

СІТ 2021

WORKSHOP

ШКОЛА-СЕМІНАР

МОЛОДИХ ВЧЕНИХ І СТУДЕНТІВ

КОМП'ЮТЕРНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

15 грудня
2021 року



м. Тернопіль,
вул. Чехова 8

fcit.wunu.edu.ua



ЗНУ ЗАХІДНОУКРАЇНСЬКИЙ
НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ



АСОЦІАЦІЯ ФАХІВЦІВ
КОМП'ЮТЕРНИХ
ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

ОРГАНІЗАТОРИ:

- Західноукраїнський національний університет
- Факультет комп'ютерних інформаційних технологій
- Асоціація фахівців комп'ютерних інформаційних технологій

Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Асоціація фахівців комп'ютерних інформаційних технологій

МАТЕРІАЛИ ШКОЛИ-СЕМІНАРУ
МОЛОДИХ ВЧЕНИХ І СТУДЕНТІВ

КОМП'ЮТЕРНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

COMPUTER INFORMATION TECHNOLOGIES

15 ГРУДНЯ 2021 року

CIT'2021

Тернопіль
ЗУНУ
2021

ББК 32.97

УДК 004.2-3+004.9+51.7+519.6-8

Організатори школи-семінару:

Західноукраїнський національний університет

Факультет комп'ютерних інформаційних технологій

Асоціація фахівців комп'ютерних інформаційних технологій

32.97 *Комп'ютерні інформаційні технології: Матеріали школи-семінару молодих вчених і студентів СІТ'2021. – Тернопіль: ЗУНУ, 2021.*

У матеріалах семінару опубліковані результати наукових досліджень і розробок науковців та студентів факультету комп'ютерних інформаційних технологій ЗУНУ, а також інших навчальних і наукових закладів України з таких напрямків: математичні моделі об'єктів та процесів, комп'ютерні мережеві технології; спеціалізовані комп'ютерні системи; системи штучного інтелекту; інженерія програмного забезпечення; комп'ютерні технології інформаційної безпеки та управління проектами. Матеріали призначені для наукових співробітників, викладачів, інженерно-технічних працівників, аспірантів та студентів.

Відповідальний за випуск:

Дивак М. П., д. т. н., професор, декан факультету комп'ютерних інформаційних технологій

Відповідальність за достовірність, стиль викладення та зміст надрукованих матеріалів несуть автори.

©ЗУНУ, 2021

© колектив авторів, 2021

ГОЛОВНИЙ РЕДАКТОР:

ШЕВЧУК Руслан Петрович

к.т.н., доцент

ЗАСТУПНИК ГОЛОВНОГО РЕДАКТОРА:

