

УДК 004.9

Р. Гасько<sup>1</sup>, В. Висоцька<sup>1</sup>, Л. Чирун<sup>2</sup>

Національний університет “Львівська політехніка”,

<sup>1</sup>кафедра інформаційних систем та мереж,

<sup>2</sup>кафедра систем автоматизованого проектування

## ОСОБЛИВОСТІ КОНТЕНТ-АНАЛІЗУ КОРИСТУВАЦЬКОЇ ІНТЕРНЕТ-ДІЯЛЬНОСТІ ДЛЯ ФОРМУВАННЯ ЗРІЗУ ПСИХОЛОГІЧНОГО СТАНУ ОСОБИСТОСТІ

© Гасько Р., Висоцька В., Чирун Л., 2017

Розроблено систему для аналізу психологічного та емоційного стану особистості. Метою є оцінка особистості за допомогою соціальних мереж та визначення практичних рекомендацій. Проаналізовано системи оцінювання даних, досліджено проблематику цього напрямку та актуальність створення цієї системи. Наведено діаграми, які описують структуру та логіку функціонування системи. Описано системні вимоги згідно з методологією RUP та створено прототип програми, який моделює діяльність системи аналізу особистості.

**Ключові слова:** інформаційний ресурс, комерційний контент, контент-аналіз, контент-моніторинг, контентний пошук.

A system for the analysis of the psychological and emotional state of the individual is developed. The aim is to assess the individual through social networks and the practical recommendations. The analysis of assessment data, the problems of this area and the relevance of the system were studied. The diagrams are developed that describe the structure and logic of the system. Description of system requirements according to RUP methodology was done and a prototype application that simulates the activity of individual analysis system was created.

**Key words:** information resources, commercial content, content analysis, content monitoring, content search.

### Вступ. Загальна постановка проблеми

Одним з фундаментальних принципів сучасної психології є диспозиційна модель особистості людини, яка містить 5 відносно незалежних диспозицій [1–2]:

- екстраверсія/інтроверсія – спрямованість людини до зовнішнього світу, балакучість, товариськість або занурення в світ уяви та роздумів;

- доброзичливість – здатність до взаємодопомоги, спільної роботи і симпатія до оточуючих;
- сумлінність – дисципліна, старанність і орієнтованість на результат;
- невротизм – ступінь емоційної стабільності, рівень контролю імпульсів і тривожності;
- відкритість досвіду – ступінь інтелектуальної допитливості, прагнення до нових і різноманітних речей, вражень.

Кожну особистість ранжують за 5 диспозиціями – так психологи і складають особистісну модель. Вона необхідна для прогнозування дій людини, формування висновків про її професійну придатність, перспективи професійного зростання, можливості роботи в колективі, можливості конкретних особистостей довго працювати разом і багато іншого. При прийомі на роботу деякі організації проводять обов’язкове психологічне тестування кандидатів для складання особистісної моделі кожного з них. Це доволі тривала процедура, яка вимагає певних витрат. Але можна значно здешевити і автоматизувати таке тестування. Доктор Шуотан Баї (Shuotian Bai) з Національної академії наук Китаю опублікував наукову роботу [3, 4] з описом простого способу обчислення кожної з 5 диспозицій особистості за результатами аналізу її активності в соціальній мережі. Досвідчений фахівець визначає властивості особистості людини, дослідивши його активність в Facebook або “Вконтакте” – для нього немає нічого складного. Інше питання – автоматизація цього процесу. У роботах [1–12] пояснюється, що кожну диспозицію обчислюють за чисельними параметрами активності в соціальній мережі. Наприклад, рівень сумлінності (дисципліна, старанність, орієнтованість на результат) визначають за кількістю повідомлень з питаннями, проханнями допомоги. Ознакою екстраверсії є велика кількість емотикони, частота оновлення статусу свідчить про відкритість досвіду, а рівень невротизму визначають за кількістю повідомлень, які викликали негативну оцінку інших. Автоматичне складання особистісних моделей користувачів надзвичайно корисно для соціальних мереж і веб-сервісів, оскільки дає змогу показувати ефективнішу контекстну рекламу, поліпшити систему рекомендацій і зробити якіснішими служби знайомств [13–22]. Досконале знання аудиторії має вирішальне значення для бізнесу та рекрутингу.

### **Аналіз останніх наукових публікацій**

#### **Складання моделі особистості за активністю в соціальній мережі**

Коли люди приходять на інтерв’ю, вони “одягають” спеціально підготовлену маску. Вони все вже відрепетирували, але це – фальшивка. Деякі компанії наполягають на психологічному профілюванні через оцінки. Для організацій залишається важливим розуміння особистості і ким вона є насправді (всі хочуть уникнути дорогих помилок під час найму, і ніхто не хоче мати справу з “руйнівниками команд”) [1]. У більшості випадків в соціальних медіа люди не “грають” і чесно висловлюють свої думки (тривалий час жодна людина не в змозі приховувати власні емоції, думки та поведінку), тому аналіз твітів (наприклад, Tweet Psych) дає багато цікавої інформації [5]. Спілкування – це вікно в розум людини, і те, як людина говорить, може багато чого сказати про її думки. Лінгвісти розробили два способи декодування написаного слова в профіль пізнавальних процесів людини. Перший метод – це регресивний словник зображень (RID). Ця схема кодування призначена для вимірювання кількості та типу з трьох категорій контенту: початковий (несвідоме; як думають уві сні), концептуальний (логічний і раціональний) і емоційний. Другий метод – Linguistic Inquiry and Word Count (LIWC) (лінгвістичне дослідження і підрахунок слів) вимірює когнітивні й емоційні властивості особистості на основі слів, які використовують люди [1–4].

#### **Психологічні профілі у відомих соціальних мережах**

**Tweet Psych** використовує RID і LIWC для створення психологічного портрету людини на основі контенту його твітів [6]. Метод ґрунтується на порівнянні змісту твітів користувача з базовими показниками, які зібрано на основі аналізу понад 1,5 млн. обраних навмання твітів, і виявлення областей, де виділяється користувач. Служба аналізує останню тисячу твітів (це якнайкраще працює з користувачами, які розмістили понад 1 тис. оновлень). Цей метод добре підходить для аналізу акаунтів, які належать одному користувачу, який використовує Twitter і для розмов, а не просто як платформу для поширення контенту. Метод вимірює:

- 1) страх,
- 2) оральну фіксацію,
- 3) роботу,
- 4) позитивні емоції,
- 5) негативні емоції,
- 6) соціальну поведінку,
- 7) печаль,
- 8) духовність,
- 9) лайки (погану мову),
- 10) сексуальні переваги,
- 11) сон,
- 12) спорт,
- 13) освіту,
- 14) самокритику,
- 15) гроші,
- 16) розваги.

При ефективному проведенні подібних досліджень багато рекрутерів використовували б отримані дані. Але в цій практиці є величезна кількість правових нюансів.

**LinkedIn: “Pole Position”** перебуває в кращому стані для створення поведінкових і психологічних профілів [6]. Хоча це не основна його мета, але він має доступ до аналізу:

- 1) навичок/досвіду,
- 2) досвіду роботи (компаній, в яких люди працювали, а також тривалість служби),
- 3) зростання кар’єри (тривалість перебування на певних посадах),
- 4) навчання і розвитку,
- 5) освіти,
- 6) подальшої освіти,
- 7) рекомендацій,
- 8) груп/асоціацій/членів дискусійних спільнот,
- 9) коментарів, зроблених у цих групах (можливий психологічний аналіз),
- 10) особистої інформації,
- 11) статей, які читає,
- 12) статей, які пересилає,
- 13) профілів, якими цікавиться/дружить,
- 14) розміру і кількості зв’язків (мережі),
- 15) оновлення, які робить (знову проаналізовані щодо певних даних),
- 16) де ставить “лайки” (“Like”).

### **Формулювання мети**

Необхідно розробити загальну структуру системи аналізу профілів користувачів соціальних мереж та їх активності для формування зрізу психологічного стану особистості на основі моделі “Велика п’ятірка”. Для цього доцільно використати методи та засоби опрацювання інформаційних ресурсів в Інтернет-середовищі [23–48].

### **Аналіз отриманих наукових результатів**

Метою роботи є створення ІС психологічного аналізу особистості. Ця система призначена для визначення основних рис “Великої п’ятірки” з використанням аналізу повідомлень користувачів у соціальній мережі Twitter. Основними користувачами системи є як користувачі Інтернету, так і користувачі ПК загалом, які хочуть просто та швидко дізнатися певні психологічні особливості окремого користувача соціальної мережі. Також користувачами можуть бути різні рекрутингові компанії, оскільки інформацію, зібрану під час роботи цієї системи, можна використати для пошуку

необхідної людини на певну посаду. Інтерфейс цього програмного рішення має бути простим та інтуїтивно зрозумілим навіть невпевненому користувачу ПК. Система повинна бути інтегрованою із Інтернетом для збирання повідомлень у соціальній мережі, а також інших допоміжних сервісів. Задачами цієї інформаційної системи є можливість:

- 1) авторизації користувачів для доступу до подальшої роботи із системою;
- 2) авторизації користувача у соціальній мережі Twitter;
- 3) пошуку необхідного користувача;
- 4) аналізу повідомлень необхідного користувача;
- 5) зручного перегляду результатів аналізу.

