

ІНТЕЛЕКТУАЛЬНА СИСТЕМА СОЦІАЛІЗАЦІЇ ГРУП КОРИСТУВАЧІВ ЗА ПОДІБНИМИ ІНТЕРЕСАМИ

Тарас Батюк¹, Вікторія Висоцька^{1,2}, Марина Шевченко³

¹ Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж, Львів, Україна

² Університет Оснабрюка, Інститут комп’ютерних наук, Оснабрюк, Німеччина

³ Університет Оснабрюка, кафедра міжнародної економічної політики,
Оснабрюк, Німеччина

E-mail: taras.batiuk.mnsa.2020@lpnu.ua, ORCID: 0000-0001-5797-594X,

E-mail: Victoria.A.Vysotska@lpnu.ua, ORCID: 0000-0001-6417-3689

E-mail: mshevchenko@uni-osnabrueck.de, ORCID: 0000-0003-2165-9907

© Батюк Т. М., Висоцька В. А., Шевченко М. М., 2023

Проаналізовано загальну архітектуру системи соціалізації груп користувачів за подібними інтересами та функціональні вимоги до неї. Для опрацювання великої частини інформації систему реалізовано із використанням методів нечіткого пошуку текстової інформації та машинного навчання. Зокрема, застосовано алгоритми N-грам, розширення вибірки та структурованої моделі Noisy Channel. Особливістю інтелектуальної системи є опрацювання тексту, аналіз слів у тексті та формування оцінок. Реалізація згорткової нейронної мережі призначена для визначення справжності користувачів на основі аналізу фотографій із обличчям. Використання алгоритмів нечіткого пошуку дає змогу виконувати опрацювання текстових даних різних обсягів для аналізу інформації про кожного користувача, формування певного рейтингу користувача, порівняння цього користувача з іншими користувачами для спрощення подальшої соціалізації користувачів, інтереси яких збігаються найбільше. Під час експериментальної апробації порівняно точність функціонування розробленої інтелектуальної системи через визначення відсотка подібності поточних користувачів за допомогою алгоритмів N-грам та їх поєднання. Одночасне виконання цих алгоритмів є приблизно на 15 % ефективнішим і точнішим від алгоритму N-грам і на 10 % від інших відомих алгоритмів. Проаналізовано також роботу алгоритму лінійного пошуку тегів у словнику та алгоритму Noisy Channel із використанням ВК-дерева. Завдяки цьому вдалося досягти значної переваги в роботі алгоритму, оскільки замість лінійної залежності часу пошуку отримано логарифмічну залежність. Також порівняно роботу синхронних та асинхронних методів системи. На початку роботи відмінність не дуже відчутна, але у міру зростання кількості запитів система навантажується і відмінності стають відчутнішими.

Ключові слова: згорткова нейронна мережа; нечіткий пошук; розширення вибірки; алгоритм N-грам; Noisy Channel; JWT токени.

Вступ

Тема соціалізації у суспільстві актуальна завжди, одна із основних властивостей суспільних відносин – налагодження комунікації та взаємодії з іншими людьми. Людина – істота соціальна, соціалізація і взаємодія з іншими людьми є важливим аспектом її життя, під час спілкування людина розширює своє коло інтересів, дізнається багато нового і цікавого, ділиться своїми знаннями з

іншими людьми. Окрім спрощення можливостей соціалізації та спілкування окремих користувачів, важливим завданням є спрощення соціалізації груп користувачів, тобто створення для користувачів можливості об'єднуватися в мережі відповідно до тематики, цікавої для них. Ця потреба важлива, оскільки існує безліч різних тем, якими цікавляться люди, і відповідно формується безліч спільнот користувачів, які потребують спілкування у межах певної теми.

Аналіз останніх досліджень та публікацій

Створення інтелектуальної системи, основна мета якої – взаємодія людей, потребує використання різних технологій, через значну кількість різноманітних соціальних мереж, кожна з яких виконує певні функції. Засновано багато соціальних мереж, які так і не стали популярними, не мали функціоналу, що відрізняв б їх від інших вже популярних соціальних мереж та не мали певних ознак, які б могли заохотити користувача в ній залишитися. Створення інтелектуальної системи соціалізації груп користувачів за сукупними інтересами – актуальне завдання, що потребує визначення методів, які використовуватимуться для її розроблення, щоб досягти успіху. Для того, щоб визначити, якою саме має бути інтелектуальна система, які функції необхідні, а яких краще уникнути, потрібно проаналізувати результати вже виконаних наукових досліджень. Аналіз наукових статей дав змогу дізнатися, які напрями в галузі соціальних комунікацій та створенні інтелектуальних систем соціалізації актуальні, які підходи ефективні у використанні, а яких варто уникнути через певні недоліки в реалізації або в систематизації. Також доцільно проаналізувати інформаційні системи, найближчі за функціоналом до створюваної інтелектуальної системи, завдяки чому можна проаналізувати, які переваги та недоліки наявних систем, як саме вони впливають на користувача та роботу інформаційної системи загалом.

У статті [1] автори проаналізували надзвичайно важливу тему соціалізації в популярній соціальній мережі Twitter, особливістю якої є те, що одна публікація може містити максимум 280 символів та до чотирьох медіафайлів. Це, на перший погляд, створює серйозні обмеження для користувачів такої системи, але насправді це унікальна особливість, яка є важливою перевагою соціальної мережі, оскільки змушує писати унікальні публікації, в яких дотримані ці обмеження. У статті [2] розглянуто ефективні підходи до збереження даних у базі даних та алгоритми швидкого доступу до них, які надають змогу забезпечувати швидку та безперебійну роботу соціальної мережі та створювати умови для зручного користування нею.

У публікації [3] розглянуто фактори взаємодії користувачів, оцінку позитивних повідомлень особи та відповідь на ці повідомлення вибраних користувачів, що дає змогу досліджувати закономірності та параметри даних користувачів системи, автоматизувати та оптимізувати роботу інтелектуальної системи. У статті [4] подано шаблон знань, отриманий на основі оцінки соціальних мереж абсолютно різними групами користувачів, відзначено, що обмеження парних параметрів, отримані унаслідок взаємодії людей та оцінки повідомлень, написаних за допомогою користувачів, визначено найрезультативнішими. Автор розглянули використання єдиного фреймворка певного виду для соціалізації клієнтів всередині соціальної мережі Twitter, основні переваги формування хештегів і, відповідно, використання згенерованих хештегів клієнтами з винятковими параметрами входу. У публікації зазначено, що найбільше алгоритмів, які потрібно чітко оптимізувати, щоб швидко та ефективно аналізувати дані про клієнтів у соціальних мережах, сформовано тому, що аналіз запитів може додатково вплинути на оцінки споживчої інформації в соціальних мережах.

У статті [5] описано основні стандарти побудови найтонших серверних структур, які дають змогу використовувати зручні паралельні запити та надавати відповідь до запитів асинхронно, формуючи відповідь на поточний запит частинами, зберігаючи всю необхідну проміжкову інформацію. Розглянуто прийоми зберігання записів, визначення правдивості статистичних даних, які залишає споживач, виділено більшість способів уберегтися від втрат та спотворення статистики, проаналізовано методи оптимізації роботи пристрою зберігання інформації. Автори наукової розвідки [6] описують використання сучасних нейронних мереж у контексті соціальної взаємодії людей,

ззначаючи, що через певні проміжки часу соціальні мережі стануть проблемою, розкривають сучасні види оцінки споживчого текстового контенту.

Автор статті [7] показує позитивні механізми, розроблені для взаємодії клієнтських і серверних систем, висвітлює більшість переваг передавання та зберігання проміжних записів, запобігаючи втраті фактів і створюючи проміжні запити, отримані від користувачів системи. В статті [8] проаналізовано використання сучасних нейронних мереж у популярних соціальних мережах для оптимізації та автоматизації позитивних рішень, що мають точний діапазон застосувань для оцінювання аудіо, відео, формування фотографій та статистичної інформації. Автор наукової публікації [9] проаналізував підходи до з'ясування наявності клієнтів на фото в соціальній спільноті Instagram, спроектував механізми оптимізації оцінки обмеження зображень нейронними мережами, продемонструвавши це на прикладі декількох взаємопов'язаних виділених об'єктів на потрібних фотографіях у соціальній мережі Instagram. У статті [10] визначено імперативні механізми взаємодії клієнтів між собою в різних соціальних мережах, запропоновано нейронну мережу, яка оволодіває повідомленнями клієнтів і набирає шаблон із повідомлень, використовуючи необхідні параметри. Серед нагальних проблем названо можливість всебічного вивчення нейронної мережі, що дає змогу оптимізувати та автоматизувати роботу соціальних мереж, а також використовувати зручні можливості взаємодії споживачів. Оцінювання взаємодії користувачів, їхніх запитів та входження у низку соціалізаційних структур та інтернет-магазинів, вважають дослідники, є нагальною темою [11]. У науковій публікації [12] висвітлено реалізацію механізмів налаштувань оцінювання користувачів за наявності певних нормалізованих параметрів та додаткової структури опрацьованих записів для автоматизації взаємодії між ними, а також розглянуто ключові ідеї методу медіаархівів, що завантажують користувачі, переважно йдеться про зображення та інші медіафайли, описано основні концепції асинхронного зберігання інформації.

У статті [13] висвітлено узагальнювальні алгоритми лінійного пошуку потрібної інформації про клієнтів для визначення життєво важливих параметрів взаємодії користувачів і системи, а отже, елементів, що впливають на розміщення позитивних фактів у профілях клієнтів, здійснено оцінку отриманої статистики. Автори наукової розвідки [14], розмірковуючи про важливі механізми соціалізації клієнтів, які використовують регулярні хеш-теги в соціальній мережі Twitter, показали вибірки взаємодії споживачів. Висвітлено особливості системи, що аналізує вподобання клієнтів соціальних мереж, використовуючи алгоритми нечіткого пошуку текстового контенту, чітко визначає побажання користувачів. Незважаючи на низку завдань, кількісні характеристики пошуку та методи вивчення клієнтів, на думку дослідників [15], продовжують залишатися серйозними, потребують безперервної оптимізації алгоритми для оцінювання текстового вмісту та створення нових та оптимізованих алгоритмів для постопрацювання системної інформації як результатів обміну та пошуку записів. Оцінку та опрацювання текстового вмісту записів у кількох структурах програмних пристроїв і, відповідно, визначення думок користувачів та прогнозування позитивних зрушень висвітлено у публікації [16]. Автори реалізували нейронну мережу, що визначає почуття та настрої клієнтів за допомогою перегляду багатьох фотографій користувача, нормалізації інформації та отримання попередніх даних. Дослідники висвітлюють механізми соціалізації клієнтів у соціальній мережі Facebook і відповідно їх реалізацію, оптимізовані алгоритмічні пакети взаємодії між клієнтами на певну кількість часу. В науковій публікації [17] проаналізовано особливості створення соціальної мережі для студентів, призначення якої в тому, щоб вони могли взаємодіяти між собою. Апробація системи демонструє, що можна зручно та легко реалізувати спілкування студентів та налагодити навчальні процеси, які цікаво і зручно відбуватимуться у соціальній мережі. Студенти можуть встановлювати час, створювати команди, ставити завдання та виконувати їх. Інтегровано алгоритми аналізу тексту, щоб аналізувати інформацію, яку студенти вводять у системі.

У статті [18] висвітлено можливості нейронної мережі виконувати аналіз сучасної моди на основі опрацювання фотографій у соціальній мережі Instagram. Завдяки заданим параметрам нейронна мережа аналізує одяг людей на вибраних фотографіях і, навчившись на певній вибірці даних,

прогнозує, який саме одяг буде модним через деякий час. Автори статті [19] дослідили алгоритми ефективної комунікації користувачів у соціальних мережах та месенджерах, тобто визначили саме системи чатів та внутрішнього спілкування, запропонували використання динамічних рішень, таких як фреймворк SignalR, який забезпечує динамічне спілкування користувачів за допомогою сесії, що модифікує використання вебсокетів. Це дає можливість використовувати чат так, щоб всі повідомлення зберігалися у сесії, уникаючи цим постійних запитів до сервера, та опрацьовувати повідомлення за допомогою концентраторів, спеціальних конвеєрів, за допомогою яких клієнт і сервер мають змогу звертатися один до одного.

Системи Tinder, Vadoo та Chatous відповідно до функціоналу, доступного для користувачів, можна вважати аналогом створюваної інтелектуальної системи, оскільки кожна з них багато в чому відтворює загальний алгоритм роботи та взаємодії користувачів, що уможливорює аналіз, дослідження та декомпозицію наявних частин та алгоритмів програми для розуміння їх роботи.

Розглядаючи програму Tinder (рис. 1), можна зробити висновок, що це одна з найпопулярніших програм для знайомства та соціалізації користувачів, а також найстаріша. Одна із головних переваг системи – її доступність на всіх платформах, наявність мобільного застосунку, вебверсії та версії для ПК, тобто цю програму зручно застосовувати практично на всіх платформах. Серед переваг можна також виділити використання нейронних мереж для ідентифікації обличчя користувача і можливість перевірки вже наявних користувачів системи, що надає впевненості щодо реальності користувачів і уникнення шахрайства. Відзначено зручний інтерфейс, який потребує мінімальної кількості дій для користування програмою.

Як недоліки варто виділити відсутність використання у програмі будь-яких алгоритмів соціалізації користувачів, тобто користувачі вводять лише базову інформацію про себе і програма використовує лише прості фільтри пошуку користувачів, такі як вік, стать та місцезнаходження. Тому важко шукати користувачів за інтересами. Програма дає змогу знаходити обмежену кількість користувачів за один сеанс, щоб мати змогу переглядати необмежену кількість користувачів, необхідно купити платну підписку.



Рис. 1. Виконання програми Tinder

Наступна програма, яку варто розглянути, – Vadoo (рис. 2). Одна з її основних переваг – програма доступна на всіх платформах, відповідно нею можна користуватися як у браузері ПК, так і на смартфоні, що дуже зручно. Також програма уможливорює пошук користувачів без зайвих обмежень відповідно до наявних фільтрів пошуку користувачів у системі. Також варто виділити те, що система використовує пошук обличчя користувачів, відповідно потрібно використовувати лише свою фотографію, щоб мати змогу користуватися нею, це стосується і додавання нових фотографій.



