. Тому у сучасному комп'ютеризованому світі для просування тієї чи іншої продукції, послуги або ж інформації за допомогою Інтернету, величезне значення має аналіз сайту. Він призначений, перш за все, для отримання достовірної інформації про стан інтернет-майданчика в пошукових системах.

До того ж, з часом застаріває все і сайти не є виключенням. Недостатньо просто зробити сучасний та оригінальний ресурс, його необхідно постійно поновлювати, враховуючи нові технології, нові тенденції в дизайні та нововведення конкурентів.

Якісний аналіз та аудит сайту - необхідний елемент при оптимізації. Взагалі варто проводити його ще на початковій стадії впровадження в Інтернет, щоб ресурс завжди приносив потрібний для його власника результат. Такий аналіз – це завжди великий комплекс досліджень, що охоплює як оцінки дизайну, графіки та їх взаємодії, так і оцінки технічної складової

В залежності від типу оцінки, яку потрібно отримати, існує декілька методик аналізу сайтів, це: експертний аналіз, вебометричний аналіз та веб-аналітика. Кожен з них відрізняється один від одного метою дослідження: експертне оцінювання використовується для прийняття рішення (вибору) на основі думки фахівців (експертів); вебометричне – для складення рейтингу сайтів спорідненої тематики; веб аналітика – в загальному для покращення і оптимізації ресурсу та пошуку необхідного розширення функціональних можливостей.

Метою кваліфікаційної роботи є підвищення продуктивності засобів вебометричного аналізу веб-сайтів навчальних закладів за рахунок автоматизації отримання вебометричних даних.

Задачі кваліфікаційної роботи: аналіз існуючих засобів отримання вебметричних даних сайтів вищих навчальних закладів; вибір моделі збору та аналізу вебметричних даних; розробка архітектури мобільного застосунку; розробка програмного забезпечення; тестування мобільного застосунку.

1 ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ ДОСЛІДЖЕННЯ

1.1 Дані до аналізу

Актуальний та грамотний аналіз необхідний як і початківцям в сфері веббізнесу, так і тим, хто хоче підвищити відвідуваність ресурсу, просунути в рейтингу пошуковиків, прибрати надлишковий вміст чи посилання, підвищити рівень usability (зручності використання). Крім того, варто розуміти, що це не тільки дослідження безпосередньо веб-ресурсу, його релевантності, рейтингу та актуальності вмісту. Це також вивчення бізнес-процесів, тобто специфіки конкретної області ринку.

В результаті аналізу спеціалісти шукають всі можливі помилки, надають не тільки повний звіт, але і рекомендації по оптимізації, допомагають виправити застарілі технології та забезпечити безпеку [1].

Етапи проведення прикладу комплексного аналізу веб-ресурсу схематично зображені на рис. 1.1.

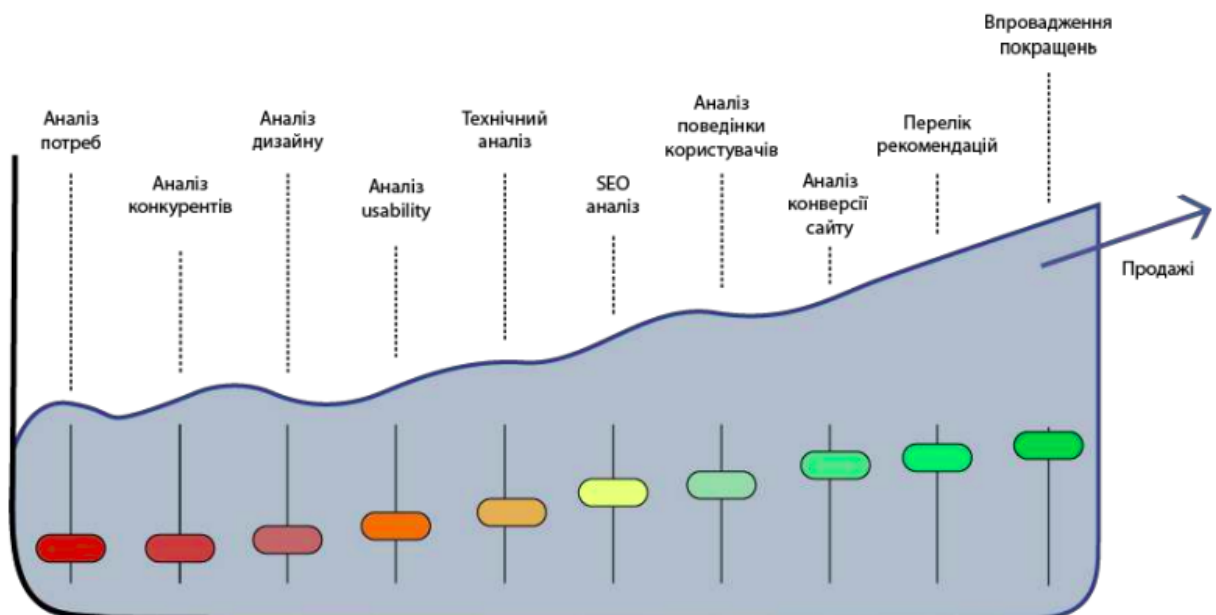


Рисунок 1.1 – Етапи проведення комплексного аналізу сайту комерційної фірми

Розглянемо детальніше ці етапи:

- вимоги – дослідження специфіки бізнесу, вимог клієнта, особливостей цільової аудиторії;
- конкуренти – порівняння позицій, наповнення, usability та оформлення сайту з аналогічними в даній області;
- дизайн – зовнішній вигляд;
- Usability – наскільки зрозумілий та зручний інтерфейс та навігація по сайту;
- технічний аналіз – код;
- SEO-аналіз – перевірка семантичного ядра, структури посилань;
- користувачі – аналіз внутрішньої статистики, переходів за посиланнями, часу, проведеному на сайт;
- аналіз конверсії – відношення кількості покупців до кількості відвідувачів;
- складання звіту та рекомендацій;
- оптимізація у відповідності з результатами перевірки [2].

Професіонали досліджують сайт за допомогою багатьох різних інструментів (рис. 1.2). Головним джерелом знань про те, що відбувається на сайті та про присутність помилок є статистична інформація. Звичайної статистики, яка по замовчуванню надається власниками недостатньо, тому існують спеціальні інструменти, розроблені саме для спеціалістів, за допомогою яких можна дізнатися, як працюють конкуренти, за якими ключами і як часто шукають конкретні товари та послуги.

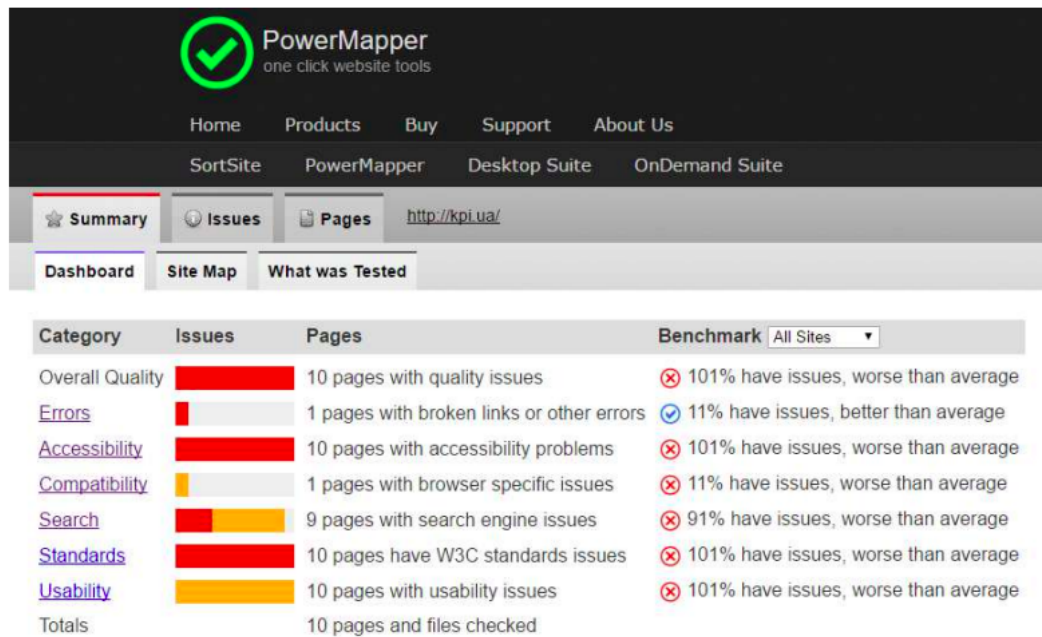


Рисунок 1.2 - Онлайн-сервіс PowerMapper для різностороннього та повного аналізу

Google Analytics (GA) – безкоштовний сервіс від Google [3] для створення детальної статистики відвідувачів веб-сайтів. Дозволяє оцінити не лише оцінити кількість продажів і конверсій, але й пропонує свіжий погляд на те, як відвідувачі використовують сайт, виявити головні джерела трафіку та що робити, щоб вони поверталися на нього (рис. 1.3).

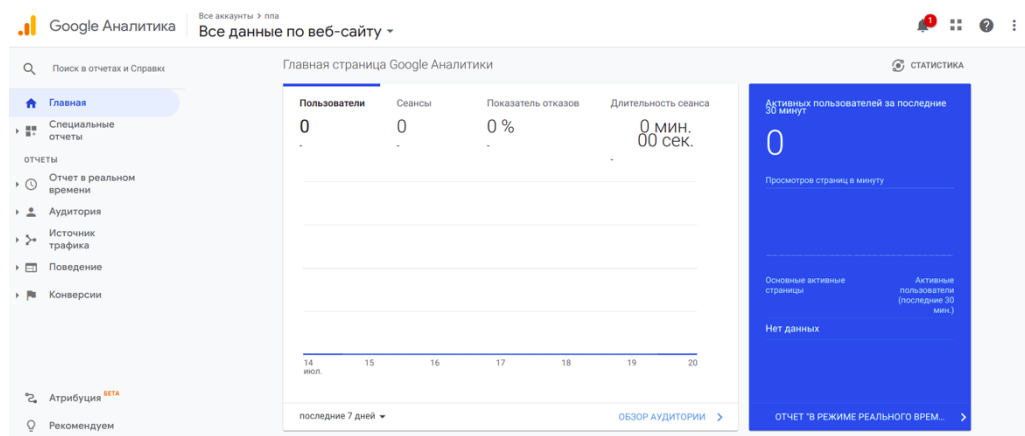


Рисунок 1.3 – Інформаційна панель платформи GA

Особливу увагу варто приділяти швидкості сайту [4], середню вартість замовлення та коефіцієнт конверсії (для підприємств, що займаються продажем товарів чи послуг).

Щоб відразу побачити, що працює ефективно, а що – ні, можна відстежувати активність на своєму сайті (рис. 2.4).



Рисунок 1.4 – Панель огляду активності на платформі GA

Список функцій GA:

- інструменти для аналізу. GA створено на основі потужної, легкої у використанні платформи звітування, щоб можна було вирішувати, які дані переглядати, і налаштовувати свої звіти усього кількома кліками;
- аналіз вмісту. Звіти про вміст допомагають виявити ефективні частини веб-сайту й популярні сторінки, щоб можна було створити кращі умови для клієнтів;
- соціальний аналіз. Мережа – це соціальне місце, а Google Analytics оцінює ефективність кожної з програм, пов'язаних із соціальними засобами комунікації. Можна проаналізувати, як відвідувачі взаємодіють із функціями обміну вмістом на сайті (наприклад, кнопка +1 у Google) і вмістом у соціальних мережах;
- аналіз мобільних пристроїв. Google Analytics допомагає оцінити вплив мобільної мережі на бізнес. Окрім того, якщо створювати

програми для мобільних пристроїв, Google Analytics пропонує пакети з розробки програмного забезпечення для пристроїв на платформі iOS і Android, щоб користувачі могли оцінити, як використовується програма;

- аналіз конверсій. Ряд функцій Google Analytics допоможе дізнатися, скільки клієнтів сайт приваблює, скільки продається товарів та як користувачі взаємодіють із сайтом;
- аналітика рекламної діяльності. Отримати якнайбільше від своєї рекламної діяльності можливо за допомогою даних про ефективність медійних оголошень, реклами в соціальній мережі, на сайтах для мобільних пристроїв. Активність на веб-сайті можна пов'язати з маркетинговими кампаніями, щоб побачити всю картину та покращити ефективність рекламної діяльності (рис. 1.5)

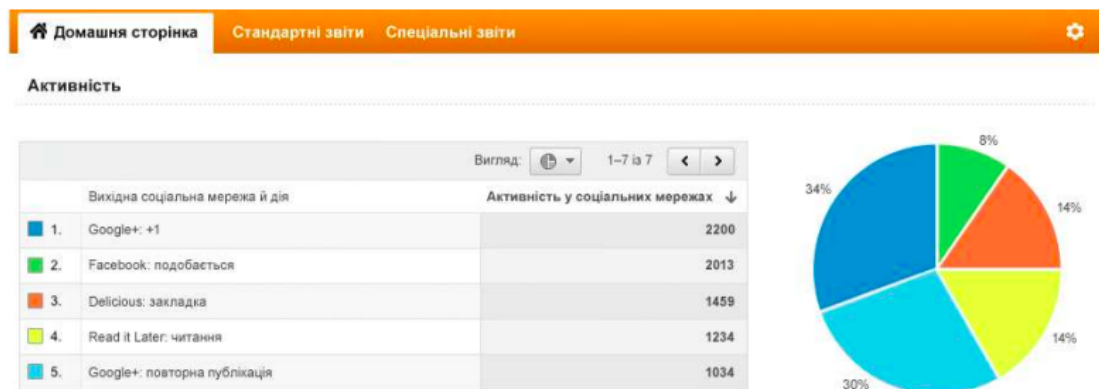


Рисунок 1.5 – Панель соціальної активності на платформі GA

Google Search Console - це веб-служба Google, яка дозволяє веб-майстрам перевіряти статус індексування, пошукові запити, помилки сканування та оптимізувати видимість своїх веб-сайтів (рис. 1.6).