Основними вхідними даними для функціонування системи є:

- 1) персональна інформація користувача: логін, пароль, дані авторизації із Twitter;
- 2) повідомлення користувача, якого аналізують у соціальній мережі Twitter.

ІС повинна мати таку можливість:

- авторизації користувачів;
- авторизації користувача у соціальній мережі Twitter;
- керування інтерфейсом користувача;
- зручного переглядання результати аналізу.

Результатом роботи є ІС, подана у вигляді десктопної програми, яка одночасно є інтернет-сервісом і дає змогу аналізувати психологічний стан певного користувача соціальної мережі Twitter за його повідомленнями. Ця система має дозволяти користувачу зберігати дані аналізу необхідного користувача, а також аналізувати інших користувачів соціальної мережі.

Головним питанням цієї роботи є аналіз повідомлень певного користувача у соціальній мережі Twitter. Існує велика кількість різноманітних методів аналізу тексту для визначення майже будь-якого аспекту характеру людини: від звичок та вподобань до конкретного і ґрунтовного аналізу особистості. Є сенс розглянути такі методи аналізу тексту:

– *контент-аналіз* – один з видів аналізу документів. Цей метод дає змогу у систематичній і надійній послідовності визначити певні елементи контенту певного масиву документів із подальшим кількісним опрацюванням отриманих даних. Застосовується для дослідження масивів однорідних документів, великих масивів тексту, найчастіше – текстів масової комунікації. Застосовують при визначенні результатів анкетування, проведення різноманітних інтерв'ю тощо.

– *морфологічний аналіз* – дає змогу виявляти морфологічні інтерпретації кожного слова тексту. Цей вид аналізу є основою для інших ґрунтовніших видів аналізу тексту.

– *семантичний аналіз* – метод, який дає змогу будувати семантичні структури, складається із семантичних вузлів та семантичних відношень.

– *інтеннт-аналіз* – доволі новий метод, який ґрунтується на тому, що поведінка кожного індивіда у соціумі є інтенціонально спрямованою і має певного адресата. Іntenція – це суб'єктивна спрямованість на певний предмет чи активність.

– *психосемантика* – сукупність методів, за допомогою яких визначають значення або зміст того чи іншого слова, предмета чи явища.

Існують певні відмінності у виявленні психологічного змісту тексту методами інтеннт-аналізу і психосемантики. Поряд з виділенням загальних факторів визначають додаткові структури, відношення між якими можуть не враховуватися при використанні цих методів окремо. Психосемантичним методом переважно виявляються несвідомі тенденції, наміри особистості. Об'єднанням цих методів можна збагатити аналіз психологічного змісту тексту.

Наступною проблемою є методи опрацювання великих масивів текстових даних. Тут варто зазначити, що для текстових даних не надто підходять звичайні алгоритми пошуку. Тут потрібно застосовувати інші методи. Нижче розглянуто деякі із них.

*Пошук рядка* формально визначають так. Нехай задано масив  $T$  з  $N$  елементів і масив  $W$  з  $M$  елементів, причому  $0 < M \leq N$ . Пошук рядка виявляє перше входження  $W$  в  $T$ , результатом вважатимемо індекс  $I$ , який вказує на перший з початку рядка (з початку масиву  $T$ ) збіг з образом

(словом). Наприклад, потрібно знайти всі входження зразка  $W = ABAА$  в текст  $T = abcabaabcabca$ . Зразок входить у текст лише один раз, із зміщенням  $S = 3$ , індекс  $I = 4$ .

*Алгоритм прямого пошуку.*

1. Нехай  $I = 1$ ;
2. Порівнюємо  $I$ -й символ масиву  $T$  з першим символом масиву  $W$ .
3. Якщо символ збігається – порівнюємо другий символ, третій символ і т. д.
4. Якщо символ не збігається, збільшуємо  $I$  на 1 і переходимо до пункту 2.

Умови завершення алгоритму:

1.  $M$  порівнянь поспіль є успішними.
2. Якщо  $I + M > N$ , то необхідне слово не знайдено.

Складність алгоритму, в найгіршому випадку становить  $O(N * M)$ .

Недоліками цього алгоритму є:

1. Висока складність алгоритму (аж до  $\Theta((N - M + 1) * M)$ );
2. Якщо символи не збігаються, то пошук починається з першого символу зразка, і тому може містити символи  $T$ , які вже раніше переглядалися.
3. Інформацію про текст  $T$ , яку ми отримуємо, перевіряючи зсув  $S$ , не використовуємо для перевірки наступних зсувів.

*Алгоритм КМП-пошуку (алгоритм Д. Кнута, Д. Моріса та В. Прата).* Цей алгоритм фактично потребує лише  $N$  порівнянь, навіть у найгіршому випадку. Після часткового збігу початкової частини масиву  $W$  з відповідними символами рядка  $T$  по факту стає відомою перша частина рядка і можна визначити деякі властивості (на основі самого масиву  $W$ ), за допомогою яких далі можна швидко пересуватися по тексті. Ідея КМП-пошуку полягає у тому, що за кожної розбіжності двох символів тексту і масив зсувається на усю вже пройдену відстань, оскільки менші зсуви не зможуть привести алгоритм до повного збігу. Особливості КМП-пошуку:

- 1) потрібно приблизно  $(N + M)$  порівнянь символів для отримання результату;
- 2) схема КМП-пошуку насправді успішна лише тоді, коли невдалим спробам передувала деяка кількість збігів. Лише у цьому випадку образ зсувається більш ніж на один символ. На жаль, збіги зустрічаються значно рідше, ніж розбіжності. Тому виграш, який можна отримати від КМП-пошуку, в більшості текстів насправді доволі незначний.

*Алгоритм Р. Боуера і Д. Мура (БМ-пошук).* На практиці алгоритм БМ-пошуку є найефективнішим, якщо масив  $W$  довгий, а потужність алфавіту є достатньо великою. В основу БМ-пошуку покладено таку ідею: порівняння символів починається з кінця масиву, а не з початку, тобто порівняння певних окремих символів відбувається справа наліво. Потім за допомогою деякої евристичної процедури ми обчислюємо величину зсуву вправо  $s$ . Потім починаємо знову порівнювати символи, починаючи з кінця зразка. Цей метод не лише покращує обробку навіть у найгіршому випадку, але і дає виграш для проміжних ситуацій. У більшості випадків, крім спеціально складених прикладів, БМ-пошук вимагає значно менше  $N$  порівнянь. За найбільш сприятливих обставин, коли останній символ зразка завжди потрапляє на символ тексту, який не збігається, кількість порівнянь  $= (N / M)$ , у гіршому випадку –  $O((N - M + 1) * M + p)$ , де  $p$  – потужність алфавіту. Отже, для опрацювання повідомлень користувача було обрано, по-перше, комбінування інтен-аналізу та психосемантики як найефективніших засобів визначення певних рис користувача, по-друге, було застосовано алгоритм БМ-пошуку, який показав себе найефективнішим з усіх вищеперелічених алгоритмів та не є надто складним у реалізації.

Метод формування зрізу психологічного стану особистості складається з таких алгоритмів:

1. Персоналізація (авторизація/ідентифікація користувача системи, вибір соціальної мережі, де буде провадитися аналіз, пошук та авторизація/ідентифікація аналізованої особистості в цій мережі та персоналізація користувача системи в цій мережі для надання можливості доступу до даних).
2. Пошук людей за певними типами чи вподобаннями
3. Пошук та отримання доступу до персоналізованої інформації особистості та історії діяльності в соціальній мережі

4. Доступ до даних та порівняння даних з різних соціальних мереж
5. Збирання та первинний аналіз даних аналізованої особистості (історія, профіль, пости, коментарі, лайки, спільноти тощо)
6. Форматування та зберігання даних, які опрацьовані на всіх етапах
7. Статистичне опрацювання та фільтрування даних
8. Контент-аналіз тексту для знаходження маркованих слів
9. Навчання системи згідно з проведеним контент-аналізом
10. Попередній аналіз даних особистості та побудова диспозиційної моделі особистості за основними параметрами великої п'ятірки
  - a. Аналіз екстраверсії/інтроверсії аналізованої особистості;
  - b. Аналіз доброзичливості аналізованої особистості;
  - c. Аналіз сумлінності аналізованої особистості;
  - d. Аналіз невротизму аналізованої особистості;
  - e. Аналіз відкритості досвіду аналізованої особистості.
11. Основний аналіз за результатами попереднього аналізу та наявністю маркованих слів та активністю в соціальній мережі
  - a. Характеру особистості.
  - b. Темпераменту особистості
  - c. Емоційності особистості
  - d. Самооцінки особистості
  - e. Вольових якостей особистості
  - f. Соціальних якостей особистості
12. виправлення можливих помилок аналізу
13. Порівняння з шаблонами та коригування результатів згідно з результатами аналізу
14. Визначення психологічного типу за З. Фрейдом
15. Визначення психологічного типу за Юнгом
16. Резервне копіювання даних
17. Інформування про похибку дослідження
18. Формування загальних висновків та рекомендацій

Основні результати формування загальних висновків та рекомендацій залежать від коректного проведення контент-аналізу тексту для знаходження маркованих слів, алгоритм якого:

**Етап 1.** Визначення набору критеріїв для аналізованого текстового контенту.

*Крок 1.* Формування набору критеріїв як тип джерела (тип соціальної мережі, публічний і/або приватний контент, спільноти тощо); тип контенту (діалоги чату, історія, дані профілю, множина зашареного контенту, активність залучення друзів, пости, коментарі, лайки, активність у спільнотах тощо); особистість, яку аналізують.

