

ПРОТОТИП ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ТА КОНТЕНТ-АНАЛІЗУ ЗВЕРНЕНЬ МЕШКАНЦІВ “РОЗУМНОГО МІСТА”

Юліан Огоновський¹, Андрій Берко¹, Любомир Чирун², Софія Чирун¹

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

² Львівський національний університет імені Івана Франка,
кафедра прикладної математики, Львів, Україна

E-mail: yulian.ohonovskyi.knm.2018@lpnu.ua, ORCID: 0000-0003-3853-4158

E-mail: Andrii.Y.Berko@lpnu.ua, ORCID: 0000-0003-2892-9519

E-mail: Lyubomyr.Chyrun@lnu.edu.ua, ORCID: 0000-0002-9448-1751

E-mail: sofii.chyrun.sa.2022@lpnu.ua, ORCID: 0000-0002-2829-0164

© Огоновський Ю. Ю., Берко А. Ю., Чирун Л. В., Чирун С. Л., 2023

Однією із сучасних тенденцій комп'ютеризації є активний розвиток технологій “розумного міста” (“Smart city”), зокрема, спрямованих на супровід адміністративних е-послуг, розроблення транспортних систем, використання екологічно чистих джерел енергії, розвиток екології, модернізацію цифрового обладнання міста, поліпшення життя у місті, пришвидшення усіх процесів міської інфраструктури. Весь цей комплекс робіт стає максимально доступним за правильного використання технологій “Smart city”. “Розумні міста” сприяють сталому розвитку. Розумні технології допоможуть містам підтримувати зростання, щоб забезпечити добробут громадян і ефективність управління в містах у найближчі роки. Поряд з перевагами Smart city є проблеми, спричинені широким використанням різних мобільних застосунків, великих даних та інформаційних технологій. Щоб ризики не переважували позитивні переваги, всі “розумні” міські ініціативи повинні бути орієнтовані на потреби споживачів. Доступ адміністративних даних підвищить рівень залученості громадян в громадські та міські процеси. Противники концепції “розумного міста” висловлюють заклопотаність загрозою недотримання вимог належного зберігання конфіденційних і персональних даних, забезпечення їх безпеки. Мета статті – аналіз переваг інформаційної системи підвищення швидкості реакції комунальних служб “розумного міста” на різні проблеми, що виникають в ньому. А мешканці міста зможуть повідомляти про різні екстрені ситуації відповідні служби, які вживатимуть заходів для швидкого їх усунення. Об'єктом дослідження є зменшення часу реагування комунальних служб на різноманітні надзвичайні ситуації в місті. Предметом дослідження є аналіз швидкості реагування комунальних служб на скарги мешканців. Практичне значення одержаних результатів полягає у розробленні програмного застосунку для збирання та аналізу проблем у місті.

Ключові слова: Smart city; розумне місто; скарги; концепція; інформаційні технології; економічна ефективність; конкурентоспроможне місто; інформаційна система.

Вступ

В сучасних умовах перенаселеності міст, руйнування пам'яток архітектури для будівництва офісних центрів актуальна проблема оптимального розвитку міського розвитку. Це пов'язано насамперед із тим, що на терезах – благополуччя громадян, загострення криміногенної ситуації,

тотальне забруднення навколишнього середовища, що іноді призводить до трагічних ситуацій. За прогнозами WEF (World Economic Forum, <https://www.weforum.org/agenda/2019/01/the-anatomy-of-a-smart-city/>), до 2040 р. 60 % населення житиме в містах [1]. Зокрема, сьогодні 80 % населення США проживає у мегаполісах проти 60 % 50 років тому. IDC (International Data Corporation компанія маркетингових досліджень) прогнозує, що в 2018 р. глобальні витрати на інтелектуальні міста досягнуть 80 мільярдів доларів, з яких – 22 мільярди тільки в США; і зросте до 135 мільярдів доларів до 2021 р. [2–4]. “Розумні міста” сприяють сталому розвитку країни. “Розумні” технології допоможуть містам підтримувати зростання і підвищувати ефективність побутування, щоб забезпечити добробут їхніх громадян і ефективність управління містами в найближчі роки. Противники концепції “розумного міста” висловлюють заклопотаність загрозою недотримання належного зберігання конфіденційних і персональних даних, забезпечення їх безпеки. Крім того, наявність давачів і камер спостереження може трактуватися як вторгнення в приватне життя або налагодження системи тотального нагляду, які описано в книзі “1984” Д. Орвелла. Основна мета статті – аналіз переваг інформаційної системи для підвищення швидкості реакції комунальних служб “розумного міста” на різні проблеми в ньому. А жителі міста з використанням інформаційної системи зможуть повідомляти про різні екстрені ситуації спеціальним службам, щоб вони вжили заходів, спрямованих на швидке їх усунення. Об’єкт дослідження – зменшення часу реагування комунальних служб на різноманітні надзвичайні ситуації в місті. Предмет дослідження – аналіз швидкості реагування на скарги мешканців. Практичне значення одержаних результатів полягає у розробленні інформаційної системи для збирання та аналізування проблем у місті.

Аналіз останніх досліджень та публікацій

Упродовж 2020 р. державна установа “Урядовий контактний центр” налагодила функціонування у цілодобовому режимі урядової “гарячої лінії”, спеціалізованих “гарячих ліній”, зокрема для громадян з вадами слуху за допомогою відеозв’язку; приймання електронних звернень, які подають заявники через Урядовий портал, вебсайт Офіційного інтернет-представництва Президента України, вебсайт Центру, офіційну сторінку в соціальній мережі Facebook, а також в онлайн-режимі (чати) через месенджери. Працівники Центру в 2020 р. забезпечили прийняття 1 374 770 дзвінків; зареєстровано 1 780 490 звернень (у 2019 р. – 1 225 189), що надійшли через засоби телефонного зв’язку, зокрема за допомогою спеціалізованих “гарячих ліній” – 1 624 741, і технології відеозв’язку – 155, а також інтернет – 112 741 електронних звернень (через Урядовий портал – 12 371, вебсайт Офіційного інтернет-представництва Президента України – 13 083, вебсайт Центру – 87 287) та через чати – 42 824 [5]. На фоні впровадження нових спеціалізованих “гарячих ліній” та розширення комунікації з громадянами Центр зафіксував надходження найбільшої за останні п’ять років кількості звернень за допомогою чатів для онлайн-консультування, що пояснюється епідемією коронавірусу. Окрім того, працівники Центру опрацювали 1 129 685 запитів громадян, отриманих за проектом “Цифрова держава” через чат “Дія” та через інші спеціалізовані чати. Загалом Центр зареєстрував та опрацював 2 910 175 звернень та запитів від громадян. Упродовж останнього часу запроваджено роботу голосового чат-бота та сервісу для перевірки стану розгляду звернень за допомогою месенджерів [6, 7]. Найактуальнішими у зверненнях заявників до повномасштабної війни в Україні (з урахуванням опрацьованих Центром) були питання [7]:

- соціального захисту населення (362 087 звернень проти 303 961 у 2019 р.);
- діяльності органів виконавчої влади та органів місцевого самоврядування (213 792 проти 63 959);
- комунального господарства (179 262 проти 203 679); пов’язані з коронавірусом (120 115);
- пенсійного забезпечення (77 021);
- роботи органів юстиції (65 859 проти 40 353);

- охорони здоров'я (47 980 проти 37 642);
- діяльності посадових і службових осіб, корупції (45 635).

Статистичні дані свідчать, що “гарячі лінії” часто використовують мешканці міста [8], але вони не дуже зручні. Можна виділити три найпопулярніші методи надсилання повідомлень за допомогою: телефонного зв'язку; різних чатів та ботів; програми “Дія”. Одним із прикладів вирішення проблеми можуть слугувати самі “гарячі лінії”, але запропоновано прототип із кращою реалізацією. Проаналізуємо наявні на ринку аналогічні застосунки.

Citizen – мобільний застосунок, який у режимі реального часу надсилає користувачам попередження про небезпеку на основі визначення місцезнаходження. Функціонал застосунку дає користувачам змогу читати поточні звіти, що постійно оновлюються, транслювати відео в реальному масштабі часу та залишати коментарі. Застосунок використовує радіоантени, встановлені у великих містах, для моніторингу зв'язку 911, а працівники фільтрують звук для створення сповіщень [9].

У 2016 р. нью-йоркська корпорація sp0n. Inc. запустила застосунок із назвою Vigilante. Мета роботи програми, основаної на маркетингу, – дати мешканцям змогу оперативно повідомляти про місцеві злочини в їх районі та транслювати пряме відео з місця події, що може стати чинником, який стримуватиме злочинців. Проте Vigilante вилучили із магазинів застосунків, щоб люди перестали ризикувати життям заради створення контенту.

Через декілька років sp0n замінив цей застосунок новою версією Citizen. Citizen вніс зміни у формулу, зокрема постійні повідомлення про те, що необхідно уникати небезпеки під час повідомлення про злочини [10]. Окрім цього, Citizen – це застосунок, що забезпечує надсилання повідомлень про злочини, свідками яких виявились мешканці.

Citizen створено для того, щоб повідомляти користувачів про небезпеку, яка може підстерігати їх у районі проживання. Мешканцям надано можливість сповіщати користувачів у режимі реального часу про злочини, аварії та інші види небезпечних подій у районі проживання. Всі сповіщення генеруються зі змісту реальних дзвінків на номер 911. Застосунок також може надавати інформацію про інші події, такі як перекриття доріг чи дії поліції. Отже, використовуючи застосунок, мешканець знатиме, чому певна вулиця перекрита або чому тут летить вертоліт [11].

Ці повідомлення, що оновлюються в реальному часі, можуть бути корисними для з'ясування уточнювальних даних про інцидент. Citizen також використовували для відстеження контактів хворих на COVID-19: користувачі мали змогу одразу згадати, коли саме у них був небезпечний контакт, щоб вжити необхідних запобіжних заходів. Ця програма допомагає набагато швидше дізнатись, що відбувся якийсь інцидент, а свідки можуть надати інформацію для миттєвого вирішення проблеми.