Рис. 2. Виконання програми Badoo

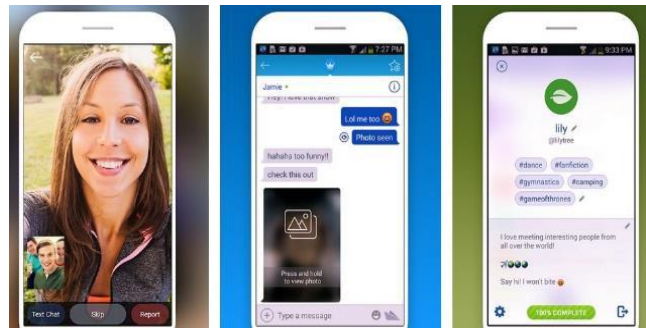


Рис. 3. Виконання програми Chatous

Серед недоліків варто виділити те, що програма хоч і безкоштовна, але містить чимало рекламних повідомлень, які дуже часто заважають користуватись системою, оскільки цю рекламу неможливо прибрати. Ще одним недоліком є відмова від використання алгоритмів соціалізації, тому доводиться витратити багато часу на перегляд дописів користувачів, з якими немає спільних інтересів.

Ще одним застосунком є Chatous (рис. 3), серед переваг якого варто виділити можливість безкоштовного її використання. Програма не пропонує платних підписок, також містить рекламні повідомлення, але їх достатньо мало, вона подається без зайвого нав'язування користувачу. Система має множину фільтрів для пошуку користувачів і цьому плані гнучкіша за аналоги, також використовує нейронні мережі для пошуку облич користувачів для реєстрації. Серед недоліків варто виділити її доступність лише на смартфонах, її не можна використовувати в браузері або в програмі для робочого стола, що зменшує кількість користувачів системи. Також зазначимо, що система є неоптимізованою і часто працює із затримками, що може бути незручно для багатьох користувачів. Основний її недолік той самий, що й інших програм, а саме відсутність алгоритмів соціалізації користувачів, наявність лише базових фільтрів, яких хоч і більше, ніж в інших програмах-аналогах, але все ж замало для зручної та швидкої соціалізації користувачів. Для того, щоб знайти користувача зі спільними інтересами, необхідно, окрім налаштування фільтрів вручну, гортати сторінки підібраних системою користувачів.

Для того, щоб детальніше виділити всі наявні переваги та недоліки програм-аналогів, створено порівняльну табл. 1.

Таблиця 1

Порівняння програм аналогів інтелектуальної системи

	Tinder	Badoo	Chatous
Необмежені множини вподобань	–	+	+
Відсутність реклами	+	–	–
Необмежена кількість користувачів	–	+	+
Кросплатформність	+	+	–
Необмежена кількість повідомлень	–	+	+
Платна версія	+	–	–
Базові фільтри пошуку	+	+	+
Аналіз фотографій	+	+	+
Алгоритми соціалізації	–	–	–

Проаналізувавши дані табл. 1, можна зробити висновок, що основним недоліком оглянутих програм є відсутність алгоритму соціалізації, наявні лише базові фільтри, із використанням яких можна оглядати список користувачів системи. Можливість створення списку користувачів системи є однозначною перевагою. Основними перевагами зазначених систем, які варто відтворити, є

можливість використання на різних платформах, відсутність або мінімальна кількість реклами та реалізація нейронних мереж, завдяки якій здійснюється розпізнавання фотографій користувачів і мінімізується кількість шахраїв.

Формулювання мети та постановка задачі

Інтелектуальна система, покликана сприяти соціалізації користувачів, повинна складатися з великої кількості послідовних елементів взаємодії з користувачем, починаючи від звичайної реєстрації та перевірки справжності користувача до створення груп користувачів відповідно до відсотка подібності, створення можливості додавання та обговорення публікацій, які стосуються певних тем відповідно до інтересів та зацікавлень окремих користувачів. Така інтелектуальна система повинна бути максимально простою у використанні, оскільки користувачі намагаються вибирати програму, найзручнішу і найпростішу у використанні. Складний інтерфейс і незрозуміла послідовність дій можуть відлякати потенційного користувача, відповідно програма має максимально спрощувати соціалізацію і взаємодію користувачів і водночас бути простою у використанні.

Мета полягає у розробленні методів та засобів соціалізації груп користувачів за подібними інтересами на основі нечіткого пошуку текстової інформації та машинного навчання.

Для реалізації поставленої мети необхідно розробити: загальні функціональні вимоги до системи соціалізації груп користувачів за подібними інтересами; загальну типову архітектуру системи соціалізації груп користувачів за подібними інтересами; метод підтвердження існування (справжності) користувача через аналіз фотографій з обличчям на основі реалізації згорткової нейронної мережі; метод аналізу інформації про кожного користувача на основі алгоритмів нечіткого пошуку для опрацювання текстових даних різних обсягів; метод формування рейтингу користувача на підставі зібраних даних; метод порівняння одного користувача з іншими для спрощення подальшої соціалізації користувачів, інтереси яких збігаються найбільше.

Поки що не існує аналогічних систем для соціалізації користувачів, які б виконували повноцінний аналіз усієї текстової інформації щодо користувача і формували кінцевий ідентифікатор подібності поточного користувача системи з уже наявними користувачами. Відповідно вперше буде створено інформаційну систему, яка практично повністю автоматизує соціалізацію користувачів.

Удосконалено алгоритм для нечіткого пошуку, який у цій системі використовується не в базовому вигляді, а оптимізований для паралельного опрацювання одночасно декількох наборів текстової інформації, що відповідно дає змогу набагато швидше і зручніше опрацьовувати всю необхідну інформацію і формувати результат. Також вперше застосовано не лише один певний алгоритм, використано різні алгоритми, а саме алгоритм N-грам, алгоритм розширення вибірки та реалізацію моделі Noisy Channel. Всі ці алгоритми схожі та ставлять одну мету – аналіз слів у тексті та формування оцінок, але кожен з них підходить для аналізу різних видів текстів та типів інформації, яка подана неструктуровано. Відповідно спочатку відбувається поділ тексту на чотири різні вибірки текстових даних, після чого кожен алгоритм здійснює опрацювання тих даних, для яких він найбільше підходить, тим самим оптимізуючи весь процес нечіткого пошуку. Надалі формується кінцевий ідентифікатор подібності на основі отриманих оцінок.

Основним призначенням інтелектуальної системи соціалізації груп користувачів за сукупними інтересами із використанням методів нечіткого пошуку текстової інформації та машинного навчання є розумна соціалізація її користувачів. Завдання полягає у тому, щоб створити таку інтелектуальну систему, яка максимально оптимізує та автоматизує дії користувача так, щоб він міг здійснювати соціалізацію настільки зручно, наскільки це можливо. Інтелектуальна система повинна бути захищеною, тобто користувач має бути впевнений у надійності збереження персональних даних та у тому, що фотографії користувачів інтелектуальної системи є справжніми і відповідно не належать шахраям, що можна реалізувати за допомогою згорткової нейронної мережі. Також за соціалізацію відповідають алгоритми нечіткого пошуку, злагоджена та ефективна робота яких забезпечує аналіз та опрацювання вхідних даних для того, хто користується інтелектуальною системою. Також варто згадати про необхідність реалізації чатів із користувачами в режимі реального часу та збереження повідомлень для оптимізації спілкування користувачів між собою із використанням сучасних інформаційних технологій.

Виклад основного матеріалу

У функціональності інтелектуальної системи, покликаної сприяти соціалізації користувачів, доцільно виділити чотири основні процедури “Передача даних”, “Аналіз даних”, “Збереження даних” та “Формування даних”. “Передача даних” полягає у підтримці постійного, швидкого та ефективного передавання різноформатних даних (текстових та медіаданих, листування користувачів між собою, повідомлень тощо) з підтримкою захисту, убезпеченні інтелектуальної системи від перевантаження за великої кількості одночасних запитів, та оптимальному доставлянні даних у місце запити.

“Аналіз даних” завдяки реалізації нейронної мережі та алгоритмів нечіткого пошуку дає змогу виконувати аналіз даних, тобто аналіз вхідної інформації з подальшим використанням результатів аналізу. Нейронна мережа забезпечує процедуру аналізу користувацьких фотографій для пошуку обличчя на фотографіях і порівняння завантажених фотографій з тими, що вже збережені в хмарному сховищі, щоб впевнитись, що аналогічну фотографію не завантажив раніше інший користувач. В алгоритмах нечіткого пошуку аналіз текстових даних є важливою частиною, оскільки для ефективної соціалізації необхідно сформувати список користувачів рейтингованого за відсотком подібності інтересів поточного користувача з іншими користувачами системи.

“Збереження даних” – це оптимальне використання бази даних та хмарного сховища.

“Формування даних” – подання результатів аналізу даних. Від результату роботи нейронної мережі залежить, чи знайдено на фотографії обличчя користувача і чи перевірено її наявність у базі даних, від правильної роботи алгоритмів нечіткого пошуку залежить поточна функціональність всієї системи, оскільки завдяки цим алгоритмам формується відсоткове значення подібності інтересів користувачів між собою. Важливе визначення тегів публікації, яку написав користувач, оскільки в результаті аналізу система отримує певні значення слів для порівняння зі словником, сформованим у системі. Саме правильне визначення текстових даних дає змогу підібрати правильні концепти, завдяки яким можна коректно визначити теги публікації, щоб зацікавити користувачів.

Система буде створюватися з використанням MVC алгоритму, тобто відбуватиметься чіткий поділ на модель даних, контролер, який займається опрацюванням даних і відображенням даних. Тобто інтелектуальна система буде доволі гнучкою і легкою в модифікації, наприклад, у вигляді відображення даних буде використовуватися окрема клієнтська програма, яка надсилатиме запити до серверної програми. Якщо говорити про модель даних, то можна виділити роботу з базою даних, міграцію відповідної бази даних, створення Data Transfer Object, тобто окремих об’єктів, що формуються на основі таблиць бази даних, але формуються окремо і ніяк не впливають на таблиці в базі даних. Тобто так зручно визначати, які саме дані потрібно передавати у відповідь на запит. Наприклад, на запит даних користувача можна не передавати певні дані про клієнта, тим самим уникнувши будь-якого шансу перехоплення цих даних. Розроблена інтелектуальна система реалізує два основні завдання: створення згорткової нейронної мережі, яка визначає, чи є обличчя користувача на фотографії, щоб контролювати процедуру додавання фотографій; реалізацію комбінації алгоритмів нечіткого пошуку, завдяки яким здійснюється ефективна соціалізація користувачів інтелектуальної системи.

У результаті системного аналізу можна зробити висновок, що інтелектуальна система соціалізації груп користувачів за сукупними інтересами із використанням методів нечіткого пошуку текстової інформації та машинного навчання має одну зовнішню сутність, а саме “Користувач” і чотири основні внутрішні сутності, до яких належать: “Браузер”, “Контролер системи”, “База даних”, “Сервіс системи”. В ході роботи інтелектуальної системи здійснюється постійна взаємодія між зовнішніми і внутрішніми сутностями: під час ініціалізації роботи системи зовнішня сутність “Користувач” входить у систему за допомогою внутрішньої сутності – “Браузер”. За допомогою сутності “Браузер” здійснюється авторизація користувача в системі, якщо він вже зареєстрований, або реєстрація в іншому випадку. І авторизація, і реєстрація відбуваються за допомогою внутрішньої сутності “Контролер системи”, після успішної реєстрації чи авторизації сутність “Контролер

системи” надсилає сесійний токен у сутність “Браузер” та зберігає поточний токен всередині браузера на час існування сесії. Після цього сутність “Користувач” здійснює завантаження власної фотографії для підтвердження своєї справжності за допомогою сутності “Браузер”, що додає фотографію за допомогою сутності “Контролер системи”. Сутність “Контролер системи” за допомогою нейронних мереж перевіряє наявність обличчя на фотографії сутності “Користувач”, а також наявність аналогічних фотографій в системі, після чого сутність “Контролер системи” зберігає всі отримані дані про користувача та надсилає сформовані дані в іншу внутрішню сутність системи, а саме “Базу даних”, для збереження всієї наданої інформації.

Після цього сутність “База даних” зберігає всю інформацію на допомогою внутрішньої сутності “Сервіс системи”. Далі “Сервіс системи” починає опрацьовувати цю інформацію та створює список унікальних користувачів системи. На наступному кроці сутність “Сервіс системи” за допомогою алгоритмів нечіткого пошуку здійснює аналіз даних користувача, який у цей момент працює у системі, та інших зареєстрованих користувачів, а у результаті роботи алгоритмів нечіткого пошуку сутність “Сервіс системи” створює список користувачів, сформований відповідно до даних поточного користувача і за зменшенням відсоткової подібності між користувачами. Далі сутність “Сервіс системи” здійснює кінцеву перевірку одержаної інформації за допомогою сутності “База даних”, виявляючи наявність пошкоджених та недостовірних даних, Після цього за допомогою користувацького запиту сформовані дані з сутності “База даних” надсилаються сутності “Користувач”. “Користувач” здійснює перегляд наданої інформації та вибирає профіль користувача, який йому найбільше сподобався, скориставшись сутністю “Сервіс системи”. Відтак сутність “Користувач” здійснює проставлення вподобання іншому користувачу, щоб інший користувач системи міг побачити, що його вибрав для спілкування поточний користувач. Здійснивши проставлення вподобання, сутність “Сервіс системи” за допомогою сутності “Контролер системи” розпочинає листування з іншим користувачем і тим самим формує діалог, який за допомогою сутностей “Контролер системи” та “База даних” зберігається як необхідна інформація у вигляді потоку повідомлень. Цей потік зберігається в сесії, що працює в системі, щоб динамічно відображати всі дані та повідомлення, якими обмінюються користувачі у цій системі та сесії, яка працює у конкретний момент часу для відображення даних.