*Крок 2.* Визначення розміру (межі обсягу/довжини), частоти появи, способу/місця розповсюдження та час появи контенту.

*Крок 3.* Визначення змісту та масиву маркованих слів згідно з категорією психологічного аналізу за основними параметрами великої п'ятірки.

*Крок 4.* Створення, формування, заповнення, модифікація словників маркованих слів, заблокованих слів, правил класифікації тексту за психологічними параметрами.

*Крок 5.* Визначення правил фільтрування контенту за сформованими критеріями.

**Етап 2.** Первісний аналітико-статистичний відбір та формування вибіркової сукупності контенту за критеріями обмеженої вибірки з більшого масиву.

**Етап 3.** Вибір лінгвістичної одиниці аналізу тексту для ідентифікації в ньому маркованих слів за основними параметрами великої п'ятірки

*Крок 1.* виявлення змістовних лінгвістичних одиниць аналізу текстового контенту (слово, словосполучення, речення, тема, лайк, пост, підтвердження дружби, ідея, автор, персонаж, соціальна ситуація, частина тексту, кластеризована за змістом категорії аналізу).

*Крок 2.* Формування вимог до вибору лінгвістичної одиниці аналізу: достатньо велика для інтерпретації значення; достатньо мала, щоб не інтерпретувати багато значень; легко ідентифікується; кількість одиниць достатньо велика для проведення вибірки.

*Крок 3.* Формування правил проведення морфологічного аналізу ідентифікованих лінгвістичних одиниць для визначення основ слів з врахуванням частоти їх появи в тексті.

*Крок 4.* Формування правил проведення синтаксичного аналізу ідентифікованих лінгвістичних одиниць з врахуванням особливості мови аналізованого тексту.

*Крок 5.* Формування правил проведення семантичного аналізу ідентифікованих лінгвістичних одиниць з врахуванням особливостей утворення маркованих словосполучень.

*Крок 6.* Формування правил ідентифікації частовживаних маркованих слів серед множини виявлених лінгвістичних одиниць аналізованого тексту без врахування заблокованих слів.

*Крок 7.* Формування правил ідентифікації заблокованих слів серед часто вживаних лінгвістичних одиниць.

**Етап 4.** Виділення одиниць рахунку аналізу текстового контенту.

*Крок 1.* Якщо одиниці рахунку збігаються з одиницями аналізу, то знаходять частоти появи виділеної змістовної одиниці, інакше перейти до кроку 2.

*Крок 2.* Модератор на основі аналізованого контенту висуває одиниці рахунку, наприклад, протяжність текстів; площа тексту, заповнена змістовними одиницями; кількість рядків (абзаців, знаків, колонок тексту); розмір/вид файла; кількість рисунків з певним змістом/сюжетом тощо.

**Етап 5.** Порівняння змістовних одиниць аналізу з одиницями рахунку.

*Крок 1.* Класифікація за угрупованнями із оціненням ваги змістовних категорій у загальному обсязі тексту. Класифікатором є загальна таблиця, до якої зведено всі категорії аналізу і одиниці аналізу. Фіксують одиниці виразу категорій.

*Крок 2.* Статистичні розрахунки зрозумілості та атрактивності контенту.

**Етап 6.** Розроблення інструменту контент-аналізу тексту.

*Крок 1.* Створення закодованого протоколу контенту для компактності подання даних та швидкого порівняння результатів аналізу різного контенту.

*Крок 2.* Заповнення протоколу контенту властивостями (автор, час публікації, обсяг тощо).

*Крок 3.* Заповнення протоколу контенту маркованими словами за параметрами великої п'ятірки.

*Крок 4.* Заповнення протоколу контенту підсумками його аналізу (кількість вживання в ньому певних одиниць аналізу і висновки щодо категорій аналізу). Протокол кожного контенту заповнюють на основі підрахунку даних всіх його реєстраційних карток.

**Етап 7.** Розроблення таблиці контент-аналізу. Тип таблиці визначають у вигляді системи скоординованих і субординованих категорій аналізу: кожна категорія (питання) передбачає ряд ознак (відповідей), за якими квантифікується зміст тексту.

**Етап 8.** Розроблення кодувальної матриці контент-аналізу.

*Крок 1.* Якщо обсяг вибірки  $\geq k$  одиниць, то аналізується набір матричних листів, інакше виконати крок 2.

*Крок 2.* Якщо вибірка  $< k$  одиниць, то проводиться двовимірний аналіз. У цьому випадку для кожного контенту формується кодувальна матриця.

**Етап 9.** Аналіз тексту згідно зі створеними кодувальними матрицями.

**Етап 10.** Інтерпретація результатів. Виявляють і оцінюють характеристики контенту на основі статистичного набору підрахованих коефіцієнтів за певний період часу на визначену категорію. Охоплює всі здобуті фрагменти тексту, висновки спираються не на частину результатів, а враховуються всі без винятку.

Пропонуємо використати алгоритми аналізу синтаксису україномовних та англійськомовних текстів опрацювання та контент-аналізу (етап 3 алгоритму) великих масивів текстових даних для знаходження та аналізу маркованих слів.

Ця ІС орієнтована на застосування у мережі Інтернет, тому при виборі інструментів та засобів реалізації було звернено увагу саме на ті технології, які дадуть змогу реалізувати необхідний програмний продукт. У процесі розроблення ІС можуть виникнути певні проблеми, адже достатньо довго усі інтернет-технології не мали одного окремого і визнаного стандарту. Звісно, з приходом

HTML5 та CSS3 та випуском Microsoft браузера MS Edge ситуація кардинально змінюється, проте все ж залишається проблема зворотної сумісності сайтів. Вибираючи архітектурне рішення, необхідно використати дві частини: серверну та клієнтську.

*Серверна частина* – це комп'ютер або сервер, який містить базу даних, у якій зберігається необхідна інформація та програмні компоненти, які реалізовуватимуть ті чи інші функції системи.

*Клієнтська частина* – це середовище (браузер), яке дає змогу відобразити запитовану інформацію. У цьому випадку вона дозволяє відобразити веб-сайт, його сторінки, вміст тощо.

Обидві частини тісно пов'язані одна з однією та функціонують як одне ціле.

У цьому випадку апаратне забезпечення серверної частини являтиме собою віддалений сервер. Тобто немає необхідності створювати окремий сервер для проекту, адже в процесі роботи програми відсутні дані, які можна обробляти на стороні сервера. Просто в майбутньому його реалізація необхідна.

Клієнтська частина міститиметься на пристрої користувача (ПК, ноутбук, планшет чи інший пристрій) та відображатиметься у браузері.

Розробляючи цей сервіс, варто зважати на те, як він поводитиметься на різних пристроях, із різними діагоналями екранів та різними браузерами. Сьогодні особливої популярності набули планшети та смартфони як наймобільніші пристрої. Це призвело до того, що для перегляду контенту на сайті довелося використовувати масштабування, яке часто відображало вміст некоректно. Тому для вирішення цих проблем застосовують так звані кросплатформеність і кросбраузерність.

Кросплатформеність дає змогу адаптувати веб-сайт до екранів із невеликими діагоналями і зосереджувала на сайті лише найнеобхідніший контент. Це зменшує час завантаження веб-сторінки, що стало одним із ключових моментів. Досягти кросплатформеності можна за допомогою медіазапитів та використання “резинових” макетів.

Кросбраузерність призначена для того, щоб сайт на усіх популярних браузерах відображався однаково, адже кожен браузер мав свої технології реалізації певних функцій. Тому застосовуються різні окремі скрипти та запити для того, аби сайт виглядав однаково у будь-якому браузері.