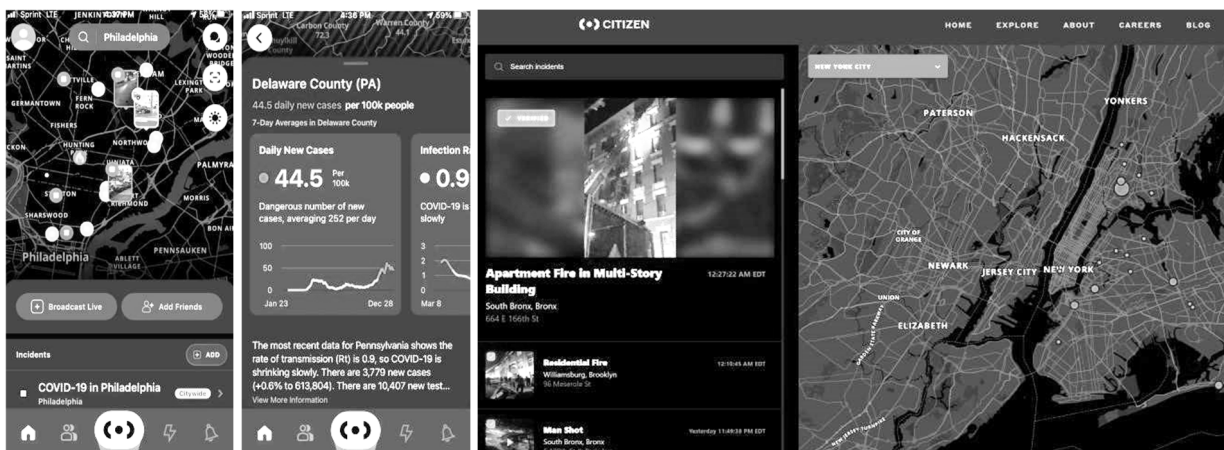


Рис. 1. Вигляд додатка Citizen

Основна функція, необхідна в розроблюваному застосунку, – це інтерактивна карта [12–16]. Ця функція, за допомогою якої зручно переглядати карту міста, використовуватиме мапи Google. Статус функції – увімкнена, пріоритет критичний, оскільки ця функція є одним із базових способів використання програми, рівень трудомісткості високий, ризик низький, стабільність висока (будуть використовуватись готові карти, а функція буде реалізована в першій робочій версії продукту).

Наступною функцією, яку потрібно реалізувати в цьому програмному застосунку, – додавання користувачем на карту місця, в якому трапився інцидент. Ця функція буде використовувати технологію GPS, яка допомагає користувачеві точніше встановити своє місцезнаходження для швидшого виявлення місця події. Статус функції – увімкнена, пріоритет критичний, оскільки вона є одним з базових способів використання програми, рівень трудомісткості високий, ризик низький, стабільність середня, бо у користувачів неоднакові пристрої та швидкість інтернет-з'єднання; функція буде реалізована в першій робочій версії продукту.

Додатковою функцією буде можливість створення прямої трансляції, що дасть змогу користувачу в реальному часі повідомляти про подію. Пропонована функція, пріоритетна, корисна, трудомісткість середня, ризик високий, стабільність висока. Наступною функцією, яка повинна бути в системі, є функція реєстрації користувачів. Статус увімкнена, пріоритет функції критичний, рівень трудомісткості середній, ризик реалізації доволі високий, стабільність висока, тому що користувачу лише потрібно ввести дані про подію. Додатковою функцією буде формування інтерактивного рядка інцидентів, які відбуваються поблизу користувача. Користувач зможе визначити радіус території, в межах якого хоче отримувати сповіщення про інциденти. Також додатковою функцією, яка розширить можливості програмного застосунку, є функція VR, які генеруватимуть різні підказки з доповненою реальністю, зокрема підказки щодо дорожніх знаків.

База даних є важливою частиною будь-якої інформаційної системи, тому застосунок повинен містити функцію роботи з базою даних. В базу даних вноситиметься основна інформація про міські проблеми, про які повідомили мешканці [17–19]. Також функція забезпечуватиме очищення застарілих або неперевіжених даних. Функції затверджувана, пріоритет важливий, трудомісткість оцінюється середнім рівнем, ризик середній, стабільність середня. Функція збирання статистичних даних та надсилання звітів комунальним службам міста доволі корисна, бо дасть змогу в зручній формі переглядати всі проблемні місця у визначеному районі. Статистичні дані будуть збирати й опрацьовувати в міру використання попередніх функцій. Статус функції пропонується, пріоритет корисний, трудомісткість низька, ризик незначний, бо реалізація функції не залежить від основної роботи застосунку, стабільність висока, цільова версія продукту залежатиме від реалізації основних функцій.

Функція співпраці із різноманітними пристроями “розумного міста” [20–25] передбачає надання доступу до відеокамер або різних датчиків у місті, а також інтеграцію в інші середовища. Виконання цієї функції сприяє збиранню статистичних даних і у випадку визначення підозрілих факторів передаватиме інформацію про них оператору. Функція пропонується, пріоритет корисний, трудомісткість висока, ризик високий, стабільність низька, функція буде реалізована у майбутніх версіях.

Функціонал застосунку сприятиме прийняттю ефективних рішень, під цією процедурою розуміють всю сукупність процесів, що приводять до знаходження рішень. Більшість етапів процесу прийняття рішення – це якісний аналіз інформації. Прямі та зворотні зв'язки між етапами переважно якісні. Схематично перелік основних запитань, відповіді на які допомагають прийняти рішення, виглядають так: “хто”, де, як, коли і що потрібно. Ці етапи прийняття рішень реалізуються як інтерактивні процедури, поєднані зв'язками, утворюють певну послідовність. Зазвичай виділяють декілька етапів.

Виявлення та описання проблемної ситуації означає розмежування і формулювання проблеми на основі попереднього аналізу даних. Збирають та аналізують потрібну внутрішню і зовнішню інформацію. Релевантна інформація використовується як основа прийняття рішення і потребує максимальної точності й відповідності проблемі.

Формулювання завдання. На цьому етапі реалізуються виділення і описання проблемної ситуації, визначають час прийняття рішень, матеріальні та трудові ресурси. На цьому етапі також описують інформаційні обмеження, з урахуванням того, що інформація – це дані, що стосуються лише конкретної проблеми. У вартості інформації треба враховувати вартість часу, який витратив децидент і підлеглі.

Поведінкові обмеження. Децидент може отримувати різні визначення тієї самої проблеми залежно від підрозділу. З погляду витрат часу треба враховувати, що практично всі рішення приймаються в умовах його дефіциту. Для прийняття ефективного рішення необхідно проаналізувати матеріальні та трудові ресурси. Підготовка прийняття рішення – це всебічне дослідження проблемної ситуації, причин її виникнення та визначення способів розв'язання.

Формулювання і структурування мети. Теорія прийняття рішень дає можливість подолати або зменшити суперечності системи цілей і забезпечити прийняття рішення, які найкраще відповідають цілям, що ставляться.

Виявлення та генерування альтернатив. Варіанти рішень розглядаються як наперед визначені, суть проблеми зводиться до вибору найкращого.

Описання можливих станів зовнішнього середовища. Цей етап аналогічний етапу визначення альтернатив. Як зовнішнє середовище розглядають фактори, якими не можна керувати, але під дією яких формується результат розв'язання проблемної ситуації.

Оцінювання станів зовнішнього середовища. Теорія реалізує особливості побудови найкращої альтернативи залежно від знання, розподілу вірогідностей, виникнення конкретних станів середовища, а також за умов невизначеності та активності середовища.

Виявлення можливих результатів дій. Виявляють не лише всі наявні компоненти вектора результатів реалізації альтернатив, але і передбачають місце і час їхньої появи. На кроці описання і оцінювання реалізації альтернатив конкретизують результати якісного аналізу, отримані на попередніх етапах для кожного окремого варіанта дій. Для кожного стану зовнішнього середовища, для якого є альтернативи, описують кількісні та якісні характеристики.

Вибір критерію оцінювання здійснюється одразу після етапу конкретизації мети. Проблема узгодженості цілей є основною.

Оцінювання відповідності результатів поставленим цілям. Використовують два підходи: попарне зіставлення результатів або розроблення техніки оцінювання ефективності результату. Перший підхід використовують для прийняття стандартних рішень з обмеженою кількістю порівняльних альтернатив і стану зовнішнього середовища. Другий підхід скерований на розроблення функцій корисності на основі дослідження об'єктивних закономірностей функціонування процесу без вивчення суб'єктивної поведінки і суб'єктивних переваг децидента в процесі управління.

Оцінювання очікуваного ефекту дій. Оцінювання рівня відповідності не може бути основою для вибору найкращої альтернативи. Теорія прийняття рішень керується певною кількістю правил, які дають змогу оцінити очікувану корисність альтернатив.

Порівняння альтернатив за очікуваними ефектами і вибір найкращої. Оцінюючи рішення, децидент визначає переваги та недоліки кожного з них, можливості впливу на загальні наслідки. Під час оцінювання варіантів рішень децидент намагається спрогнозувати те, що відбудеться в майбутньому. Якщо проблему правильно визначено, а альтернативні рішення ретельно оцінено, це спрощує процедуру вибору. Децидент вибирає альтернативу за найсприятливішими загальними наслідками. Проблеми виникають у випадку багатовимірності системи критеріїв для визначення

якості системи і досягнення мети. Якщо як мету виявлено багатовимірний випадок, то оцінки ефекту за відповідними критеріями є суперечливими і в результаті дослідження можливо тільки побудувати множину недомінованих рішень (паретооптимальних розв'язків), а для остаточного вибору необхідна додаткова інформація від децидента.

Етап прийняття рішення. Децидент доповнює результат неформальними знаннями про об'єкт управління.

Етап реалізації рішення. Для розв'язання проблеми рішення має бути реалізоване. Рівень ефективності впровадження рішення підвищиться, якщо його визнають ті, кого воно стосується. Повне впровадження рішень потребує задіявання всього процесу управління, особливо його організаційної та мотиваційної функції.