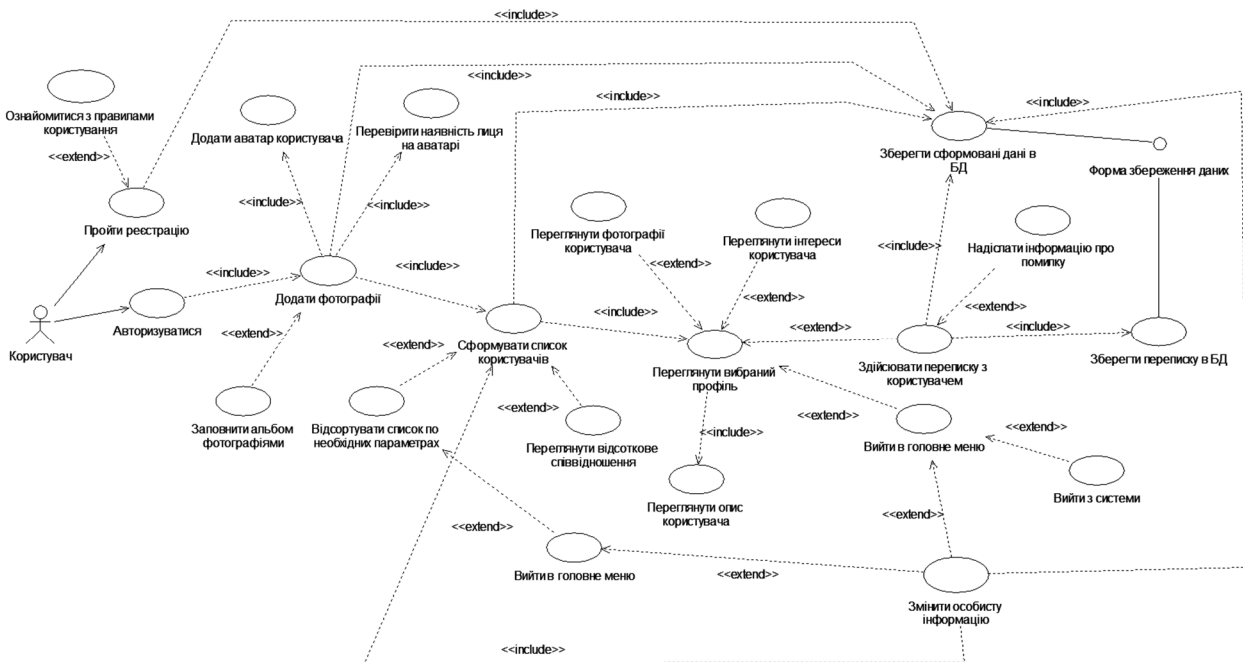


Рис. 4. Діаграма варіантів використання системи

Сутність “Сервіс системи” надсилає запит сутності “База даних” для перевірки наявності даних, оскільки одним з найважливіших аспектів системи є безпека користувачів та безпека їхньої інформації, відповідно час від часу сутність “Сервіс системи” надсилає сутності “База даних” запит, щоб перевірити цілісність даних та надіслати повідомлення про помилку у разі певних проблем з опрацюванням даних. Після того як сутність “Сервіс системи” сформувала запит потоку повідомлень, вона відправляє запит сутності “Контролер системи”, яка запитує токен діалогу в сутності “Сервіс системи”, відповідно сутність “Сервіс системи” відправляє поточний токен, в якому містяться дані про поточну сесію. Далі сутність “Контролер системи” через сутність “Браузер” відображає сформований діалог сутності “Користувач”, після чого “Користувач” або продовжує роботу, або виходить з системи. Система має одну дійову особу “Користувач”, яка поєднується із варіантами використання “Пройти реєстрацію” та “Авторизуватися” за допомогою відношення асоціації. Система складається з певних послідовних варіантів використання, які поєднані між собою відношенням включення, а саме “Додати фотографії”, “Сформувати список користувачів”, “Переглянути вибраний профіль”, “Здійснювати листування з користувачем”, “Вийти в головне меню” та “Вийти з системи”. Також варто звернути увагу на інтерфейс “Форма збереження даних” системи, який використовується з варіантами “Зберегти сформовані дані в БД” та “Зберегти листування в БД”. Варто виділити додаткові варіанти використання (рис. 4), які поєднані з основними за допомогою відношення асоціації, такі як “Додати аватар користувача”, “Перевірити наявність обличчя на аватарі”, “Переглянути опис користувача”, “Змінити особисту інформацію”. Також варто показати додаткові варіанти використання, поєднані з основними за допомогою відношення розширення, такими як “Ознайомитися з правилами користування”, “Заповнити альбом фотографіями”, “Відсортувати список за необхідними параметрами”, “Переглянути відсоткове співвідношення”, “Переглянути фотографії користувача”, “Переглянути інтереси користувача”, “Вийти в головне меню в процесі перегляду списку користувачів”, “Надіслати інформацію про помилку”.

На діаграмі класів подано основні класи інтелектуальної системи (рис. 5), а саме “Користувач”, “Збереження даних”, “Сервер системи”, “Взаємодія користувачів”, “Аналіз даних”, “Додавання фотографій”, “Підбір користувачів”. Всі класи поєднані між собою: клас “Користувач” поєднаний із класом відношення асоціації “Сервер системи”, а “Користувач” та “Сервер системи” у відношенні композиції з класом “Підбір користувачів”. Відповідно клас “Збереження даних” має відношення агрегації з класом “Підбір користувачів”, а клас “Взаємодія користувачів” – відношення агрегації з класом “Підбір користувачів” і також відношення асоціації з класом “Сервер системи”; клас “Додавання фотографій” – відношення асоціації з класом “Користувач”. Також класи “Взаємодія користувачів” та “Додавання фотографій” разом мають відношення композиції відносно класу “Аналіз даних”. Кожен клас має своє поле-ідентифікатор, яким він однозначно поданий у вигляді цілого числа і є приватним, оскільки не повинен бути доступним ззовні. Також приватними полями є всі поля, які потребують захисту і не мають бути доступними, такі як паролі, хеш-коди, значення параметрів, бали користувача тощо. Більшість з методів відповідно публічні, але є і приватні, які виконують лише певні специфічні внутрішні функції всередині класу.

Система має одну головну дійову особу, це “Користувач”. Також варто виділити основні об’єкти системи (рис. 6): це “Браузер”, “Контролер системи”, “База даних”, “Сервіс системи”, вони сполучені за допомогою зв’язків. Між всіма наявними об’єктами встановлено такі зв’язки, як “увійти в систему”, “авторизуватися”, “zareєструватися”, “зберегти токен”, “додати фотографію”, “перевірити наявність обличчя”, “надіслати сформовані дані”, “додати інформацію”, “створити список користувачів”, “перевірити відсоткове співвідношення”, “зберегти дані”, “показати профіль користувача”, “переглянути вибраний профіль”, “вподобати користувача”, “почати листування з

користувачем”, “надіслати запит перевірки даних”, “надіслати запит збереження даних”, “відповісти на запит”, “отримати проміжні дані”, “надіслати запит сервісу”, “перевірити наявність нових даних”, “зберегти дані”, “перевірити збережені дані”, “сформуванати запит повідомлень”, “запит токена діалогу”, “відповісти на запит токена”, “відобразити сформований діалог користувачу”, “надіслати дані та опрацювати дані”. Усі зв’язки між об’єктами передбачають передавання за допомогою них певних повідомлень, усі повідомлення за замовчуванням прості. Але для певних завдань використано асинхронні повідомлення для асинхронної роботи з даними, повідомлення з очікуванням для гарантованого повернення результату, якщо такий є. Функціонал інтелектуальної системи передбачає варіант повернення і опрацювання помилки, також використано виклик процедур та повернення із процедур, тобто щоб було явно видно, які саме дані надходять у певну процедуру і які вже опрацьовані, або повністю нові сформовані дані виходять з вибраної процедури. На рис. 6 зображено діаграму кооперацій.



Рис. 5. Діаграма класів

Інтелектуальна система має одну головну дійову особу, це “Користувач”, також варто виділити основні об’єкти системи: “Браузер”, “Контролер системи”, “База даних”, “Сервіс системи”. Вони всі взаємодіють за допомогою повідомлень (рис. 7). Більшість повідомлень відправляються асинхронно, завдяки чому система працює набагато швидше, їй не потрібно очікувати повернення даних, але є і синхронні повідомлення, в яких важливо дочекатися повернення результату. Між всіма об’єктами встановлено такі зв’язки, як “увійти в систему”, “авторизуватися”, “зареєструватися”, “зберегти токен”, “додати фотографію”, “перевірити наявність обличчя”, “надіслати сформовані дані”, “додати інформацію”, “створити список користувачів”, “перевірити відсоткове співвідношення”, “зберегти дані”, “показати профіль користувача”, “переглянути вибраний профіль”, “вподобати користувача”, “почати листування з користувачем”, “надіслати запит перевірки даних”, “надіслати запит збереження даних”, “відповісти на запит”, “отримати проміжні дані”, “надіслати запит сервісу”, “перевірити наявність нових даних”, “зберегти дані”, “перевірити збережені дані”, “сформуванати запит повідомлень”, “запит токена діалогу”, “відповісти на запит токена”, “відобразити сформований діалог користувачу”, “надіслати дані та опрацювати дані”.

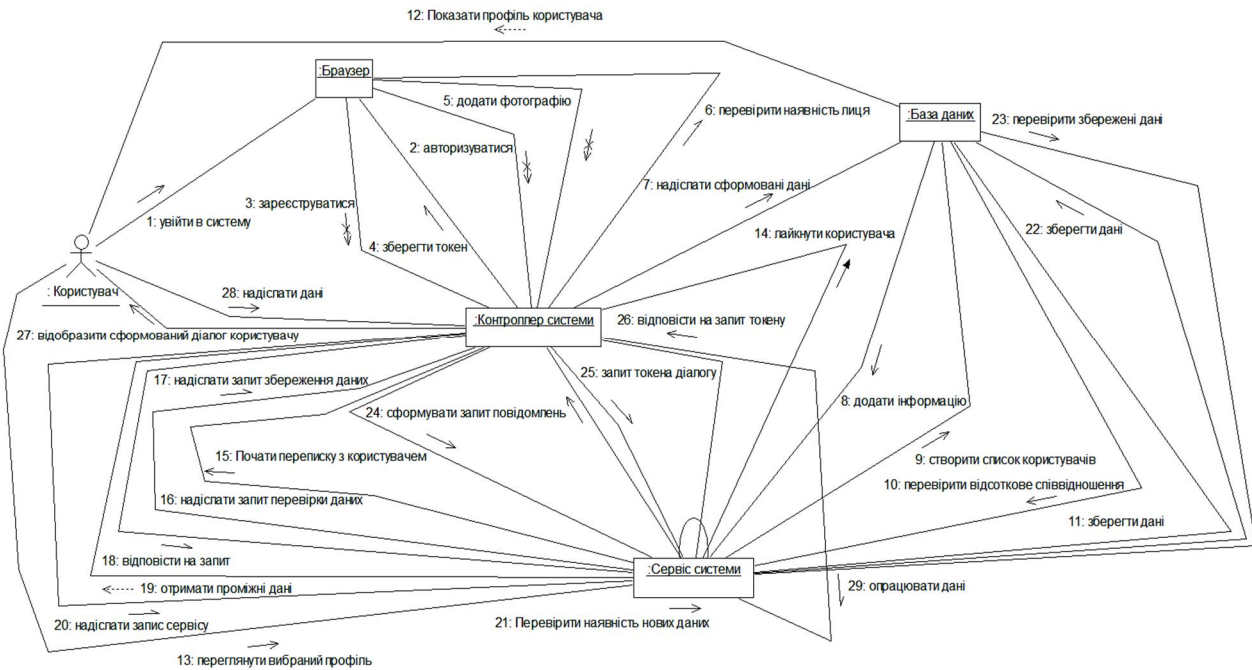


Рис. 6. Діаграма кооперації

На діаграмі є початковий та кінцевий стани (рис. 7). Виділено такі стани “Залишатися в системі”, “Додати фотографію”, “Перевірити наявність обличчя”, “Внести інформацію в БД”, “Прийняти запит”, “Створити список користувачів”, “Перевірити користувача”, “Зареєструватися, авторизуватися”, “Зберегти дані в БД”, “Вивести кінцеву інформацію користувачу”, “Перевірити наявність вже існуючої фотографії користувача в БД”, “Перевірити наявність нових даних”, “Надіслати сформовані дані”, “Внести змінені дані в БД”, “Вибрати користувача”, “Вподобати сторінку користувача”, “Розпочати діалог з користувачем”, “Додати потік повідомлень у сесію”, “Надіслати повідомлення”, “Відобразити повідомлення користувачу”. Також було створено два складені стани. Перший стан – “Зберегти інформацію на сервері” на вхід отримує інформацію, здійснює опрацювання даних на сервері, формує відповідь та на вихід посилає сформовану відповідь на запит клієнта. Другий стан – “Відсортувати список за вказаними параметрами”: на вхід завантажуються дані з сервера, всередині виконується основна робота, тобто перевірка даних на відповідність, із урахуванням запиту та сортування списку за наданими параметрами, після чого на вихід подається готовий відсортований список, який надалі буде збережений у БД.

На діаграмі подано такі стани дії (рис. 7), як “Увійти в систему”. Після виконання умови вибирається “Авторизуватися” чи “Зареєструватися”, “Надіслати сформовані дані”, циклічне “Зберегти сформовані дані в БД”, “Додати фотографію”, “Перевірити наявність обличчя на аватарі”, “Додати інформацію”, “Створити список користувачів”, “Перевірити відсоткове співвідношення”, “Оглянути вибраний профіль”. Процес “Оглянути вибраний профіль” розпаралелюється на “Показати профіль користувача” та “Змінити персональну інформацію”, що за певної умови переходить у “Зберегти сформовані дані в БД”, “Перевірити збережені дані”, “Оглянути новий список”, “Переглянути профіль користувача”, “Вподобати користувача”, “Почати листування з користувачем”, “Додати новий потік повідомлень”, який розпаралелюється на “Зберегти дані” та “Перевірити наявність нових даних”.

Потім йдуть стани дії “Сформувати повідомлення користувачу”, “Надіслати повідомлення користувачу”, “Показати повідомлення в застосунку”. Завершується робота кінцевим станом системи.

Інтелектуальна система має основні зовнішні об'єкти системи (рис. 9). Це “Користувач”, “Системний контролер”, “База даних”, “Система”. Вони всі взаємодіють за допомогою станів дій. Також варто виділити особливі для цієї діаграми внутрішні об'єкти, такі як “Запит” зі своїм внутрішнім станом “Опрацьовано”, “Дані” зі своїм внутрішнім станом “Збережено”, “Сформовані дані” зі своїм внутрішнім станом “Отримано”, “Повідомлення” зі своїм внутрішнім станом “Сформоване”, “Повідомлення користувачу” зі своїм внутрішнім станом всередині цього об'єкта “Коректно надіслане”.