Сьогодні існують такі популярні браузери:

1. Google Chrome
2. Mozilla Firefox
3. Opera
4. Safari
5. Internet Explorer
6. Microsoft Edge.

Google Chrome та Opera (починаючи з 10 версії) – це по суті один і той самий браузер, адже вони використовують один спільний рушій. Сьогодні це найпопулярніші засоби для перегляду сайтів. Internet Explorer, особливо до 8 версії, не підтримував жодних сучасних стандартів, що призводило до того, що сайти, які коректно відображались у всіх інших браузерах, у цих версіях відображались із помилками. Проте ця ситуація нарешті буде змінена із випуском Windows 10 та нового браузера Microsoft Edge. Новий браузер обіцяє обходити у всьому навіть теперішнього лідера – Google Chrome, та підтримуватиме усі сучасні WEB-стандарты.

Під час розроблення будь-якого веб-сервісу буде використано мову гіпертекстової розмітки (HTML) та таблиці каскадних стилів (CSS). Оскільки останнім часом практично всі браузери почали підтримувати основні останні стандарти, варто звернути увагу на останні версії цих технологій, а саме HTML5 та CSS3, які дають змогу не використовувати на сайті застарілих технологій, таких як Flash. Для написання програм, тобто скриптів вибір було зупинено на мові JavaScript, яка в останні роки набирає шаленої популярності серед веб-розробників. Її перевагами є простота освоєння та гнучкість у роботі. Також для цієї мови написано величезну кількість готових фреймворків, які спрощують розроблення веб-застосувань. Отже, для створення макета сторінки (інакше називається front-end) було обрано такі технології: HTML5, CSS3, JavaScript. Також було обрано фреймворк AngularJS, який дає змогу значно розширити можливості JavaScript.



Для створення серверної частини (back-end) було обрано платформу Node.js. Ця платформа дозволяє використовувати мову JavaScript на серверній частині, отже, для реалізації веб-застосунку не потрібно додатково вчити інші мови програмування, як от PHP чи Java.

Проаналізувавши усі необхідні вимоги та обравши конкретні мови для реалізації програми, було вирішено використовувати середовище виконання Electron. Це середовище базується на браузері Google Chrome та містить рушій V8, який дає змогу швидко виконувати серверний JavaScript, а також використовувати технології HTML5 та CSS3. Це середовище виконання є кросплатформним, а його реалізація на базі браузера Google Chrome дає змогу відкинути підтримку інших браузерів, таких, як морально застарілий Internet Explorer.

Принцип роботи Electron оснований на процесах двох типів.

Перший тип – основний процес, який відповідає за інтеграцію та взаємодію з інтерфейсом операційної системи (ОС). Це поняття означає інтеграцію в док у OS X, або панель задач на Windows, а також згортання в трей, повноекранний режим та інші звичайні для будь-якої ОС речі. Такий процес може бути запущено лише один раз до завершення виконання програми.

Другий тип – процес рендерингу, що відповідає за відображення вікна браузера, в якому за допомогою одного рядка може бути відкрита сторінка додатка або будь-яка інша веб-сторінка. Таких процесів є довільна кількість. За створення процесу рендерингу відповідає основний процес. Для того, щоб створити основний процес, використовують таку схему народження програми:

Electron читає спеціальний файл, який називається `package.json` і шукає в ньому секцію `main`, в якій визначено основний файл програми. Цей файл називають “точкою входу”. Потім відбувається обробка “точки входу” і створюється основний процес, який, своєю чергою, відкриває певну сторінку або сторінки, тобто створює вікно браузера, або процес рендерингу. На рис. 1 графічно зображено як виглядає життя додатка, побудованого із застосуванням платформи Electron. У результаті отримуємо кросплатформний веб-застосунок, який працюватиме на більшій частині десктопних операційних систем, а також, за потреби, зможе працювати як звичайний веб-сайт. При цьому зміни у структурі і робочому коді для цього будуть мінімальними, адже середовище виконання Electron за своєю суттю є мініфікованою версією браузера Google Chrome останньої стабільної версії, яка підтримує усі сучасні веб-стандарти.

Реалізацію програмного засобу буде проведено за допомогою таких трьох програмних компонентів: HTML, CSS та Javascript. HTML – це мова розмітки документів. Вона необхідна для створення структури сайту. CSS – це мова, яка описує зовнішній вигляд документа, тобто задає візуальний вигляд веб-сайту. Javascript – це мова програмування, яка в цьому випадку знадобиться для реалізації того, чого неможливо досягти у CSS, та для надання сайту інтерактивності.

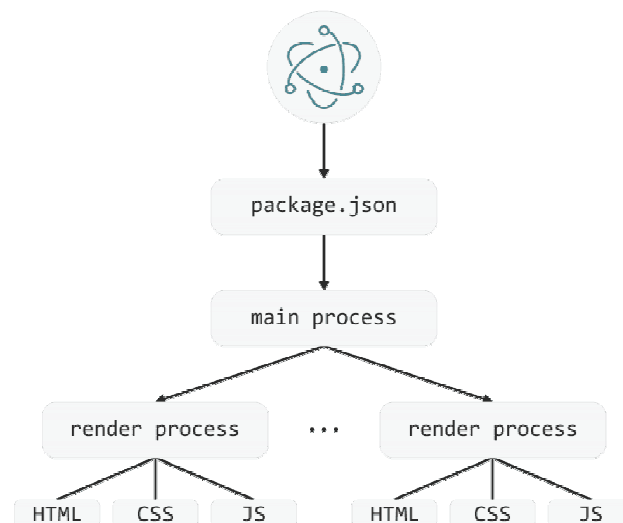


Рис. 1. Принцип роботи середовища виконання Electron

Першим кроком під час реалізації програмного засобу буде налаштування робочого оточення. Для цього у системі повинен бути встановлений Node.js. Також необхідно встановити пакет electron-prebuilt, який дасть змогу автоматично завантажувати актуальну версію Electron і запускати додатки без необхідності попереднього збирання. Для цього у командному рядку набираємо наступний код: `$ npm i -g electron-prebuilt`. Після установки пакет автоматично завантажить поточну стабільну версію Electron. Тепер для того, щоб запустити додаток, необхідно лише виконати команду `electron` у директорії проекту. Для подальшої автоматизації процесу розроблення додамо цю команду до секції скриптів у спеціальному файлі `package.json`:

```
...“scripts”: {“start”: “electron.”},...
```

Цей запис дасть можливість ініціювати запуск програми при команді `npm start`. Далі переходимо до створення власне програмного засобу. Першим кроком на початку розроблення будь-якого Node.js застосунку буде створення файлу `package.json`. Його можна створити в напівавтоматичному режимі за допомогою команди `npm init`, відповідаючи на питання, чи ж вручну, додавши лише найпотрібніші пункти:

```
{“name”: “AppSMA”, “version”: “0.1.0”, “main”: “./main.js”, “scripts”: {“start”: “electron.”}}
```

Наступним кроком є створення `html`-файла, який повинен мати ім'я `index.html`. Він буде головним вікном програми, яке ми побачимо після запуску веб-застосунку. Файл `main.js` є “точкою входу” для всієї програми. У ньому будуть створюватися вікна і оброблятися системні події. Спочатку необхідно підключити модуль `app`, що відповідає за управління життєвим циклом додатка, і `browser-window`, що створює нове вікно браузера.

```
var app = require('app');
var BrowserWindow = require('browser-window');
var path = require('path');
```

Потім створюють посилання на об'єкт `Window`. Це робиться для того, щоб вікно не закривалося автоматично, коли об'єкт буде оброблений складальником сміття.

```
var mainWindow = null;
```

```
index.html
1 <!DOCTYPE html>
2 <html ng-app="AppSMA">
3 <head>
4 <meta charset="utf-8">
5 <meta http-equiv="X-UA-Compatible" content="IE=edge">
6 <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1" />
7 <title>SMA App</title>
8 <!-- <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/font-awesome/4.5.0/css/font-awesome.min.css" -->
9 <!-- <script src="https://ajax.googleapis.com/ajax/libs/jquery/1.12.2/jquery.min.js"></script> -->
10 <!-- Global styles -->
11 <script>
12 window.$ = window.jQuery = require('./js/jquery.min.js');
13 window.hammer = require('./bower_components/materialize/js/hammer.min.js');
14 </script>
15 <script type="text/javascript" src=".js/main.js"></script>
16 <script>
17 $(function() {
18 |   app.global.init();
19 | })
20 </script>
21 <link href="css/style.css" rel="stylesheet" type="text/css">
22 <!-- Google material icons -->
23 <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
24
25 <script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.3.11/angular.min.js"></script>
26 <!-- Auth0 lock script -->
27 <script src="http://cdn.auth0.com/js/lock-9.1.min.js"></script>
28
29 <!-- Setting the right viewport -->
30
```

a

```
main.js
1 const electron = require('electron');
2 // Module to control application life.
3 const {app} = electron;
4 // Module to create native browser window.
5 const {BrowserWindow} = electron;
6
7 // Keep a global reference of the window object, if you don't, the window will
8 // be closed automatically when the Javascript object is garbage collected.
9 let win;
10
11 function createWindow() {
12 // Create the browser window.
13 win = new BrowserWindow({width: 800, height: 700, minHeight: 640, minWidth: 480, frame: false}); /*, frame: false*/
14
15 // and Load the index.html of the app.
16 win.loadURL('file://$$_dirname/index.html');
17
18 // Open the DevTools.
19 win.webContents.openDevTools();
20
21 // Emitted when the window is closed.
22 win.on('closed', () => {
23 // Dereference the window object, usually you would store windows
24 // in an array if your app supports multi windows, this is the time
25 // when you should delete the corresponding element.
26 win = null;
27 });
28
29 //win.on('maximize', function() {
30 console.log('maximized!');
```