До 20 травня 2015 року сервіс мав назву Google Webmaster Tools. В 2018

році Google представив нову версію пошукової консолі зі змінами в інтерфейсі користувача.

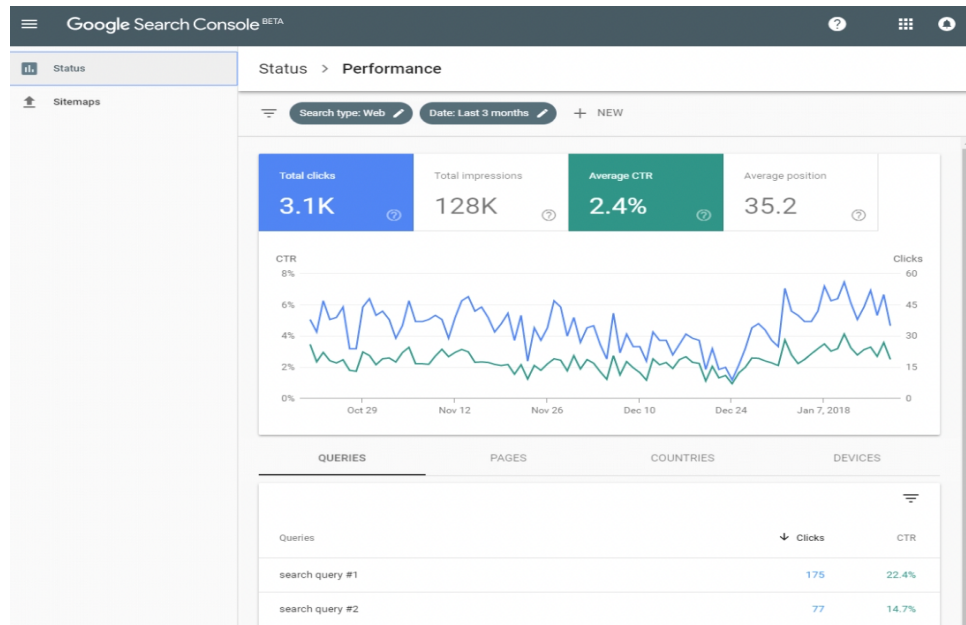


Рисунок 1.6 – Інтерфейс сервісу Google Search Console

Особливістю сервісу є те, що він містить дуже багато інструментів для веб-майстрів. Одні з найбільш відомих інструментів є:

- перевірка карти сайту;
- швидкість сканування та перегляд статистики Googlebot;
- перевірка файлу robots.txt, щоб допомогти виявити сторінки, які можуть бути заблоковані в файлі;
- дозволяє переглядати звіти про швидкість роботи сайту зі звіту про взаємодію з користувачем Chrome;
- надає доступ до API для додавання, зміни та видалення списків і помилок сканування списків;
- виділяє в пошуку Google елементи структурованих даних, які використовуються для збагачення пошукових запитів (випущено в грудні 2012 року під назвою Google Data Highlighter);
- дозволяє перевіряти проблеми з безпекою веб-сайту, якщо вони є.

Google Search Console тісно взаємодіє з пошуковою системою Google.

По суті, вона працює в три етапи:

- crawling - Google шукає в Інтернеті за допомогою автоматизованих програм, які називаються сканерами, шукаючи нові або оновлені сторінки. Google зберігає ці адреси сторінок (або URL-адреси сторінок) у великому списку, щоб переглянути їх пізніше. Потім, знаходить сторінки багатьма різними методами, але основний метод полягає в переході за посиланнями зі сторінок;
- indexing - Google відвідує сторінки, про які він дізнався за допомогою сканування, і намагається проаналізувати, про що йдеться на кожній сторінці. Google аналізує вміст, зображення та відеофайли на сторінці, намагаючись зрозуміти, про що йдеться на сторінці. Ця інформація зберігається в індексі Google , величезній базі даних;
- serving search results - коли користувач виконує пошук Google, Google намагається визначити результати найвищої якості. «Найкращі» результати залежать від багатьох факторів, у тому числі таких факторів, як місцезнаходження користувача, мова, пристрій (комп'ютер або телефон) і попередні запити.

Google Scholar - пошукова система, яка індексує повний текст наукових публікацій всіх форматів і дисциплін.

У листопаді 2004 року Google випустив першу бета-версію сервісу. Засновниками сервісу є Алекс Верстак та Анураг Ачарья.

У 2007 році Ачарья оголосив, що Google Scholar почав програму з оцифрування та хостингу журнальних статей за угодою з видавцями, окремо від Google Books, чії скани старих журналів не включають метадані, необхідні для пошуку конкретних статей у конкретних областях.

Google Scholar дозволяє здійснювати пошук цифрової або фізичної копії статей, онлайн або в бібліотеках. Результати пошуку генеруються з

використанням посилань з повнотекстових журнальних статей, технічних звітів, дисертацій, книг та інших документів, у тому числі обраних веб сторінок.

Scopus - бібліографічна і реферативна база даних та інструмент для відстеження цитування статей, опублікованих у наукових виданнях (рис. 1.7).



Рисунок 1.7 – Інтерфейс сервісу Scopus

Scopus базується видавничою корпорацією Elsevier. Сама ж корпорація є найбільшою у світі універсальною реферативною базою даних з можливостями відстеження наукової цитованості публікацій. Розробниками корпорації було заявлено, що база даних корпорації стане найбільш повним та вичерпним ресурсом для пошуку наукової літератури.

Станом на середину 2009 р. SCOPUS включала 38 млн записів наукових публікацій, в тому числі, 19 млн записів ресурсів, опублікованих після 1996 р.

Класифікаційна система SciVerse Scopus включає такі тематичні розділи:

- фізичні науки (У 2012 році становило 32%, в 2017 році - 41%);

- медичні науки (У 2012 році - 31% , в 2017 році - 40%);
- науки про життя (У 2012 році -20%, в 2017 році 24%);
- соціогуманітарні науки (У 2012 році - 17%, в 2017 році - 12%).

У 2009 році, в Україні почали з'являтися проекти щодо використання науко метричної бази даних Scopus для оцінки наукового потенціалу держави в цілому та окремих суб'єктів наукової діяльності в Україні на рівні вищих органів державної влади.

Універсальна аналітика (Universal Analytics) від GA дозволяє скористатися сегментуванням.

Сегмент – це підмножина даних Analytics. Наприклад, можна об'єднати в один сегмент усіх користувачів із певної країни або міста, а в інший – користувачів, які купують визначену групу товарів або відвідують відповідні сторінки сайту.

За допомогою сегментів можна виділяти й аналізувати такі окремі підмножини даних, щоб вивчати тенденції у своїй комерційній діяльності та відповідно реагувати на них. Наприклад, якщо в певному географічному регіоні обсяг продажів товарів скоротився порівняно з попередніми періодами, можна перевірити, чи не пропонує інша конкурентна компанія такі самі товари за нижчими цінами. Якщо так, можна запропонувати вигідну знижку, щоб повернути своїх попередніх клієнтів.

На основі сегментів можна також створювати аудиторії. Наприклад, можна створити сегмент користувачів, які відвідують сторінки сайту з чоловічим одягом, і націлити на них компанію ремаркетингу, що рекламує нові товари на цих сторінках.

В Google Analytics для збору інформації про відвідувачів використовуються 5 різних cookies, кожен з яких містить інформацію про відвідувача: ключове слово, джерело трафіка, назва кампанії, номер сторінки в сесії на сайті та деякі інші показники. Кожен раз, коли відвідувач переходить на сторінку сайту, код відслідковування Google Analytics зчитує цей масив

інформації з cookies і передає її на сервер збору даних.

В Universal Analytics дещо інший підхід до виконання тієї ж задачі. В браузері відвідувача зберігається тільки одна cookie, яка містить в собі фактично один єдиний унікальний ідентифікаційний номер відвідувача сайту (client ID). Оновлений код відслідковування зчитує цю cookie і передає на сервери Google не увесь великий масив показників, перерахований в попередньому абзаці, а лише унікальний номер відвідувача. В чому різниця? На сервери Universal Analytics передається значно менший об'єм даних. Далі Google вже на своїх серверах підраховує, яка ця сторінка по рахунку в користувацькій сесії, чи був відвідувач на сайті раніше, чи присвоєна цьому відвідувачу яка-небудь користувацька змінна, чи ні і так далі. За запевненнями Google повний перехід на Universal Analytics дозволить на 5% збільшити середню швидкість роботи світової мережі (враховуючи той факт, що на 80% всіх сайтів в світі встановлений Google Analytics). Маючи доступ до унікального ID відвідувача, стає можливим кросплатформне відстеження одного і того ж самого відвідувача. Якщо раніше, відвідувач, переходячи на сайт спочатку з домашнього комп'ютера, а потім, наприклад, і мобільного пристрою був фактично для сайту відразу двома відвідувачами, то зараз у нас з'являється можливість в той момент, коли відвідувач авторизується на сайті, повідомити системі аналітики, що на сайт повернувся той самий відвідувач и прив'язати всю інформацію про інше відвідування до того ж відвідувача.

Статистичні дані – лише одна сторона медалі. Величезна маса інформації знаходиться у відвідувачів. Потрібно зрозуміти, що потрібно клієнтам, як вони оцінюються рівень обслуговування, чого їм не вистачає.

Основні джерела необхідної інформації:

- служба підтримки. Деяким аналогом служить поле пошуку. Варто звернути увагу, якщо відвідувачі часто шукають функціональні елементи, такі як «допомога», «контакти» та ін. за допомогою пошуку;

- звернення до клієнтів напямую (в крайньому випадку);
- створення опитування (безпосередньо на сайті після покупки чи перед виходом з ресурсу, при повторному відвідуванні сайту, на цільовій сторінці);
- інструменти.

Схеми переходів (рис. 1.7) та теплові карти забезпечать інформацією, яка допоможе зрозуміти поведінку відвідувачів на сайті. Що саме приваблює їх увагу, які об'єкти мають першочергове значення, що не потрапляє в їх поле уваги.

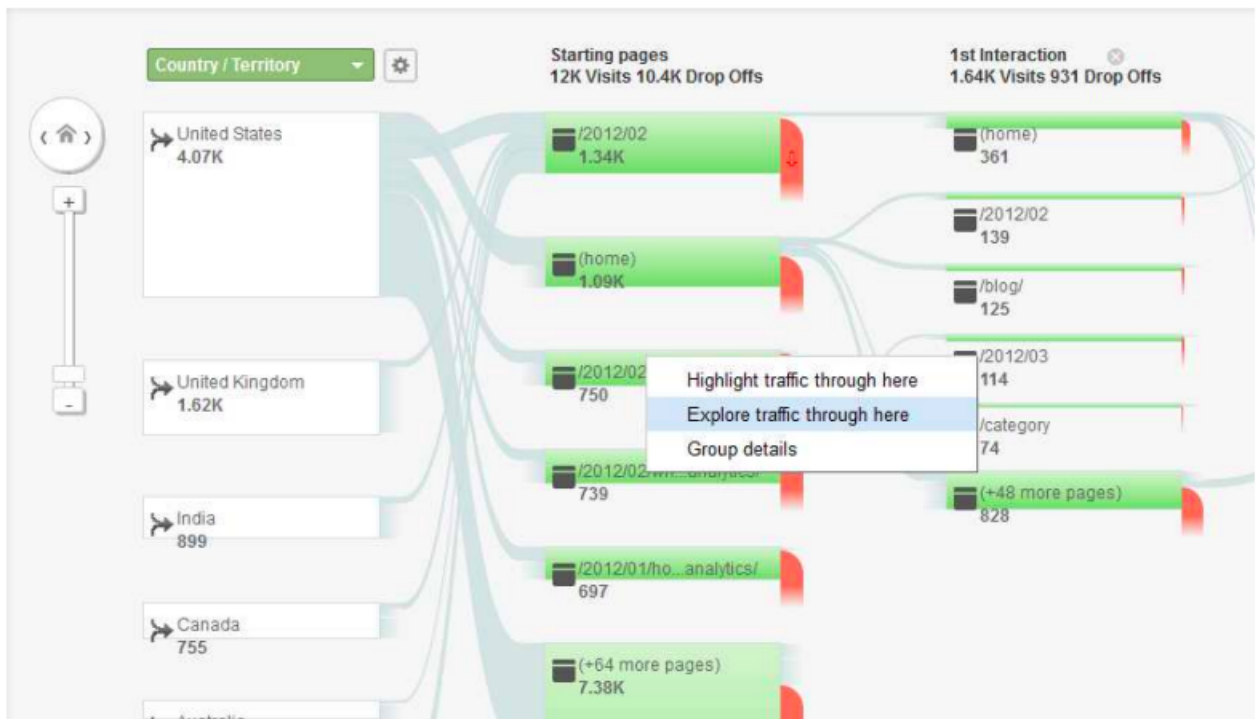


Рисунок 1.7 – Карта переходів від GA

Для отримання такої інформації можна користуватися сервісами Google чи інших сервісів, які надають схожі послуги. В основі таких карт лежить відслідковування рухів миші користувача під час відвідування сайту.