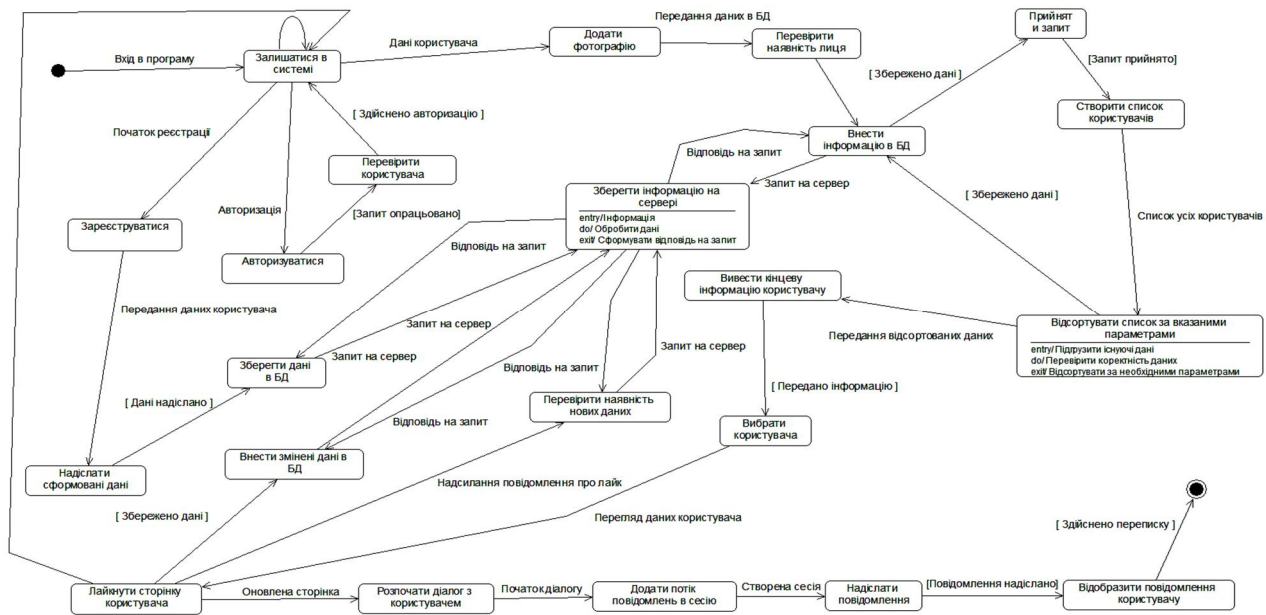


Рис. 7. Діаграма станів

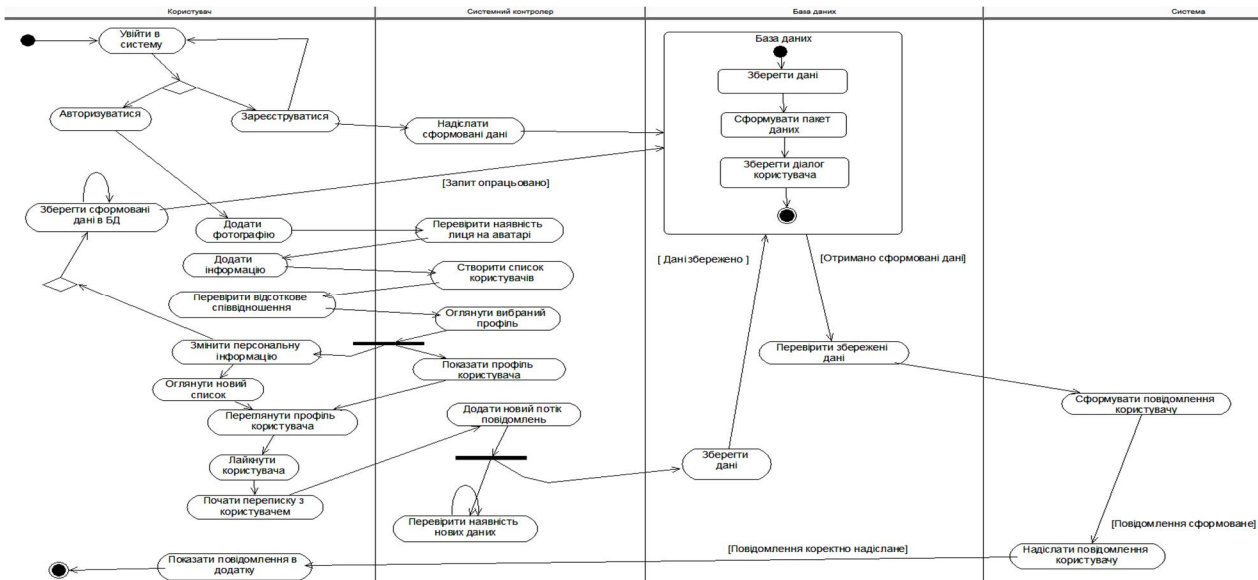


Рис. 8. Діаграма діяльності

Також на створеній діаграмі чітко видно складений стан дій “База даних”, який має окрему доріжку, за допомогою якої взаємодіє з іншими станами дій. Насамперед варто звернути увагу на початковий та кінцевий стани всередині складеного стану, тобто можна зробити висновок, що кожен

запит до бази даних має чіткий початок та кінець і відповідно база даних починає працювати лише тоді, коли до неї здійснено необхідний запит. Всередині було створено кінцевий автомат, який складається з конкретних станів, а саме “Зберегти дані”, “Сформувати пакет даних”, “Зберегти діалог користувача”. Використано стани, а не стани дій, тому що в цій конкретній ситуації ми працюємо із базою даних. Важливим елементом роботи з даними є саме їхня цілісність та незмінність, тому потрібно бути впевненим у тому, що дані під час роботи не будуть ні в якому разі знищені чи пошкоджені. Тому всі дані додатково зберігаються і перевіряються під час кожного стану всередині описуваного у цей момент стану дії.

Головною функцією є функція входу в систему “Main.exe” (рис. 10). Надалі варто виділити специфікації підпрограм, до яких належать “Надіслати повідомлення користувачу”, “Авторизуватися”, “Опрацювання даних”, “Форма повідомлення”. На діаграмі є також тіла підпрограм: “Оновити систему”, “Сформувати список користувачів”, “Перевірка наявності даних”. На діаграмі виділено базу даних, яка відповідає за зберігання та опрацювання даних. Також на діаграмі чітко видно частини основної програми, а саме “Оновлення системи”, “Зареєструватися”, “Вибрати користувача”, “Створити сесію”.

Також на діаграмі визначено основні компоненти інтелектуальної системи, до яких належать “Користувач” та “Система”. Варто показати наявні в системі тіла пакетів, до яких належать “Надіслати запит на зберігання токена”, “Переглянути інформацію про користувача”, “Створити потік повідомлень”, “Оновити дані”, “Додати фотографії”, “Надіслати повідомлення”, “Надіслати запит на створення сесії”, “Заповнити форму”. Відповідно частиною тіла пакета є його специфікація, така як “Переглянути вхідні дані”, “Вибрати параметри діалогу”, “Зберегти потік повідомлень у БД”, “Прочитати опис користувача”, “Переглянути інтереси користувача”, “Зберегти дані користувача в БД”, “Внести дані”, “Ознайомитися з правилами користування”, “Додати аватар користувача”, “Перевірити наявність обличчя”, “Заповнити альбом фотографіями”, “Надіслати інформацію про помилку”, “Оновити існуючу сесію”, “Зберегти токен”, “Перевірити параметри з’єднання”, “Оновити дані” та “Перевірити дані”.

На діаграмі розгортання (рис. 11) можна виділити два основні пристрої. Це “Сервер” та “Хмарне сховище”, а також три процесори, це “Користувач”, “Сервіс системи” та “Контролер системи”. Також вони поєднані між собою за допомогою відношень зв’язків: процесор “Користувач” поєднаний з процесором “Сервіс системи”, відповідно процесор “Сервіс системи” із процесором “Контролер системи” та з пристроєм “Сервер”, а пристрій “Сервер” із пристроєм “Хмарне сховище”.

За допомогою пунктирних ліній показано зв’язок процесорів та пристроїв із примітками. Кожен процесор або пристрій на діаграмі виконують певну унікальну функцію, наприклад, процесор “Користувач” – користувач системи, який здійснює функції пошуку інших користувачів за спільними інтересами, він виконує основну роботу із інтелектуальною системою, тобто реєструється, завантажує фотографію, шукає користувачів за спільними інтересами з використанням усіх необхідних фільтрів та здійснює листування із користувачем.

Процесор “Сервіс системи” – основний сервіс інтелектуальної системи, що опрацьовує запити користувача, відповідь сервера та отриману інформацію. Процесор “Контролер системи” здійснює перевірку даних, форматує дані для подальшого відображення користувачу або надсилання на сервер. Пристрій “Сервер” зберігає усю інформацію, яка міститься у базах даних інтелектуальної системи. Пристрій “Хмарне сховище” – це хмарний онлайн-сервіс, пов’язаний із пристроєм “Сервер”, а саме із базою даних, у якій містяться ідентифікатор і посилання на фотографії користувачів, які зберігаються у хмарному сховищі.

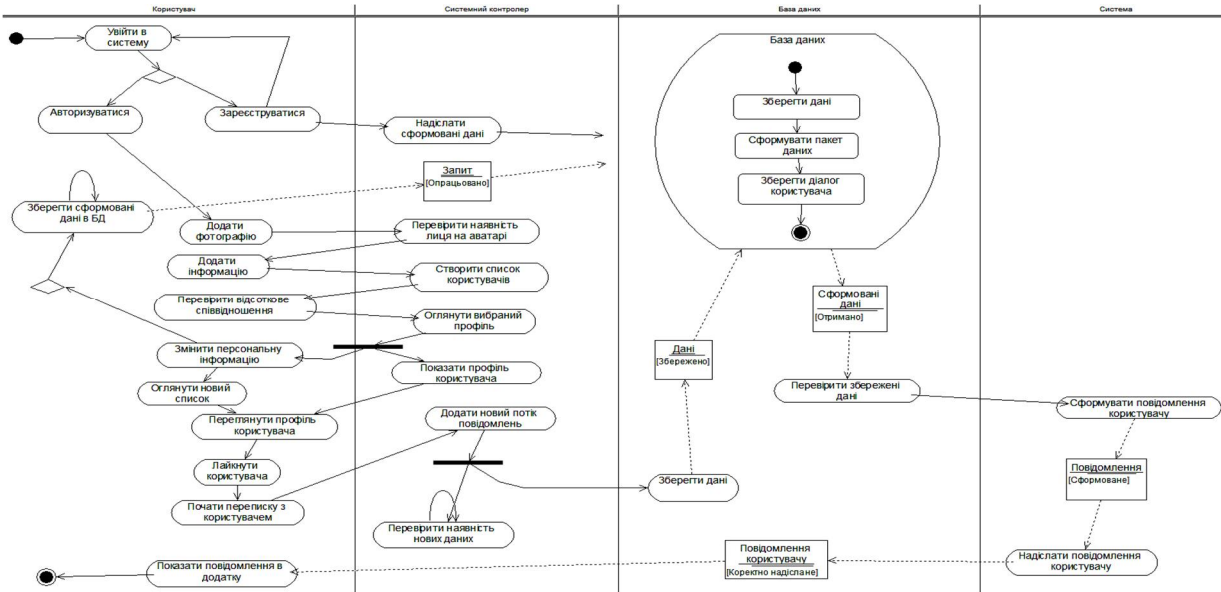


Рис. 9. Діаграма діяльності системи для моделі бізнес-процесу фотографії

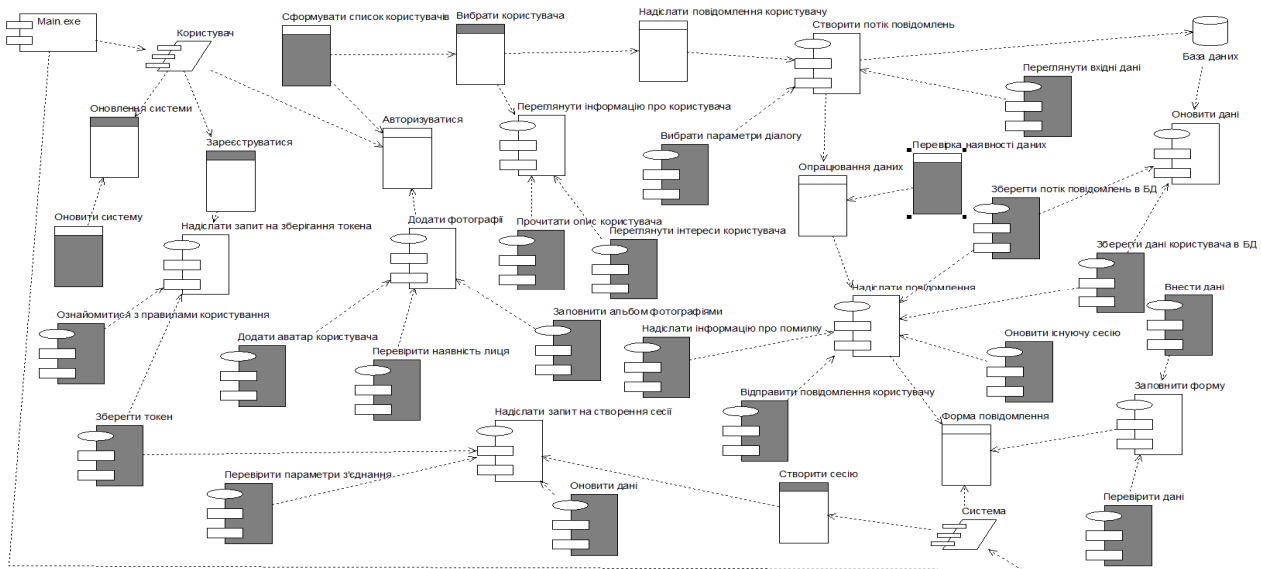


Рис. 10. Діаграма компонентів

Основним методом створення інтелектуальної системи є шаблон MVC, тобто Model-View-Controller. Завдяки використанню цього шаблону інтелектуальна система буде масштабованою та правильно структурованою, оскільки умовно поділена на три різні частини, де кожна частина безпосередньо взаємодіє одна з одною і виконує визначені функції. Так, під Model розуміють модель даних, тобто це база даних, яка містить всю інформацію про користувачів та іншу системну інформацію, хмарний сервіс, в якому міститься вся медіаінформація користувачів, і, звичайно, вибрана ORM система. Користуючись нею, не потрібно працювати напряму з таблицями і SQL запитами до бази даних, а з відповідними класами та методами всередині програми, оскільки база даних створюватиметься з використанням підходу Code First. Спочатку будуть створені всі необхідні класи для представлення інтелектуальної системи і всі зв'язки між класами, після чого відбувається мапінг, коли на основі вже наявних класів створюють базу даних, таблиці та зв'язки між ними.