б

Рис. 2. Файл `index.html` (а) та `main.js` (б)

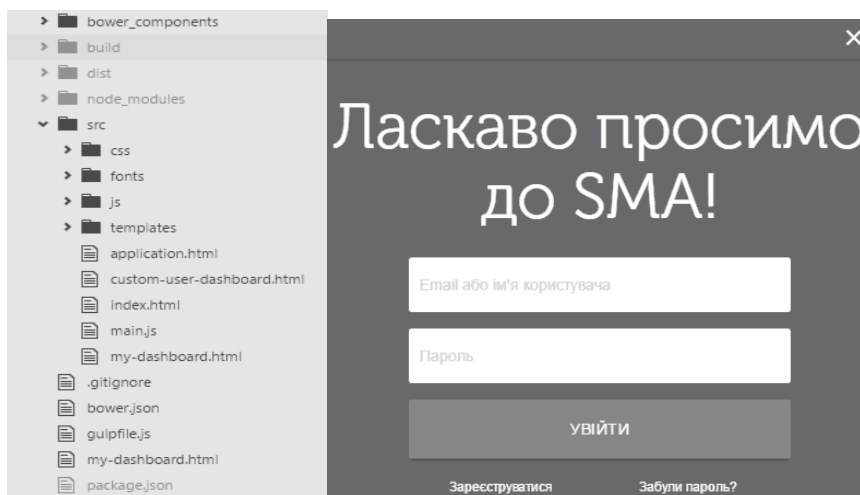
При закритті всіх вікон програми з неї вийти. В OS X ця подія є загальною для додатків та їхніх меню-барів, тому тут присутня умова, що відкидає цю платформу.

```
app.on ('window-all-closed', function () {
  if (process.platform !== 'darwin') {
    app.quit ();}});
```

Нижче наведено код, який створює нове вікно браузера розмірами 800 на 600 пікселів. Потім в цьому вікні завантажуватиметься раніше створений нами html-файл.

```
app.on ('ready', function () {
  mainWindow = new BrowserWindow ({width: 800, height: 600});
  mainWindow.loadUrl (path.join ('file://', __dirname, '/index.html'));
  mainWindow.on ('closed', function () {
    mainWindow = null;});});
```

Загальна структура проекту виглядатиме ось так:



а

б

Рис. 3. Загальна структура проекту (а) та головне вікно програми (б)

У кінці встановлюють обробник події `closed`, який виконується тоді, коли вікно браузера вже було закрито. Після того, як отримали цю подію, необхідно видалити посилання на вікно і більше його не використовувати. Після виконання усіх вищенаведених кроків можемо запустити додаток, використовуючи консольну команду `npm start`. Після створення загального макета застосунку розпочинаємо його наповнення та додаємо йому інтерактивності – реалізуємо програми, необхідні для виконання певних функцій. Як вже згадувалося, для цього використовуємо JavaScript (рис. 4).

Для створення таких функцій, як авторизація користувача та підрахунок статистики за словами необхідно використовувати JavaScript, адже у процесі виконання змінюється візуальний вигляд та наповнення деяких блоків на сайті. Це неможливо реалізувати за допомогою лише каскадних стилів. У процесі роботи створюються певні значення та ідентифікатори, які необхідні для коректного функціонування програми. Нижче наведено реалізацію авторизації користувача.

```
function submit_ (log, pas) {if (log === 123 && pas === 123) return true;
else {alert ('ERROR'); return false}}
```

Для повноцінної авторизації необхідно підключити базу даних (рис. 5).

У цій базі даних ми маємо 4 таблиці, а саме “Користувач”, “Дані акаунта у Twitter”, “Твіти” та “Результати аналізу”. Зв’язок “один до одного” існує між таблицями “Користувач” та “Дані акаунта у Twitter” а також між таблицями “Дані акаунта у Twitter” та “Результати аналізу”.

Реєстрація користувача здійснюється з використанням запити

```
INSERT INTO Користувач (PriKey, Description)
VALUES (123, 'Mark', 'ru94@ukr.net', 'e98tve4m75em5g8c4rse');
```

```

var sign = 0;
function Collector() {
  sign = 1;
}
function Analyzer() {
  if (sign == 1) {
    var words = [];
    words.length = 100;
    var letters = [];
    letters.length = 100000;
    var analyze = 1;
    if (analyze == true) {
      subAnalyzer();
      return;
    } else {
      for (var i = 0, l = words.length; i < l; i++) {
        var word = words[i], letters = word.split(""), cur = trie;

```

Рис. 4. Структура JavaScript-файла

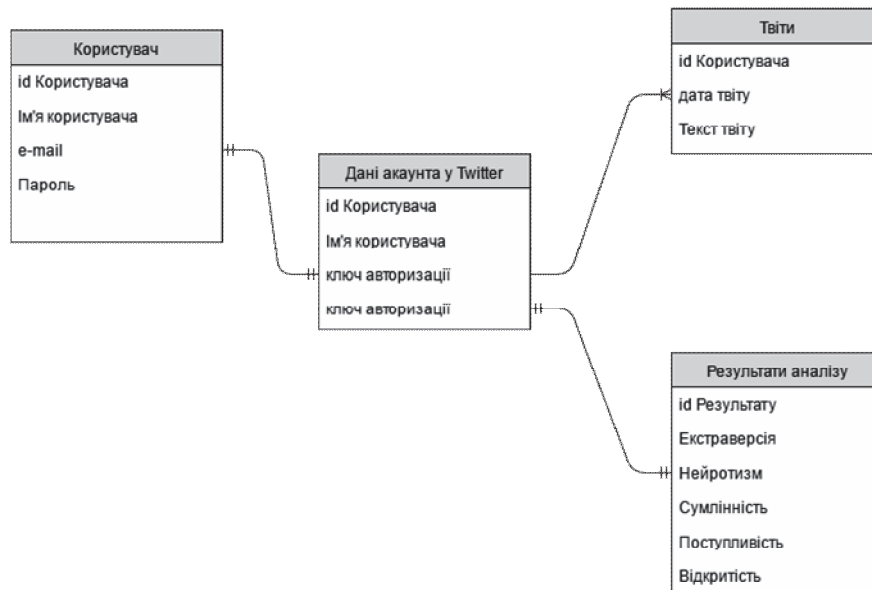


Рис. 5. ER-діаграма бази даних

Для авторизації користувача у Twitter використовується схожий до попереднього запит. Для вибору усіх твітів користувача використовуємо запит `SELECT * FROM Твіти`. Використовуючи дані з цих трьох таблиць та на підставі аналізу формуємо таблицю “Результати аналізу”. Алгоритм пошуку необхідних слів у твітах подано на рис. 6. Після реалізації основної частини програми переходимо до збирання застосунку. Для цього використовуємо пакет `electron-packager`. Спочатку встановимо його глобально: `$ npm i -g electron-packager`.

```

function SubstringSearch(sub, str)
{
  var i, j, n = sub.length,
      N = str.length - n + 1;

  for (i = 0; i < N; i++)
  {
    j = 0;
    while (j < n && sub.charAt(j) === str.charAt(i+j)) j++;
    if (j === n) return i;
  }
  return -1;
}

```

Рис. 6. Алгоритм пошуку слів у твітах

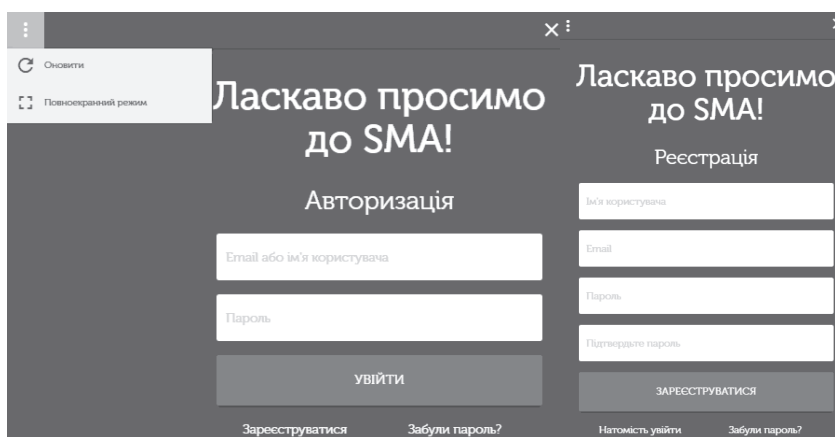
Створимо новий скрипт у package.json

```
... "scripts": { "start": "xo && electron.",  
  "build": "electron-packager. myApp --platform=win32 --arch=x64 --  
  version=0.33.3 --app-version=0.1.0 --out=dist --ignore=dist --prune"},...
```