КРЕПИЧ Світлана Ярославівна

к.т.н., доцент

ЧЛЕНИ РЕДАКЦІЙНОЇ КОЛЕГІЇ

ВОЙТЮК Ірина Федорівна

к.т.н., доцент

ГОНЧАР Людмила Іванівна

к.е.н., доцент

КРЕПИЧ Світлана Ярославівна

к.т.н., доцент

МАРЦЕНЮК Євгенія Олексіївна

к.т.н., доцент

МАНЖУЛА Володимир Іванович

к.т.н., доцент

МЕЛЬНИК Андрій Миколайович

к.т.н., доцент

МАСЛЯК Юрій Богданович

к.т.н., старший викладач

ОЛІЙНИК Ірина Степанівна

к.т.н., доцент

ПУКАС Андрій Васильович

к.т.н., доцент

ПОРПЛИЦЯ Наталія Петрівна

к.т.н., доцент

СПІВАК Ірина Ярославівна

к.т.н., доцент

ШЕВЧУК Руслан Петрович

к.т.н., доцент

ШПІНТАЛЬ Михайло Ярославович

к.т.н., доцент

ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

ПУКАС Андрій Васильович

д.т.н., доцент

КРЕПИЧ Світлана Ярославівна

к.т.н., доцент

ПАПА Олександр Андрійович

к.т.н., викладач

ШЕВЧУК Руслан Петрович

к.т.н., доцент

ЗМІСТ

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РЕАЛІЗАЦІЇ АЛГОРИТМУ СТРЕЛЕЦЬКИХ ВИПРОБУВАНЬ Марценюк Є.О., Деменко М.О.	1
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ Марценюк Є.О., Павлишин Т.В.	2
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНИТОРИНГУ COVID - 19 Марценюк Є.О., Соляр О.М.	3
ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ УКРАЇНСЬКОЇ СОЦМЕРЕЖІ LINSTAGRAM Гончар Л.І., Мацук А.Р., Дем'янюк Д.Б.	4
АНТИПАТЕРНИ В ПРОГРАМНИХ СИСТЕМАХ ІЗ СЕРВІСНО-ОРІЄНТОВАНОЮ АРХІТЕКТУРОЮ Кіселичник І.М.	5
ПІДХІД ДО ОЦІНКИ ЮЗАБІЛІТІ ІНТЕРФЕЙСУ САЙТІВ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ Кошицький Н.В.	7
МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПОШУКУ КОНТЕНТУ НА ОСНОВІ АЛГОРИТМУ A* Тимчишин Б.С., Шпінталь М.Я.	10
АНАЛІЗ І ПІДТРИМКА ОСВІТНЬОЇ ТА НАУКОВОЇ ДІЯЛЬНОСТІ ПРАЦІВНИКІВ ЗВО НА ОСНОВІ ХМАРНИХ СЕРВІСІВ Колода М.К., Манжула В.В., Коваль Д.В.	11
МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПОШУКУ ДОСТУПНИХ МАРШРУТІВ ДЛЯ ЛЮДЕЙ З ОСОБЛИВИМИ ПОТРЕБАМИ Войтюк І.Ф., Бойчук Ю.Ю.	13
СИСТЕМА ОБМІНУ ПОВІДОМЛЕННЯМИ ДЛЯ ОС ANDROID Беркищук В.Б., Шпінталь М.Я.	14
МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛЮВАННЯ ГАЗООБМІНУ В ДИХАЛЬНИХ ШЛЯХАХ ЛЮДИНИ Войтюк І.Ф., Панасюк К. А.	16
ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МОДУЛЯ ШИФРУВАННЯ ДАНИХ МЕСЕНДЖЕРА ... Шаповал Д.А.	17
МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ У СОЦІАЛЬНИХ МЕРЕЖАХ Фатюк В.І., Міклешевський Е.Р., Сушко С.С., Гарчар Р.Р.	18
МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ РОЗПОДІЛУ РЕСУРСІВ В ХМАРНИХ ОБЧИСЛЕННЯХ Гончар Л.І., Порученко В.Ю., Мельниченко В.В., Лукянчук В.А.	21
МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ФОРМУВАННЯ ПОРТФЕЛЯ ЦІННИХ ПАПЕРІВ В УМОВАХ ОБМЕЖЕНОЇ ВИБІРКИ Магера С.Р., Костик Б.П., Романюк М.С., Кушнірчук Р.М.	23
ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ НЕДОСТОВІРНОЇ ІНФОРМАЦІЇ НА ВЕБ- РЕСУРСАХ Сидорчук І.А., Пасічник М.П., Карпишин Ю.В., Костик Б.П.	25
МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АНАЛІЗУ КОДІВ ПРОГРАМ НА ОСНОВІ СЕМАНТИЧНИХ МОДЕЛЕЙ Тарасюк В., Кравчук А., Матвєєв В.А.	27

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ РЕАЛІЗАЦІЇ АЛГОРИТМУ СТРІЛЕЦЬКИХ ВИПРОБУВАНЬ

Марценюк Є.О.¹⁾, Деменко М.О.²⁾

Західноукраїнський національний університет

¹⁾ к.е.н., доцент; ²⁾ магістрант

I. Постановка проблеми

На сьогоднішній день, коли на Сході країни відбуваються бойові дії та Збройні Сили України потребують кваліфікованих та професійно підготовлених кадрів, важливим завдання є стрілецькі навчання та тренування військовослужбовців.

Одним із критеріїв професіоналізму є рівень точності попадань у ціль та їх аналіз, у залежності від точності випробувань, їх кількості та умов, за яких вони були проведені. Тому розробка програмного забезпечення для реалізації алгоритму стрілецьких випробувань є актуальною [1].

II. Мета роботи

Метою даної роботи є створення програмного забезпечення для реалізації алгоритму стрілецьких випробувань.

III. Програмне забезпечення для реалізації алгоритму стрілецьких випробувань

На рисунку 1 представлена блок-схема алгоритму співставлення реальних та теоретичних результатів випробувань і їх аналітика в умовах обмеженого доступу до мережі.