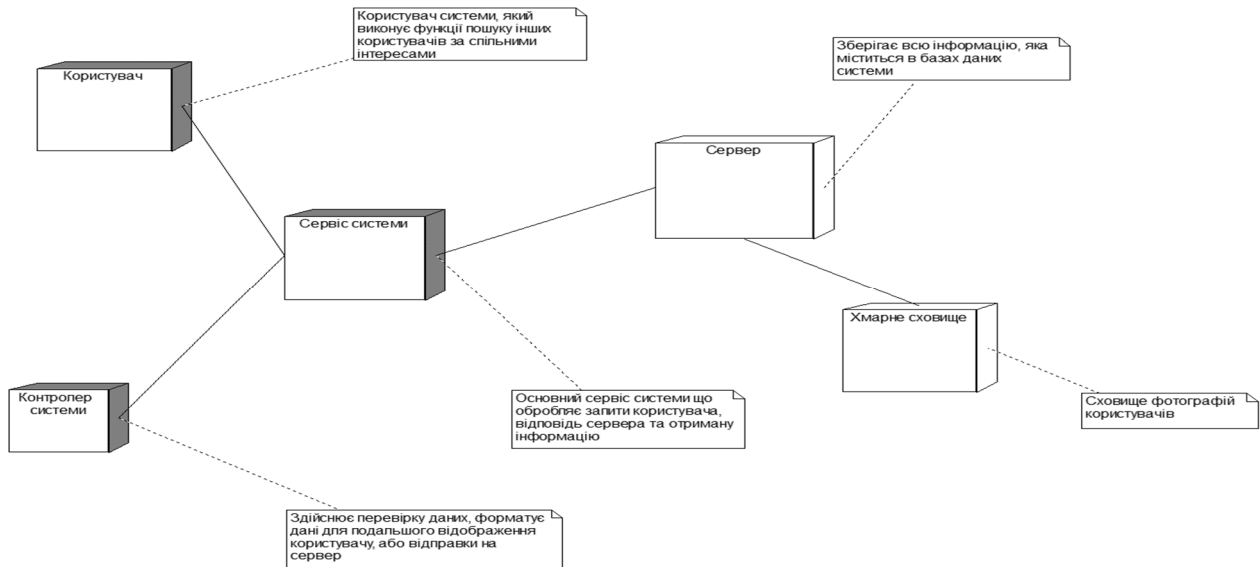


Рис. 11. Діаграма розгортання

Ще однією важливою частиною є Controller, тобто підсистема, яка керує основними компонентами інтелектуальної системи. До елементів контролера можна зарахувати репозиторії – спеціальні класи, призначені для роботи з БД, які реалізують основні базові методи роботи з даними – відображення, створення, оновлення та зміну даних. Завдяки їм також можна створювати інші методи для роботи з наявними в БД частинами даних, для вибирання певної особливої частини даних та їх подальшого опрацювання. Уся робота з даними безпечна і виконується за допомогою вибраної ORM на фоні. Не потрібно працювати безпосередньо з даними, достатньо уявити, як виглядатимуть сама база й основні запити до неї, і реалізувати все це за допомогою класів та методів усередині репозиторіїв. Іншими важливими елементами контролера є сервіси системи, в системі існує безліч різних класів, що взаємодіють між собою та виконують певні бізнес-функції, тобто це різноманітні шаблони та алгоритми, які здійснюють підрахунки, опрацьовують дані, формують нові дані, міняють дані у певний спосіб відповідно до завдання системи. Кожен із таких сервісів має свій інтерфейс, який описує основні властивості та методи класу, який надалі реалізуватиме цей інтерфейс. Відповідно кожен сервіс унікальний і працює лише в межах певних завдань, завдяки реалізації механізму ін'єкції залежностей. Клас використовує лише методи, призначені для нього.

Останньою частиною є View, тобто відображення сформованих даних користувачу, який відповідно з цими даними може працювати, оскільки ця інтелектуальна система реалізовується у вигляді клієнт-сервера. Увесь клієнт буде реалізований у вигляді окремої програми і являтиме собою View, для більшої зручності View буде реалізовано з використанням шаблону SPA (тобто Single Page Application), отже, застосунок працює на одній сторінці. Під цим розуміють, що клієнтська частина інтелектуальної системи завантажуватиметься лише один раз на початку роботи з нею у вигляді окремої сторінки, а всі елементи системи – це окремі компоненти, які завантажуються всередину цієї сторінки як її частина, а не як повноцінна вебсторінка. Тому можна сказати, що клієнтська частина програми буде реалізована практично повністю асинхронною і не вимагатиме перезавантажень сторінки там, де у цьому немає потреби, тобто не будуть витрачатися зайві ресурси інтелектуальної системи на роботу, яка має виконатися лише один раз. Отже, у разі реалізації асинхронної роботи класів та модулів система працюватиме швидше. Оскільки браузеру доведеться опрацьовувати значно менше синхронних подій, через що всі запити між клієнтською та серверною частиною програми працюватимуть швидше та оптимальніше, дані в межах компоненти сторінки можна завантажувати не всі одразу, а з використанням методів ініціалізації вебкомпонентів, завдяки

чому можна завантажувати лише частину даних для відображення сторінки, а решту вже під час взаємодії користувача з інтелектуальною системою, що мінімізує шанс уповільнення її роботи через важкі запити.

Перед тим, як розпочати роботу над створенням інтелектуальної системи, необхідно чітко визначити усі засоби для вирішення поточної проблеми. Спочатку варто визначити, яку саме мову програмування використовувати, вибираючи переважно між C#, node.js, Java та Python. У результаті вибрали мову програмування C#. Основним чинником вибору стало те, що вже є певний досвід написання програм мовою C# і вона дуже зручна для створення інтелектуальної системи. Основною перевагою є те, що мова C# повністю ООП, відповідно зручно використовувати класи, інтерфейси та структури мови. Крім того, C# має велику бібліотеку вже готових для використання класів і зазвичай внутрішній збирач сміття, який спрощує роботу з пам'яттю і водночас дає змогу здійснювати доступ до певних ділянок пам'яті в певний момент часу. Також мова C# ідеально підходить для створення великих масштабованих програм за допомогою механізму абстракцій та ін'єкції залежностей, завдяки чому можна так розподіляти компоненти системи, щоб кожен виконував свою визначену функцію і не залежав від інших компонентів системи. До недоліків C# можна зарахувати те, що вона повільніша, ніж низькорівневіші аналоги і складніша у використанні, ніж слабкотипізовані аналоги, але всі ці недоліки незначні.

Звичайно мова програмування не має ніякого значення. Головне – вибрати backend-фреймворк, у цьому випадку буде використовуватися ASP .NET Core Web API, який ідеально підходить для створення серверної частини інтелектуальної системи, оскільки передбачає зручну процедуру роботи контролерів та запитів, можливість створювати проміжний код із використанням мови програмування C# та розширювати вже наявні класи необхідними для конкретної задачі методами. Також варто відзначити механізм інтеграції інтерфейсів та абстрактних класів на рівні програмної рефлексії. Важливо сказати, що ASP .NET Core Web API має інтегровану систему авторизації та реєстрації користувачів – Identity, яка надає базові методи для реалізації основних компонентів зручної та безпечної реєстрації, такі як набір класів для роботи з JWT access та refresh токенами, які є найсучаснішими.

Для створення клієнтського застосунку вибрано мову програмування TypeScript, яка працює разом із мовою JavaScript, відповідно можна створювати додаток з використанням TypeScript, а в необхідних компонентах застосовувати JavaScript і комбінувати різні шаблони програмування. Основні переваги мови TypeScript в тому, що вона, як і C#, строго типізована і формує всі типи системи на етапі компіляції, а також містить внутрішній збирач сміття, який виконує велику частину роботи й очищає використані посилання. Окрім мови програмування TypeScript, необхідний зручний фреймворк, такий як Angular. Основні його переваги у тому, що він має компонентну структуру і чіткий розподіл обов'язків між компонентами системи, тобто кожен компонент має свій HTML та CSS клас, який є доповненням до зовнішніх компонентів, всередині яких містяться внутрішні компоненти, та додає стилі лише там, де потрібно. Відповідно можна виконувати лише певну частину завдання і бути впевненим, що ніде більше в програмі не буде змін, оскільки використовується шаблон SPA – Single Page Application, кожен компонент якого продекларований окремими селекторами, що визначають межі дії компонента та його стилів.

В наш час неможливо писати код без використання спеціалізованих програмних засобів. Найзручніші спеціальні програми, які дають змогу писати код, додавати бібліотеки, шукати помилки в коді та відстежувати використання системних ресурсів. Такі програми називаються IDE і для написання серверного ASP .NET Core Web API застосунку найкраще підходять три різні IDE – Visual Studio 2022, Rider, MonoDevelop. Кожна з наведених програм має певні переваги та недоліки у використанні (табл. 2). У результаті аналізу всіх трьох серверних IDE вибрано найзручнішу та функціональну для створення цієї інтелектуальної системи – Visual Studio 2022.

Таблиця 2

Порівняння наявних серверних IDE та наявних клієнтських IDE

Параметр	серверних			клієнтських		
	Visual Studio	Rider	MonoDevelop	Visual Studio Code	Sublime Text	Atom
Засоби аналізу системи	–	+	–	+	–	+
Потреба потужного ПК	–	+	–	–	–	+
Важкість програми	+	–	–	+	–	–
Безкоштовне розповсюдження	+	–	+	+	–	+
Кросплатформність	–	+	+	+	+	+
Наявність документації	+	+	+	+	+	+
Наявність Debugger	+	+	+	+	+	+
Наявність розширень	+	+	–	+	+	+
Користувацькі бібліотеки	+	–	–	+	–	–
Open-source				+	–	–

Для тестування створеної серверної частини інтелектуальної системи та запитів до всіх необхідних методів застосовують програму Postman, яка є легкою у використанні, безкоштовною і надає безліч можливостей функціоналу, необхідного для тестування всіх API-запитів та пошуку різноманітних серверних помилок. Для створення клієнтської частини інтелектуальної системи з використанням фронтенд фреймворку Angular необхідно підібрати IDE, яка буде найзручнішою для написання програмного продукту. Здійснюється вибір серед трьох основних програм: це Visual Studio Code, Sublime Text та Atom. Кожна із цих програм має певні переваги та недоліки (табл. 2). В ході аналізу всіх трьох клієнтських IDE ми вибрали найзручнішу та функціональну для створення цієї інтелектуальної системи – Visual Studio Code. Окрім програмних засобів для написання коду необхідно вибрати правильний програмний засіб для створення і роботи із базою даних, серед основних варіантів варто виділити такі СУБД, як Microsoft SQL Server, MySql, PostgreSQL (табл. 3).

Таблиця 3

Порівняння описаних СУБД

	Microsoft SQL Server	PostgeSql	MySql
Швидке виконання	+	–	+
Проста у використанні	–	–	–
Робота з іншими БД	+	–	+
Базова наявність	+	–	+
Користувацькі інтерфейси	+	+	+
Підтримка JSON	+	+	+
Підтримка OLAP	–	–	–
Підтримка XML	–	+	–
Базові функції	+	+	+
Наявність документації	+	–	+
Безкоштовне розповсюдження	+	+	+
Кросплатформність	–	+	+

У ході аналізу всіх трьох СУБД вибрано найзручнішу та найфункціональнішу для створення цієї інтелектуальної системи – Microsoft SQL Server. Вибрано той хмарний сервіс, який найбільше підходить для створення інтелектуальної системи соціалізації груп користувачів за сукупними інтересами з використанням методів нечіткого пошуку текстової інформації та машинного навчання,

це хмарний сервіс Cloudinary. Серед переваг варто відзначити, що він має безкоштовну підписку, що ідеально підходить для тестування самого сервісу і для створення мінімального набору системних даних для збереження, а також зручне API для мови програмування С#. Достатньо встановити бібліотеку для роботи з сервісом і можна отримати доступ до основних методів додавання, видалення, оновлення та зберігання медіафайлів користувачів системи. Серед недоліків можна виділити відсутність чіткої документації та простоту самого хмарного сервісу, в якому відсутні деякі функції. Механізми авторизації та реєстрації реалізовано за допомогою механізму Identity, який надає основні класи та їх методи для надійної реалізації всіх потрібних механізмів. Для цього завдання використано класи `UserManager<AppUser>` та `SignInManager<AppUser>`, кожен з них ініціалізується з використанням конструктора та механізмів вставляння залежностей. Для реалізації реєстрації користувача використано механізм JWT. Це JSON Web Token, спеціальний захешований ідентифікатор користувача, що зберігається у Local Storage браузера. Використання refresh та access tokenів зображено на рис. 12. Сам token створюють із визначенням механізмів Claims, тобто вимог для користувача системи, що авторизується, вказують допустимі ролі користувача та додають у список доступу дозволів токена.

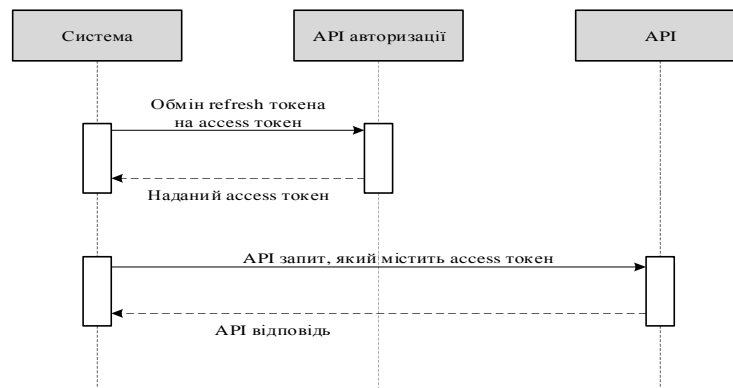


Рис. 12. Робота refresh та access tokenів

Наступний крок – створення `SigningCredentials`, тобто хеш-значення, яке отримане з використанням `HMACSHA512`. Це надійний криптографічний механізм, що перевіряє цілісність інформації, який хешує ідентифікатори даних і надає гарантію цілісності інформації. Далі для токена створюють дескриптор, у який додають всі вимоги для користувача, час дії токена і отримані `SigningCredentials`, після чого вже створюють сам об'єкт токена з переданим в конструктор дескриптором. Після цього token зберігається і повертається у метод реєстрації. В методі реєстрації асинхронно за допомогою методу `CreateAsync` створюється новий об'єкт користувача, перевіряється цілісність цього об'єкта і також перевірка на `NULL`, після чого повертається готовий об'єкт.

Механізм авторизації користувача починається із пошуку даних користувача в базі даних, коли користувача було знайдено, за допомогою Identity методу `CheckPasswordSignInAsync` перевіряється правильність пароля. Якщо пароль неправильний, виводиться відповідна інформація, якщо правильний, то створюється новий `Data Transfer Object` – об'єкт передавання даних, який містить нікнейм користувача, асинхронно створений JWT token, знайдений аватар користувача разом із посиланням на фотографію в хмарному сервісі та інші інформаційні поля системи.