Формулювання мети та постановка задачі

Аналіз функціонала застосунку, що розглядається у цій роботі, починається з побудови дерева цілей. В основі цього дерева – процес логічного мислення, який, своєю чергою, тісно пов'язаний із теорією обмежень. Один із найважливіших принципів та правил побудови графіка під назвою “дерева цілей” – це принцип, або ж правило “повноти редукції”. Повнотою редукції вважають процес зведення частини системи, однієї із компонент до ще роздрібненіших та декомпонованіших складових. Під час побудови дерева цілей основну увагу зосереджують на верхній (основній) цілі. Основна ціль декомпонується, утворюються підкатегорії, що слугують способами для досягнення верхньої цілі. Під час побудови дерева цілей важливо дотримуватись правил, які виконують роль вимог для побудови цих типів графіків:

- 1) цілі кожного рівня мають відрізнятись за обсягом релевантності та важливістю;
- 2) цілі та вимоги повинні бути сформовані за правилом SMART (чітко, з прив'язкою до часу, можливості виконання);
- 3) основною вимогою до побудови дерева рішень є дотримання принципу повноти редукції, який полягає в тому, що кожен мету розкладають на низку інших, які слугують засобами, що допоможуть рухатись деревом рішень, для досягнення кінцевої цілі;
- 4) описуючи ціль, визначають не способи її можливого досягнення, а бажаний результат;
- 5) підцілі, розташовані на одному рівні, мають бути автономними, незалежними від інших цілей цього ж рівня. Крім того, вони не виходять із сусідніх цілей;
- 6) дерево вважається побудованим тоді, коли сформовано поняття, які можна розглядати як кінцеві альтернативні варіанти досягнення мети проєктованої системи;
- 7) між цілями, що містяться на різних рівнях дерева, не повинно бути жодних змістових конфліктів;
- 8) декомпозиція компонентів дерева цілей відбувається на всіх суб'єктах дерева однаковим методом;
- 9) дерево цілей повинно буде забезпечене правильними та релевантними зв'язками у його структурі. Варто не забувати, що у дереві цілей існують як вертикальні, так і горизонтальні зв'язки.

Для проєктованої системи побудовано дерево цілей (рис. 2), подане у вигляді чотирирівневої ієрархічної структури:

- *перший рівень* – це основна мета цієї роботи, що полягає у створенні проєкту застосунку, моніторингу та контент-аналізу звернень і скарг мешканців “розумного міста”;
- *другий рівень* – це другий рівень дерева цілей, підцілей головної мети, виконання яких необхідне для подальшої реалізації системи моніторингу та контент-аналізу звернень і скарг мешканців “розумного міста”;
- *третій рівень* – надає конкретизації попереднього, другого рівня, декомпонуючи його на складові батьківської мети, необхідні для її реалізації;
- *четвертий рівень* – це вихідні цілі, вони ж критерії якісного функціонування системи.

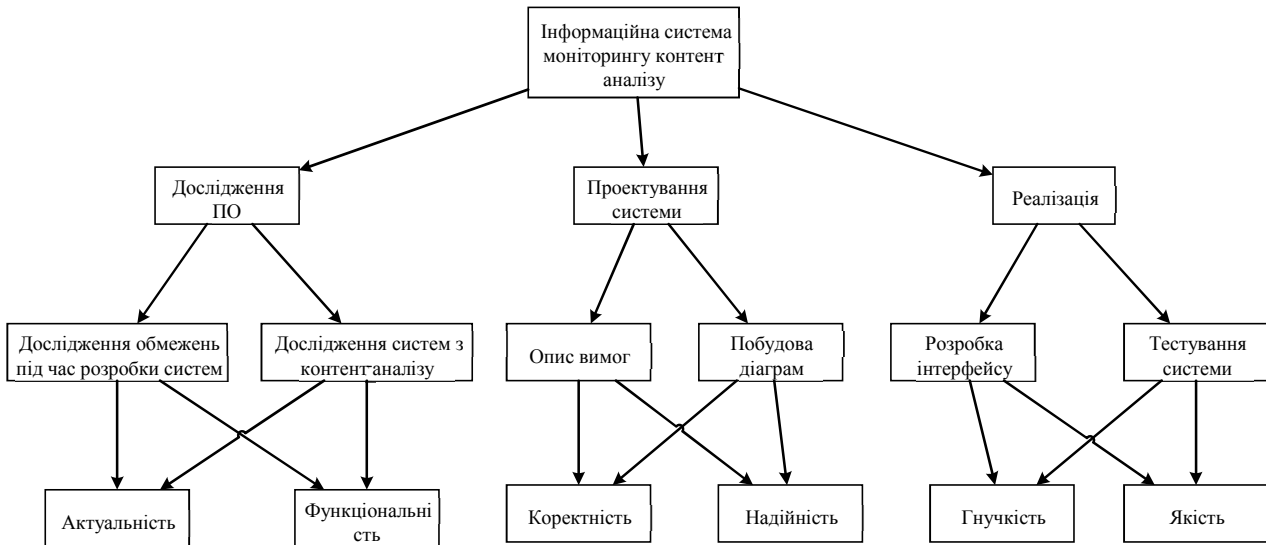


Рис. 2. Дерево цілей реалізації проекту інформаційної системи

Дослідження ПО. Цей аспект охоплює загальне дослідження наявних систем, виявлення обмежень для систем із використанням методів машинного навчання під час моніторингу та контент-аналізу звернень і скарг мешканців “розумного міста”. Також цей аспект передбачає визначення актуальності такого застосунку та спробу попередньо передбачити популярність розроблюваного продукту на ринку.

Наступним аспектом створення системи моніторингу та контент-аналізу звернень і скарг мешканців “розумного міста” є проектування самої системи. Цей етап передбачає розроблення діаграм та продумування аспектів роботи застосунку.

Останнім аспектом, зображеним на побудованому дереві цілі, є аспект під назвою “Реалізація”. Цей аспект охоплює нюанси процесу імплементації вже спроектованої системи, урахуваючи всі основні завдання, необхідні для реалізації задуму в програмній репрезентації.

Ці підаспекти визначають три основні критерії правильної роботи спроектованого застосунку: “гнучкість” – критерій, що відповідає за гнучкість налаштувань застосунку, проектування різноманітних карт проходження перешкод, “сучасність” – критерій забезпечення програмного коду та методів вирішення проблем сучасними технологіями та “якість” – критерій, відповідальний за те, щоб кінцева версія розробленого програмного продукту задовольняла вимоги та звела можливу кількість дефектів до мінімуму.



Рис. 3. Ієрархія МАІ для проекту ІС

Визначення типу системи здійснюватиметься за методом аналізу ієрархій (МАІ). Серед типів інформаційних систем розглядатимуться:

- інформаційно-довідкова (А1) – її функція полягає у збиранні та опрацюванні інформації, яку потім зберігають та надають користувачам;
- інформаційно-пошукова (А2) – її функція – у здійсненні пошуку інформації за ключовими словами чи фільтром, але пошук не враховує зміст контексту;
- інформаційно-аналітична (А3) – її функція в опрацюванні та аналізі інформації;
- інформаційна система підтримки прийняття рішень (А4) – її функція – здійснення впливу на прийняття та ухвалення рішень на підставі зібраної та проаналізованої інформації.

Перший етап методу аналізу ієрархій – побудова ієрархії. Ієрархію проєктованого застосунку наведено на рис. 3.

Наступним кроком буде розгляд матриць попарних порівнянь усіх критеріїв системи, із розрахунком власних чисел та власних векторів. Для виконання обчислень потрібно використати шкалу відносної важливості пріоритетів експертної оцінки, де: 1 – рівноцінні; 2 – несуттєвий пріоритет; 3 – слабкий пріоритет; 4 – помірний пріоритет; 5 – значний пріоритет; 6 – істотний пріоритет; 7 – сильний пріоритет; 8 – дуже сильний; 9 – безумовний пріоритет. Матриця порівнянь критеріїв відображає важливість кожного критерію порівняно із іншими (табл. 1).

Таблиця 1

Матриця порівнянь критеріїв

	К1	К2	К3	К4	К5	К6	ВЧ	ВВ
К1	1,00	3,00	0,50	0,67	2,00	0,25	0,89	0,13
К2	0,33	1,00	0,25	0,25	1,00	0,25	0,42	0,06
К3	2,00	4,00	1,00	2,00	3,00	0,50	1,70	0,24
К4	1,50	4,00	0,50	1,00	2,00	0,50	1,20	0,17
К5	0,50	1,00	0,33	0,50	1,00	0,67	0,62	0,09
К6	4,00	4,00	2,00	2,00	1,50	1,00	2,14	0,31

Таблиця 2

Матриці попарних порівнянь для критеріїв

	А1	А2	А3	А4	ВЧ	ВВ
1	2	3	4	5	6	7
Матриця попарних порівнянь для критерію 1 – актуальність						
А1	1,00	0,50	0,25	0,33	0,45	0,10
А2	2,00	1,00	0,33	0,33	0,69	0,15
А3	4,00	3,00	1,00	1,00	1,86	0,39
А4	3,00	3,00	1,00	1,00	1,73	0,37
Матриця попарних порівнянь для критерію 2 – функціональність						
А1	1,00	0,33	0,20	0,25	0,36	0,08
А2	3,00	1,00	0,25	0,33	0,71	0,15
А3	5,00	4,00	1,00	2,00	2,51	0,53
А4	4,00	3,00	0,50	1,00	1,57	0,33
Матриця попарних порівнянь для критерію 3 – коректність						
А1	1,00	1,00	0,20	0,33	0,51	0,11
А2	1,00	1,00	0,25	3,00	0,93	0,20
А3	5,00	4,00	1,00	4,00	2,99	0,63
А4	3,00	0,33	0,25	1,00	0,71	0,15

Продовження табл. 2

1	2	3	4	5	6	7
Матриця попарних порівнянь для критерію 4 – надійність						
A1	1,00	0,33	0,50	0,67	0,58	0,12
A2	3,00	1,00	0,25	0,33	0,71	0,15
A3	2,00	4,00	1,00	4,00	2,38	0,50
A4	1,50	3,00	0,25	1,00	1,03	0,22
Матриця попарних порівнянь для критерію 5 – гнучкість						
A1	1,00	2,00	0,60	0,60	0,92	0,19
A2	0,50	1,00	0,50	0,67	0,64	0,14
A3	1,67	2,00	1,00	1,33	1,45	0,31
A4	1,67	1,50	0,75	1,00	1,17	0,25
Матриця попарних порівнянь для критерію 6 – якість						
A1	1,00	0,50	0,25	0,33	0,45	0,10
A2	2,00	1,00	0,67	0,50	0,90	0,19
A3	4,00	1,50	1,00	4,00	2,21	0,47
A4	3,00	2,00	0,25	1,00	1,11	0,23

Тепер побудовано матрицю порівнянь альтернатив на основі попередніх обчислених власних чисел та власних векторів (табл. 3).