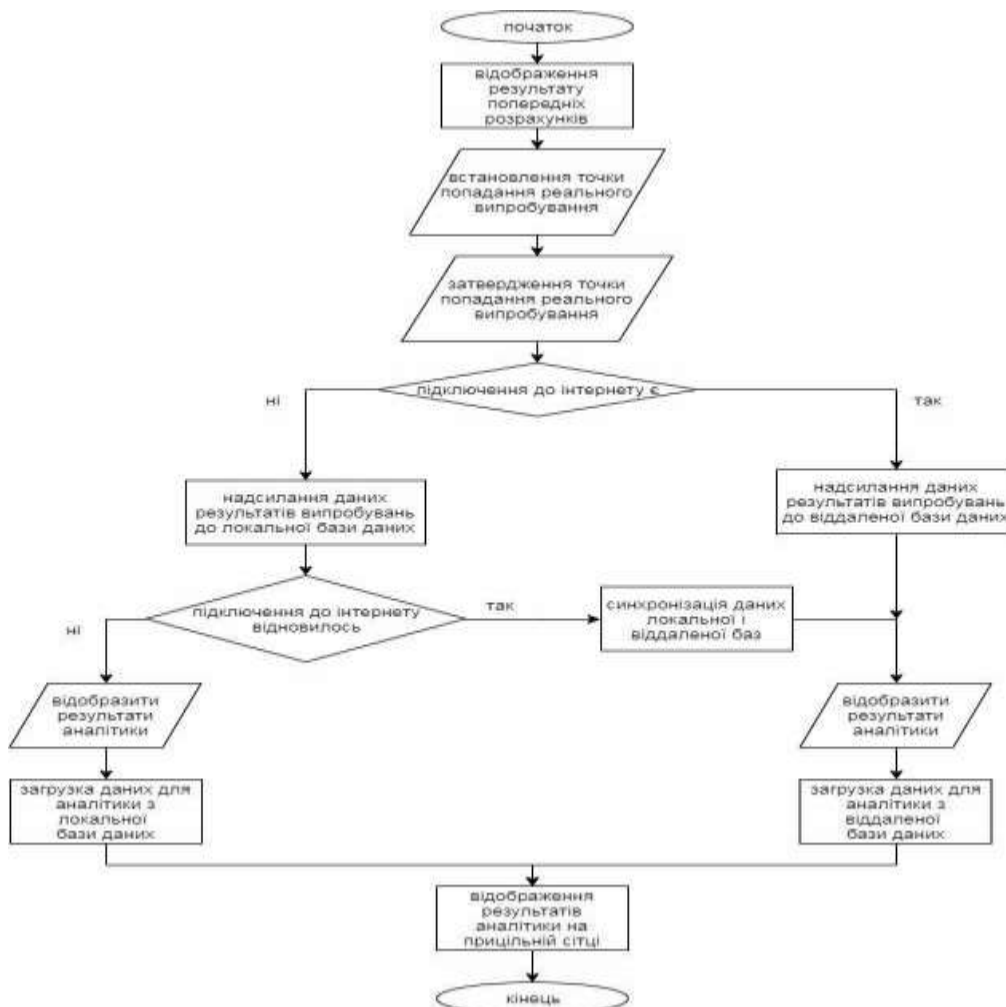


Рисунок 1 - Блок-схема алгоритму реєстрації об'єктів для синхронізації

Користувач опиняється на сторінці, де візуально зображуються результати попередніх розрахунків і вони мають бути відміченими на віртуальній прицільній сітці. Далі користувач сам повинен вказати результати вже реальних випробувань на тій же прицільній сітці.

Якщо підключення до мережі інтернет існує, то дані відправляються, шляхом використання Firestore API, до віддаленої бази даних. Після цього користувач відразу може загрузити аналітику проведених стрілецьких випробувань, за цим вона може бути відображена вже візуально для практичної наочності. Якщо підключення до мережі немає, дані надсилаються до локальної бази даних пристрою. Якщо з'єднання відновлюється, то відображення аналітики проводиться із застосуванням вхідних даних, які беруться вже із синхронізованої віддаленої бази даних, після чого результати аналітики відображаються користувачу.

Висновок

Розроблено програмне забезпечення для реалізації алгоритму стрілецьких випробувань, яке дасть можливість в значній мірі покращити навчання та тренування військовослужбовців.

Список використаних джерел

1. Відхилення кулі неоднорідним вітром [Електронний ресурс]. – Режим доступу: <http://www.hpbt.org/articles/veter.htm>.