Візуалізація даних забезпечує якісний аналіз плюсів та мінусів сайту. До того ж, можна налаштувати ці інструменти таким чином, щоб отримувати саме ту інформацію і від того типу користувачів, які потрібні.

- usability тестування

Перед таким тестуванням ставиться задача оцінки зручності та зрозумілості інтерфейсу сайту. Основні типи тестування:

- Якісне тестування спрямоване на оцінку поведінки користувача. Дає можливість перевірити дієздатність елементів і ступінь їх зручності для користувача;
- порівняльне тестування – застосовується для порівняння ключових показників по сайту і сайтам конкурентів;
- крос-культурне тестування дозволяє оцінити наскільки продукт адаптований для зарубіжного користувача;

1.2 ВебOMETричний аналіз

ВебOMETричний аналіз проводиться з наступними цілями:

- розробка дієвих методів аналізу та синтезу існуючих систем;
- аналітична робота по збору статистичних даних на основі вебсередовища;
- збір інформації щодо функціонування веб-ресурсу;
- проведення аналітичних досліджень будь-якої складності (статистичний аналіз, аналіз гіперпосилань, прогнозування);
- забезпечення представлення у тематичних та загальних вебOMETричних рейтингах, підвищення позицій у рейтингах;
- дослідження сучасних методик та технологій вебOMETричного аналізу з метою застосування у практиці розбудови веб-системи;
- розробка практичних рекомендацій щодо покращення вебOMETричних показників веб-ресурсів;
- аналіз методик світових вебOMETричних рейтингів з метою розроблення методичних рекомендацій щодо ефективного

- представлення вебресурсів у них;
- розроблення методик складання вебметричних рейтингів;
- проведення тренінгів та семінарів щодо будь-яких веб-орієнтованих питань, можливість організації корпоративного дистанційного навчання;
- консультування з питань веб-аналітики, вебметрики, представлення у міжнародних вебметричних рейтингах та інші види консультування.

Наразі існують три глобальних вебметричних рейтингів вищих навчальних закладів, які є досить популярними:

- webometrics Ranking of World Universities;
- 4 International Colleagues Universities (4ICU);
- Edu Route.

По-перше, вони мають найбільший рівень публічності, адже всі джерела, за якими проводяться виміри, є публічними. По-друге, вебметричні рейтинги максимально орієнтовані на незалежне об'єктивне оцінювання на відміну від тих, що ґрунтуються на формалізованій статистичній інформації, яка, як правило, далеко не повною мірою відображає якісні характеристики діяльності. До речі, перевага оцінювання результату порівняно з оцінюванням вхідних ресурсів є однією з вимог Берлінських принципів ранжування вищих навчальних закладів. Нарешті, у XXI ст. web-простір все більш ставатиме відображенням усіх сфер діяльності вищих навчальних закладів, що робить його навіть певною альтернативою класичним джерелам інформації про вищих навчальних закладів. Через зворотний ефект вебметричні рейтинги також виконують дві важливі функції: вимагаючи якомога більшого представлення діяльності у web-просторі, вони впливають на розвиток інноваційної складової діяльності вищих навчальних закладів, її ІТ-компоненти та, ураховуючи відкритий доступ – на якість навчальних і наукових матеріалів, сприяють підвищенню активності в міжнародному

науково-освітньому просторі.

Об'єктами рейтингування у внутрішньо інституційних вебметричних рейтингах мають бути під домени структурних підрозділів у межах головного домена навчального закладу. На жаль, серед українських вищих навчальних закладів поширена практика використання структурними підрозділами неінституціональних доменів. Це, безумовно, послаблює їхні позиції в глобальних вебметричних рейтингах. Тому при побудові внутрішньо-інституційних вебметричних рейтингів обов'язковою умовою має бути приналежність домену структурного підрозділу головному домену навчального закладу.

Необхідно розглянути основні вимоги до вебметричної моделі даних та їхнє відображення в запропонованій структурі моделі вебметричного рейтингу.

В табл. 1.1 вказані основні індикатори, що входять до глобальних вебметричних рейтингів, тобто відповідають умові глобальності.

Отже, модель вебметричного рейтингу структурних підрозділів вищих навчальних закладів повинна відповідати таким умовам:

- відповідність методикам глобальних вебметричних рейтингів (умова глобальності);
- урахування національних особливостей науково-освітньої діяльності (умова локалізації);
- об'єкти рейтингування мають бути розділені на групи за ознаками розміру, місії тощо (умова багатомірності);
- органічне взаємодоповнення з іншими складовими внутрішньо інституційних рейтингів, виключення при цьому зайвого дублювання тощо (умова інтеграції);
- збір вебметричних даних і розрахунок рейтингових значень мають бути максимально автоматизованими.

Таблиця 1.1 - Можливі оцінки індикаторів вебметричного рейтингу структурних підрозділів навчального закладу і провайдери вебметричних даних.

№ з/п	Найменування оцінки	Вага, %	Провайдер даних
1. Оцінки індикаторів масштабу представлення домена у web-просторі			
1.1	Кількість наявних web-сторінок	10	google.com
1.2	Кількість наявних файлів для завантаження (форматів pdf, doc, docx, ppt, pptx, ps, eps)	10	google.com
1.3	Кількість наявних наукових робіт	20	scholar.google.com
2. Оцінки індикаторів популярності домена у web-просторі			
2.1	Кількість унікальних візитів протягом звітного періоду	10	інституціональні сервери
2.2	Місце у світі за рейтингом <i>Alexa Traffic Rank</i>	10	alexa.com
3. Оцінки індикаторів авторитетності домена у web-просторі			
3.1	Кількість зовнішніх гіперпосилань, з яких:	5	majesticseo.com
3.1.1	актуальних зовнішніх гіперпосилань	10	majesticseo.com
3.2	Кількість доменів-джерел зовнішніх гіперпосилань, з яких:	5	majesticseo.com
3.2.1	доменів-джерел актуальних зовнішніх гіперпосилань	10	majesticseo.com
3.3	Місце у світі за рейтингом <i>Google PageRank</i>	5	google.com

У науковій літературі ефективність веб-ресурсу трактується по різному, оскільки власники сайтів та компанії з їх просування в Інтернеті вкладають у цей процес різні значення. Але всі вони сходяться до думки, що створення власного веб-ресурсу та Інтернет-інтеграція бізнесу пов'язані з реалізацією і досягненням ринкових цілей підприємства – продажі товарів або послуг та, як наслідок, отримання прибутку.

Статистика показує, що переважна більшість користувачів шукають інформацію в Інтернеті за допомогою пошукових систем та каталогів. За різними даними через пошукові системи на ресурси заходять до 95 % цільових користувачів, 55 % покупок і замовлень здійснюються на сайтах, знайдених через пошукові системи. Від того, на якому місці в результатах цільових запитів пошукових систем знаходиться той чи інший ресурс, залежить

кількість споживачів, що його відвідають.

Постає питання що вважати показником ефективності пошукового просування веб-ресурсу і як оцінити його мережеву видимість?

Закон Р. Меткалфа доводить наявність зв'язку між економічною цінністю застосування ІТ у підприємницькій діяльності та кількістю комунікацій, здійснюваних через Інтернет. Іншими словами, економічний результат веб-ресурсу збільшуватиметься у квадратній залежності від числа його користувачів, тобто числа комунікацій. Інтернет при цьому розглядаються як середовище для обміну інформацією між користувачами через електронні ресурси. Для підприємств сутність цього закону полягає у нарощуванні показників економічної ефективності ІТ для бізнесу за рахунок збільшення числа комунікацій, здійснюваних через електронний ресурс та залучення нових користувачів (табл. 1.2).

Таблиця 1.2 - Визначення індивідуальної та загально мережевої цінності веб-ресурсу

Кількість комунікацій на 1 людину через веб-ресурс	Індивідуальна цінність користувача веб-ресурсом	Загальномережева цінність веб-ресурсу
1	0	0
2	V	$2V = 2 \cdot 1 \cdot V$
3	2V	$6V = 3 \cdot 2 \cdot V$
...
26	25V	$650V = 26 \cdot 25 \cdot V$
N	$(n - 1)V$	$n(n - 1)V = (n^2 - n)V$

У табл. 1.2 показана залежність між кількістю комунікацій через веб-ресурс підприємства та економічною цінністю від підприємницької діяльності на основі ІТ. В основу економічної ефективності Інтернет-комунікацій, зокрема, вебресурсу, покладено два показники – індивідуальну цінність кожного окремого відвідувача і загальномережеву цінність веб-ресурсу для підприємства.

Найпростішим вираженням першої цінності є, коли кожний комунікаційний зв'язок користувача через електронний ресурс завжди приносить йому однакову цінність:

$$V \cdot (n - 1);$$

де V – разова цінність для користувача;

n – загальний розмір мережі (кількість відвідувачів веб-ресурсу);

$(n - 1)$ – число інших учасників, з якими користувач може вступати у комунікацію через веб-ресурс.

Формула демонструє, що індивідуальна цінність для відвідувача веб-ресурсу збільшується лінійно з ростом кількості користувачів на ньому. До числа основних складових, що визначають індивідуальну цінність ми можемо віднести: кількість інформації доступної користувачу на сайті, зручність її організації та розміщення на веб-сторінках ресурсу, доступність та легкість пошуку тощо. Підприємство на основі ІТ виробляє інформацію нової якості, використовуючи яку споживачу легше вирішувати свої проблеми, задовольняти власні потреби, більш раціонально та обґрунтовано приймати рішення.

Не менш важливим показником залежності відвідуваності від кількості комунікацій, здійснюваних через веб-ресурс, є загально мережева цінність, яка являє собою суму всіх індивідуальних цінностей. У цьому випадку значення загально мережевої цінності веб-ресурсу зі збільшенням кількості його відвідувачів зростатиме набагато швидше, ніж індивідуальна цінність кожного користувача. Кожен новий відвідувач отримає від приєднання до мережі Інтернет індивідуальну користь, привносячи своїм приєднанням у неї додаткову цінність та розширюючи можливості наявних користувачів ресурсом.

При цьому, одержувана користь від використання Інтернет-технологій повинна перевищувати витрати, пов'язані зі створенням та обслуговуванням веб-ресурсу, та мати такий вигляд: $V_i \cdot (n - 1) > C$;

Цю вимогу виконати тим легше, чим більше користувачів матиме електронний ресурс (n):

$$V_i = C / (n - 1);$$

Співвідношення на рис. 1.8 показує, що при великому трафіку, а саме збільшенні кількості відвідувачів, ефективність сайту як носія маркетингових комунікацій зростатиме пропорційно, що виправдує розширення бізнес діяльності в електронний простір. Чим більшим відвідувачів заходитиме на сайт, тим вищим буде його економічна цінність. Для окремої компанії ця залежність трансформується в підвищення продуктивності та зростання прибутків.

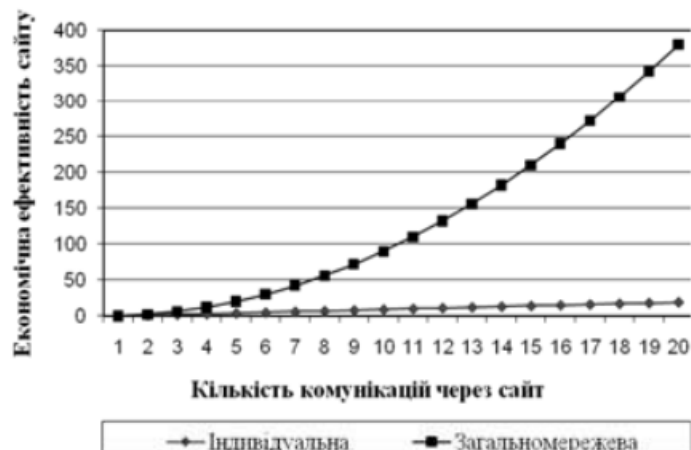


Рисунок 1.8 – Залежність економічної ефективності електронного ресурсу від кількості користувачів, що здійснюють комунікацію через нього.

Важливою складовою кожної веб-орієнтованої системи є її оптимізованість. Оптимізованість — це кількість часу, за який веб-сторінка повністю завантажується. Швидкість завантаження розраховується з моменту, коли користувач натискає на посилання, до моменту, коли система повністю завантажується, включаючи зображення, відео тощо. Швидкість завантаження системи має дуже велике значення, що підтверджує наступна статистика:

- 1 з 4 відвідувачів залишає веб-орієнтовану систему, завантаження

якої займає більше 4 секунд;

- 46 відсотків користувачів не повертаються на неоптимізовані системи;
- сайти з найвищим коефіцієнтом конверсії повністю завантажуються менш ніж за 2 секунди.

Крім цього швидкість завантаження є важливим фактором рейтингу Google. Цей факт було підтверджено у 2018 році, коли в компанії Google швидкість завантаження сторінки зробили одним із найважливіших факторів рейтингу для пошуку з мобільних пристроїв. Збільшення конверсій є кінцевою метою кожного бізнесу. Конверсія — пропорція або залежність числа покупок від кількості відвідувачів. Швидкість завантаження сторінки є вирішальним фактором для прийняття рішення про покупку. Рекомендований час завантаження системи компанією Google становить менше 2 секунд. Для визначення швидкості завантаження системи існують спеціальні інструменти, серед яких слід відзначити наступні:

- page speed insights вказує, наскільки ефективна сторінка, і пропонує оптимізацію;
- lighthouse чудовий інструмент для перевірки продуктивності, доступності та іншого веб-сторінок; Швидкість завантаження може залежати від великої кількості факторів.