Якщо виконати команду `npm run build` отримаємо складання застосунку під операційну систему Windows. Якщо ж нам буде необхідне складання під всі три платформи, то команда матиме вигляд:

```
electron-packager. myApp --all --version=0.33.3 --app-version=0.1.0 --out=dist --ignore=dist --prune.
```

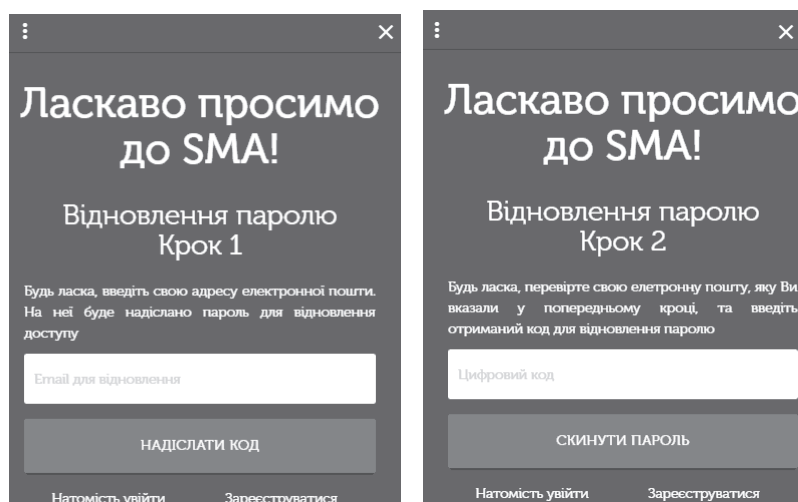
У результаті отримуємо складання веб-застосунку під усі необхідні платформи. Кожен містить у теці: `app-darwin-x64`, `app-linux-ia32`, `app-linux-x64`, `app-win32-ia32`, `app-win32-x64`. Разом усі необхідні файли матимуть розмір приблизно 500 мегабайтів, тобто приблизно по 100 мегабайтів під кожну платформу. Після розроблення інтерфейсної частини та її тестування роблять загальний огляд працездатності застосунку. Відкриваючи ресурс, користувач потрапляє на сторінку автентифікації. Тут знаходиться форма для авторизації на ресурсі. За невдалої авторизації користувач побачить помилку. Якщо користувач новий, то він може зареєструватися, перейшовши за необхідним посиланням. Якщо вже відомий користувач забув пароль, то він може відновити його, якщо перейде на сторінку відновлення паролю, яка складатиметься з двох кроків: введення електронної пошти для відновлення паролю, та сторінки, на якій користувач повинен ввести код підтвердження скидання паролю, який приходить йому на електронну пошту, вказану у попередньому кроці. Після вдалої авторизації користувач власне потрапляє у сам додаток.



а

б

Рис. 7. Сторінка авторизації (а) та реєстрації (б)



а

б

Рис. 8. Сторінка відновлення паролю

Після вдалої автентифікації користувач потрапляє на наступну сторінку ресурсу. На цій сторінці користувачу буде запропоновано пройти автентифікацію у соціальній мережі Twitter для подальшої роботи у додатку (рис. 9, а). Наступний крок: користувач потрапляє на сторінку, де йому пропонується обрати з двох варіантів: або аналізувати власний профіль у соціальній мережі, або ввести ім'я існуючого користувача для аналізу його твітів (рис. 9, б). Якщо користувача не існує, то відобразиться помилка.

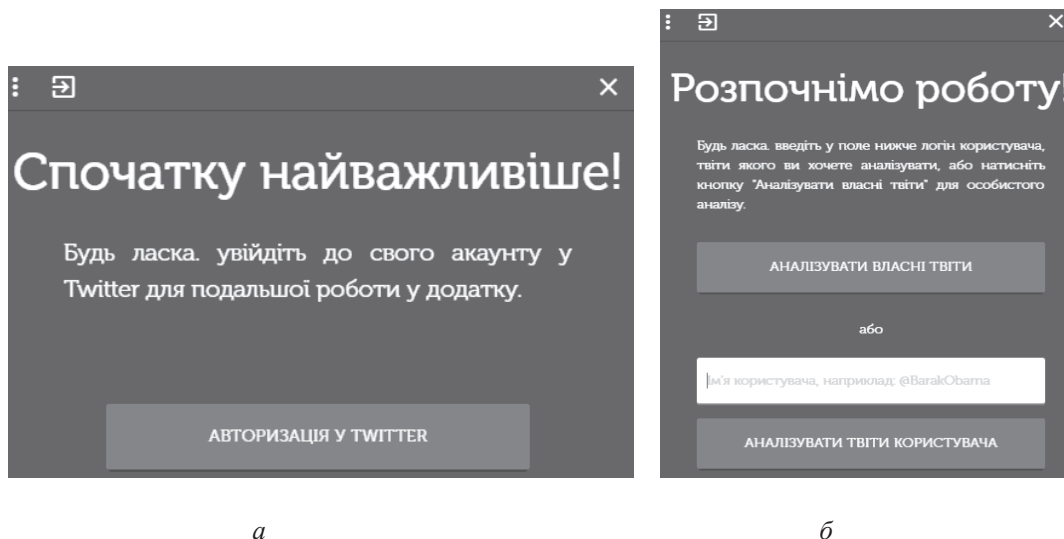
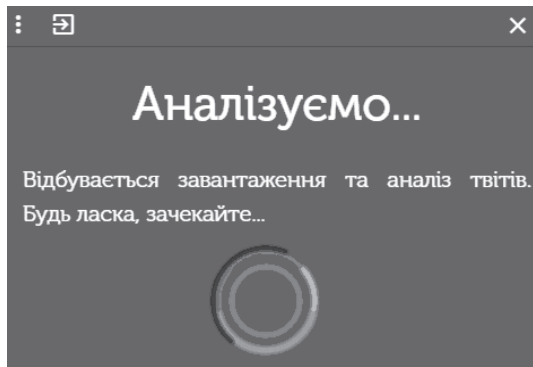


Рис. 9. Сторінка авторизації у Twitter (а) та з вибором користувача для аналізу (б)

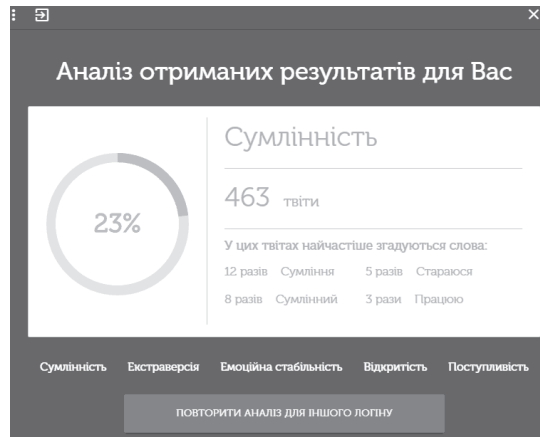
Коли необхідний пункт обрано, користувача скеровують на сторінку очікування. Водночас система шукає необхідного користувача, збирає дані цього користувача, формує архів твітів. Також аналізується текст для визначення п'яти емоційних критеріїв. Усі зібрані результати записуються до бази даних, а після виконання усіх необхідних дій користувача перескеровують на сторінку з відображенням результатів аналізу. На цій сторінці відображається відсоткове співвідношення для кожної риси, кількість твітів, а також найуживаніші слова, пов'язані із цими рисами (рис. 10). На цій сторінці користувач може перейти до аналізу будь-якого іншого користувача, натиснувши відповідне посилання. Також угорі розташовано кнопку для виходу зі свого профілю. Варто також зазначити, що цей застосунок може працювати у повноекранному режимі, перехід до якого реалізовано у спеціальному меню у верхньому лівому куті. Після перегляду застосунку можна сказати, що він працює коректно, без помилок та було досягнуто бажаного результату. Також він правильно працюватиме і на інших операційних системах, таких, як Linux та MacOS, адже є кросплатформеним. Отже, застосунок містить лише необхідну інформацію та не створює труднощів користувачу під час роботи із ним. Всі функції є чіткими та зрозумілими.

#### Висновки та перспективи подальших наукових розвідок

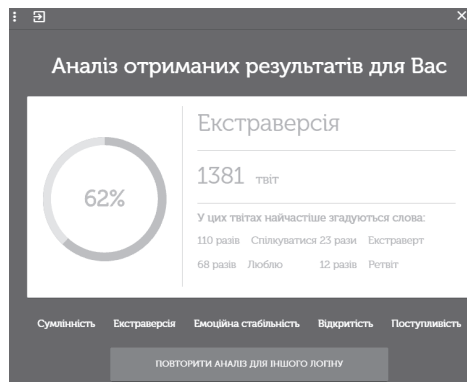
Під час виконання цієї роботи було створено та розроблено інформаційну систему, за допомогою якої можна проводити психологічний аналіз особистості, використовуючи її повідомлення із соціальної мережі Twitter. Система допомагає автоматизувати проект збирання інформації та отримання та збереження результатів. Проаналізовано предметну область, розглянуто різноманітні інформаційні системи, створені за певним принципом. Було проаналізовано загальновідомі методи та критерії розроблення інформаційних систем, а також проаналізовано системи на функціональність. Було передбачено, які функції існують у системі, та їх корисність. Здійснено системний аналіз інформаційної системи психологічного аналізу особистості, описано проблему предметної області та визначено мету проекту. За допомогою основних принципів системного підходу описано вимоги до створюваної інформаційної системи.