УДК 62.503.55

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПРОГНОЗУВАННЯ КУРСУ КРИПТОВАЛЮТ

Марценюк Є.О.¹⁾, Павлишин Т.В.²⁾

Західноукраїнський національний університет

¹⁾ к.е.н., доцент; ²⁾ магістрант

I. Постановка проблеми

На даний час все ширша цікавиться відносно новою формою фінансового активу - криптовалютою. Природно, разом з популярністю нових валют зростає попит на сервіси, що дозволяють відстежувати їхню динаміку, відповідаючи на запитання, якими є найближчі перспективи електронних грошей та, з точки зору інвестування, яку стратегію доцільно обрати. Тому, розробка програмного забезпечення для прогнозування курсу криптовалют є актуальною задачею [1].

II. Мета роботи

Метою даної роботи є створення програмного забезпечення для прогнозування курсу криптовалют.

III. Програмне забезпечення для прогнозування курсу криптовалют

У роботі розроблене програмне забезпечення, яке показує як система прогнозує курс криптовалют. Генеруються дані для графіку регресії відносно обраного алгоритму та періоду, виводяться дані курсу валют з сервера в пам'ять, ініціалізується плагін відображення графіків.

Попередній графік видаляється з інтерфейсу. Дані курсу криптовалют завантажуються в бібліотеку графіка та виводиться на екран у вигляді регресії, дати вказуються згідно обраного періоду разом з кількістю днів прогнозу.

На рисунку 1 зображено вікно програми, де відображено прогноз курсу криптовалюти Bitcoin, яка є найпопулярнішою на даний час.

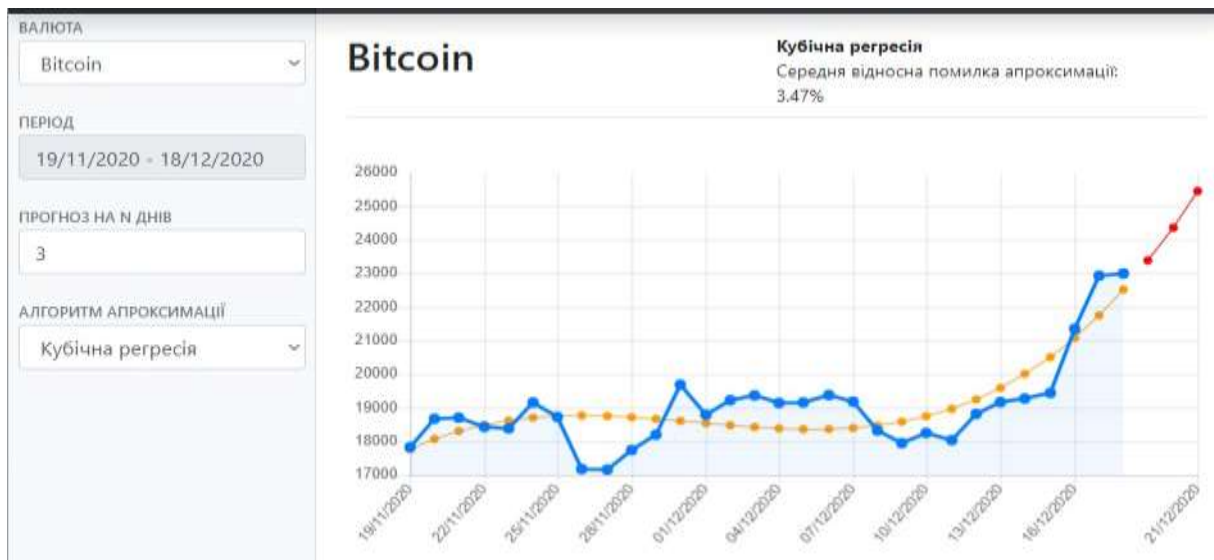


Рисунок 1 - Прогноз курсу криптовалют

Висновок

Розроблено програмне забезпечення для процесу прогнозування курсу криптовалют на певний період, яке дасть можливість спланувати вкладення коштів без втрат.