Під час авторизації в самому клієнті подається запит до сервера системи, за допомогою механізму мапінгу послідовних запитів перевіряють, чи пройшов запит та чи існує користувач системи; якщо поточний користувач існує, то потрібно зберегти token. За допомогою спеціального класу `AuthGuard` та визначеного методу `canActivate` перевіряють валідність отриманого токена, якщо token правильний, то він зберігається в Local Storage браузера. Після цього готовий об'єкт користувача системи передається із сервісу в компоненти системи. Всі дані зберігаються в базі даних

(рис. 13). AspNetUsers зберігає дані про користувачів системи, вона містить ідентифікатор користувача, нікнейм, логін, згенерований хеш користувацького пароля, місто та країну користувача, дату народження, стать, дату останнього входження користувача в систему. Також там міститься інформація про користувача, його інтереси та зацікавлення, відомості про користувачів, яких він шукає. Є і поле ідентифікатора подібності користувача і багато додаткових полів, таких як email та значення його підтвердження, номер телефону користувача та його підтвердження. Оскільки база даних реляційна, то всі таблиці поєднані між собою зв'язками: так, таблиця AspNetUsers має відповідно зв'язок “один до багатьох” з Likes, Photos, Messages та Posts. Відповідно таблиця Posts має зв'язок “один до багатьох” з такими таблицями, як PhotoOfPosts та Comments. Таблиця Likes складається із двох полів, це унікальні ідентифікатори користувачів. Одне є ідентифікатором користувача, який самостійно поставив вподобайку, а інше є ідентифікатором користувача, якому поставив вподобайку інший користувач. Таблиця Photos містить користувацькі фотографії, їхній ідентифікатор, значення чи фотографія є аватаром користувача, а також ідентифікатор та посилання на фотографію у хмарному сервісі, щоб мати постійний доступ. Таблиця Messages містить повідомлення користувачів, ідентифікатори відправників та отримувачів повідомлення, сам вміст повідомлення, дату відправлення та дату прочитання повідомлення, інформацію про того, хто видалив повідомлення, якщо його видалили.

Posts містить інформацію про публікації користувача інтелектуальної системи, а саме вміст публікації, ідентифікатор користувача, ідентифікатор публікації та дату, коли публікація була створена. PhotoOfPosts містить зображення, завантажені в публікації, а саме посилання на зображення у хмарному сервісі та створений унікальний ідентифікатор. Comments містить коментарі користувачів, а саме ідентифікатор користувача і коментаря, вміст і дату публікації цього коментаря.

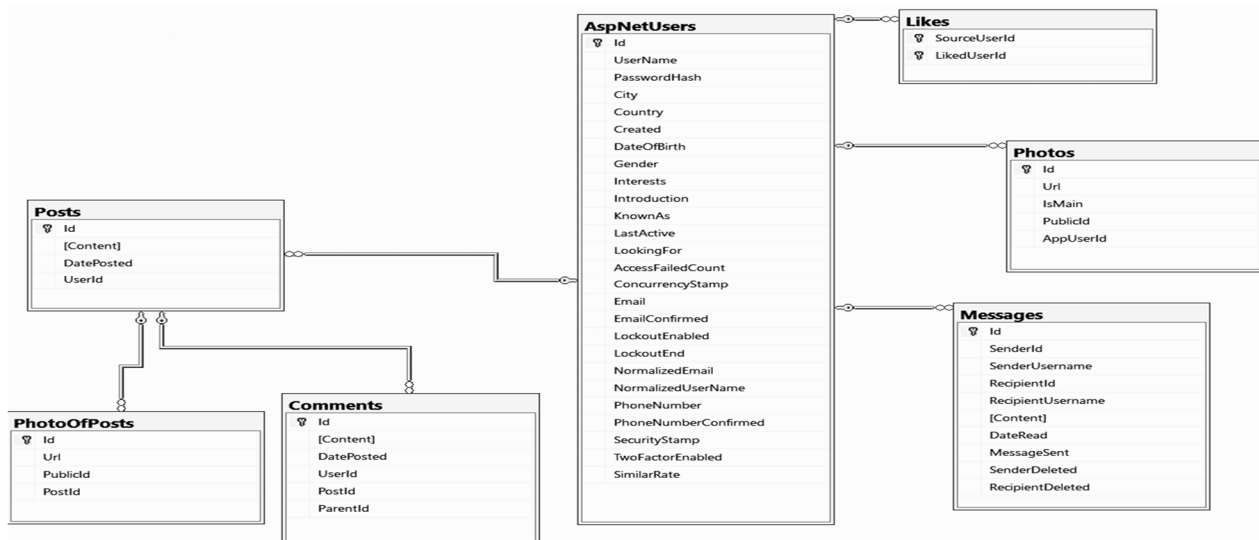


Рис. 13. Схема бази даних інтелектуальної системи

Одним із основних завдань є створення згорткової нейронної мережі, яка в середині інтелектуальної системи може визначати, чи є в користувача на кольоровій фотографії обличчя і чи є у базі даних системи такий користувач і його фотографія. Для цього найкраще підходить згорткова нейронна мережа, яка містить декілька прихованих шарів, а також шар згорток. Головне завдання ЗНМ – класифікація зображень, тобто отримання зображення на вхід і вибір одного з наявних класів на виході. В нашому випадку це наявність чи відсутність обличчя.

На вхід подається зображення, яке ЗНМ бачить у вигляді пікселів. Кожен піксель набуває значення від 0 до 255, яке описує його інтенсивність у певній точці. Відповідно на виході повинно бути єдине число, яке відображає шанс того, до якого класу належить подане зображення, відповідно

сам аналіз зображення відбувається за допомогою покрокового пошуку даних в отриманій матриці чисел характеристик базового рівня, таких як викривлення та нелінійності поданого зображення, а також за допомогою використання абстрактніших концепцій, таких як групи згорткових шарів. Можна виділити три групи шарів згорткової нейронної мережі: P-Net, або згорткові шари, або Convolution Layers, відтак R-Net, або об'єднані шари, або Pooling Layers, останніми йдуть O-net, або ущільнені шари, або Dense Layers (рис. 14, 15), відповідно кожен із шарів виконує свою певну функцію.

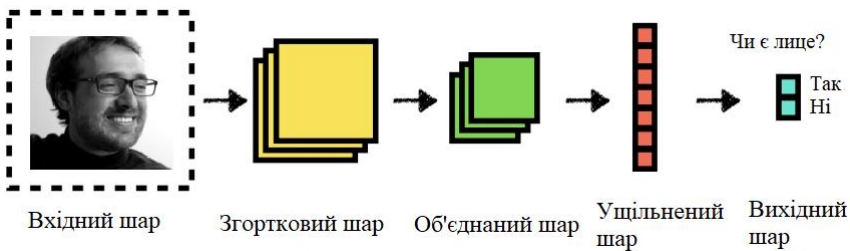


Рис. 14. Покрокова робота згорткової нейронної мережі

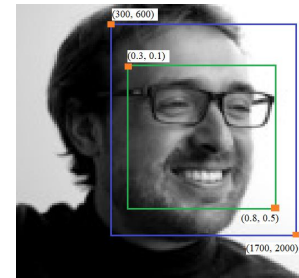


Рис. 15. Робота згорткових шарів

Шар згортки відповідає за згортання фільтрів поданого зображення з урахуванням результату попереднього шару. Фільтр можна використати як менше зображення, тобто меншу матрицю, ніж подану на вхід. Якщо частина зображення відповідає тому, що очікує фільтр, вихідне отримане значення буде великим. Тобто в результаті роботи застосовуємо два фільтри, один для горизонтальних і один для вертикальних країв. Потім об'єднуємо обидва результати, щоб отримати зображення, на якому вказані як горизонтальні, так і вертикальні краї. В цьому прикладі синє поле відображає ядро 32×32 , яке зменшилося за розміром до поданого зображення. Обчислено висоту та ширину ядра: $1700 - 300 = 1400$, $2000 - 600 = 1400$. Висота та ширина, вказані тут, – це висота та ширина ядра, зведені до наданого розміру. Потім перемножили координатні рамки на 1400 : $0,3 \times 1400 = 420$, $0,1 \times 1400 = 140$, $0,8 \times 1400 = 1120$, $0,5 \times 1400 = 720$. Отримані значення додають до верхньої лівої координати цього ядра для того, щоб зафіксувати розраховані координати зображеного граничного поля: $(300 + 420, 600 + 140)$ і $(300 + 1120, 600 + 720)$ або $(720, 740)$ та $(1420, 1320)$. Отримані результати передають у наступний об'єднаний шар. Об'єднання у пул даних зображення реалізовано за допомогою застосування спеціальної функції фільтрації для вибору розміру ядра та значення кроку так, щоб програми фільтрів не перекривали одна одну. Таку техніку називають максимальним об'єднанням. У максимальному об'єднанні вибирається максимальне значення підregionу для нашого результату підвибірки. На рис. 16 подано результат застосування максимального об'єднання вхідної матриці 4×4 до 2×2 ; на рис. 17 – результат подвійної вибірки вихідних даних шару згортки. В результаті об'єднання якість зображення погіршилась, але це не важливо, оскільки дані вже були проаналізовані раніше шарами згортки. Отримана матриця зберігається і передається в ущільнений шар. Ущільнені шари з'єднують кожен нейрон від попереднього шару до наступного. У контексті згорткової нейронної мережі вони утворюють класифікаційну частину мережі. Нейрони в ущільнених шарах дізнаються, з яких ознак складається кожен клас. Ущільнені шари складніші з погляду підбору параметрів, ніж шари згортки. Фільтр із ядром 4×4 із шару згортки має 26 параметрів, що не залежать від кількості вхідних нейронів. Повністю зв'язаний шар із шістьма об'єднаннями із 32×32 нейронами на попередньому шарі вже має $32 \times 32 \times 6 = 6144$ нейронів. Останній крок – упакування усіх даних у кортеж з трьома полями: “поле”, “шанс” і “ключові точки”. Поле “поле” зберігає координати отриманого обмежувального поля,

“шанс” характеризується рівнем довіри ЗНМ до кожної комірки, а “ключові точки” зберігають координати кожного елемента обличчя користувача (ніс, очі та кінцеві точки рота).

5	9	10	10
2	15	3	3
1	0	2	1
0	0	2	0

→
2x2
максимум

15	10
1	2



Рис. 16. Приклад максимального об'єднання

Рис. 17. Зміни після максимального об'єднання

Головним завданням інтелектуальної системи соціалізації груп користувачів за сукупними інтересами з використанням методів нечіткого пошуку текстової інформації та машинного навчання є пошук максимально схожих користувачів для подальшої соціалізації, тому було створено оптимізований алгоритм нечіткого пошуку для опрацювання текстових даних. Створений алгоритм є комбінацією чотирьох різних алгоритмів, до яких належать алгоритм розширення вибірки, алгоритм N-грам та алгоритм Noisy Channel на основі ВК-дерева. Для початку визначено, які саме дані подаються на вхід. У нашому випадку це загальний опис користувача, його інтереси та текстовий опис того, що саме шукає цей користувач. Всі дані формуються у вигляді кортежу, який складається з назви поданих текстових даних та самих масивів текстових даних.

Спочатку текстова інформація опрацьовується за допомогою алгоритму розширення вибірки, він добре підходить для цієї задачі, оскільки в середньому текстових даних у користувача не дуже багато, відповідно не ускладнене розширення. Суть алгоритму полягає в тому, що він вибирає слово зі сформованого масиву слів користувача і формує на його основі список помилкових слів, для яких здійснює пошук у словнику всіх відповідників. Для цієї задачі алгоритм було модифіковано так, що він звіряє згенеровані слова лише з тими словами, які збігаються із заданим шаблоном з початковим словом. У такий спосіб відкидаються всі зайві слова, всі зайві символи з масиву значень, а також всі слова, які технічно є словами, але не мають ніякого принципового значення для користувача і не можуть бути використані для порівняння зі словами інших користувачів, як, наприклад, прикметники та інші подібні частини мови. Завдяки роботі цього алгоритму формується розширена вибірка, яка ділиться на дві підвибірки, визначається середня довжина всіх отриманих слів і відповідно одна підвибірка містить довші слова, а інша коротші. Завдяки цьому алгоритм доволі просто і водночас ефективно здійснює розподіл слів, оптимізується робота двох описаних нижче створених алгоритмів.

Перша сформована підвибірка, яка складається здебільшого зі слів, довжина яких більша від середнього, опрацьовується за допомогою алгоритму N-грам. Сам по собі він доволі старий і часто ним нехтують через повільність роботи, але для цього випадку він підходить ідеально, оскільки потрібно ділити великі за довжиною слова на менші частинки. Суть алгоритму приблизно можна описати так: якщо перше слово збігається з другим словом, з урахуванням наявності деяких помилок, значить, у них буде хоча б один спільний підрядок, що має довжину N. Такі рядки і називають N-грамами. Найчастіше використовують три символи, тому їх деколи називають три-грамами. Суть полягає у тому, що всі наявні підвибірки розподіляють на триграми, після чого відбувається асинхронне порівняння того, що ми отримали після розподілу, тобто порівняння безпосередньо три-грам. Три-грами найзручніше використовувати, оскільки в середньому у такий спосіб здійснюється найзручніший поділ, особливо враховуючи те, що ми працюємо з підвибіркою, яка сама по собі містить великі за обсягом слова. Після того, як алгоритм здійснив порівняння всіх три-грам даних поточного користувача, формується тимчасова вибірка знайдених три-грам і

здійснюється відповідний підрахунок того, скільки три-грам збігаються і між двома користувачами, після чого ставлять оцінку від 1 до 10. Приклад роботи такого алгоритму N-грам подано на рис. 18. Також реалізовано механізм публікацій, які користувач може додавати самостійно або коментувати публікації інших користувачів, які теж посортовані відповідно до відсоткового співвідношення. Додатково подано механізм тегів, які автоматично підставляються до створених користувачем публікацій. Цей функціонал реалізовано за допомогою алгоритму Noisy Channel на основі ВК-дерева. Ця реалізація представляє метричні дерева, алгоритм створення таких дерев оснований на властивостях метрик відповідати нерівностям трикутників. Ці властивості дають змогу цим метрикам створювати метричні простори необхідної довільної розмірності. Суть алгоритму полягає в бінарному пошуку відповідних тегів у словнику відносно тих слів, які користувач вказав у своїй публікації. Завдяки алгоритму Noisy Channel вибране слово підбирається зі словника, коли знайдено максимальний бал подібності зі значенням у словнику. Приклад роботи алгоритму Noisy Channel на основі ВК-дерева наведено на рис. 19.