Деякі приклади включають типи веб-хостингу та сервера, код і бази даних, які використовує веб-сайт, вибір елементів дизайну, кількість і розмір зображень, відео та файлів на кожній сторінці, версії браузера, поведінку користувачів тощо.

Основний обсяг контенту сайту займають медіафайли. Для вирішення цієї проблеми можна скористатися програмним забезпеченням, яке зменшить обсяг пам'яті, що займають медіа-файли без втрати якості. Іншим способом вирішення такої проблеми є оптимізація таких файлів під час завантаження їх

на сервер. Для цього в кожній мові програмування існують готові модулі. Після оптимізації розмір кожного файлу може зменшитись у декілька разів. Але крім зменшення розміру файлів слід скористатися можливістю асинхронного завантаження.

Асинхронне завантаження — це призупинення завантаження медіа-контенту до моменту, коли не буде завантажено всю гіпертекстову розмітку сторінки. Для цього в гіпертекстовій розмітці існує спеціальний атрибут під назвою `loading`. Ще одним способом оптимізації медіа-контенту є використання вбудованих можливостей серверу. Наступним кроком оптимізації веб-орієнтованої системи є зменшення обсягу коду, який транспортується мережею. Зазвичай для цього використовуються мініфікація цього коду. Мініфікація— це процес з'єднання великої кількості файлів в один із видаленням форматування.

Іншою важливою складовою кожної веб-орієнтованої системи є її пошукова оптимізованість. Пошукова оптимізація веб-орієнтованої системи (англ. *search engine optimization*) — це перелік заходів та дій, що включають в себе редагування наповнення контенту , структури, метою якого є підняття рейтингу системи в результатах пошуку за певними запитами користувачів. Чим вище позиція сайту в результатах пошуку, тим більша ймовірність, що відвідувач перейде на нього з пошукових систем, оскільки люди зазвичай йдуть за першими посиланнями (рис. 1.9).

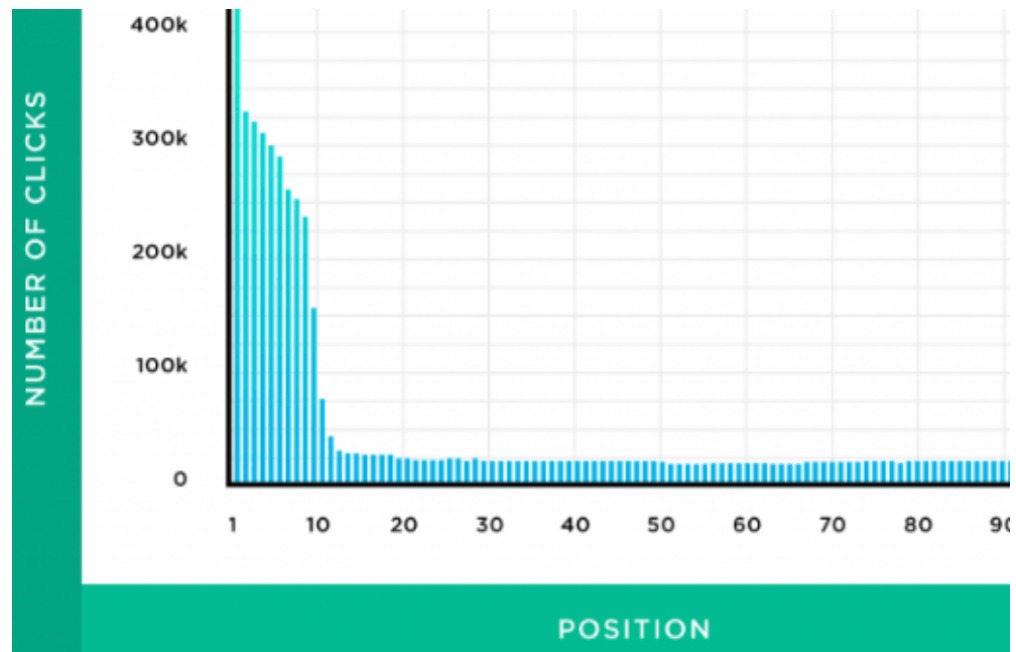


Рисунок 1.9 - Приклад залежності рейтингу сторінки в пошукових запитах з її відвідуваністю.

Пошукова оптимізація з'явилася із появою перших пошукових систем в середині 90-х років 20-го століття. Перші методи пошукової оптимізації передбачали редагування контенту та мета-тегів. Внаслідок таких заходів високі місця у пошуку стали отримувати сайти, котрі не містили корисного змістового навантаження, а лише популярні пошукові запити. У той час пошукувачі надавали великого значення тексту на сторінці та іншим внутрішнім чинникам, якими власники сайтів могли легко маніпулювати. Це призвело до того, що у видачі багатьох пошукувачів перші декілька сторінок зайняли так звані «сайти смітники», що різко знизило якість роботи пошукувачів і спричинило занепад багатьох із них. З появою технології PageRank більшої уваги стали приділяти зовнішнім чинникам, що допомогло Google вийти в лідери пошуку у світовому масштабі, ускладнивши оптимізацію за допомогою одного лише тексту на сайті.

Одними із основних термінів пошукової оптимізації є релевантність та ранжування. Релевантність у інформаційній науці і інформаційному пошуку означає ступінь відповідності знайденого документа або набору документів інформаційним потребам користувача. Ранжування — це процес розміщення

даних у певному порядку за ступенем важливості, значущості. У процесі ранжування сайту пошуковими системами беруться до уваги більш ніж 200 факторів.

Основними факторами, що впливають на видачу результату в пошукових системах є:

- внутрішня оптимізація сторінки;
- технічна оптимізація сайту (використовувана CMS, чистота коду,
- швидкість завантаження сайту, мобільна версія тощо);
- якісні зовнішні посилання на сайт;
- вік сайту;
- контент для людей (ключових слів може в статті не бути, але вона буде на першій позиції через релевантність і цінність);
- наявність ключових слів в адресі сайту.

До факторів, що знижують рейтинг сайту, належать:

- завелика кількість ключових слів у контенті;
- технології, які пошукові машини розглядають як спам.

Біла оптимізація — оптимізаторська робота над ресурсом без застосування офіційно заборонених кожною пошуковою системою методів розкручування ресурсу — без впливу на пошукові алгоритми сайтів. Вона включає в себе роботу над самим сайтом, тобто над його внутрішньою навігацією і вмістом, і роботу з зовнішнім середовищем сайту, тобто просуванням цього сайту шляхом оглядів, прес-релізів, реєстрації в соціальних закладках, партнерських програм тощо із зазначенням посилань на сайт. Слід зазначити, що навіть якщо який-небудь метод оптимізації не є офіційно забороненим, це не означає, що його можна застосовувати.

Сіра оптимізація. До сірої пошукової оптимізації можна віднести додавання великої кількості ключових слів в текст сторінки, часто з втратою читаємості для людини. Проте в деяких випадках при використанні сірої оптимізації допускається помилка під назвою переоптимізація, що призводить до переходу з сірої оптимізації в чорну. При цьому завдання SEO-копірайтера

— написати оригінальний текст таким чином, щоб подібна оптимізація була якомога менш помітна «живому» читачеві. Сіра оптимізація відрізняється від чорної тим, що вона офіційно не заборонена, але її використання все одно може бути розцінене як неприродне завищення популярності сайту. Деякі пошукові системи, наприклад, Google, можуть тимчасово або назавжди заблокувати такий сайт. Тобто, остаточне рішення про те, чи є методи просування законними чи ні, приймає фахівець — модератор пошукової системи, а не програма.

Чорна оптимізація. До чорної оптимізації відносяться всі методи, які суперечать правилам пошукових систем. Серед них можна виділити наступні: використання дорвеїв (сторінок і ресурсів, створених спеціально для роботів пошукових систем, найчастіше з великою кількістю ключових слів на сторінці), прийом під назвою клоакінг (користувачеві віддається одна сторінка, що легко читається, а пошуковому роботу — інша, оптимізована під які-небудь запити), використання прихованого тексту на сторінках сайту.

Таким чином, основним мотивом для приєднання до Інтернету окремої людини є індивідуальна цінність, для підприємства таким спонукальним чинником є загально мережева цінність. Остання також пояснює, чому усе більше компаній прагнуть мати доступ до Світової мережі, адже, чим більшим є її розмір, тим вищою є її загально мережева цінність.

Виходячи з вищесказаного, основним показником ефективності маркетингових комунікацій підприємства на основі ІТ є нарощування відвідуваності через залучення більшої кількості користувачів.

1.4 Постановка завдання

Однією з основних речей вищих навчальних закладів є сайт, на якому відображається уся діяльність університету, кафедри, факультету. Також, сюди відносить обслуговування сайту, наповнення контентом, виявлення помилок, відвідуваність, доступність, а також моніторинг, обробка та збір вебметричних даних в реальному часі.

Об'єкт дослідження у роботі – процес збору вебметричних даних сайтів вищих навчальних закладів з різних доступних сервісів.

Предмет дослідження – моделі, методи та процедури збору, обробки та відображення вебметричних даних сайтів вищих навчальних закладів з використанням мобільних технологій.

Сутність роботи – проектування зручного мобільного застосунку для відображення діаграм даних в рамках отримання та обробки вебметричних моделей даних сайтів навчального закладу.

Мобільний застосунок має забезпечувати отримання, обробку та відображення доступних вебметричних даних сайтів вищих навчальних закладів в реальному часі.

2 ВИБІР МОДЕЛІ ЗБОРУ ТА АНАЛІЗУ ВЕБОМЕТРИЧНИХ ДАНИХ

Вебометричний рейтинг університетів світу (англ. Webometrics ranking of world's universities) — це рейтингова система для університетів світу на основі зведеного показника, який враховує як обсяг веб-вмісту (кількість веб-сторінок і файлів), їх наочність, а також вплив цих веб-публікацій за кількістю вхідних посилань на зовнішні ресурси, з яких вони отримані. Рейтинг публікується лабораторією кіберметрики («Cybermetrics Lab») — науково-дослідною групою Національної дослідницької ради Іспанії («Spanish National Research Council»), яка розташована в Мадриді.

Метою рейтингу є підвищення наявності порталів академічних і науково-дослідних установ в Інтернет-просторі, а також сприяти відкритому доступу публікації їхніх наукових результатів. Рейтинг почав публікуватися в 2004 році й оновлюється щороку в січні і липні. На сьогоднішній день аналізуються понад 20 000 вищих навчальних закладів і визначається їхнє місце відповідно до ступеня представлення своєї діяльності в Інтернеті, за допомогою власної методології оцінки. Можна критикувати цю методологію, але всі провідні вищі навчальні заклади світу знаходяться попереду. Лабораторія кіберметрики використовує кількісні методи і показники, які допомагають оцінити наукову діяльність у веб-середовищі України і світу. Методологію здійснення рейтингу розроблено за методикою, яка базується на Берлінських принципах визначення рейтингу вищих навчальних закладів (Berlin Principles on Ranking of Higher Education Institutions), які розробило ЮНЕСКО.

Webometrics є рейтинговою системою, яка заснована на університетській веб-присутності, видимості і доступу в мережу Інтернет. Центральною гіпотезою цього підходу є те, що веб-присутність є надійним індикатором глобальної продуктивності і престижу університетів. Це є непрямим способом вимірювання всіх місій університету (навчання, дослідження, передавання). Веб-технології є одними з найважливіших

інструментів для наукового спілкування, але ці показники ще дуже рідко використовуються для оцінки науково-дослідної й академічної успішності університетів. Найчастіше вебметричні показники наводяться, щоб показати прихильність установ для веб-публікацій.

Кращі університети світу публікують мільйони сторінок наукових видань, які розроблені десятками відділів і служб, сотнями наукових колективів і тисячами вчених. Постійна присутність в Інтернеті чітко інформується найрізноманітнішими факторами про глобальні якості вишу:

- широку доступність комп'ютерних ресурсів;
- глобальну інтернет-грамотність;
- політику просування демократії і свободи слова;
- конкуренцію за міжнародну видимість;
- підтримку відкритого доступу ініціатив.

Первісною метою рейтингу було залучення дослідників і науковців до веб-публікацій. Підтримка відкритого доступу, електронного доступу до наукових публікацій та інших навчальних матеріалів є основною метою лабораторії кіберметрики. Однак веб-показники є також дуже корисними в цілях рейтингу, оскільки вони не ґрунтуються на кількості відвідувань сторінки або її дизайні, а на світовій продуктивності і видимості університетів. Рейтинг охоплює не тільки формальні (електронні журнали, сховища), а й неофіційні наукові комунікації. Веб-публікації набагато дешевші, ніж друковані видання, але вони так само підтримують високі стандарти якості з огляду експертів. За допомогою Інтернету можна досягти набагато більшої потенційної аудиторії, надаючи доступ до наукових знань дослідникам й установам, розташованим у країнах, що розвиваються, а також третім особам, які зацікавлені у співтоваристві.

У вебметричному рейтингу враховується багато показників і видів діяльності ніж у інших подібних рейтингах, бо він не зосереджений тільки на результатах досліджень. Він також враховує інші критерії, які можуть краще

відображати глобальні якості вченого і науково-дослідних інститутів у всьому світі. Основні показники вебметричного рейтингу висвітлені в табл. 2.1.