Таблиця 3

Матриця порівнянь альтернатив

КРИТЕРІЇ	K1	K2	K3	K4	K5	K6	УЗАГАЛЬНЕНІ ПРІОРИТЕТИ
A1	0,13	0,06	0,24	0,17	0,09	0,31	0,11
	0,10	0,08	0,11	0,12	0,19	0,10	
A2	0,15	0,15	0,20	0,15	0,14	0,19	0,17
A3	0,39	0,53	0,63	0,50	0,31	0,47	0,49
A4	0,37	0,33	0,15	0,22	0,25	0,23	0,23

Із отриманих результатів методу аналізу ієрархії можна виділити найбільший показник інформаційно-аналітичної системи, а саме 0,49. Отже, реалізація системи здійснюватиметься за цим типом.

Виклад основного матеріалу

Опис розгорнутого сценарію прецеденту за стандартом RUP

1. Зацікавлені особи прецеденту та їх вимоги:

- Користувач: інформаційно-аналітична система повинна швидко та зручно повідомляти про ситуацію (халатність).
- Місто: система повинна збирати інформацію про проблеми та швидко її доносити до користувача.

- Модератор: система повинна забезпечити зручну модерацію і верифікацію інцидентів.

2. Користувач інформаційно-аналітичної системи, тобто основний актор цього прецеденту:

- Це людина, яка встановила інформаційно-аналітичну систему та як свідок може надати трохи інформації для швидкого донесення проблеми до відповідних служб.

3. Передумови прецеденту (preconditions):

- Клієнт повинен зареєструватись в інформаційно-аналітичній системі.

- Клієнт повинен успішно пройти процедуру автентифікації в інформаційно-аналітичній системі.

4. Основний успішний сценарій:

- Клієнт завантажує інформаційно-аналітичну систему на свій пристрій.
- Клієнт вводить свої персональні дані (ПІБ, дату народження тощо).
- Користувач успішно заходить в інформаційно-аналітичну систему й отримує можливість бути проінформованим, якщо в його районі виникне загрозна ситуація.
- Інформаційно-аналітична система раз на тиждень надаватиме згруповану статистику щодо інцидентів.

5. Розширення основного сценарію або альтернативні потоки:

- Клієнт не може сам зареєструватись.
- 1. Модератор верифікує клієнта і тим збільшує його можливості.
- 2. Клієнт підтверджує опрацювання персональних даних.
- 3. Клієнт під час постингу ситуації надає доступ до GPS.
- Клієнт користується застосунком із телефону.
- 1. Інформаційно-аналітична система пропонує користувачеві надати доступ до модулів пристрою.

2. Користувач отримує доступ до додаткових функцій програми.

6. Постумови (postconditions):

- Збереження всіх випадків у базі даних інформаційно-аналітичної системи.

7. Спеціальні СВ:

- Знаки та застереження системи повинні бути простими та зрозумілими.
- Необхідно забезпечити 100 % конфіденційність даних користувача.
- Потрібно забезпечити можливість зв'язку інформаційно-аналітичної системи з іншими приладами "розумного міста".

8. Список необхідних технологій та додаткових пристроїв:

- Користувач повинен володіти персональним комп'ютером або смартфоном.
- Пристрій повинен підтримувати технологію GPS.

Зовнішні сутності: модератор, користувач, програмне забезпечення. Процеси:

1. Верифікація.
2. Додавання ситуації у базу даних.
3. Аналіз списку.
4. Введення інформації в базу даних.
5. Введення персональних даних користувача.
6. Збирання інформації про користувачів.
7. Збирання даних GPS.
8. Заповнення даних інциденту на мапу.
9. Ресстрація.
10. Розширення можливості застосунку.
11. Надсилання даних про проблемні випадки.
12. Перевірка ситуації.
13. Передавання інформації у відповідні служби.
14. Отримання відомостей про список проблем.
15. Створення спеціальних можливостей.

На основі наведених процесів побудовано діаграму потоків даних (рис. 4). Детальний опис наведених процесів подано в табл. 4.

Опис процесів

Верифікація	Розширення можливостей програми	Процес 1
Створення спеціальних можливостей	Розширення можливостей програми	Процес 2
Надсилання даних про проблемний випадок	Запощення даних інциденту на мапу	Процес 3
Перевірка ситуації	Запощення даних інциденту на мапу	Процес 4
Перевірка ситуації	Додавання ситуації в базу даних	Процес 5
Передавання інформації у відповідні служби	Отримання даних про список проблем	Процес 6
Аналіз списку	Отримання даних про список проблем	Процес 7
Надсилання даних про проблемний випадок	Збирання даних GPS	Процес 8
Збирання даних GPS	Збирання інформації про користувачів	Процес 9
Внесення персональних даних користувача	Внесення інформації в базу даних	Процес 10

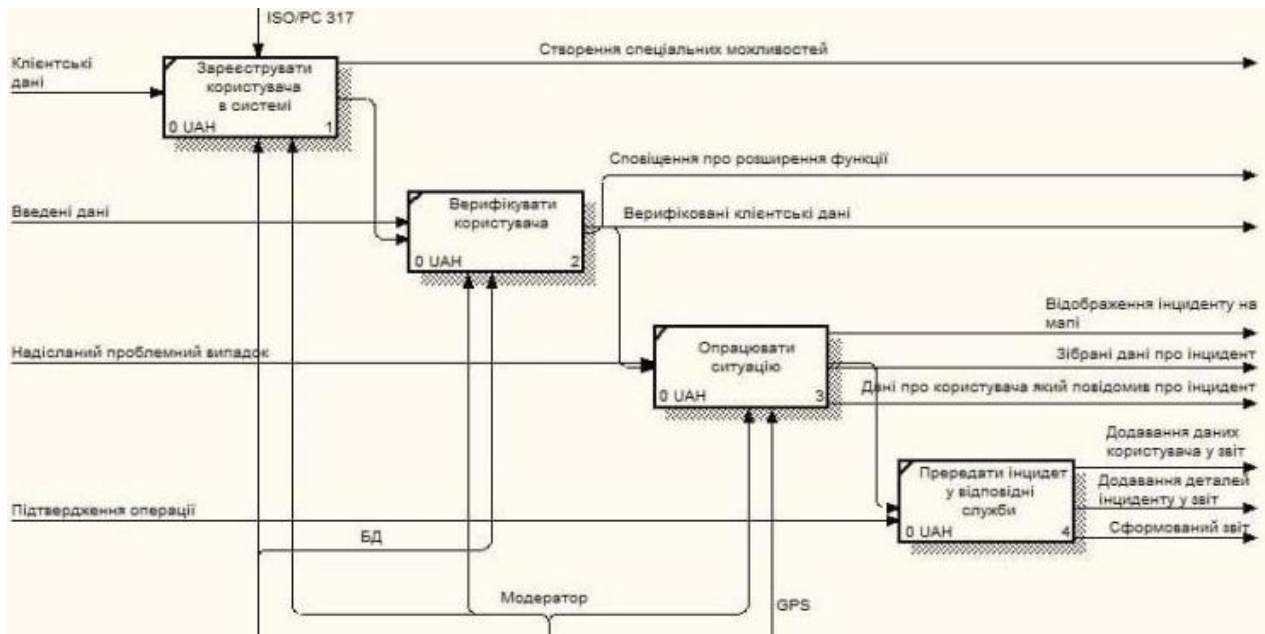


Рис. 4. Діаграма потоків даних

Сховище даних призначене для відображення статичної структури моделі системи.

Опис сховища даних

Клас	Стереотип	Атрибути	Операції	Короткий опис
1	2	3	4	5
Користувач	entity	Ім'я; Прізвище; По батькові; Дата народження	Зареєструватися; Авторизуватися; Ввести дані; Надіслати додаткову верифікацію	Користувач, який вводить свої персональні дані, а пізніше на їх основі реєструється і авторизується
Контролер вводу	control	Логін; Пароль	Перевірити логін та пароль; Перевірити правильність даних	Контролер, необхідний для перевірки даних входу та перевірки правильності реєстраційних даних

Продовження табл. 5

1	2	3	4	5
База даних	domain	–	Збереження даних; Редагування та оновлення даних; Видалення даних; Підключення до БД; Зберегти звіт про ситуації	Сутність, необхідна для збереження та аналізу всіх інцидентів
Модератор	entity	Ім'я; Прізвище; По батькові; Дата народження	Авторизується; Модерування поста та мапи подій; Верифікування	Сутність являє собою персональні дані модератора
Інтерактивна мапа	entity	Сигнал GPS; Список інцидентів	Редагування	Зберігається інформація про місцезнаходження
Пост	entity	Інтерактивна мапа; Користувач; Оцінка; Дата; Коментар модератора	Видалення; Редагування	Групується вся статистика про ситуацію, за необхідності з коментарем модератора
Звіт про ситуації	entity	Користувач, що повідомив; Деталі ситуації; Пост	Видалення	Звіт про ситуації містить узагальнену статистику для подальшого аналізу