Список використаних джерел

1. Товкун Л. В. Характерні особливості криптовалюти: окремі теоретичні та практичні аспекти [Електронний ресурс] / Л. В. Товкун, В. В. Черниш // Порівняльно-аналітичне право: електрон. наук. фах. вид. – Ужгород, 2020. – № 4. – С. 561–567. – Режим доступу: http://pap-journal.in.ua/wp-content/uploads/2020/08/PAP-4_2020.pdf

УДК 62.503.55

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ МОНІТОРИНГУ COVID - 19

Марценюк Є.О.¹⁾, Соляр О.М.²⁾

Західноукраїнський національний університет

¹⁾ к.е.н., доцент; ²⁾ магістрант

I. Постановка проблеми

В теперішніх умовах пандемії важливим завдання є необхідність мати систему, яка дозволить відстежувати вогнище захворювань, небезпечні місця, а також отримувати повідомлення про контакти з COVID-інфікованим. Мобільні додатки, які були створені на замовлення урядів багатьох країн, орієнтовані на інформаційну функцію та функцію контролю дотримання карантину.

Тоді як попередження про контакти з інфікованими може скоротити кількість інфікованих та полегшити стан перебігу хвороби у тих, хто, все ж таки, був заражений. Тому створення програми для моніторингу COVID - 19, який дасть можливість уникнути небезпечних місць зараження та стежити за станом свого здоров'я, коли все таки вже мали контакт з інфікованими.

II. Мета роботи

Метою роботи є створення програмного забезпечення для моніторингу COVID - 19.

III. Програмне забезпечення для моніторингу COVID - 19.

У роботі розроблено мобільний додаток спрямований на побудову теплової карти поширення COVID - 19 (рисунок 1).

Теплова карта дозволяє у зручній формі переглядати карту контактів з інфікованими особами. Кольором проводиться ранжування за тривалістю контактів. Ранжування необхідне, оскільки зі зростанням тривалості контакту з інфікованою людиною зростає і ймовірність інфікування інших осіб.

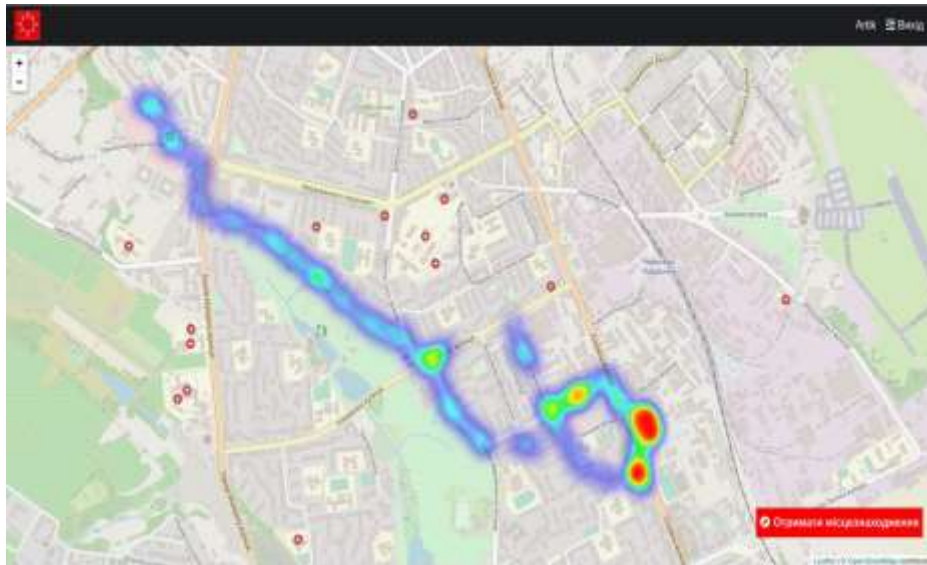


Рисунок 1 - Теплова карта можливих контактів

Теплова карта дозволяє мати як ретроспективну, так й предиктивну функцію – користувач може визначити які місця є місцями найбільшої концентрації вірусу з метою цілеспрямованого уникнення таких місць.

Висновок

Розроблено програмне забезпечення для моніторингу COVID – 19 та побудови теплової карти, що дасть можливість уникнути місця зараження, а якщо побували в місцях скупчення інфікованих, то вчасно звернутись до лікаря.

Список використаних джерел

1. CDC. 2019 Novel Coronavirus, Wuhan, China. CDC. Available at <https://www.cdc.gov/coronavirus/2019-ncov/about/index.html>. January 26, 2020; Accessed: January 27, 2020.

УДК 62.503.55

ОСОБЛИВОСТІ РЕАЛІЗАЦІЇ УКРАЇНСЬКОЇ СОЦМЕРЕЖІ LINSTAGRAM

Гончар Л.І.¹⁾, Мацук А.Р.²⁾, Дем'янюк Д.Б.³⁾

Західноукраїнський національний університет

¹⁾ к.е.н., доцент; ^{2,3)} студент

I. Постановка проблеми

Сьогодні кількість активних користувачів соціальних мереж наближається до 15 млрд [1]. Практично кожен користувач мережі Інтернет має, як мінімум, один профіль в соціальній мережі.