Таблиця 2.1 - Визначення індивідуальної та загально мережевої цінності веб-ресурсу

	Індикатор		Зміст	Джерела	Частка
Видимість, 50%	Вплив (Impact)	Зворотні посилання	Кількість зовнішніх посилань	Majestic SEO, Yahoo Site Explorer, Google (Page Rang, Webmaster Tools)	50 %
		Домени	Кількість доменів, з яких здійснюються зворотні посилання.		
Активність, 50%	Присутність (Presence)	Розмір (Size, web-page)	Кількість сторінок, яку надають пошукові ресурси	Google, Yahoo, Live, Search, Exealed	20 %
	Відкритість (Openness)	Мультимедійні файли (Rich Files)	Обсяг присутніх на сайті університету різних форматів файлів: Adobe Acrobat (.pdf), Adobe PostScript (.ps), Microsoft Word (.doc) та Microsoft Powerpoint(.ppt). Зазвичай ці формати автори використовують для написання і публікації своїх робіт. Присутність великої кількості документів у таких форматах на порталі університету свідчить про те, що там зберігаються не тільки адміністративні	Google Scholar, Google Books, Yahoo, Bing, Exealed	15 %

			матеріали, але й наукові публікації		
	Висока якість Excellence	Академія (Scholar)	Google Академія надає інформацію про кількість наукових матеріалів та їхню цитованість для кожного академічного домену. Враховують публікації, звіти та інші академічні матеріали за 5 років	SCImago SIR	15 %
			Кількість сторінок на Scimago.		

За цими вебметричними даними ми можемо відстежити положення сайтів вищих навчальних закладів України. Так на рис. 2.1 зображено вебметричний рейтинг сайтів вищих навчальних закладів України на липень 2022 року. А на рис. 2.2 зображено вебметричний рейтинг сайтів вищих навчальних закладів України на липень 2021 року.

ranking	World Rank	University	Det.	Impact Rank*	Openness Rank*	Excellence Rank*
1	1320	National Taras Shevchenko University of Kyiv / Київський національний університет Тараса Шевченка	→	1999	1458	1504
2	1443	Sumy State University / Сумський державний університет	→	1680	1293	2030
3	1515	National Technical University of Ukraine Kyiv Polytechnic Institute / Національний технічний університет України Київський політехнічний інститут	→	1286	1185	2531
4	1904	National Aviation University (Kyiv International University of Civil Aviation) / Національний авіаційний університет	→	1518	1706	3067
5	2144	Kharkiv National University VN Karazin / Харківський національний університет В Н Каразіна	→	4594	1315	2344
6	2307	Kharkiv National University of Radio Electronics / Харківський національний університет радіоелектроніки	→	3722	1711	2854
7	2569	(1) National Technical University Kharkiv Polytechnical Institute / Национальный технический университет Харьковский политехнический институт	→	4803	1282	3104
8	2664	National University of Life and Environmental Sciences of Ukraine (National Agricultural University) / Національний університет біоресурсів і природокористування України	→	5590	1526	2969
9	2897	Lviv Polytechnic National University / Національний університет Львівська політехніка	→	5194	7619	1474
10	2944	Ternopil National Medical University / Тернопільський Національний Медичний Університет	→	5666	2257	3263

Рисунок 2.1 – Рейтинг сайтів вищих навчальних закладів України на липень 2022 року

Ukraine						
ranking	World Rank 	University	Det.	Impact Rank*	Openness Rank*	Excellence Rank*
1	1152	National Taras Shevchenko University of Kyiv / Київський національний університет Тараса Шевченка		1857	1093	1404
2	1590	National Technical University of Ukraine Kyiv Polytechnic Institute / Національний технічний університет України Київський політехнічний інститут		1321	1111	2851
3	1755	Sumy State University / Сумський державний університет		2963	1238	2461
4	2037	National Aviation University (Kyiv International University of Civil Aviation) / Національний авіаційний університет		1146	1619	3980
5	2331	National Technical University Kharkiv Polytechnical Institute / Национальный технический университет Харьковский политехнический институт		4581	1191	3283
6	2363	National Aerospace University Kharkiv Aviation Institute / Национальный авиационный университет Н.Е. Жуковского		3506	2038	3305
7	2424	National University of Life and Environmental Sciences of Ukraine (National Agricultural University) / Національний університет біоресурсів і природокористування України		4959	1433	3266
8	2471	Dnipro University of Technology / Національний гірничий університет		2626	2524	3723
9	2496	Kharkiv National University of Radio Electronics / Харківський національний університет радіоелектроніки		6916	1660	2727
10	2549	Kharkiv National University VN Karazin / Харківський національний університет В.Н. Каразіна		3459	5118	2213

Рисунок 2.2 – Рейтинг сайтів вищих навчальних закладів України на липень 2021 року

Отже, для дослідження, було обрана загально відома вебметрична модель для отримання, аналізу та обробки вебметричних даних сайтів вищих навчальних закладів. Отримати дані, ми можемо за допомогою перерахованих сервісів вебметричного аналізу.

3.1. Мова програмування.

Kotlin – статично типізована мова розроблена компанією Jet Brains, яка працює поверх віртуальної машини Java. Свою назву мова отримала від острова Котлін у Фінській затоці.

Головну ціль компанія Jet Brains ставила перед собою створити лаконічнішу та безпечнішу мову ніж Java, та простіше ніж Scala. Також позитивною зміною стала швидка компіляція проекту та краща підтримка.

Новину про початок розробки було анонсовано у 2010 році, а публічно представлена у 2011 році. 15 лютого 2016 року відбувся перший офіційний реліз який містив плагін для IDEA.

17 травня 2017 року на конференції Google I/O командою розробників було анонсовано новину, що компанія Google рекомендує Kotlin для розробки під операційну систему Android.

Jet Brains повідомила, що де працює Java– повинен працювати і Kotlin. Тобто використовувати Kotlin можливо і в змішаних проектах, в яких використовується декілька мов програмування. Тобто, мова повністю сумісна з Java, що дозволяє розробникам поступово перейти з Java на Kotlin. Зокрема, в Android мова вбудовується за допомогою Gradle, що дозволяє для існуючого Android-додатки впроваджувати нові функції на Kotlin без переписування програми цілком.

Одним з основних місць застосування Котліна є розробка під Android. Платформа на деякий час застрягла на Java 7 (деякі сучасні мовні функції стали доступними завдяки використанню Retrolambda або інструментарію Jack), а Kotlin вносить багато поліпшень для програмістів, таких як null-безпечність, функції розширення та інфіксна нотація [11]. Завдяки повній сумісності з Java та підтримкою IDE (Android Studio), він покликаний покращити читання коду,

полегшити використання класів Android SDK та загально прискорити розробку.

Користувачі нової мови програмування Kotlin, виділили основні особливості, на відміну від інших мов:

- Extension Functions – це розширюючі методи, які надають можливість розширити клас без наслідування з новою функціональністю [13];
- @Nullable та @NotNull – забезпечує захист від null. Основна та найважливіша перевага від Java. Загалом використовуються при отриманні інформації з джерела, коли невідомо прийде очікуване значення або null.

3.2 Середовище розробки.

Аналізуючи поставлену задачу та методи її вирішення, було вирішено розроблювати додаток, який є гнучким, сучасним та легким у використанні.

Розробку застосунків для Android можна вести мовою Java та Kotlin. Офіційним середовищем розробки є Android Studio, представлене компанією Google в 2013. Android Studio зображена на рис. 1.9.

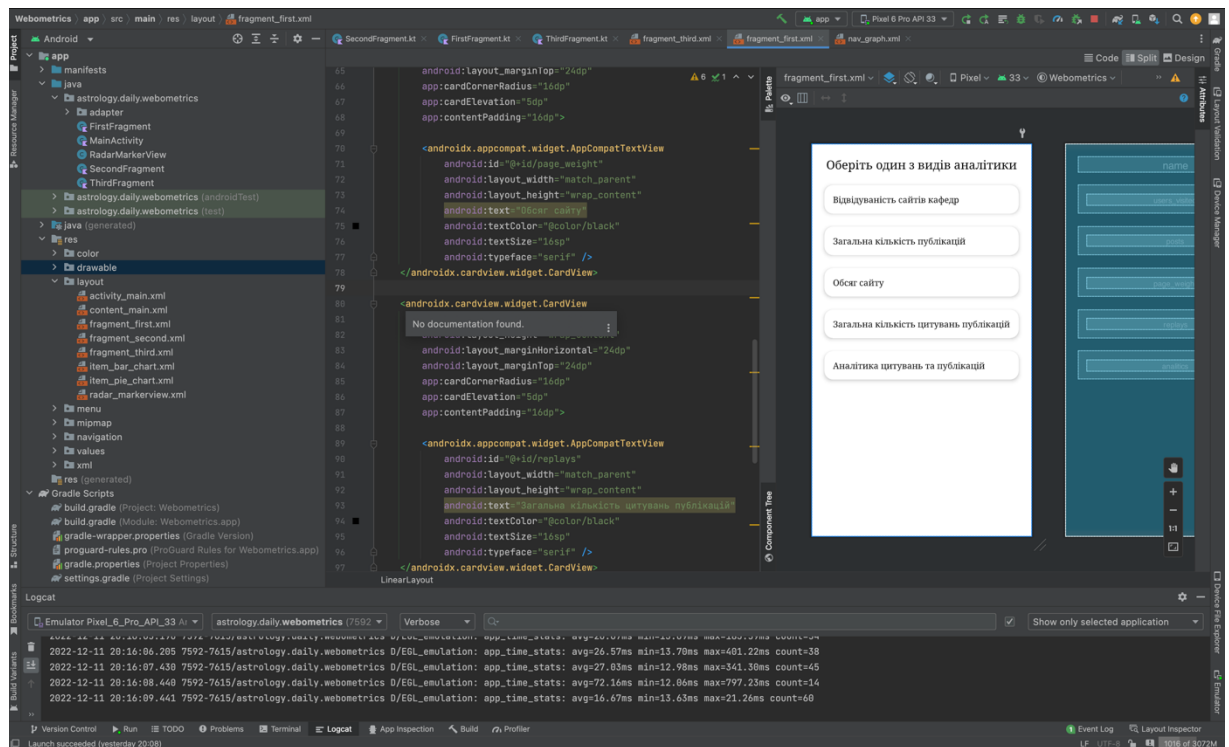


Рисунок 1.9 – Середовище розробки Android Studio

Крім цього існує плагін для Eclipse — «Android Development Tools», призначений для Eclipse версій 3.3-3.7. Для IntelliJ IDEA також існує плагін, який полегшує розробку Android-застосунків.. Для середовища розробки NetBeans розроблено плагін, який починаючи з версії Netbeans 7.0 перестав бути експериментальним. Крім того існує Motodev Studio for Android, що являє собою комплексне середовище розробки, засноване на базі Eclipse і дозволяє працювати безпосередньо з Google SDK. Крім того в 2009 році на застосунок до ADT був опублікований Android Native Development Kit, пакет інструментів і бібліотек дозволяє вести розробку застосунків мовою C/C++. NDK рекомендується використовувати для розробки ділянок коду, критичних до швидкості.

Основним середовищем розробки серверної частини на Java/Kotlin є IntelliJ IDEA, що розроблена компанією JetBrains і має єдину платформу із Android Studio IDE.

Дане середовище існує у двох модифікаціях: безкоштовна та повна. Для комерційної розробки серверних додатків необхідно використовувати саме

повну 21 версію середовища. У ньому наявні додаткові плагіни для роботи із базами даних та проектування архітектури додатку, а також підтримка різних фреймворків.

InjelliJ IDEA постійно розвивається, і функції, що в ній з'являються, пізніше переходять і до Android Studio.

3.3 Організація клієнт-серверної взаємодії зі сторони мобільного додатку.

Для виконання запитів до серверу було обрано веб-клієнт OkHttp та його зручну обгортку Retrofit.

Веб-клієнт OkHttp зарекомендував себе як швидкий, універсальний та конфігурований веб-клієнт, що дозволяє робити запити як асинхронно, так і синхронно. Також зручною особливістю даного веб-клієнту є можливість логувати усі запити та відповіді, тип самим значно полегшується процес відладки застосунків.

Даний веб-клієнт може бути використаним у будь-якій програмі на мові Java, але популярністю від користується саме в контексті програмування під Android.

Обгортка веб-клієнту Retrofit дозволяє значно полегшити конфігурування запитів веб-клієнту та обробки відповідей, що надходять з веб-клієнту, завдяки набору анотацій, згідно з якими на етапі компіляції генерується код конфігурування запитів веб-клієнту, тим сам код стає прозорішим, а програміст не витрачає час на написання подібного шаблонного коду [14].

До основних анотацій фреймворку відносяться:

- POST, GET, PUT, DELETE — методи запитів. У якості параметрів приймають абсолютний або відносний шлях до API сервера;
- Query — url-параметр запиту. У якості параметру приймає ключ параметру у запиті;

- Path — параметр шляху запиту;
- Body — json-тіло запиту;
- Headers — додаткові хедери запиту.

Дана бібліотека також має можливість автоматично переводити об'єкти в json об'єкти (або xml) і навпаки, використовуючи адаптери інших бібліотек, щоспрямовані на серіалізацію та десеріалізацію об'єктів у json/xml.

На даний момент бібліотека має адаптери на наступні серіалізатори:

- Gson;
- Jackson;
- Moshi;
- Protobuf;
- Wire;
- Simple XML.