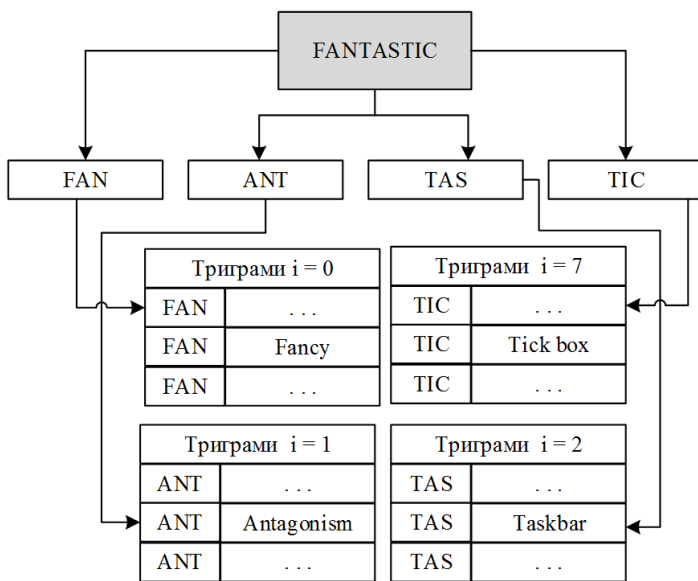


Рис. 18. Приклад роботи алгоритму N-грам

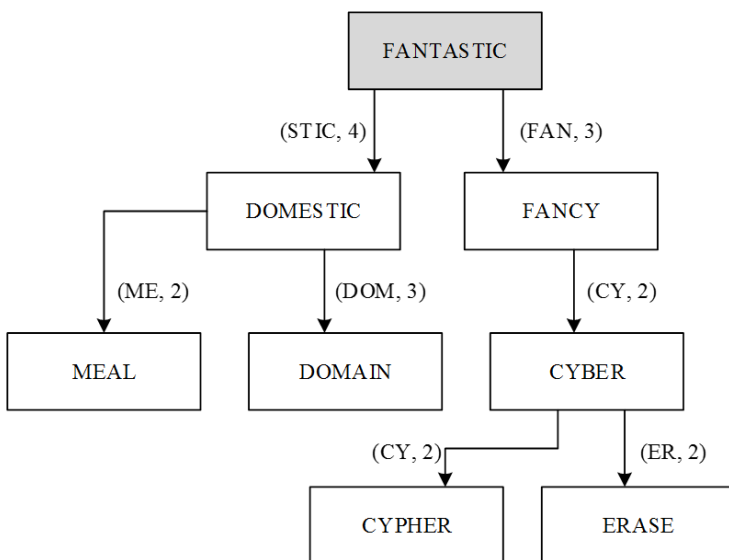


Рис. 19. Приклад роботи алгоритму Noisy Channel на основі ВК-дерева

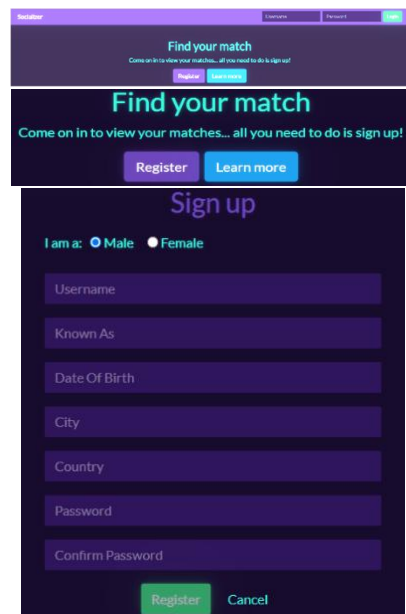


Рис. 20. Вхід у систему

Після реалізації інтелектуальної системи наведено контрольний приклад її роботи відповідно до поставленого завдання. На рис. 20 зображено початкове вікно системи, кнопки для реєстрації і авторизації, поля для реєстрації. На рис. 21, 22 наведено відображення залогіненого користувача, випадне вікно з параметрами користувача, поля змін користувачьких даних та вікно підтвердження змін даних.

На рис. 23 наведено додавання нових фотографій користувача, завантаження цих фотографій та додані фотографії, серед яких ті фотографії, на яких не було знайдено обличчя, відповідно їх не можна використати як аватар користувача.

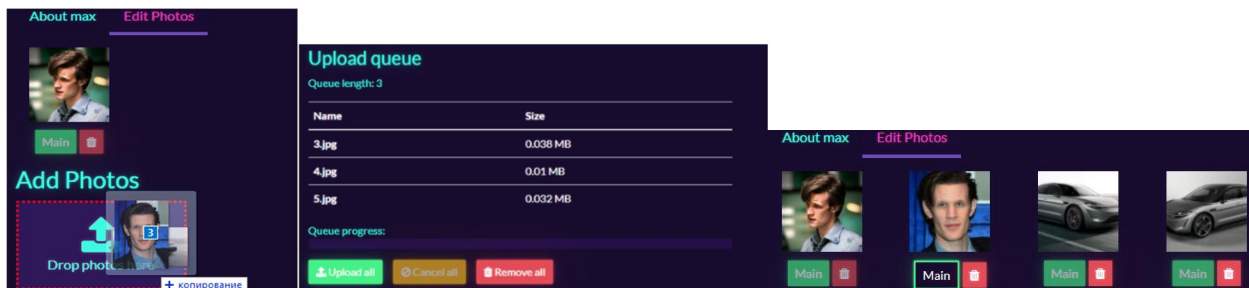


Рис. 21. Додавання нових фотографій користувача, їх завантаження та додані фотографії

В інтелектуальній системі подано сформований список користувачів, посортований за зменшенням значення відсоткового співвідношення подібності, список останніх активних користувачів, список найновіших користувачів, список користувачів та відфільтрований список. На рис. 22–25 наведено список вподобаних користувачів та список користувачів, що поставили вподобайку поточному користувачу, а також отримані/надіслані його повідомлення.

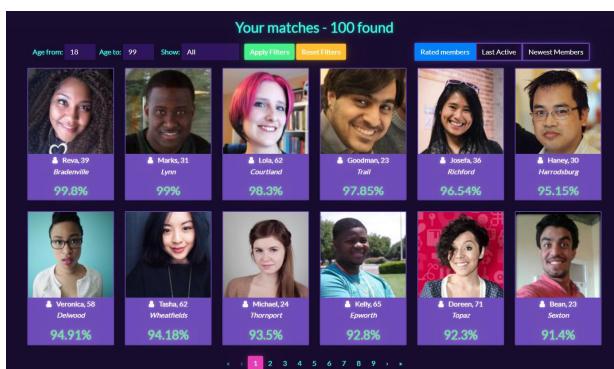


Рис. 22. Список користувачів за відсотком подібності

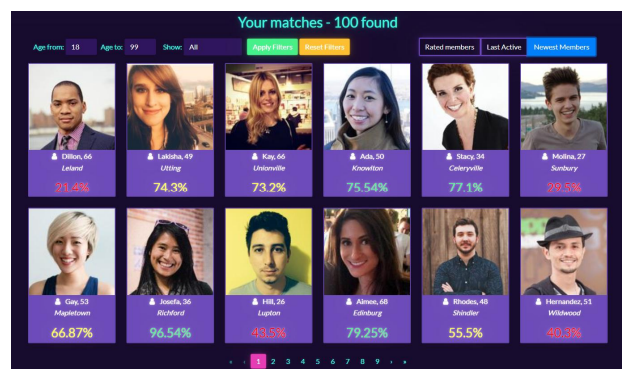


Рис. 23. Останні активні користувачі

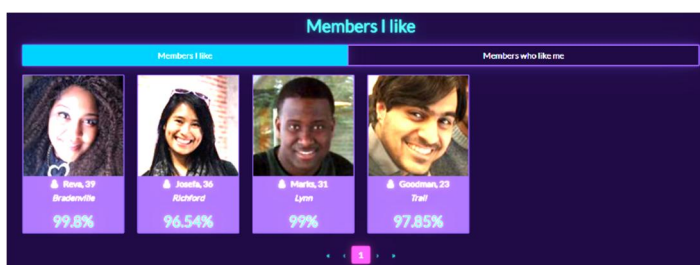


Рис. 25. Надіслані повідомлення

На рис. 26–28 подано коментарі та вподобайки від інших користувачів та вікно додавання вподобайки.



Рис. 26. Коментарі до публікації

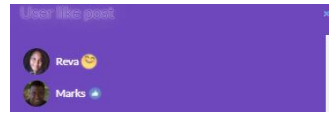


Рис. 27. Вподобання

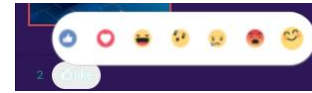


Рис. 28. Доступні вподобання

На рис. 29–32 подано загальну інформацію користувача, користувацькі інтереси та фотографії користувача. На рис. 33 зображено тих користувачів, які перебувають онлайн у конкретний момент часу.

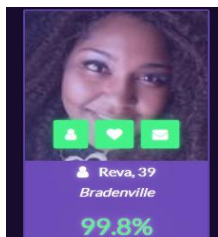


Рис. 29. Користувач зі списку користувачів

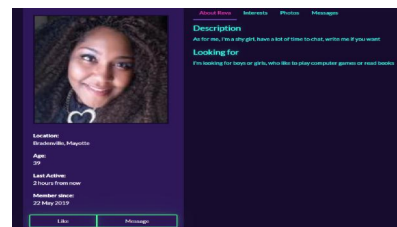


Рис. 30. Вкладка із загальною інформацією

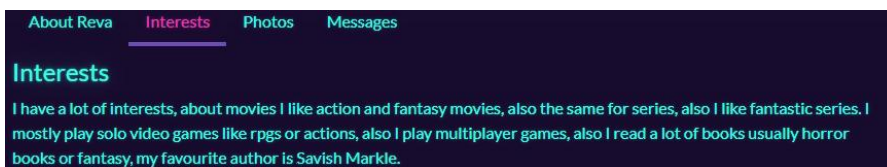


Рис. 31. Вкладка з інтересами користувача

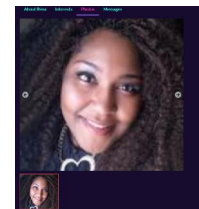


Рис. 32. Вкладка з фотографіями користувача

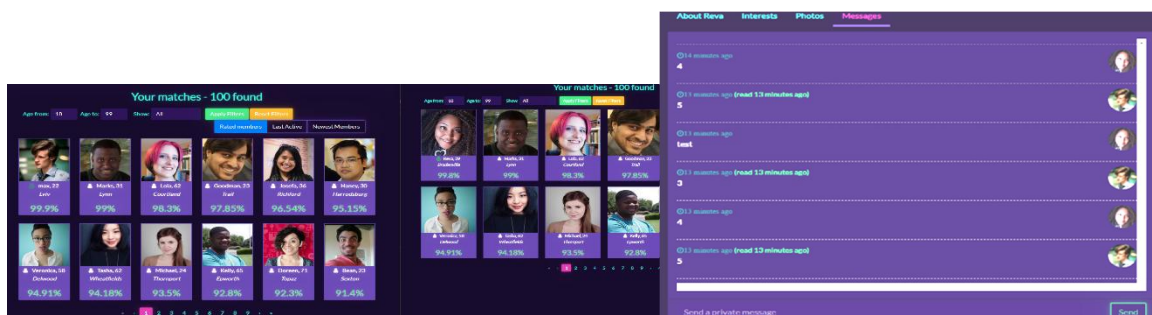


Рис. 33. Відображення онлайн-користувачів та приклад листування користувачів

На рис. 33 відображено листування з користувачем.

Проаналізовано роботу системи із 100 випадкових користувачів з кола знайомих у соціальній мережі одного зі співавторів. Порівняно точність роботи (рис. 34), тобто визначено відсоток

подібності поточних користувачів системи за допомогою алгоритму N-грам, порівняно з комбінацією алгоритмів, використаною у створеній інтелектуальній системі. Комбінація алгоритмів є ефективнішою і точнішою на 10 % від інших алгоритмів і на 15 % від алгоритму N-грам. Проаналізовано також роботу алгоритму лінійного пошуку тегів у словнику та алгоритму Noisy Channel з використанням ВК-дерева, завдяки чому вдалося досягти значної переваги в роботі алгоритму, оскільки замість лінійного часу пошуку отримано логарифмічний (рис. 35). Також порівняно роботу синхронних та асинхронних методів системи. Як видно з результатів на рис. 35, спочатку відмінностей не видно, але чим більше запитів, тим швидше навантажується система і починає довше на них відповідати, на відміну від асинхронних методів.

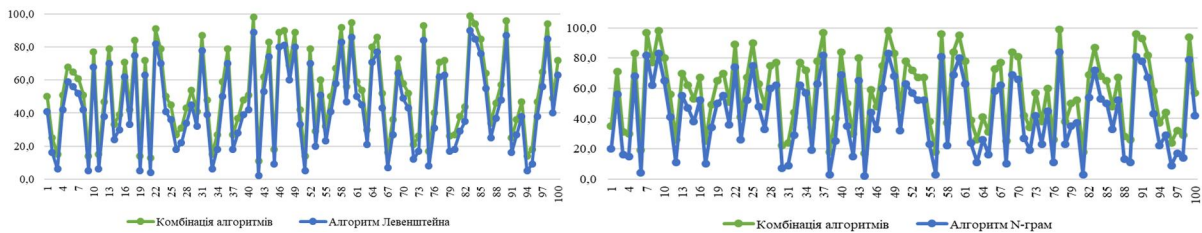


Рис. 34. Аналіз точності вихідних даних: Y – подібність користувачів, %; X – номер користувача

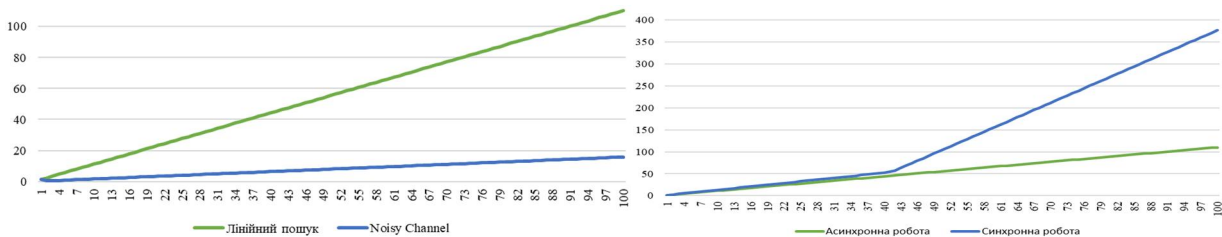


Рис. 35. Аналіз алгоритмів пошуку тегів та час відповіді на запит користувача:
 Y – час виконання, мс; X – номер користувача

Щоб оцінити ефективність розв'язання задачі класифікації, варто було б використати більше метрик (а не лише точність роботи, яка, зокрема для незбалансованої вибірки, не використовується), що ми плануємо зробити в наступних дослідженнях. Також варто надалі виконати оцінювання оптимальності параметрів роботи системи, зокрема нейронної мережі.