Актуальність теми дослідження обумовлена, насамперед, тим, що уявити сучасну комунікацію без соціальних мереж неможливо. За даними Brand Analytics, щодня в соціальних мережах з'являється близько 30 млн нових повідомлень і 35 млн осіб залишають хоча б одне публічне повідомлення на місяць [2].

II. Мета роботи

Метою даного дослідження є розробка української соцмережі LINKSTAGRAM.

III. Особливості реалізації соціальної мережі

Для розроблення веб-порталу із функціоналом соціальної мережі була обрана клієнт-серверна архітектура та використані технології React JS, SCSS/SASS, Redux, Figma. Розроблений портал є адаптивний та характеризується зручним та простим інтерфейсом (рисунок 1).

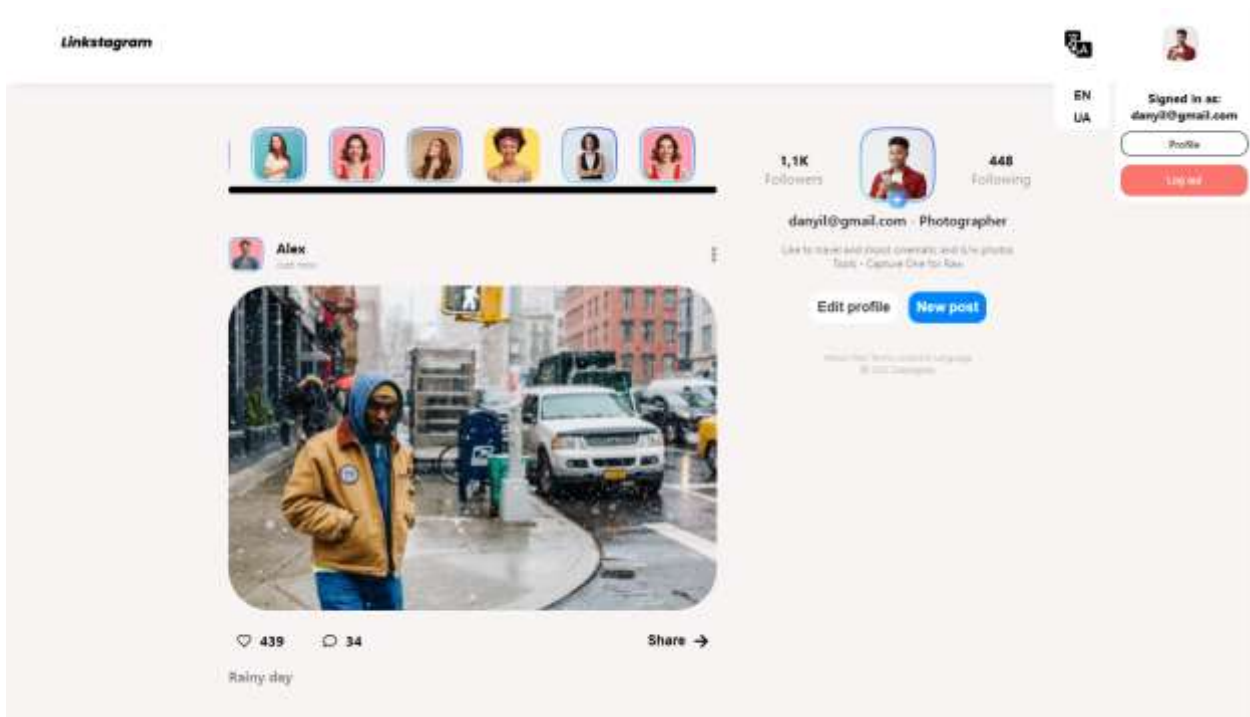


Рисунок 1 – Домашня сторінка соцмережі LINKSTAGRAM

Висновок

У роботі розроблено веб-сайт із функціоналом соціальної мережі LINKSTAGRAM.