Також є можливість писати свої адаптери та парсери. Одним з розповсюджених прикладів використання кастомних адаптерів є обробка полів в яких зберігається long-змінна, що відповідає за час у секундах/мілісекундах. Подібні поля на стороні клієнта зручніше зберігати у виді об'єктів класу Date.

3.4 Firebase Cloud Storage.

Аналізуючи поставлені задачі, виникла необхідність обрати сервіс для зберігання отриманих даних. Якщо розглядати доступні засоби, то можна побачити що великим попитом користується саме Firebase Cloud Storage – потужна, проста та економічно ефективна служба зберігання об'єктів, створена для масштабу Google. Пакети Firebase SDK для хмарного сховища додають захист Google для завантажень і завантажень файлів для програм Firebase, незалежно від якості мережі.

Firebase Cloud Storage дає можливість завантажувати файли з мобільних клієнтів через Firebase SDK для хмарного сховища. Крім того, надає можливість виконувати обробку на стороні сервера, наприклад фільтрацію зображень або перекодування відео, використовуючи API Google Cloud Storage.

Хмарне сховище масштабується автоматично, що означає, що немає необхідності переходити до іншого постачальника.

Ключові можливості Firebase Cloud Storage:

- надійні операції - Firebase SDK виконує завантаження незалежно від якості мережі. Завантаження надійні, тобто вони перезапускаються там, де зупинилися, заощаджуючи час і пропускну здатність користувачів;
- сильна безпека - Firebase SDK інтегрується з Firebase Authentication, щоб забезпечити просту та інтуїтивно зрозумілу автентифікацію для розробників. Можна використовувати Firebase декларативну модель безпеки, щоб дозволити доступ на основі імені файлу, розміру, типу вмісту та інших метаданих;
- висока масштабованість - Cloud Storage створено для ексабайтного масштабу, коли додаток стає «вірусним».

4 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

При розробці додатку під ОС Android для побудови користувацького інтерфейсу та структури додатку використовують методи, які надає Android SDK [6].

При виборі архітектури проекту, враховувались фактори використання, тестованості, гнучкості та чистого коду. Взнявши до уваги данні фактори, було обрано розробляти проект на архітектурі Model View Controller (MVC) для мобільного додатку [8] (рис. 4.1).

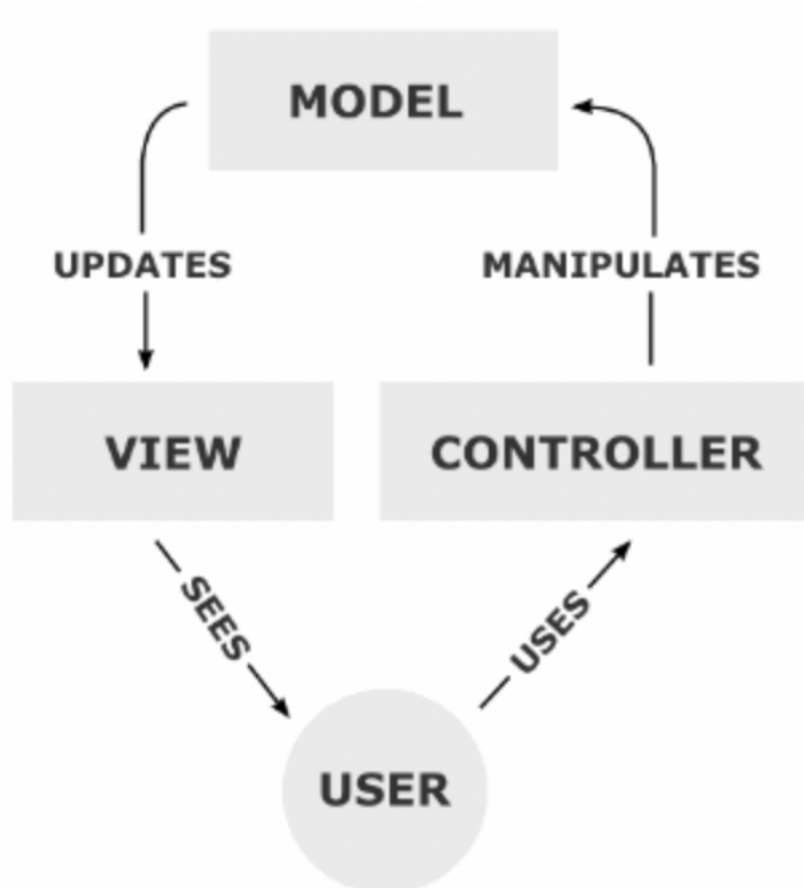


Рисунок 4.1 – Схема архітектури Model View Controller

- модель (Model) - надає дані та реагує на команди контролера, змінюючи свій стан;
- вид (View) - відповідає за відображення даних моделі користувачу, реагуючи на зміни моделі;

- контролер (Controller) - інтерпретує дії користувача, сповіщаючи модель про необхідність змін.

Для побудови користувацького інтерфейсу в розробці використовують групу представлень (ViewGroup).

ViewGroup [7] – це спеціальне представлення, яке може містити інші представлення (ViewGroup). Група представлень є базовим класом для контейнерів макетів і представлень. Цей клас також визначає клас LayoutParams який слугує базовим класом для параметрів макетів (рис. 4.2).

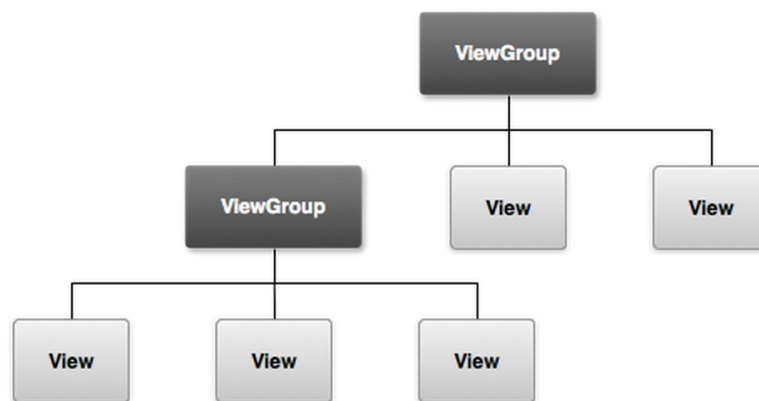


Рисунок 4.2 – Діаграма групи представлень ViewGroup

Представлення (View) – клас являє собою основний будівельний блок для компонентів для користувача інтерфейсу. Представлення займає прямокутну область на екрані і відповідає за малювання і обробку подій. Представлення - це базовий клас для віджетів, які використовуються для створення інтерактивних компонентів для користувача інтерфейсу: кнопок, текстових полів.

Для побудови інтерфейсу користувача у даному додатку використовували саме ViewGroup з взаємодією з View.

4.1 Структура головного екрану

Головний екран (рис. 4.3) має назву «Home». На ньому відображено перелік доступних кнопок, на яких вказано, які саме аналітичні данні будуть відображатись, якщо користувач натисне. Список реалізований за допомогою Android компоненту Recycler View (Лістинг 4.1). Користувач має можливість натиснути на потрібний елемент списку, після чого перейдемо на екран деталей.

Лістинг 4.1 – Реалізація класу Recycler View для відображення списку

```
class ChartAdapter : ListAdapter<ChartModel, ChartViewHolder>(ChartDiffUtils()) {
    companion object {
        private const val PIE_CHART = 0
        private const val BAR_CHART = 1
    }

    override fun getItemViewType(position: Int): Int {
        return if (position == 0) {
            PIE_CHART
        } else {
            BAR_CHART
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ChartViewHolder {
        val view = when (viewType) {
            PIE_CHART -> ItemPieChartBinding.inflate(LayoutInflater.from(parent.context))
            BAR_CHART -> ItemBarChartBinding.inflate(LayoutInflater.from(parent.context))
            else -> ItemPieChartBinding.inflate(LayoutInflater.from(parent.context))
        }
        return ChartViewHolder(view.root)
    }

    override fun onBindViewHolder(holder: ChartViewHolder, position: Int) {
        holder.bind(currentList[position])
    }
}
```

```
class ChartViewHolder(private val view: View) : RecyclerView.ViewHolder(view) {  
    fun bind(item: ChartModel) {  
    }  
}
```

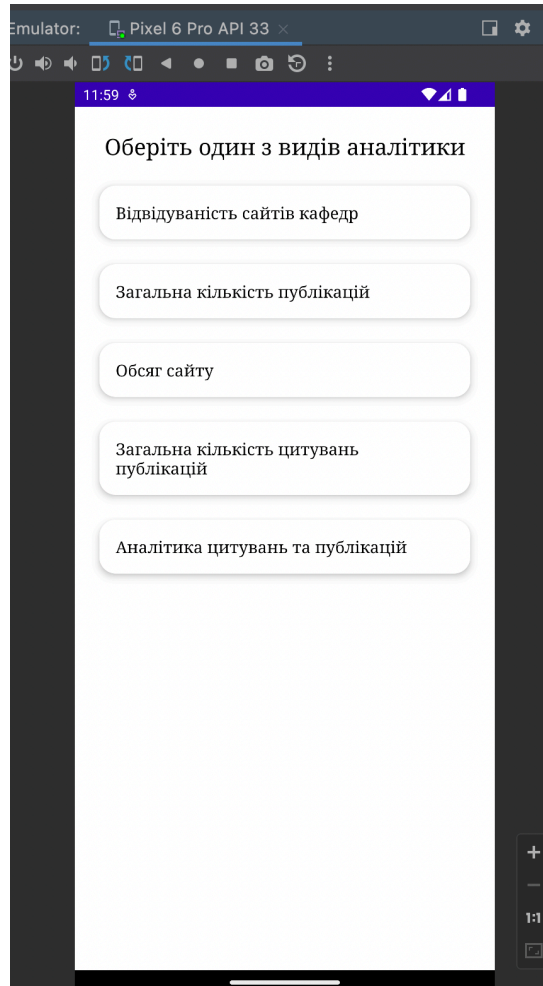


Рисунок 4.3 – Зовнішній вигляд головного екрану

У лістингу 4.2 показано як будуються візуальні блоки головного екрану, в xml файлі головного екрану. Що дозволяють відображати елементи як показано на рис. 4.4.

Лістинг 4.2 – Клас візуального відображення окремого елемента

```

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="24dp"
    android:layout_marginTop="24dp"
    app:cardCornerRadius="16dp"
    app:cardElevation="5dp"
    app:contentPadding="16dp">

<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/page_weight"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="Обсяг сайту"
    android:textColor="@color/black"
    android:textSize="16sp"
    android:typeface="serif" />
</androidx.cardview.widget.CardView>

```

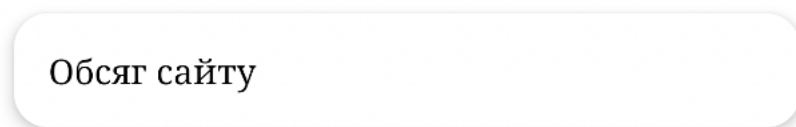


Рисунок 4.4 – Елемент списку на головному екрані

Взаємодія з елементом списку відбувається за допомогою метода відстежування натискання, що відображено в лістингу 2.3. Коли користувач натиснув на один з елементів списку, система виконує навігацію на потрібний екран з відображенням вебметричної аналітики.

Лістинг 4.3 – Метод відстежування натискання на елементі головного екрану

```

binding.pageWeight.setOnClickListener {
    val directions = FirstFragmentDirections.actionFirstFragmentToSecondFragment(
        dataPageWeight,

```

```

        "Обсяг сайтів"
    )
    findNavController().navigate(directions)
}

```

Текст на кожному елементі головного екрану відображається статично, за допомогою текстового представлення `AppCompatActivity` пакету `androidx`. Тінь та більш динамічне відображення елемента надає клас `CardView`.

4.2 Структура екрану порівняльної аналітики `BarChart` та `PieChart`

Назва екрану «Analytics» (рис. 4.5). Цей екран слугує для відображення конкретної аналітики з сайтів:

- АПОТ;
- ВМ;
- Інформатики;
- ІІІ;
- ПМ;
- Фізики.

На даному екрані було прийнято рішення відображати аналітику з деяких сайтів одночасно, для порівняння в реальному часі. На рис. 4.4 можна побачити, що для відображення аналітики використовується дві діаграми, для більш комфортної візуалізації.

Для відображення першої діаграми використовується представлення `BarChart` з бібліотеки `MPAndroidChart` (Лістинг 4.4).

Лістинг 4.4 – Представлення `BarChart`

```

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="10dp"
    android:layout_marginTop="16dp"

```

```

app:cardCornerRadius="16dp"
app:cardElevation="5dp">

<com.github.mikephil.charting.charts.BarChart
    android:id="@+id/barChart"
    android:layout_width="match_parent"
    android:layout_height="400dp" />
</androidx.cardview.widget.CardView>

```

Для відображення кругової діаграми також використовується представлення з бібліотеки MPAndroidChart, яке має назву PieChart (Лістинг 4.5).

Лістинг 4.5 – Представлення PieChart

```

<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="10dp"
    android:layout_marginTop="16dp"
    app:cardCornerRadius="16dp"
    app:cardElevation="5dp">

    <com.github.mikephil.charting.charts.PieChart
        android:id="@+id/pieChart"
        android:layout_width="match_parent"
        android:layout_height="300dp" />

</androidx.cardview.widget.CardView>

```

Бібліотека MPAndroidChart слугує для відображення різних типів діаграм, в залежності від потреб клієнтів. Також, бібліотека надає багато інструментів для змінення графіків під конкретні задачі, зміна кольору, відображення заголовків, зміна шрифтів тощо.