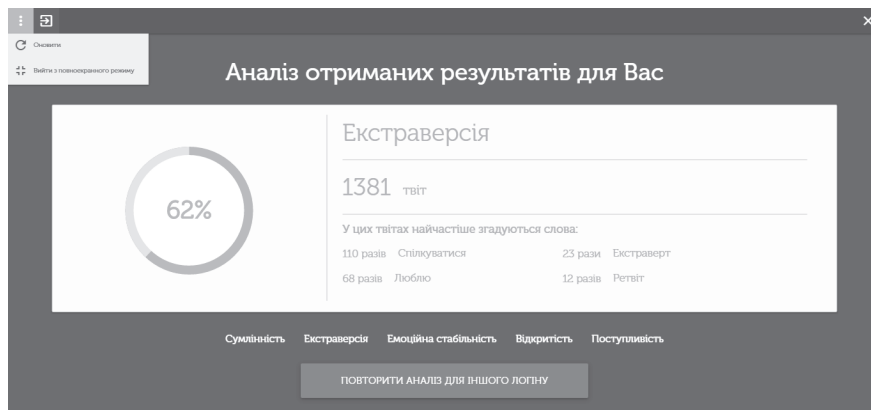
а



б



в



г

Рис. 10. Сторінка очікування (а), та з результатами аналізу для критерію “Сумлінність” (б) і “Екстраверсія” (в), повноекранний режим, контекстне меню та кнопка виходу з додатку (г)

Застосовано RUP-методологію, за допомогою якої було визначено та описано основні прецеденти системи, а також її актори. Побудовано діаграми варіантів використання, діяльності, пакетів та послідовності. Описано функції та проаналізовано використання системи групою користувачів. Описано основний алгоритм роботи цієї інформаційної системи. Описано основні методи вирішення поставленої проблеми. Порівняно різноманітні методики аналізу великих обсягів тексту та упорядковано великі масиви текстової інформації. Для опрацювання повідомлень користувача у Twitter було обрано, по-перше, комбінування інтент-аналізу та психосемантики як найефективніших засобів визначення особистісних рис користувача; по-друге, було застосовано

алгоритм БМ-пошуку як найефективніший з усіх вищеперелічених алгоритмів та не надто складний у реалізації. Також було проаналізовано та обрано необхідні засоби вирішення проблеми. Було обрано необхідну платформу (Electron) для побудови веб-застосунку, а також мови програмування, які використовуватимуть у процесі розроблення програмного рішення.

1. Lovakov A. Оценка личности по активности в социальных сетях, или Big Data приходят в психологию [Электронный ресурс] / А. Lovakov. Режим доступа: [http://psyresearchdigest.blogspot.com/2013\\_11\\_01\\_archive.html](http://psyresearchdigest.blogspot.com/2013_11_01_archive.html). – Назва з екрана. 2. Ализар Анатолий. Составление модели личности по активности в социальной сети [Электронный ресурс] / Анатолий Ализар. Режим доступа: <https://xakep.ru/2012/04/26/58618/>. – Назва з екрана. 3. Shuotian Bai. Big-Five Personality Prediction Based on User Behaviors at Social Network Sites [Электронный ресурс] / Shuotian Bai, Tingshao Zhu, Li Cheng. Режим доступа: <http://arxiv.org/pdf/1204.4809v1.pdf>, <http://arxiv.org/abs/1204.4809>. – Назва з екрана. 4. Shuotian Bai. List of computer science publications by Shuotian Bai [Электронный ресурс] / Shuotian Bai. Режим доступа: <http://dblp.uni-trier.de/pers/hd/b/Bai:Shuotian>. – Назва з екрана. 5. Соловьев Дмитрий. Потребности и поведение людей в социальных сетях. Теория “лайков” [Электронный ресурс] / Дмитрий Соловьев. Режим доступа: <http://www.cossa.ru/234/13291/>. – Назва з екрана. 6. Джефери Мэтью. Рекрутинг 5.0: Психологические профили в социальных сетях [Электронный ресурс] / Мэтью Джефери, Перевод Татьяна Горбань. Режим доступа: <http://www.hr-portal.ru/blog/rekruting-50-psihologicheskie-profiliv-socialnyh-setyah>. – Назва з екрана. 7. Прохоров Александр. Социальные сети: психология, социология, бизнес [Электронный ресурс] / Александр Прохоров. Режим доступа: <http://compress.ru/article.aspx?id=23890>. – Назва з екрана. 8. Global Web Index. Social Web Involvement [Электронный ресурс] / Global Web Index. Режим доступа: <file:///H:/Global-Map-of-Social-Web-Involvement-Global-Web-Index-2009.pdf>. – Назва з екрана. 9. Bennett Jeff. Visualization Critique [Электронный ресурс] / Jeff Bennett. Режим доступа: <http://vizthinker.com/visualization-critique/>. – Posted on October 28, 2012. 10. Kluemper D. H., Social Networking Websites, Personality Ratings, and the Organizational Context: More Than Meets the Eye? / D. H. Kluemper, P. A. Rosen, K. W. Mossholder // *Journal of Applied Social Psychology*. – 2012 – С. 42(5), С. 1143–1172. 11. Schwartz, H. A., Personality, Gender, and Age in the Language of Social Media: The Open-Vocabulary Approach. / H. A. Schwartz, J. C. Eichstaedt, M. L. Kern // *Plos One*. – 2013 – С. 8(9). 12. Kosinski, M., Private traits and attributes are predictable from digital records of human behavior. / M. Kosinski, D. Stillwell, T. Graepel // *Proceedings of the National Academy of Sciences*. – 2013 – С. 110(15), 5802–5805. 13. StatSoft. Скоринг [Электронный ресурс] / StatSoft. Режим доступа: <http://www.statsoft.ru/solutions/tasks/scoring/>. – Назва з екрана. 14. Як банки оцінюють позичальників? – Режим доступа: <http://kredytor.pp.ua/yak-banku-otsinyuyut-pozychalnykiv.html>. – Назва з екрана. 15. Скиба С. А. Социальный скоринг [Электронный ресурс] / С. А. Скиба // Методический журнал “Банковский ритейл”. – Режим доступа: [http://www.reglament.net/bank/retail/2013\\_4/get\\_article.htm?id=2934](http://www.reglament.net/bank/retail/2013_4/get_article.htm?id=2934). – Назва з екрана. 16. Schaefer Marketing Solutions. Get ready. Social scoring will change your life [Электронный ресурс] / Schaefer Marketing Solutions. – Режим доступа: <http://www.businessesgrow.com/2010/11/22/get-ready-social-scoring-will-change-your-life/>. – Назва з екрана. 17. Скиба С. А. Социальный скоринг [Электронный ресурс] / С. А. Скиба, В. И. Лойко // Журнал Политематический сетевой электронный научный журнал Кубанского государственного аграрного университета. – Вып. № 91 / 2013. – <http://cyberleninka.ru/article/n/sotsialnyu-skoring>. – Назва з екрана. 18. Рукосуева И. Банкиры рассматривают перспективы кредитного скоринга в социальных сетях [Электронный ресурс] / И. Рукосуева // Профессиональный финансовый портал Bankir.ru. – Режим доступа: <http://bankir.ru/publikacii/20131209/pustaya-istoriya-10004344/>. – Назва з екрана. 19. Филонов Д. Лайк от банка: как интернет помогает проверять надежность заемщиков [Электронный ресурс] / Д. Филонов // Финансово-экономический журнал Forbes. – Режим доступа: <http://www.forbes.ru/tekhnologii/internet-i-svyaz/256483-laik-ot-banka-kak-internet-pomogaet-proveryat-nadezhnost-zaemshc>. – Назва з титулки екрану. 20. HR-Portal. HR-специалистов изучают