Висновки

Соціалізація груп користувачів за сукупними інтересами є важливим процесом. Під час виконання досліджень було здійснено аналітичний огляд наукових джерел. Завдяки цьому визначено, які із сучасних підходів найоптимальніші в сфері алгоритмів нечіткого пошуку та які реалізації згорткових нейронних мереж вже існують. Проаналізовано загальну поведінку користувачів у соціальних мережах, характер та особливості спілкування і здійснення різних дій, завдяки чому з'ясовано, як краще здійснювати побудову інтелектуальної системи, щоб нею було зручно користуватись саме середньостатистичному користувачеві популярних соціальних мереж. Також виділено вже наявні програми-аналоги, їхні переваги, які реалізовано в інтелектуальній системі, а також виділено недоліки, яких потрібно уникнути під час реалізації. Далі здійснено системний аналіз проблемної ситуації, проаналізовано основні компоненти, які необхідні системі, основні бізнес-процеси та потоки даних. Для концептуального відображення інтелектуальної системи використано об'єктні нотації UML, для максимальної деталізації цієї системи створено діаграми варіантів використання, класів, кооперації, станів, діяльності, моделювання бізнес-процесів, компонентів та розгортання. Також загалом сформульовано проблему, описано входи та виходи інтелектуальної системи та загальну структуру. Відтак вибрано основні методи та засоби вирішення проблеми

створення інтелектуальної системи соціалізації груп користувачів за сукупними інтересами, це важливе завдання для реалізації системи. Для створення інтелектуальної системи вибрано мови програмування JavaScript та C#, їх основні фреймворки, а саме Angular та ASP .NET Core і середовища розроблення ПЗ Visual Studio Code та Visual Studio, що дало змогу створити асинхронну систему із поділом на клієнтську і серверну частини. Проаналізовано особливості реалізації інтелектуальної системи. Її умовно поділено на три частини, а саме: робота з даними, реалізація алгоритмів та технічна реалізація системи. Роботу з базою даних реалізовано з використанням СУБД MSSQL Server та ORM Entity Framework Core, розроблено схему бази даних та описано особливості збереження даних, такі як збереження медіафайлів у хмарному середовищі. Реалізовано алгоритми N-грам, Noisy Channel, за допомогою яких здійснюється нечіткий пошук у тексті й забезпечується основна функція системи, а саме аналіз текстових даних про користувача та формування ідентифікатора подібності поточного користувача на інших користувачів системи, відповідний підбір публікацій користувачів, які будуть цікаві лише для певних груп користувачів зі схожим ідентифікатором. Реалізовано оптимізовану згорткову нейронну мережу, яка здійснює пошук обличчя користувача на фотографії. У ході дослідження виконано основні завдання та досягнуто поставленої мети. Створена інтелектуальна система дає користувачу змогу здійснити реєстрацію та авторизацію у системі, завантажити своє фото, отримати ідентифікатор подібності з іншими користувачами, здійснювати листування з декількома користувачами за допомогою реалізованого динамічного чату й обговорювати різні теми з групами користувачів за допомогою реалізованого механізму, публікувати, оцінювати та коментувати публікації.

Список літератури

1. Chu S. C. (2016). Using a consumer socialization framework to understand electronic word-of-mouth (eWOM) group membership among brand followers on Twitter. *Electronic Commerce Research and Applications*, No. 14 (4), 251–260.
2. De-Gregorio F., Sung Y. (2010). Understanding attitudes toward and behaviors in response to product placement. *Journal of Advertising*, No. 39 (1), 83–96. DOI: <http://doi.org/10.2753/JOA0091-3367390106>.
3. Elaheebocus S. M., Weal M., Morrison L. (2018). Peer-based social media features in behavior change interventions: Systematic review. *Journal of Medical Internet Research*, No. 20(2), 1–20. DOI: <http://doi.org/10.2196/jmir.8342>.
4. Ferrara E., Interdonato R., Tagarelli A. (2014). Online popularity and topical interests through the lens of Instagram. *Hypertext and Social Media*, No. 2, 24–23. DOI: <http://doi.org/10.1145/2631775.2631808>.
5. Geurin-Eagleman A. N. (2015). Communicating via photographs: A gendered analysis of Olympic athletes' visual self -presentation on Instagram. *Sport Management Review*, No. 19 (2), 133–145. DOI: <http://doi.org/10.1016/j.smr.2015.03.002>.
6. Hanna R., Rohm A., Crittenden V. L. (2011). We're all connected: The power of the social media ecosystem. *Business Horizons*, No. 54 (3), 265–273. DOI: <http://doi.org/10.1016/j.bushor.2011.01.007>.
7. Salganik M. (2019). Social Research in the Digital Age. *Journal of Interactive Marketing*, No. 2 (9), 345–358.
8. Guidry J. D., Messner M., Jin Y. (2015). From McDonalds fail to Dominos sucks: An analysis of Instagram images about the 10 largest fast food companies. *Corporate Communications: An International Journal*, No. 20 (3), 344–359.
9. Kim E. (2014). Brand followers' retweeting behaviour on Twitter: How brand relationship influence brand electronic word-of-mouth. *Computers in Human Behavior*, No. 38 (8), 18–25.
10. Kudeshia C., Sikdar P., Mittal A. (2016). Spreading love through fan page liking: A perspective on small scale entrepreneurs. *Computers in Human Behavior*, No. 8 (19), 257–270. DOI: <http://doi.org/10.1016/j.chb.2015.08.003>.
11. Lueg J. E. (2007). Interpersonal communication in the consumer socialization process: Scale development and validation. *Journal of Marketing Theory and Practice*, No. 15 (1), 25–39. DOI: <http://doi.org/10.2753/MTP1069-6679150102>.

12. Mousavijad M. (2017). The effect of socialization factors on decision making of teenagers consumers in schools. *Journal of School Administration*, No. 5 (1), 217–234.
13. Schnell R. (2015). Enhancing Surveys with Objective Measurements and Observer Ratings. *Journal of Interactive Marketing*, No. 44 (1), 288–302.
14. Quan-Haase A., Sloan L. (2017). Introduction to the Handbook of Social Media Research Methods: Goals, Challenges and Innovations. *The Sage Handbook of Social Media Research Methods*, No. 10 (5), 606–859. DOI: <http://doi.org/10.4135/9781473983847.n1>.
15. Murphy S. T. (2001). Affect, cognition, and awareness: Affective priming with optimal and suboptimal stimulus exposures. *Journal of Personality and Social Psychology*, No. 8 (3), 723–739. DOI: <http://doi.org/10.1037/0022-3514.64.5.723>.
16. Ntale P. D. (2019). Word of mouth communication: A mediator of relationship marketing and customer loyalty. *Cogent Business and Management*, No. 6 (18), 3–36.
17. Ozdemir A., Tozlu B., Şen E., Ateşoğlu A. (2016). Analyses of word-of-mouth communication and its effect on students' university preferences. *Procedia – Social and Behavioral Sciences*, No. 8 (5), 22–35.
18. Park J., Ciampaglia G. L., Ferrara F. (2016). Style in the age of Instagram: Predicting success within the fashion industry using social media. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, No. 22 (8), 64–72. DOI: <http://doi.org/10.1145/2818048.2820065>.
19. Wang X., Yu C., Wei Y. (2012). Social media peer communication and impacts on purchase intentions: A consumer socialization framework. *Journal of Interactive Marketing*, No. 26 (4), 198–208.

References

1. Chu S. C. (2016). Using a consumer socialization framework to understand electronic word-of-mouth (eWOM) group membership among brand followers on Twitter. *Electronic Commerce Research and Applications*, No. 14 (4), 251–260.
2. De-Gregorio F., Sung Y. (2010). Understanding attitudes toward and behaviors in response to product placement. *Journal of Advertising*, No. 39 (1), 83–96. DOI: <http://doi.org/10.2753/JOA0091-3367390106>.
3. Elaheebocus S. M., Weal M., Morrison L. (2018). Peer-based social media features in behavior change interventions: Systematic review. *Journal of Medical Internet Research*, No. 20 (2), 1–20. DOI: <http://doi.org/10.2196/jmir.8342>.
4. Ferrara E., Interdonato R., Tagarelli A. (2014). Online popularity and topical interests through the lens of Instagram. *Hypertext and Social Media*, No. 2, 24–23. DOI: <http://doi.org/10.1145/2631775.2631808>.
5. Geurin-Eagleman A. N. (2015). Communicating via photographs: A gendered analysis of Olympic athletes' visual self-presentation on Instagram. *Sport Management Review*, No. 19 (2), 133–145. DOI: <http://doi.org/10.1016/j.smr.2015.03.002>.
6. Hanna R., Rohm A., Crittenden V. L. (2011). We're all connected: The power of the social media ecosystem. *Business Horizons*, No. 54 (3), 265–273. DOI: <http://doi.org/10.1016/j.bushor.2011.01.007>.
7. Salganik M. (2019). Social Research in the Digital Age. *Journal of Interactive Marketing*, No. 2 (9), 345–358.
8. Guidry J. D., Messner M., Jin Y. (2015). From McDonalds fail to Dominos sucks: An analysis of Instagram images about the 10 largest fast food companies. *Corporate Communications: An International Journal*, No. 20 (3), 344–359.
9. Kim E. (2014). Brand followers' retweeting behaviour on Twitter: How brand relationship influence brand electronic word-of-mouth. *Computers in Human Behavior*, No. 38 (8), 18–25.
10. Kudeshia C., Sikdar P., Mittal A. (2016). Spreading love through fan page liking: A perspective on small scale entrepreneurs. *Computers in Human Behavior*, No. 8 (19), 257–270. DOI: <http://doi.org/10.1016/j.chb.2015.08.003>.
11. Lueg J. E. (2007). Interpersonal communication in the consumer socialization process: Scale development and validation. *Journal of Marketing Theory and Practice*, No. 15 (1), 25–39. DOI: <http://doi.org/10.2753/MTP1069-6679150102>.
12. Mousavijad M. (2017). The effect of socialization factors on decision making of teenagers consumers in schools. *Journal of School Administration*, No. 5 (1), 217–234.
13. Schnell R. (2015). Enhancing Surveys with Objective Measurements and Observer Ratings. *Journal of Interactive Marketing*, No. 44 (1), 288–302.

14. Quan-Haase A., Sloan L. (2017). Introduction to the Handbook of Social Media Research Methods: Goals, Challenges and Innovations. *The Sage Handbook of Social Media Research Methods*, No. 10 (5), 606–859. DOI: <http://doi.org/10.4135/9781473983847.n1>
15. Murphy S. T. (2001). Affect, cognition, and awareness: Affective priming with optimal and suboptimal stimulus exposures. *Journal of Personality and Social Psychology*, No. 8 (3), 723–739. DOI: <http://doi.org/10.1037/0022-3514.64.5.723>
16. Ntale P. D. (2019). Word of mouth communication: A mediator of relationship marketing and customer loyalty. *Cogent Business and Management*, No. 6 (18), 3–36.
17. Ozdemir A., Tozlu B., Şen E., Ateşoğlu A. 2016. Analyses of word-of-mouth communication and its effect on students' university preferences. *Procedia-Social and Behavioral Sciences*, No. 8 (5), 22–35.
18. Park J., Ciampaglia G. L., Ferrara F. (2016). Style in the age of Instagram: Predicting success within the fashion industry using social media. *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, No. 22 (8), 64–72. DOI: <http://doi.org/10.1145/2818048.2820065>.
19. Wang X., Yu C., Wei Y. (2012). Social media peer communication and impacts on purchase intentions: A consumer socialization framework. *Journal of Interactive Marketing*, No. 26 (4), 198–208.

INTELLIGENT SYSTEM FOR USER GROUPS SOCIALIZATION WITH SIMILAR INTERESTS

Taras Batiuk¹, Victoria Vysotska^{1,2}, Maryna Shevchenko³

¹ Lviv Polytechnic National University,

Information Systems and Networks Department, Lviv, Ukraine

² Osnabrück University, Institute of Computer Science, Osnabrück, Germany

³ Osnabrück University, International Economic Policy Chair, Osnabrück, Germany

E-mail: taras.batiuk.mnsa.2020@lpnu.ua, ORCID: 0000-0001-5797-594X,

E-mail: Victoria.A.Vysotska@lpnu.ua, ORCID: 0000-0001-6417-3689

E-mail: mshevchenko@uni-osnabrueck.de, ORCID: 0000-0003-2165-9907

© Batiuk T., Vysotska V., Shevchenko M., 2023

The article develops a general architectural system of socialization of groups of users with similar interests and functional requirements for it. To process a large part of the information, the system is implemented using the methods of fuzzy text information search and machine learning. thus, N-gram, selection expansion and structured Noisy Channel models are applied. A feature of the implementation is the processing of the text, the analysis of words in the text and the formation of evaluations. A convolutional neural network implementation is designed to determine user authenticity based on facial photo analysis. implementation of fuzzy search algorithms – for processing text data of various volumes to analyze information about each user, form a certain user rating, compare this user with other users to promote further socialization of users whose interests coincide the most. When experimentally checking the accuracy of the developed system by determining the percentage of similarity of current users with the help of N-grams and their connections. Running these algorithms simultaneously is about 15 % more accurate than the N-gram algorithm and about 10 % more efficient and accurate than the others algorithm. The operation of the algorithm for linear search of tags in the dictionary and the operation of the Noisy Channel algorithm using the BK-tree are also analyzed. Thanks to which it was possible to achieve significant advantages in the work algorithm, instead of a linear view of the search time, a logarithmic dependence was obtained. A system of synchronous and asynchronous methods also works. At first, the difference is not visible, but the more requests, the faster the system loads and tries to respond to them more by displaying from asynchronous methods.

Key words: convolutional neural network; fuzzy search; sample extensions; N-gram algorithm; Noisy Channel; JWT tokens.