Під час проектування системи дуже важливо розуміти важливість завдань та послідовність їх виконання. Саме для цього варто використовувати ієрархію задач. Тут всі завдання поділено на підзадачі й завдяки візуалізації їх можна краще зрозуміти. Ієрархічний аналіз задач використовують віддавна – з 1960-х років або навіть раніше. Він найдоцільніший для аналізу завдань із чітко визначеною структурою, тобто завдань, для яких характерна тенденція подібного виконання кожен раз, а не тих, які мають дуже вільну структуру. Ієрархія передбачає описання завдання у термінах ієрархії завдання – підзадача та набору планів, що визначають, у якій послідовності підзавдання можуть виконуватися або за яких обставин певні підзадачі взагалі виконуються. На діаграмі добре видно, які завдання є головними, бо розміщені на вершині, а всі завдання, розташовані нижче, обов'язково мають бути виконані для того, щоб досягнути головної цілі. Їх можна виконати у довільній послідовності, але обов'язково усі вони мають бути реалізовані. В іншому випадку головна мета не буде досягнута. Діаграма ієрархії задач у AllFusion Process Modeler створена для відображення усіх завдань, що розроблялися протягом роботи над діаграмами декомпозиції, на одному аркуші. Так простіше побачити їх усіх, не замилюючи очі даними, потоками і сховищами. Відповідно для створення ієрархії потрібно обов'язково вибрати головний процес, який виконує застосунок, декомпонувати його та зобразити деревом вузлів. Застосунок за замовчуванням зображає головні процеси у вигляді прямокутників, а процеси нижнього рівня подаються лише списком завдань під головними процесами. У AllFusion Process Modeler є можливість вибирати стиль зображення процесів: він може бути або діагональним, або ортогональним. Це забезпечує зображення всіх процесів на одній діаграмі. Для того, щоб зобразити інформаційну систему, вибрали побудову діаграму ієрархії задач у вигляді дерева вузлів (рис. 5). Головну задачу та підзадачі першого рівня декомпозиції зображено у вигляді прямокутників із заокругленими кутами. А всі підзадачі нижчого рівня зображено у вигляді списків під блоками головних задач. Основна задача – “Здійснити збір скарг громадян міста (розумного міста)”. Її поділено на чотири

підзадачі: зареєструвати користувача в системі; верифікувати користувача; опрацювати ситуацію; передати інцидент у відповідні служби.



Рис. 5. Ієрархія задач проекту

Інтерактивне картографування передбачає використання карт, що дають змогу збільшувати та зменшувати масштаб, панорамувати, ідентифікувати конкретні особливості, запитувати базові дані, за темою чи певним показником (наприклад, соціально-економічний статус), створювати звіти та інші способи використання або візуалізації вибраної інформації на карті. Інтерактивне картографування використовує Глобальну інформаційну систему (ГІС) для показу точних даних на карті. Працюючи в системі шарів, різні рівні географічної інформації розміщують один на одному [13–15]. На відміну від статичних карт, інтерактивні карти мають переваги – низку функцій, призначених для кращого відображення великої кількості складних даних [16–21].

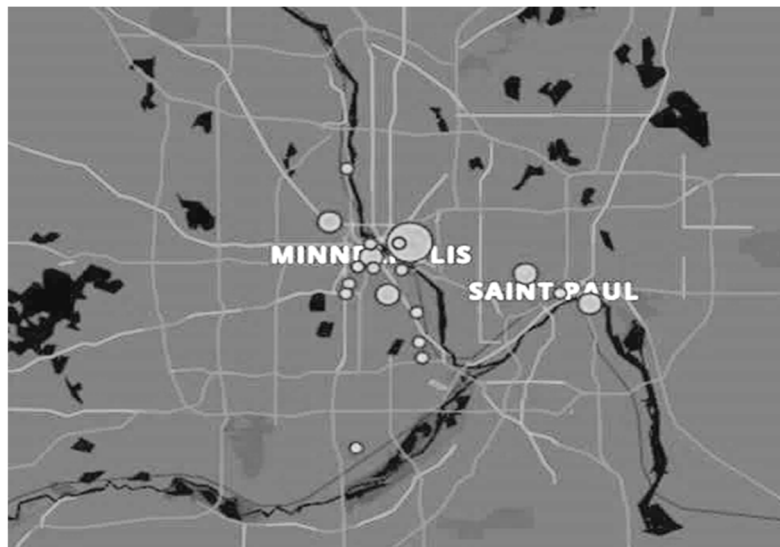


Рис. 6. Приклад проекту застосунку інтерактивного мапування

Червоні точки можуть легко підказати, де відбувається та чи інша ситуація, а за розміром точки можна зрозуміти масштабність або екстремність події.

Програмою для виконання цих дій вибрано безплатну програму з відкритим кодом для запису відео і потокового мовлення – SQL MMSS. Сховище даних (СД) – це складна комп'ютерна система. Як правило, архітектуру СД утворюють такі компоненти:

- засоби отримання даних із різних БД OLTP-систем, успадкованих систем та інших зовнішніх джерел даних;

- засоби перетворення і очищення даних: перед тим, як помістити дані в сховище, їх необхідно очистити;
- програмне забезпечення БД: як правило, це високопродуктивна СУБД;
- засоби для з'єднання джерел даних зі сховищем, а також клієнтів із сервером.

Крім цього, необхідні спеціальні програмні засоби проектування сховища, засоби роботи з репозиторієм метаданих і засоби оперативної аналітики (OLAP-засоби). Усе це – складне спеціальне програмне забезпечення, вартість якого обчислюється десятками і сотнями тисяч доларів. На вибір архітектури СД і методи його проектування впливають вид і масштаб задач аналізу даних в організації. З одного боку, СД створюють для вирішення конкретних, строго визначених завдань аналізу і відтворення нових даних, з іншого боку – СД має забезпечувати корпоративну звітність в межах всієї організації. Отже, визначальним моментом у побудові СД є задачі опрацювання і аналізу даних, побудови та доставки звітів.

Бажано вибирати архітектуру СД до початку його реалізації, однак на практиці не завжди дотримуються цього правила. На цей вибір впливають кілька різних чинників: інфраструктура організації, виробниче та інформаційне середовище організації, управління і контроль, масштаби проекту, можливості апаратно-технологічного забезпечення, готовність персоналу та наявні ресурси. На прийняття рішень щодо вибору способу реалізації впливає кілька факторів: час, відведений на проєкт; повернення інвестицій; швидкість упровадження СД; потреби користувачів; потенційні ризики під час розроблення проекту; вимоги до ресурсів, необхідних у певний момент часу; вибрана архітектура СД; сукупна вартість СД.

У деяких рішеннях певні компоненти схеми в архітектурі СД можуть бути відсутні.

Розглянемо компоненти типової архітектури сховища даних.

- Програмне забезпечення проміжного шару, основне призначення якого полягає у забезпеченні доступу до мережі й доступу до даних. До цього ПЗ належать мережеві комунікаційні протоколи, драйвери, системи обміну повідомленнями тощо. Підтримку такого ПЗ зазвичай забезпечують ІТ служби організації.

- Бази даних систем оперативного опрацювання даних (OLTP) і дані зовнішніх джерел. Для OLTP-систем характерна цільова спрямованість на ефективне опрацювання структур даних порівняно невеликої кількості чітко визначених типів транзакцій. Спрямованість на швидке виконання транзакцій робить такі системи малопридатними для вирішення аналітичних завдань. Транзакції для побудови аналітичних вибірок за природою відрізняються від транзакцій OLTP-систем. І виконання таких вибірок в OLTP-системах призводить до зниження продуктивності.

- Попереднє опрацювання та завантаження даних. Попереднє опрацювання, пов'язане з фільтрацією, очищенням і перетворенням даних з OLTP-систем та зовнішніх джерел, як правило, виконується в деякому проміжному файлі, який називається завантажувальною секцією. Після такого опрацювання дані завантажуються в СД.

- Сховище даних є ядром системи бізнес-аналітики. Це можуть бути один або декілька серверів БД, які підтримують СД.

- Метадані є репозиторієм, який виконує роль довідника про дані. Він містить термінологію предметної області, відомості про джерела даних, опис джерел вихідних даних, відомості про алгоритми опрацювання вихідних даних.

- Рівень доступу до даних охоплює ПЗ, яке забезпечує взаємодію кінцевих користувачів із даними СД. Нині універсальними засобами спілкування є SQL і його розширення.

- Рівень інформаційного доступу. Такими засобами можуть слугувати стандартні пакети MS Office, Lotus Notes або спеціальні програмні продукти.

- Рівень адміністрування, компоненти якого відстежують виконання процедур оновлення даних у сховищі; до них належать процедури підзавантаження даних, індексування, підсумовування та агрегації даних, реплікацію даних у розподіленому обчислювальному середовищі, авторизацію користувача і розмежування доступу.

Глобальне СД проєктують і конструюють на основі потреб аналітичної інформаційної підтримки організації загалом. Його можна розглядати як загальний репозиторій для даних, які забезпечують прийняття рішень. Глобальне СД не обов'язково має бути реалізоване фізично як централізоване. Термін "глобальне" використовують для відображення масштабу використання і доступу до даних у межах всієї організації. Глобальне СД може бути фізично як централізованим, так і розподіленим. Наприклад, окремі частини розподіленого СД можуть управлятися в межах підрозділів або напрямів бізнесу. Управління СД визначає, хто вирішує:

- які дані повинні надходити в СД;
- коли дані повинні надходити в СД;
- коли дані повинні оновлюватися;
- кому дозволено доступ до даних в СД.

Отже, для глобального СД існують два основних архітектурних рішення. Перевагою глобального СД є надання кінцевим користувачам доступу до інформації в масштабах підприємства, недоліком – високі витрати на реалізацію, зокрема витрати часу на створення СД. Незалежні вітрини даних містять автономні або незалежні вітрини даних (Stand-alone Data Marts), які управляються робочими групами, відділами або напрямками бізнесу і розробляються виключно для реалізації аналітичних потреб останніх. Цілком можливо, що немає жодного зв'язку між ними. Наприклад, дані для таких вітрин можуть генеруватися безпосередньо в самих підрозділах організації. Дані можуть вилучатись із OLTP-систем, зокрема, за допомогою ІТ-служб організації. ІТ-служби можуть підтримувати обчислювальне середовище для вітрин даних, але не управляти інформацією в них. Дані у вітрини можуть надходити і з глобального СД. Для організації незалежних вітрин даних потрібні деякі професійні та технічні навички.

Як правило, для їх створення виділяють ресурси і персонал у межах того підрозділу, для якого їх створюють. Такий тип реалізації СД мінімально впливає на інформаційні ресурси організації і може бути виконаний дуже швидко.