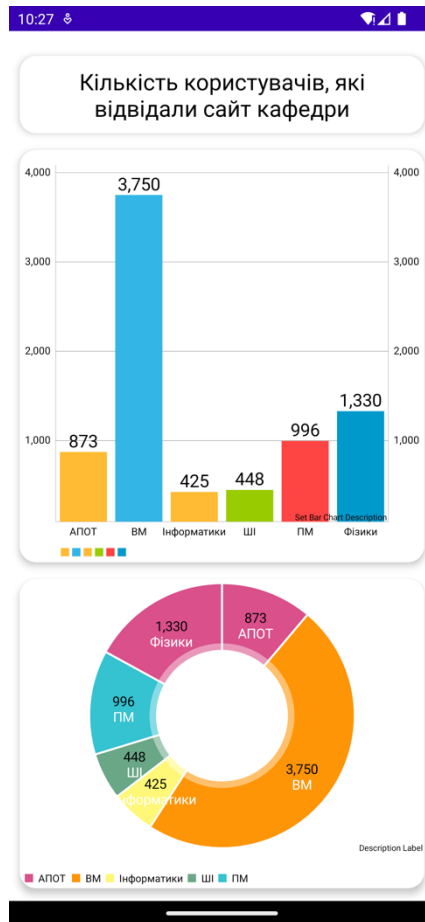


Рисунок 4.5 – Структура екрану виводу аналітики

Для отримання даних аналітики, які відображаються на екрані застосовується Firebase Cloud Storage, в якому зберігається усі доступні вебметричні дані, які ми отримали з сайтів. На рис. 4.6 відображена схема бази даних Firebase.

На схемі ми можемо побачити що в нас зберігається колекція кафедр, вебметричну аналітику яких, вдалося отримати за допомогою сервісів які були розглянуті. В елементі кожної кафедри, розміщені поля з значеннями. Назва полів залежить від типу аналітики, по стандартам сервісів.

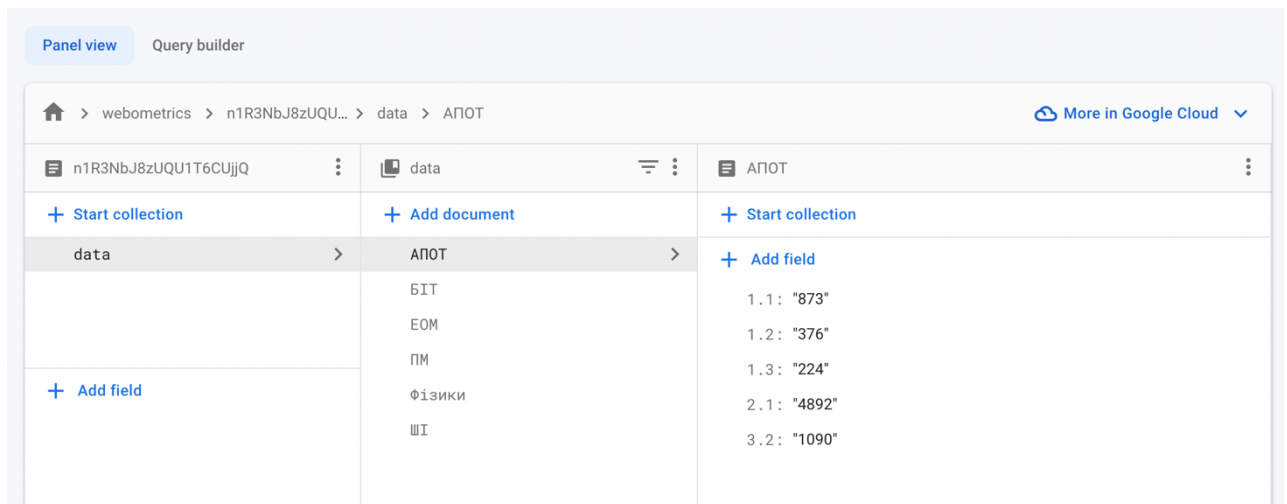


Рисунок 4.6 – Структура екрану виводу аналітики

Для того щоб отримати ці дані в додатку, необхідно звернутись до Firebase, та запросити дані з таблиці, які нам необхідно відображати. Код виконання даного запиту до Firebase [9] відображений в лістингу 4.6.

Лістинг 4.6 – Код запиту на отримання даних з Firebase

```
private suspend fun downloadFileInputStream(storageReference: StorageReference): Result<InputStream>
    try {
        suspendCancellableCoroutine { continuation ->
            storageReference.stream.addOnSuccessListener {
                continuation.resume(Result.success(it.stream))
            }.addOnFailureListener { exception ->
                Timber.d("downloadFileInputStream $exception")
                continuation.resumeWithException(exception)
            }
        }
    } catch (e: Exception) {
        Result.failure(e)
    }
}

private suspend fun getVersion(storageReference: StorageReference): Result<Long> = try {
    suspendCancellableCoroutine { continuation ->
        storageReference.metadata.addOnSuccessListener {
            continuation.resume(Result.success(it.updatedTimeMillis))
        }.addOnFailureListener { exception ->
            Timber.d("getVersion $exception")
            continuation.resumeWithException(exception)
        }
    }
} catch (e: Exception) {
    Result.failure(e)
}
```

Щоб отримати аналітичні дані з Google Analytics, Google Search Console або Google Scholar необхідно використовувати API запити, які надає Google.

Google API надає єдиний метод для запиту даних. Це відображено в лістингу 4.7. Цей метод доступний у різних клієнтських бібліотеках і має мовні інтерфейси для встановлення параметрів запиту [10]. API також можна запитувати як REST-full.

Лістинг 4.7 – Метод Google API для запиту даних.

```
analytics.data.ga.get()
```

Запит до Google API [5] можна уніфікувати за допомогою спеціальних параметрів (рис. 4.7). Після уніфікації, запит може мати вигляд як показано в лістингу 4.8.

Ім'я	Значення	вимагається	Резюме
<code>ids</code>	string	так	Унікальний ідентифікатор таблиці у формі <code>ga:XXXX</code> , де <code>XXXX</code> – це ідентифікатор представлення (профілю) Analytics, для якого запит отримуватиме дані.
<code>start-date</code>	string	так	Дата початку отримання даних Analytics. Запити можуть вказувати дату початку у форматі <code>YYYY-MM-DD</code> , або як відносну дату (наприклад, <code>today</code> , <code>yesterday</code> або <code>NdaysAgo</code> де <code>N</code> додатним цілим числом).
<code>end-date</code>	string	так	Кінцева дата отримання даних Analytics. Запит може вказувати кінцеву дату у форматі <code>YYYY-MM-DD</code> , або як відносну дату (наприклад, <code>today</code> , <code>yesterday</code> або <code>NdaysAgo</code> де <code>N</code> додатним цілим числом).
<code>metrics</code>	string	так	Список показників, розділених комами, наприклад <code>ga:sessions</code> , <code>ga:bounces</code>
<code>dimensions</code>	string	немає	Список параметрів, розділених комами, для даних Analytics, як-от <code>ga:browser</code> , <code>ga:city</code>
<code>sort</code>	string	немає	Список параметрів і показників, розділених комами, що вказує на порядок і напрямок сортування для повернутих даних.
<code>filters</code>	string	немає	Фільтри параметрів або показників, які обмежують дані, що повертаються за вашим запитом.
<code>segment</code>	string	немає	Сегментує дані, повернуті на ваш запит.
<code>samplingLevel</code>	string	немає	Бажаний рівень дискретизації. Дозволені значення: <ul style="list-style-type: none"> <code>DEFAULT</code> – Повертає відповідь із розміром вибірки, який збалансовує швидкість і точність. <code>FASTER</code> – Повертає швидко відповідь із меншим розміром вибірки. <code>HIGHER_PRECISION</code> – Повертає точнішу відповідь із використанням великого розміру вибірки, але це може призвести до сповільнення відповіді.
<code>include-empty-rows</code>	boolean	немає	За замовчуванням значення <code>true</code> ; якщо встановлено значення <code>false</code> , рядки, у яких усі значення метрики дорівнюють нулю, будуть виключені з відповіді.
<code>start-index</code>	integer	немає	Перший рядок даних для отримання, починаючи з 1. Використовуйте цей параметр як механізм розбиття на сторінки разом із <code>max-results</code> параметром.
<code>max-results</code>	integer	немає	Максимальна кількість рядків для включення у відповідь.
<code>output</code>	string	немає	Бажаний тип вихідних даних для даних Analytics, які повертаються у відповідь. Допустимі значення <code>json</code> та <code>dataTable</code> . За замовчуванням: <code>json</code> .

Рисунок 4.7 – Параметри запиту в Google API

Лістинг 4.8 – Уніфікований метод Google API для запиту даних.

```
GET https://www.googleapis.com/analytics/v3/data/ga
?ids=ga:12345
&start-date=2008-10-01
&end-date=2008-10-31
&metrics=ga:sessions,ga:bounces
```

В додатку є можливість відображати різні типи аналітики в порівнянні з іншими сайтами. Не дивлячись що аналітичні дані різні, екран відображення аналітики постійно один і той самий. Це обумовлено тим, що для оптимізації коду, було прийняти рішення пере використовувати один і той самий екран, для відображення різних вебметричних даних. Це дозволяє підтримувати чистий код та уникати дуплікації[9]. На рис. 4.8 відображений граф переходів екранів, що свідчить про це.



Рисунок 4.8 – Граф навігації в додатку Webometrics

4.3 Структура екрану порівняльної аналітики RadarChart

В додатку також є можливість відобразити аналітику використовуючи ще одне представлення з бібліотеки MPAndroidChart. Це представлення має назву RadarChart. У лістингу 4.9 відображено як таке представлення розмістити в додатку.

Лістинг 4.9 – Клас представлення RadarChart

```
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="400dp"
    android:layout_margin="24dp"
    android:layout_marginTop="24dp"
    app:cardCornerRadius="16dp"
    app:cardElevation="5dp">

<androidx.appcompat.widget.AppCompatTextView
    android:id="@+id/users_visited"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:layout_marginHorizontal="16dp"
    android:layout_marginTop="5dp"
    android:textAlignment="center"
    android:textColor="@color/black"
    android:textSize="16sp" />

<com.github.mikephil.charting.charts.RadarChart
    android:id="@+id/radar1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_marginTop="40dp" />

</androidx.cardview.widget.CardView>
```

Для завантаження даних в представлення RadarChart, використовується код написаний в класі цього екрану (Лістинг 4.10). Він дозволяє записати

отримані дані в діаграми, та методом `invalidate()` перерисувати діаграми вже з переданими даними.

Лістинг 4.9 – Метод для передачі даних в RadarChart

```

{
    val set1 = RadarDataSet(
        entries1,
        "Загальна кількість цитувань публікацій НПП кафедри в базі Google Scholar"
    )
    val set2 = RadarDataSet(
        entries2,
        "Загальна кількість цитувань публікацій НПП кафедри в базі Scopus"
    )
    val set3 = RadarDataSet(entries3, "Кількість публікацій штатних співробітників кафедр")
    val set4 = RadarDataSet(
        entries4,
        "Органічний пошук сторінок користувачами через пошукову систему Google"
    )

    set1.setDrawFilled(true)
    set1.fillAlpha = 180
    set1.lineWidth = 2f
    set1.isDrawHighlightCircleEnabled = true
    set1.setDrawHighlightIndicators(false)

    set2.setDrawFilled(true)
    set2.fillAlpha = 180
    set2.lineWidth = 2f
    set2.isDrawHighlightCircleEnabled = true
    set2.setDrawHighlightIndicators(false)

    set3.setDrawFilled(true)
    set3.fillAlpha = 180
    set3.lineWidth = 2f
    set3.isDrawHighlightCircleEnabled = true
    set3.setDrawHighlightIndicators(false)

    set4.setDrawFilled(true)
    set4.fillAlpha = 180
    set4.lineWidth = 2f
    set4.isDrawHighlightCircleEnabled = true
    set4.setDrawHighlightIndicators(false)

    val sets1 = java.util.ArrayList<IRadarDataSet>()
    val sets2 = java.util.ArrayList<IRadarDataSet>()
    val sets3 = java.util.ArrayList<IRadarDataSet>()
    val sets4 = java.util.ArrayList<IRadarDataSet>()
    sets1.add(set1)
    sets2.add(set2)
    sets3.add(set3)
    sets4.add(set4)

    binding.radar1.data = data1
    binding.radar2.data = data2
    binding.radar3.data = data3
    binding.radar4.data = data4
    binding.radar1.invalidate()
    binding.radar2.invalidate()
    binding.radar3.invalidate()
    binding.radar4.invalidate()
}

```

На наступному екрані (рис. 4.9) показано, як відображається аналітика кафедр АПОТ, БІТ, БМІ, ЕОМ. Використовуючи сервіси для збору вебметричних даних, вдалося отримати для цього екрану таку аналітику:

- загальна кількість цитувань публікацій НПП кафедри в базі Google Scholar;
- загальна кількість цитувань публікацій НПП кафедри в базі Scopus;
- кількість публікацій штатних співробітників кафедр;
- органічний пошук сторінок користувачами через пошукову систему Google.