Список використаних джерел

1. Shevchuk R. Improve the Security of Social Media Accounts / R. Shevchuk, Y. Pastukh // Proceedings of the 2019 9th International Conference “Advanced Computer Information Technologies” – Ceske Budejovice, Czech Republic. – June 5–7, 2019 –P.439-442.
2. Мавріна М. О., Корж І. Ю. Основні тенденції та перспективні технології Web-розробки. Новітні інформаційні системи та технології. 2016. № 5. URL: <http://journals.pntu.edu.ua/mist/article/view/562> (дата звернення: 19.11.2021).
3. Створення сайтів: цілі та методи. URL: <https://pbb.lviv.ua/statti-i-novyny/statti-shchodo-stvorennia-saitu/tsili-saitiv> (дата звернення: 19.11.2021).

УДК 004.67

АНТИПАТЕРНИ В ПРОГРАМНИХ СИСТЕМАХ ІЗ СЕРВІСНО-ОРІЄНТОВАНОЮ АРХІТЕКТУРОЮ

Кіселичник І.М.

Західноукраїнський національний університет, магістрант

І. Актуальність проблеми

Сервісно-орієнтований стиль програмних систем дуже широко поширений у промисловій розробці, оскільки дозволяє впроваджувати гнучкі та масштабовані розподілені системи за конкурентоспроможною ціною. Результатом розробки є автономні, багаторазові та незалежні одиниці платформи-сервіси, які можна використовувати через будь-яку мережу, включаючи Інтернет [1]. Цей підхід набуває широкого поширення в індустрії інженерії програмного забезпечення, оскільки дозволяє реалізовувати розподілені системи, що характеризуються високою якістю.

Атрибути якості можуть стосуватися системи (наприклад, доступність, змінюваність), аспектів, пов'язаних з бізнесом (наприклад, час виходу на ринок) або архітектури (наприклад, правильність, узгодженість) [2]. Підтримка атрибутів якості на високому рівні є критичною проблемою, оскільки сервісно-орієнтовані системи не мають центрального контролю та повноважень, мають обмежену

наскрізну видимість сервісів, піддаються непередбачуваним сценаріям використання та підтримують динамічний склад системи [2].

Постійна еволюція сервісно-орієнтованих систем може легко погіршити загальну архітектуру системи, і, таким чином, можуть з'явитися неправильні рішення, відомі як антипатерни SOA [3].

Це ті моделі, яких слід уникати. Якщо ми їх вивчаємо і вміємо розпізнавати, то ми повинні мати можливість їх уникати. Якщо ми не знаємо про антипатерни, ми ризикуємо повторити помилки, які інші робили кілька разів. Формально фіксуючи повторювані помилки, можна розпізнати симптоми, а потім працювати над тим, щоб не потрапити в проблемну ситуацію. Антипатерни говорять нам, як ми можемо перейти від проблеми до поганого рішення. Насправді вони виглядають як гарне рішення, але при застосуванні вони мають зворотний ефект. Знання поганої практики, можливо, так само цінне, як і знання хорошої практики. Маючи ці знання, ми можемо здійснити рефакторинг рішення, якщо ми рухаємося від антипатерна. Як і у випадку з патернами, також доступні каталоги антипатернів.

Антипатерни в програмних системах, заснованих на сервісах, є «поганими» рішеннями, що повторюються проблемами проектування. На відміну від патернів дизайну, антипатерни є добре перевіреними рішеннями, яких інженерам слід уникати. Антипатерни також можуть бути введені як наслідок різних змін, таких як нові вимоги користувача або зміни операційного середовища. Враховуючи очевидний негативний вплив антипатернів SOA, існує очевидна й нагальна потреба в техніках та інструментах для їх виявлення.

II. Мета дослідження

Метою цієї роботи є дослідження антипатернів у програмних системах на основі SOA, тобто виявлення таких частин системи (підсистеми), які беруть участь у певному антипатерні. З цієї точки зору виявлення антипатернів може сприяти досягненню різних цілей, наприклад для реінженерії систем [4], виявлення патернів проектування [5], ідентифікації компонентів у застарілих системах [6] тощо.

III. Класифікація антипатернів SOA

Сервісно-орієнтована архітектура – це архітектурний шаблон (див. рис. 1), у якому програма складається з програмних сервісів і користувачів програмного забезпечення (також відомих як клієнти або запитувачі сервісів).

Проблема специфікації антипатернів SOA не нова. Вже існує кілька сховищ антипатернів. Вони зібрані в книгах [7] або веб-ресурсах [8].

Одна з робіт по виявленню антипатернів в сервіс-орієнтованих архітектурах (SOA) була запропонована в Moha et al. у 2012 році [9]. Там автори запропонували підхід до визначення та виявлення великого набору антипатернів SOA, що керують такими поняттями, як гранулярність, когезія та дублювання. На додаток до підходу, автори виділили три антипатерни, а саме: сервіс-вузьке місце, ланцюг сервісів, сервіси даних.