Водночас максимальна незалежність і мінімальна інтеграція, а також відсутність глобального уявлення про дані організації можуть стати обмеженням такої архітектури. Вітрини даних можуть бути взаємозалежні або взаємопов'язані (так звані пов'язані вітрини даних). Така архітектура СД охоплює сукупність вітрин даних, які управляються робочими групами, відділами або напрямками бізнесу, але розробляються в межах єдиної для організації схеми задоволення аналітичних потреб. Для взаємопов'язаних кіосків даних типовою є розподілена архітектура реалізації. Незважаючи на те, що окремі вітрини даних реалізуються в межах робочих груп, підрозділів і напрямів бізнесу, вони можуть бути інтегровані, тобто взаємопов'язані для того, щоб забезпечити представлення даних у межах організації загалом. Фактично, на найвищому рівні інтеграції вони можуть стати глобальним СД.

У такій архітектурі користувачі одних підрозділів можуть отримувати доступ до даних інших підрозділів у межах своїх повноважень. Вимоги інтеграції даних у межах архітектури взаємопов'язаних вітрин даних роблять реалізацію СД складнішою, порівняно з незалежними вітринами даних. Наприклад, необхідно вирішити питання, хто керуватиме даними у вітринах і хто підтримуватиме середовище. Важливе питання про те, що робити з даними, які є загальними для кількох вітрин даних, а також як розробити схему розмежування доступу користувачів до вітрин даних у межах всієї організації. Реалізація такої архітектури не ставить високих вимог до програмно-апаратного забезпечення, і вартість її може бути невеликою. Однак час реалізації буде більшим порівняно із незалежними вітринами даних. Зростають також складність і вартість процедур

проектування. Також можна створювати так звані віртуальні СД, які працюють над OLTP системами, СД з багаторівневою архітектурою і вбудовані СД, які імплементують в наявну систему опрацювання даних організації. Підходи до реалізації сховища даних для систем бізнес-аналітики такі самі, як і для реалізації будь-яких типів інформаційних систем із базами даних. До СД застосовують такі основні методологічні підходи:

- “зверху вниз” (Top down design);
- “знизу вгору” (Bottom down design);
- “ізсередини” (Middle of design).

Вибір методологічного підходу до реалізації СД впливає на обсяг і ретельність проектування. Підхід “зверху вниз” потребує детального планування і проектування СД в межах ІТ-проєкту до початку виконання проєкту. Це пов'язано з тим, що необхідно залучати всіх потенційних користувачів СД для з'ясування їх інформаційних потреб в аналітичному опрацюванні даних, приймати рішення про джерела даних, безпеку, структуру даних, стандарти даних. Всі ці роботи повинні бути задокументовані й узгоджені. За такого підходу модель СД має бути розроблена до початку реалізації. Зазвичай такий підхід практикують під час створення глобального СД. Якщо вітрини даних входять у конфігурацію, їх можна побудувати пізніше. Перевагою такого підходу є отримання узгодженіших визначень даних і бізнес-правил організації на початку роботи над створенням системи бізнес-аналітики. Вартість початкового планування і проектування може виявитися досить високою. Для цього підходу характерні великі витрати часу, що відкладає початок реалізації й затримує повернення інвестицій. Підхід “зверху вниз” добре застосовувати в організаціях з чітко організованою інформаційно-обчислювальною структурою, коли програмно-апаратна платформа визначена і злагоджено працюють джерела даних. Використовуючи підхід “знизу вгору”, починають із планування і проектування вітрин даних підрозділів без попереднього розроблення глобальної інформаційно-обчислювальної інфраструктури організації. Це не означає, що така глобальна інфраструктура не буде розроблена пізніше. Такий підхід є прийнятнішим у багатьох випадках, оскільки швидше приводить до кінцевих результатів. У нього є і недоліки: дані можуть дублюватися і бути неузгодженими в різних вітринах даних. Щоб уникнути цього, необхідно ретельно планувати та проектувати. Підходи “знизу вгору” і “зверху вниз” можна комбінувати залежно від поставлених перед керівником проєкту зі створення СД цілей. Підхід “проектування ізсередини” є комбінацією перерахованих вище підходів, які застосовують по спіралі.

Спочатку створюють ядро системи (підхід “згори вниз”), а потім його поетапно нарощують за рахунок додавання нової або додаткової функціональності (підхід “знизу вгору”). Отже, на кожному витку спіралі можна використовувати кожен із двох зазначених вище підходів. Існують й інші комбінації. Вибір підходу до реалізації СД поряд з вибором архітектури СД визначає тактичні рішення у проектуванні та управлінні проєктом щодо інженерії систем бізнес-аналітики.

База даних інформаційної системи моніторингу та контент-аналізу звернень мешканців “розумного міста” на основі методів машинного навчання, NLP та геопозиції дуже велика за розміром, адже повинна забезпечувати злагоджену роботу усіх компонентів та блоків системи, вміщувати інформацію про користувачів, скарги у відповідні служби, а також запам'ятовувати усі процеси, які здійснюють менеджери у системі. Тип таблиць бази даних – InnoDB. Базу даних виконано за допомогою SQL команд. Середовище зберігання бази даних – PHP MyAdmin.

Для забезпечення оптимальної роботи ресурсу на Zend Framework та для побудови власної системи управління контентом створюємо 84 таблиці, які відповідають за розміщення інформації про компоненти ресурсу, про скарги, користувачів, запити, статті, фото тощо.

Діяльність інформаційної системи моніторингу та контент-аналізу звернень мешканців “розумного міста” на основі методів машинного навчання, NLP та геопозиції можна поділити умовно

на три напрями – аналіз скарг користувача, надання інформаційного забезпечення стосовно важливих питань і створення власного кабінету для представників різних служб, здійснення різноманітних бізнес-процесів, завантаження скарги, опис додавання фото тощо. Для створення системи використовуємо модифікований варіант Zend Framework. Із нього залучаємо модуль бази даних, кеш, модуль повідомлень, реєстрації клієнтів та модуль перекладу програмного забезпечення.

Система розрахована на роботу із великим обсягом інформації, тому розроблений фреймворк повинен працювати швидко та якісно. У дереві файлів на рис. 7 бачимо папки Core, Global та Local. У папці Core міститься ядро фреймворка – основні модулі. Папка Global також містить модулі, може містити додаткові модулі або переписувати модулі із папки Core. Вона відрізняється від папки Local тим, що може використовуватися на різних сайтах і у різних системах. Тобто вона не є унікальною. Папка Local містить ті файли, що актуальні лише для певного ресурсу. Унікальні стилі, JS файли та зображення. На рис. 7 зображений вміст папки Local – це папки css, images, javascript, layouts, library, modules, single_page. У css містяться стилі усього ресурсу, відповідно у javascript скрипти, у layouts – розмітка тих сторінок, які є шаблонними для системи і можуть дублюватися під час створення нових розділів (це, для прикладу, сторінки із описанням скарги, вулицею, темою). У library містяться файли бібліотеки, у modules – модулі та у single_page – розмітки тих сторінок, що на сайті, використані лише один раз. За роботу власного кабінету певних служб відповідають модуль DS, який призначений для здійснення бізнес-процесів у системі, та кілька сторінок із папки single_page (рис. 7). Коли менеджер заходить у власний кабінет, система запускає файл із папки views/index.php (рис. 8). І уже до цього файла підключаються інші підмодулі, такі як скарга, вирішення проблеми. Вся система чітко структурована відповідно до концепту MVC. Серверна частина ресурсу написана мовою PHP. В інформаційній системі передбачений користувач із особливими правами – Admin. Для нього у кожному модулі створюємо окремі права. Користувач із правами адміністратора автоматично отримує доступ до даних усіх зареєстрованих користувачів, може додавати, редагувати та видаляти інформацію на головній сторінці та другорядних сторінках сайта. Користувачі з іншими правами мають доступ лише до власного кабінету, який підтримується модулем DS, та можуть перейти до нього, додавши в адресному рядку /ds та ввівши свій логін і пароль. У системі передбачено, що з кабінетом однієї служби може працювати декілька користувачів. Ця функція реалізується модулем Agenda та синхронізована за допомогою Ajax із іншими компонентами. Тому під час додавання, редагування чи видалення елемента зі щоденника сторінка не перезавантажується.