соціальні мережі соискателей [Електронний ресурс] / HR-Portal // Професійний портал Сообщество HR-Менеджеров. – Режим доступу: [http://hr-portal.ru/article/53-hr-specialistov-izuchayut-socialnye-seti-soiskateley?utm\\_source=site&utm\\_medium=block&utm\\_campaign=analog65](http://hr-portal.ru/article/53-hr-specialistov-izuchayut-socialnye-seti-soiskateley?utm_source=site&utm_medium=block&utm_campaign=analog65). – Назва з титулки екрана. 21. Аликина Е. Заемщик с хитрым профилем [Електронний ресурс]/ Е. Аликина // Журнал “Коммерсантъ Деньги” №42. – Режим доступу: <http://kommersant.ru/doc/2585025>. – Назва з титулки екрана. 22. Дорохов Р. Кредит по “лайкам” [Електронний ресурс]/ Р. Дорохов // Ежедневная деловая газета. Режим доступу: <http://www.vedomosti.ru/newspaper/articles/2014/04/28/kredit-po-lajkam>. – Назва з екрана. 23. *Methods based on ontologies for information resources processing : Monograph* / [Vasyl Lytvyn, Victoria Vysotska, Lyubomyr Chyrun, Dmytro Dosyn] // LAP Lambert Academic Publishing. Saarbrücken, Germany. – ISBN-13: 978-3-659-89905-8, ISBN-10: 3659899054, EAN: 9783659899058. – 2016. – 324 с. – Access mode: <https://www.lap-publishing.com/catalog/details/store/gb/book/978-3-659-89905-8/methods-based-on-ontologies-for-information-resources-processing?locale=gb>. 24. Берко А. Ю. Системи електронної контент-комерції: монографія / А. Ю. Берко, В. А. Висоцька, В. В. Пасічник. – Львів: Вид-во Нац. ун-ту “Львівська політехніка”, 2009. – 612 с. 25. Математична лінгвістика. [Книга 1. Квантитативна лінгвістика]: навч. посібник / [В. В. Пасічник, Ю. М. Щербина, В. А. Висоцька, Т. В. Шестакевич] // Серія “Комп’ютинг”. – Львів: “Новий світ-2000”, 2012. – 359 с. 26. Висоцька В. А. Методи і засоби опрацювання інформаційних ресурсів в системах електронної контент-комерції: автореф. дис. ... канд. техн. наук: 05.13.06 – інформаційні технології / Вікторія Анатоліївна Висоцька; Нац. ун-т “Львівська політехніка”. – Львів, 2014. – 27 с. 27. Vysotska V. Web Content Processing Method for Electronic Business Systems / V. Vysotska, L. Chyrun // International Journal of Computers & Technology. – Vol. 12, No 2. – December 2013. – P. 3211–3220. – ISSN 2277-3061. – [Online] <http://cirworld.org/journals/index.php/ijct/article/view/3299>. Impacr factor 1,532. (Index Copernicus, NASA ADS, DOAJ, Google Scholar, Eyesource, EBSCO, CiteSeer, UlrichWeb, ScientificCommons, ProQuest CSA Technology Research Database). 28. Висоцька В. А. Моделювання етапів життєвого циклу комерційного web-контенту / В. А. Висоцька, Л. В. Чирун, Л. В. Чирун // Інформаційні системи та мережі. Вісник Нац. ун-ту “Львівська політехніка”. – Львів, 2011. – № 715. – С. 69–87. 29. Висоцька В. А. Особливості проектування та впровадження систем електронної комерції. / В. А. Висоцька // Інформаційні системи та мережі. Вісник Нац. ун-ту “Львівська політехніка”. – Львів, 2008. – № 631. – С. 55–84. 30. Vysotska V. Analysis and evaluation of risks in electronic commerce / V. Vysotska, I. Rishnyak, L. Chyrun // CAD Systems in Microelectronics, CADSM '07, 9th International Conference. – The Experience of Designing and Applications of CAD Systems in Microelectronics. – Lviv, 24 February 2007. – P. 332–333. 31. Vysotska V. Set-theoretic models and unified methods of information resources processing in e-business systems / V. Vysotska, L. Chyrun // Applied Computer Science. ACS journal. – Vol. 10, Number 3. – Poland, 2014. – ISSN 2353-6977 (Online), ISSN 1895-3735 (Print). – P. 5–22 [Online]. – Режим доступу: [www.acs.pollub.pl](http://www.acs.pollub.pl), <http://www.acs.pollub.pl/pdf/v10n3/1.pdf>. 32. Vysotska V. Generative regular grammars application to modeling the semantics of sentences in natural language / V. Vysotska // Комп’ютерні системи проектування. Теорія і практика: Вісник Нац. ун-ту “Львівська політехніка”. – 2014. – № 808. – С.43–56. 33. Висоцька В. А. Побудова та функціонування систем електронної контент-комерції / В. А. Висоцька, Л. В. Чирун // Комп’ютерні науки та інформаційні технології, Вісник Нац. ун-ту “Львівська політехніка”. – 2014. – № 800. – С. 217–227. 34. Висоцька В. А. Концептуальна модель процесу формування семантики речення природною мовою / В. А. Висоцька // Інформаційні системи та мережі. Вісник Нац. ун-ту “Львівська політехніка”. – 2014. – № 805. – С. 258–278. 35. Берко А. Ю. Теоретико-множинна модель систем електронної контент-комерції / А. Ю. Берко, В. А. Висоцька, Л. В. Чирун // Комп’ютерні науки та інформаційні технології, Вісник Нац. ун-ту “Львівська політехніка”. – 2014. – № 800. – С. 105–114. 36. Висоцька В. А. Архітектура систем електронної контент-комерції / В. А. Висоцька, Л. В. Чирун // Інформаційні системи та мережі. Вісник Нац. ун-ту “Львівська політехніка”. – 2014. – № 783. – С. 39–55. 37. Берко А. Ю. Створення та застосування систем електронної контент-комерції / А. Ю. Берко, В. А. Висоцька, Л. В. Чирун // Інформаційні системи та мережі. Вісник Нац. ун-ту “Львівська політехніка”. –

2014. – № 805. – С. 23–36. 38. Висоцька В. А. Особливості генерування семантики речення природною мовою за допомогою породжувальних необмежених та контекстно-залежних граматик / В. А. Висоцька // Інформаційні системи та мережі. Вісник Нац. ун-ту “Львівська політехніка”. – 2014. – № 783. – С. 271–292. 39. Berko A. Functionally logistic model of commercial content processing / A. Berko, V. Vysotska, L. Chyrun // Комп’ютерні системи проектування. Теорія і практика, Вісник Нац. ун-ту “Львівська політехніка”. – 2013. – № 777. – С. 30–38. 40. Vysotska V. Life Cycle Model of Commercial Content Processing in Electronic Commerce System / V. Vysotska, L. Chyrun // Computational Problems of Electrical Engineering. Founder and Publisher Lviv Polytechnic National University. – Vol. 3, No 2. – Lviv, 2014. – ISSN 2224-0977. P. 118–122 [Online]. – [http://vlp.com.ua/files/special/24\\_137.pdf](http://vlp.com.ua/files/special/24_137.pdf). 41. Висоцька В. А. Математичні моделі інформаційних потоків у системах електронної контент-комерції / В. А. Висоцька // Комп’ютерні науки та інформаційні технології, Вісник Нац. ун-ту “Львівська політехніка”. – 2010. – № 672. – С. 185–195. 42. Висоцька В. А. Схеми моделювання систем електронної контент-комерції / В. А. Висоцька // Комп’ютерні науки та інформаційні технології, Вісник Нац. ун-ту “Львівська політехніка”. – 2010. – № 686. – С. 112–121. 43. Берко А. Ю. Інтелектуальна система прийняття маркетингових рішень з поширення електронних видань / А. Ю. Берко, В. А. Висоцька, О. Ю. Михайлов // Комп’ютерні науки та інформаційні технології, Вісник Нац. ун-ту “Львівська політехніка”. – 2010. – № 672. – С. 7–18. 44. Чирун Л. В. Методи та засоби опрацювання інформаційних ресурсів в системах електронної контент-комерції / Л. В. Чирун, В. А. Висоцька // Наукові праці Чорноморського держ. Університету ім. П. Могили: Комп’ютерні технології. – 2010. – Т. 117. – Вип. 104. – С. 188–202. 45. Висоцька В. А. Моделювання систем електронної контент-комерції за допомогою мереж Петрі / В. А. Висоцька // Комп’ютерні науки та інформаційні технології, Вісник Нац. ун-ту “Львівська політехніка”. – 2009. – № 638. – С. 99–109. 46. Щербина Ю. М. Методи та розділи математичної лінгвістики в структурі курсу “комп’ютерна лінгвістика” / Ю. М. Щербина, В. А. Висоцька, Т. В. Шестакевич // Наукові праці Чорноморського держ. університету ім. П. Могили: Комп’ютерні технології. – 2009. – Вип. 104. – С. 203–209. 47. Висоцька В. А. Математичні моделі керування контентом в системах електронної комерції: тези науково-практичної конференції, 14 травня 2009 р. / В. А. Висоцька // III міжнар. конф. молодих вчених “Комп’ютерні науки та інженерія (CSE-2009)”. – 2009. – С. 189–193. 48. Висоцька В. А. Проектування та реалізація інтелектуальних інформаційних систем інтернет-аукціону / В. А. Висоцька, А. В. Овчаренко, Л. В. Чирун // Інформаційні системи та мережі. Вісник Нац. ун-ту “Львівська політехніка”. – 2008. – № 610. – С. 70–84.