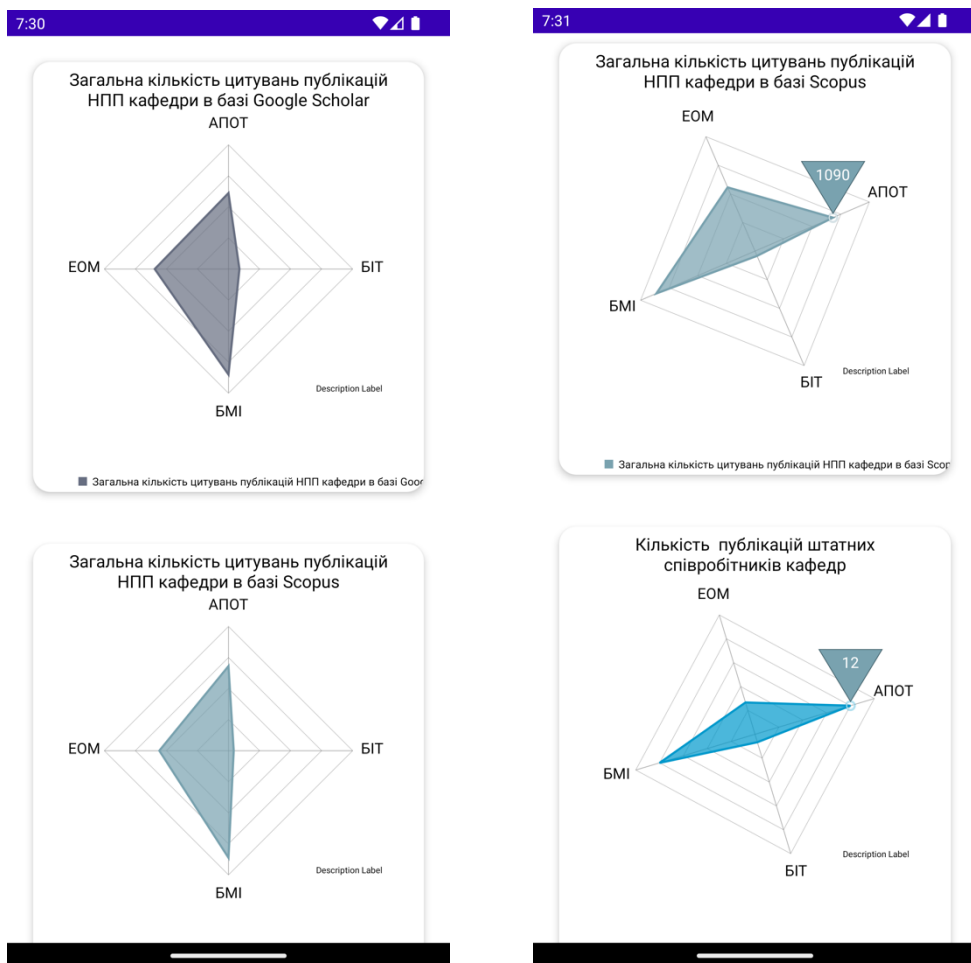


Рисунок 4.9 – Екран відображення аналітики з представленням RadarChart

5 ТЕСТУВАННЯ МОБІЛЬНОГО ЗАСТОСУНКА

Початкові умови роботи додатку не мають даних про вебметричні дані кафедр Харківського національного університету радіоелектроніки з досліджуваних сервісів. Тому на головному екрані виводиться відповідне повідомлення (рис. 5.1). Тестування будемо проводити на прикладі функціонала «Відображення різної аналітики».



Завантаження доступних даних...

Рисунок 5.1 – Головний екран в початкових умовах

На головному екрані починає працювати взаємодія з Firebase Cloud Storage і з інтегрованими сервісами для отримання вебметричних даних, які

будуть збережені локально та на Firebase Cloud Storage після завантаження. Як тільки завантаження успішно завершено, на головному екрані відображається список доступної аналітики (рис. 5.2).



Рисунок 5.2 – Головний екран після збереження даних для відображення

Після завантаження та збереження даних, натискаємо на аналітику «Відвідуваність сайтів кафедри». І нас навігує на екран відображення обраної аналітики (рис. 5.3).

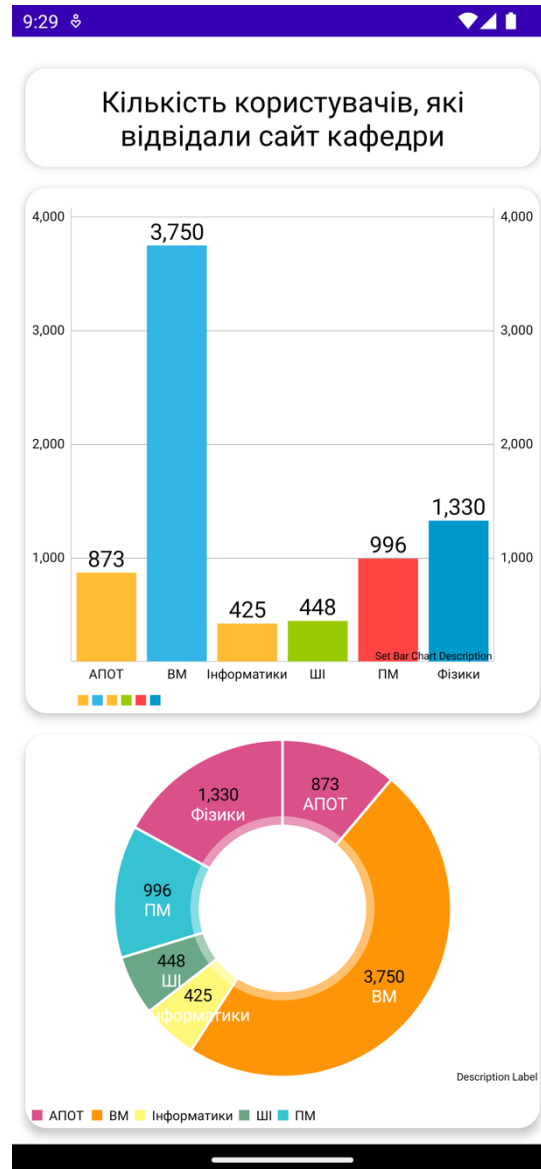


Рисунок 5.3 – Екран відображення аналітики «Відвідуваність сайтів кафедри».

Можемо побачити, що аналітика відображена правильно і не викликала ніяких помилок. Це ще раз підтверджує те, що дані збережені вже локально і дістаються по запиту для відображення.

Проведемо тестування зміни даних. Замінімо дані в базі даних в об'єкті «Відвідуваність сайтів кафедри». Такі дії проводяться для тестування відображення діаграм, при зміні вхідних даних. Отже, замінивши дані в таблиці, можемо побачити що діаграми відпрацювали коректно (рис. 5.4).

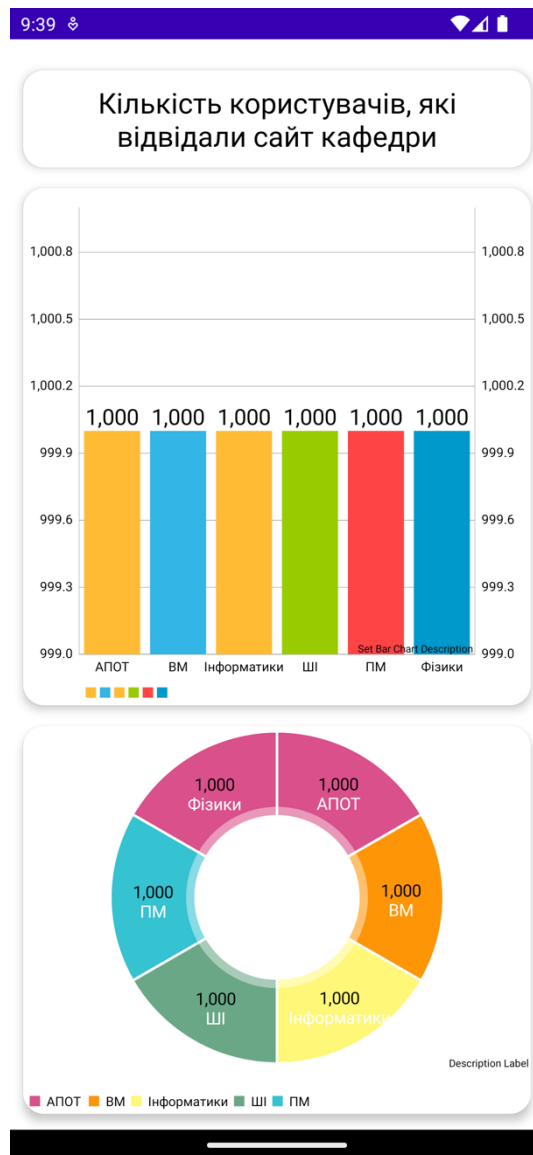


Рисунок 5.4 – Екран відображення тестування підміни даних для аналітики «Відвідуваність сайтів кафедри».

Повернемось на головний екран. З усіх видів аналітики, є ще аналітика «Загальна кількість цитувань публікацій». Це специфічний екран, для відображення аналітики декількох аналітик. Перейшовши на екран, бачимо анімацію завантаження діаграм а після, вже відображення чотирьох діаграм, кожна з яких відображає свою аналітику по чотирьом обраним кафедрам (рис. 5.5).



Рисунок 5.5 – Екран відображення аналітики «Загальна кількість цитувань публікацій».

При тестуванні, була здійснена багаторазова підміна даних, написані тести для обробки інформації, запитів, роботи з базою даних. Був проведений краш-тест додатку, в якому не було знайдено жодної помилки.

ВИСНОВКИ

В результаті виконання кваліфікаційної роботи було досягнуто мету – підвищення продуктивності засобів вебметричного аналізу веб-сайтів навчальних закладів за рахунок автоматизації отримання вебметричних даних.

Виконано аналіз існуючих засобів отримання вебметричних даних сайтів вищих навчальних закладів, вибрано моделі збору та аналізу вебметричних даних, розроблено архітектуру мобільного застосунку та програмне забезпечення, здійснено тестування мобільного застосунку.

Мобільний застосунок реалізований на базі операційної системи Android. Програмна реалізація алгоритму автоматизованого отримання вебметричних даних здійснена мовою програмування Kotlin у середовищі розробки Android Studio.

В роботі використано підхід реактивного програмування, створено інтерфейс користувача, підключено сервіси Google Analytics, Google Scholar, Google Search Console, Scopus. Розглянуто основні індикатори, що входять до глобальних вебметричних рейтингів сайтів вищих навчальних закладів. Реалізовано екрани для відображення діаграм веб аналітики сайтів навчальних закладів.

Розроблене програмне забезпечення дозволяє автоматизувати збір та обробку вебметричних даних, отримати актуальну статистику за встановленими показниками та порівняти різні сайти навчальних закладів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Басюк О. В. Експертний аналіз інтернет – сайтів турфірм як метод покращення їх економічної діяльності / О. В. Басюк // Науковий вісник Ужгородського національного університету. Серія: Міжнародні економічні відносини та світове господарство. – Ужгород: УНУ, 2016. – Вип. 7.
2. Лебеденко М. С. Вебметричний ранг як показник ефективності електронного ресурсу підприємства / М. С. Лебеденко // Економічний вісник Національного технічного університету України "Київський політехнічний інститут". - 2014. - № 11. - С. 401-408.
3. Google [Електронний ресурс]: Офіційний веб-сайт Google Analytics – Вебаналітика та звітування – Google Analytics – Режим доступу: <https://www.google.com.ua/intl/uk/analytics/index.html> - Дата доступу: 15.04.2017.
4. Google [Електронний ресурс]: Список функцій Google Analytics – Google Analytics – Режим доступу: https://www.google.com.ua/intl/uk_ALL/analytics/features/index.html - Дата доступу: 15.04.2017.
5. Google [Електронний ресурс]: Про сегменти - Analytics Дповідка – Режим доступу: <https://support.google.com/analytics/answer/3123951?hl=uk> - Дата доступу: 15.04.2017.
6. Филлипс, Б. Android. Программирование для профессионалов. 3-е издание [Текст] / Б.Филлипс, К.Стюарт, К.Марсикано – СПб.: Питер, 2017. – 688 с.
7. Жемеров, Д. Kotlin в действии [Текст] / Д. Жемеров, С. Исакова – М.: ДМК Пресс, 2016. – 402 с.
8. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения [Текст] / Р. Мартин – СПб.: Питер, 2016. – 352 с.
9. Приемы объектно-ориентированного проектирования [Текст] / Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влассидес – СПб.: Питер, 2013. – 368 с.

10. Офіційний сайт документації платформи Firebase [Електронний ресурс] / Режим доступу: www/ URL: <https://firebase.google.com/docs/>– 09.05.2020 р. – Загл. з екрану.

11. Офіційний сайт документації AndroidSDK [Електронний ресурс] / Режим доступу: www/ URL: <https://developer.android.com/>– 09.05.2020 р. – Загл. з екрану.

12. Нуркевич, Т. Реактивное программирование с использованием RxJava [Текст] / Т. Нуркевич, Б. Кристенсен – М.: ДМК-Пресс, 2017. – 358 с.

13. Дарвин, Ян. Android. Сборник рецептов: задачи и решения для разработчиков приложений. 2-е издание [Текст] / Ян Дарвин – К.: Диалектика, 2018. – 768 с.

14. Еванс, Е. Предметно-ориентированное проектирование (DDD). Структуризация сложных программных систем [Текст] / Е. Еванс – К.: Диалектика, 2016. – 448 с.