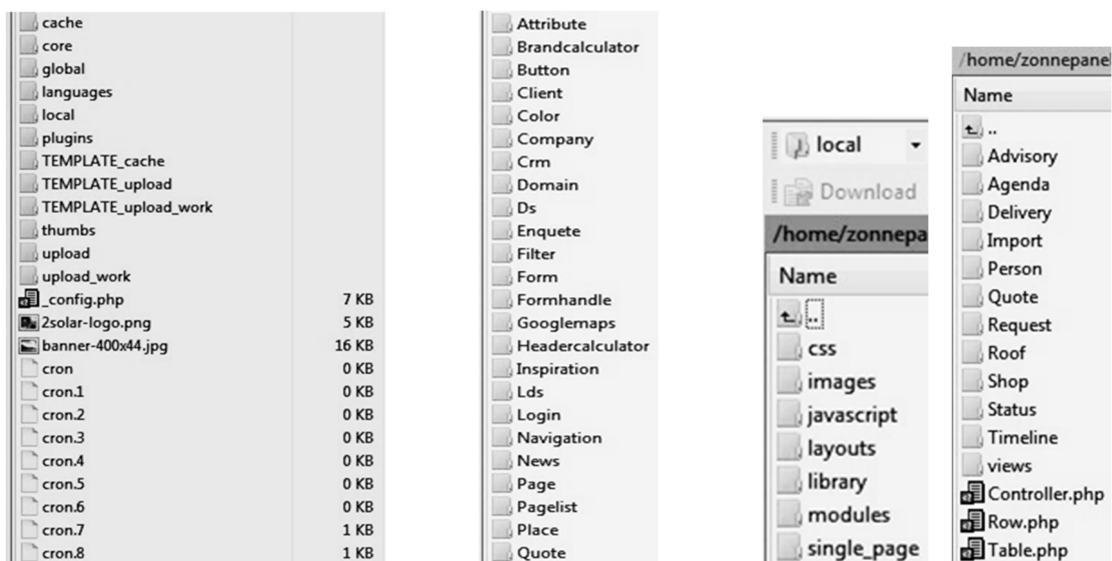


Рис. 7. Список модулів, вміст директорій Modules, Local та DS

```

<script type="text/javascript" src="<?=$this->javascript('my_first_tour.js') ?>"></script>
<div id="header" class="navbar" <?=$inverse ?>>
  <div class="navbar-inner"
  <div class="container"
  <ul class="nav"
  <? if ( $SESSION['core']['user']['group']->status == 'special' ) : ?>
  <? foreach ( $activeViews as $clientViewId => $clientViewName ) : ?>
  <? <li><a class="clientViewLink" href="/_ma/g/ds_request/a/listview/view/<?=$clientViewId ?>"><?=$clientViewName ?> <span class="label label-info" data-
  <? endforeach ?>
  <? else : ?>
  <? if ( $clientConfig->send_request_to_other_clients == 'yes' && $clientConfig->agenda_access == 'yes' ) : ?>
  <? <!-- <li><a id="agenda" style="cursor: pointer" ><?=$clientConfig->agenda_access ?> </a></li>> -->
  <? <li><a class="clientViewLink" data-agenda-link="1" href="/_ma/g/ds_agenda/a/index"><?=$clientConfig->agenda_access ?> </a></li>
  <? endif ?>
  <? if ( $clientConfig->dashboard_access == 'yes' && $clientRole->dashboard_access == 'yes' ) : ?>
  <? <li><a class="dsAdminLink" href="/_ma/g/ds_request/a/listview/view/clients" data-target="admin-listview"><?=$clientRole->dashboard_access ?> </a></li>
  <? endif ?>
  <? if ( $clientConfig->send_request_to_other_clients == 'yes' && $SESSION['core']['user']['client']->client_id == 6 ) : ?>
  <? <li id="header2"><a class="dsAdminLink" href="/_ma/g/ds_request/a/listview/view/forwarded" data-target="admin-listview"><?=$SESSION['core']['user']['client']->client_id == 6 ?> </a></li>
  <? <li><a class="dsAdminLink" href="/_ma/g/ds_request/a/listview/view/order" data-target="admin-listview"><?=$SESSION['core']['user']['client']->client_id == 6 ?> </a></li>
  <? <li><a class="dsAdminLink" href="/_ma/g/ds_request/a/listview/view/billed" data-target="admin-listview"><?=$SESSION['core']['user']['client']->client_id == 6 ?> </a></li>
  <? endif ?>
  <? if ( $clientConfig->shop_access == 'yes' ) : ?>
  <? <li><a class="clientViewLink" href="/_ma/g/ds_shop/a/listview"><?=$clientConfig->shop_access ?> <span class="label label-info" data-content="/_ma/g/ds_shop/a/count
  <? endif ?>
  <? if ( $clientConfig->shop_personal == 'yes' ) : ?>
  <? <li><a class="clientViewLink" href="/_ma/g/ds_shop/a/listviewpersonal"><?=$clientConfig->shop_personal ?> <span class="label label-info" data-content="/_ma/g/ds_s
  <? endif ?>

```

Рис. 8. Підключення підмодулів у модулі DS

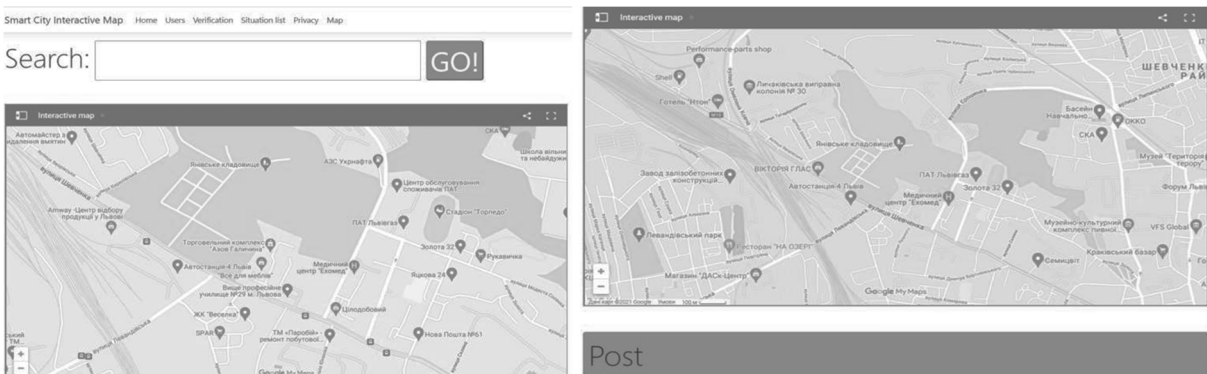


Рис. 9. Початок трансляції та вікно програми

Розроблена програма дає можливість переглядати мапу, а також через різні вкладки керувати базою даних на комп’ютері. Перед тим як запустити програму, потрібно увімкнути сервер SQL для запуску бази даних. В такому випадку для цього буде використано MS SQL, а на сайті з посиланням “https://localhost:44305/” можна переглянути сам сайт. Також під самою картою є кнопка Post і в майбутньому це буде використовуватись для збереження подій.

На цій вкладці можна виконувати різні дії з редагування даних.

Smart City Interactive Map Home Users Verification Situation list Privacy Map

Index Page

Create new customer

Search by name: Search [Back to Full list](#)

FirstName	LastName	Address	Discount	
Kelly	Buttler	Independents 19	20	Edit Details Delete
Tom	Taylor	Independents 11	10	Edit Details Delete
Smite	Butter	Independents 23	15	Edit Details Delete
Yulian	Yulian	Lviv	77	Edit Details Delete

Рис. 10. Доступ до бази даних на сайті

Рядок пошуку користувачів за ім'ям.

Рис. 11. Рядок пошуку

На цьому місці сторінки можна побачити кнопки для редагування даних у БД.

Рис. 12. Кнопки для редагування даних

Рис. 13. Можливість редагування полів

FirstName	LastName	Address	Discount	
Kelly	Buttler	Independents 19	20	Edit Details Delete

Рис. 14. Виконання пошуку

Для виконання пошуку значення поля у рядку пошуку порівнюють із даними в базі даних. На рис. 14 відображено пошук за ім'ям.

Висновки

У статті проаналізовано особливості розроблення інформаційної системи моніторингу та контент-аналізу звернень та скарг мешканців “розумного міста”. Мешканцям міста надається можливість повідомляти про різні екстрені ситуації певні служби, які вживатимуть заходів для швидкого вирішення цієї проблеми. Інформаційна система призначена для мешканців міста та комунальних служб. Схожих систем з таким самим функціоналом доволі мало, оскільки розроблення складне і залежить від вибраного алгоритму. Проте інформаційні системи, яким притаманний цей функціонал, надзвичайно успішні. Цей програмний продукт може стати популярним у користувачів, адже дає змогу швидко та просто налагодити інформування мешканців про небезпечні інциденти. До переваг програми, що роблять її конкурентоспроможною на нашому ринку, можна зарахувати локалізований українською мовою зручний та інтуїтивно зрозумілий інтерфейс і швидкий алгоритм пошуку. Для поширення необхідно вкласти кошти у маркетинг, задля просування програми в умовах конкуренції на ринку. Але в результаті інформаційна система швидко окупить як ресурси, витрачені на її створення, так і кошти, вкладені у рекламу.

Розумні технології допоможуть містам підтримувати зростання і підвищувати ефективність, щоб забезпечити добробут громадян і ефективність управління у містах в найближчі роки.

Запропонований у цій роботі прототип інформаційної системи моніторингу та контент-аналізу звернень і скарг мешканців “розумного міста” створено на основі методів машинного навчання, NLP та геолокації. Проаналізовано наукові публікації, релевантні темі дослідження, що дало змогу визначити засоби вирішення проблеми, функціональність інформаційної системи та структуру процесу прийняття рішення. Здійснено інформаційне моделювання інформаційної системи: побудовано дерево цілей, яких необхідно досягти, реалізуючи систему, створено контекстну діаграму та декомпоновано її за допомогою методології Data Flow Diagram у нотації Гайна-Сарсона. Крім того, встановлено ієрархію задач за допомогою дерева вузлів та з використанням методу аналізу ієрархій (MAI) визначено тип інформаційної системи як інформаційно-аналітичної, оскільки саме його коефіцієнт найбільший – 0,49. Детально проаналізовано засоби, використані для виконання завдань інформаційної системи, та обґрунтовано причини їх вибору. Проаналізовано можливості Google Maps та використання програмних засобів для розробки, а саме MS SQL. Розроблено підходи до реалізації інформаційно-аналітичної системи та описано аналіз контрольного прикладу її роботи, її функціональні можливості. Описано систему зберігання інформації у системі.

Список літератури

1. This is what the cities of the future could look like. URL: <https://www.weforum.org/agenda/2019/01/the-anatomy-of-a-smart-city/>
2. Smart Cities: Where the IoT and Ethics Converge. URL: <https://www.spiceworks.com/tech/iot/guest-article/smart-cities-where-the-iot-and-ethics-converge/>
3. Investments in Technologies Enabling Smart Cities Initiatives Is Forecast to Reach \$80 Billion in 2018, According to a New IDC Spending Guide. URL: <https://www.businesswire.com/news/home/20180220005118/en/Investments-in-Technologies-Enabling-Smart-Cities-Initiatives-Is-Forecast-to-Reach-80-Billion-in-2018-According-to-a-New-IDC-Spending-Guide>
4. IDC Forecasts Worldwide Spending on the Internet of Things to Reach \$772 Billion in 2018. URL: <https://www.businesswire.com/news/home/20171207005963/en/IDC-Forecasts-Worldwide-Spending-on-the-Internet-of-Things-to-Reach-772-Billion-in-2018>
5. Інформація про звернення. URL: <https://ukc.gov.ua/wp-content/uploads/2021/02/Za-rik.pdf>
6. Урядовий контактний центр. URL: <https://ukc.gov.ua/>
7. Державне управління в smart city. URL: <https://knote.edu.ua/file/NjY4NQ==/d649f05f3e9fa1e8356f3af1541ae90e.pdf>
8. Урядова “гаряча лінія” 1545. URL: <https://www.kmu.gov.ua/uryadova-garyacha-liniya-1545>
9. Citizen. About this app. Royal Media Services. URL: https://play.google.com/store/apps/details?id=newsapp.citizentv.royalmedia.citizentvnews&hl=en_US&pli=1
10. What Is Citizen? the Safety and Crime-Tracking App, Explained. URL: <https://www.businessinsider.com/what-is-citizen-app>
11. Citizen – Connect and stay safe. URL: <https://citizen.com/>
12. Dirks, S., & Keeling, M. (2009). A vision of smarter cities: How cities can lead the way into a prosperous and sustainable future. IBM Institute for business Value, 8. URL: https://web.archive.org/web/20180429121219id_/http://www-935.ibm.com/services/us/gbs/bus/pdf/ibm_podcast_smarter_cities.pdf
13. European Commission (2009). Investing in the Low Carbon Technologies. SETPlan. URL: http://ec.europa.eu/eip/smartcities/index_en.htm
14. Bentivegna, V., Curwell, S., Deakin, M., Lombardi, P., Mitchell, G., & Nijkamp, P. (2002). A vision and methodology for integrated sustainable urban development: BEQUEST. Building Research & Information, 30(2), 83–94. DOI: 10.1080/096132102753436468.
15. Against the Smart City. URL: <https://urbanomnibus.net/2013/10/against-the-smart-city/>
16. European Smart Cities. URL: <https://www.smart-cities.eu/>
17. FTTH Council Europe (2018). Smart City Guide. URL: www.ftthcouncil.eu
18. Matseliukh Y., Bublyk M., Vysotska V. (2021). Development of Intelligent System for Visual Passenger Flows Simulation of Public Transport in Smart City Based on Neural Network. In COLINS, pp. 1087–1138.
19. Al-Khoury A. M. (2012). Emerging markets and digital economy: Building trust in the virtual world. International Journal of Innovation in the Digital Economy (IJIDE), 3(2), pp. 57–69. DOI: 10.4018/jide.2012040105.

20. Krislata I., Katrenko A., Lytvyn V., Vysotska V., Burov Y. (2020). Traffic Flows System Development for Smart City. In ITPM, pp. 280–294.
21. Bublyk M., Zahreva Y., Vysotska V., Matseliukh Y., Chyrun L., Korolenko O. (2022). Information System Development For Recording Offenses In Smart City Based On Cloud Technologies And Social Networks. *Webology* (ISSN: 1735-188X), 19(2).
22. Bublyk M., Kalynii T., Varava L., Vysotska V., Chyrun L., Matseliukh Y. (2022). Decision Support System Design For Low-Voice Emergency Medical Calls At Smart City Based On Chatbot Management In Social Networks. *Webology* (ISSN: 1735-188X), 19(2).
23. Peleshko D., Rak T., Noennig J. R., Lytvyn V., Vysotska V. (2020). Drone Monitoring System DROMOS of Urban Environmental Dynamics. In ITPM, pp. 178–193.
24. Podlesna L., et al. (2020). Optimization Model of the Buses Number on the Route Based on Queueing Theory in a Smart City. In *MoMLet+ DS*, pp. 502–515.
25. Bublyk M., Matseliukh Y., Motorniuk U., Terebukh M. (2020). Intelligent System of Passenger Transportation by Autopiloted Electric Buses in Smart City. In *COLINS*, pp. 1280–1294.

References

1. This is what the cities of the future could look like. URL: <https://www.weforum.org/agenda/2019/01/the-anatomy-of-a-smart-city/>
2. Smart Cities: Where the IoT and Ethics Converge. URL: <https://www.spiceworks.com/tech/iot/guest-article/smart-cities-where-the-iot-and-ethics-converge/>
3. Investments in Technologies Enabling Smart Cities Initiatives Is Forecast to Reach \$80 Billion in 2018, According to a New IDC Spending Guide. URL: <https://www.businesswire.com/news/home/20180220005118/en/Investments-in-Technologies-Enabling-Smart-Cities-Initiatives-Is-Forecast-to-Reach-80-Billion-in-2018-According-to-a-New-IDC-Spending-Guide>
4. IDC Forecasts Worldwide Spending on the Internet of Things to Reach \$772 Billion in 2018. URL: <https://www.businesswire.com/news/home/20171207005963/en/IDC-Forecasts-Worldwide-Spending-on-the-Internet-of-Things-to-Reach-772-Billion-in-2018>
5. Information about the application. URL: <https://ukc.gov.ua/wp-content/uploads/2021/02/Za-rik.pdf>
6. Government contact center. URL: <https://ukc.gov.ua/>
7. State administration in a smart city. URL: <https://knote.edu.ua/file/NjY4NQ==/d649f05f3e9fa1e8356f3af1541ae90e.pdf>
8. Government hotline 1545. URL: <https://www.kmu.gov.ua/uryadova-garyacha-liniya-1545>
9. Citizen. About this app. Royal Media Services. URL: https://play.google.com/store/apps/details?id=newsapp.citizen.v.royalmedia.citizen.vnews&hl=en_US&pli=1
10. What Is Citizen? the Safety and Crime-Tracking App, Explained. URL: <https://www.businessinsider.com/what-is-citizen-app>
11. Citizen - Connect and stay safe. URL: <https://citizen.com/>
12. Dirks, S., & Keeling, M. (2009). A vision of smarter cities: How cities can lead the way into a prosperous and sustainable future. *IBM Institute for business Value*, 8. URL: https://web.archive.org/web/20180429121219id_/http://www-935.ibm.com/services/us/gbs/bus/pdf/ibm_podcast_smarter_cities.pdf
13. European Commission (2009). Investing in the Low Carbon Technologies. *SETPlan*. URL: http://ec.europa.eu/eip/smartcities/index_en.htm
14. Bentivegna, V., Curwell, S., Deakin, M., Lombardi, P., Mitchell, G., & Nijkamp, P. (2002). A vision and methodology for integrated sustainable urban development: BEQUEST. *Building Research & Information*, 30(2), 83–94. DOI: 10.1080/096132102753436468.
15. Against the Smart City. URL: <https://urbanomnibus.net/2013/10/against-the-smart-city/>
16. European Smart Cities. URL: <https://www.smart-cities.eu/>
17. FTTH Council Europe (2018). *Smart City Guide*. URL: www.ftthcouncil.eu
18. Matseliukh Y., Bublyk M., Vysotska V. (2021). Development of Intelligent System for Visual Passenger Flows Simulation of Public Transport in Smart City Based on Neural Network. In *COLINS*, pp. 1087–1138.
19. Al-Khoury A. M. (2012). Emerging markets and digital economy: Building trust in the virtual world. *International Journal of Innovation in the Digital Economy (IJIDE)*, 3(2), pp. 57–69. DOI: 10.4018/jide.2012040105.

20. Krislata I., Katrenko A., Lytvyn V., Vysotska V., Burov Y. (2020). Traffic Flows System Development for Smart City. In ITPM, pp. 280–294.
21. Bublyk M., Zahreva Y., Vysotska V., Matseliukh Y., Chyrun L., Korolenko O. (2022). Information System Development For Recording Offenses In Smart City Based On Cloud Technologies And Social Networks. *Webology* (ISSN: 1735-188X), 19(2).
22. Bublyk M., Kalynii T., Varava L., Vysotska V., Chyrun L., Matseliukh Y. (2022). Decision Support System Design For Low-Voice Emergency Medical Calls At Smart City Based On Chatbot Management In Social Networks. *Webology* (ISSN: 1735-188X), 19(2).
23. Peleshko D., Rak T., Noennig J. R., Lytvyn V., Vysotska V. (2020). Drone Monitoring System DROMOS of Urban Environmental Dynamics. In ITPM, pp. 178–193.
24. Podlesna L., et al. (2020). Optimization Model of the Buses Number on the Route Based on Queuing Theory in a Smart City. In *MoMLeT+ DS*, pp. 502–515.
25. Bublyk M., Matseliukh Y., Motorniuk U., Terebukh M. (2020). Intelligent System of Passenger Transportation by Autopiloted Electric Buses in Smart City. In *COLINS*, pp. 1280–1294.

INFORMATION SYSTEM PROTOTYPE FOR MONITORING AND CONTENT ANALYSIS OF COMPLAINTS FROM SMART CITY RESIDENTS

Yulian Ohonovsky¹, Andrii Berko¹, Lyubomyr Chyrun², Sofia Chyrun¹

¹ Lviv Polytechnic National University,
Information Systems and Networks Department, Lviv, Ukraine

² Ivan Franko National University of Lviv,
Applied Mathematics Department, Lviv, Ukraine

E-mail: yulian.ohonovskyi.knm.2018@lpnu.ua, ORCID: 0000-0003-3853-4158

E-mail: Andrii.Y.Berko@lpnu.ua, ORCID: 0000-0003-2892-9519

E-mail: Lyubomyr.Chyrun@lnu.edu.ua, ORCID: 0000-0002-9448-1751

E-mail: sofia.chyrun.sa.2022@lpnu.ua, ORCID: 0000-0002-2829-0164

© Ohonovsky Y., Berko A., Chyrun L., Chyrun S., 2023

One of the current computerization trends is the active development of “smart city” or “Smart city” technologies, in particular, the support of administrative e-services, the development of transport systems, the use of ecologically clean energy sources, the development of ecology, the modernization of the city’s digital equipment, in an improved state. life in the city, completion of all processes of the city infrastructure. All this complex of works is as accessible as possible with the correct use of “Smart city” technology. Smart cities contribute to sustainable development. Smart technologies fully support cities’ growth and increase efficiency to ensure the well-being of citizens and the efficiency of city governance in the years to come. Along with the advantages of Smart city, there are problems that are provided as a result of the wide use of various mobile applications, big data and various technologies. So that the risks do not outweigh the positive benefits, all smart city initiatives must be focused on the needs of consumers. Access to administrative data will increase the involvement of citizens in public and city processes. Opponents of the smart city express their dismay at the possibility of non-compliance with the proper storage of confidential and personal data, ensuring their security. The method of work is the development of an information system of a smart city to increase the speed of response of communal services to various problems in it. Residents of the city should be classified as other extreme situations and actions should be taken to quickly solve this problem. The object of the study is the process of reducing the response time of communal services to various emergency situations in the city. The subject of the study is the analysis of the speed of response to various tests. The practical significance of the obtained results is the development of a software addition to the information system for collecting and analyzing problems in the city.

Key words: Smart city; smart city; complaints; concept; information technologies; economic efficiency; competitive city; information system.