Вузьке місце – це сервіс, який використовується багатьма іншими компонентами системи, і, як наслідок, характеризується високим рівнем вхідних та вихідних з'єднань, що впливають на час відповіді сервісу. Ланцюжки сервісів виникають, коли бізнес-об'єкт досягається за допомогою довгого ланцюга послідовних викликів. Сервіс даних – це сервіс, який виконує прості операції пошуку інформації або доступу до даних, що може вплинути на час підключення компонента.

У 2012 році Rotem-Gal-Oz [10] визначив антипатерн «вузла» – невеликий набір підключених сервісів, які, однак, тісно залежать один від одного. Таким чином, антипатерн може зменшити простоту використання та збільшити час відгуку.

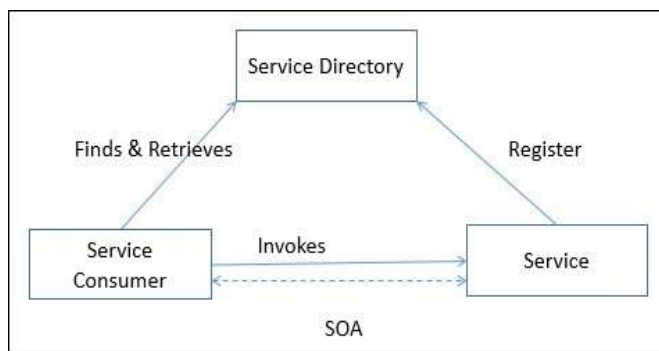


Рис. 1. Архітектурний шаблон SOA

Іншим прикладом антипатерн є «підсана купа», визначена Kr' al et al [11]. Він з'являється, коли багато невеликих сервісів використовують спільні дані, доступ до яких можна отримати через сервіс, яка представляє антипатерн «сервіс даних».

У роботі [12] запропонований антипатерн «дублікатного сервісу», який впливає на спільний доступ до сервісів, які містять подібні функції, викликаючи проблеми в процесі підтримки. У роботі [10] визначено набір антипатернів для додатків J2EE. «Мультисервісний» (Multi service) антипатерн виділяється, «крихітний сервіс» (Tiny service) і «балакучий сервіс» (Chatty service). Мультисервіс – це сервіс, який забезпечує різноманітні бізнес-операції, які практично не схожі (наприклад, належать до різних підсистем), що може вплинути на доступність сервісу та час відповіді. Tiny service – це невеликий сервіс з кількома методами, які завжди використовуються разом. Це може призвести до неможливості повторного використання. Нарешті, антипатерн «балакучий сервіс» являє собою такі сервіси, які постійно взаємодіють один з одним, передаючи невелику кількість інформації.

Висновки

У даній роботі розглянуто питання необхідності автоматизованого моніторингу програмних систем, реалізованих на основі сервісно-орієнтованої архітектури, в умовах постійного масштабування та вдосконалення, коли складність систем зростає швидше, ніж людина може впоратися з її аналізом. У контексті цієї роботи було наведено класифікацію антипатернів і для подальшого детального дослідження був обраний клас метричних специфічних антипатернів.

Список використаних джерел

1. T. Erl, Service-Oriented Architecture: Concepts, Technology, and Design. Prentice Hall PTR, August 2005.
2. L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice. Boston, MA: Addison-Wesley, 2003.
3. G. Farrow. SOA antipatterns: When the SOA paradigm breaks // IBM Developer Works [Online]. Available: http://www.ibm.com/developerworks/library/wa-soa_antipattern/
4. D. Card, R. Glass. Measure Software Design Quality. Prentice-Hall, NJ, 1990.
5. R. Marinescu. Detecting Design Flaws via Metrics in Object-Oriented Systems. In Proceedings of TOOLS USA 2001, pages 103–116. IEEE Computer Society, 2001.
6. A. Trifu. Using Cluster Analysis in the Architecture Recovery of OO Legacy Systems. Diploma Thesis, Karlsruhe and the "Politehnica" University Timisoara, 2001.
7. M. Bell. SOA Modeling Patterns for Service Oriented Discovery and Analysis. Hoboken, John Wiley & Sons, Inc., 2010.
8. SOA Antipatterns // SOFA: Service Oriented Framework for Analysis [online] [available: <http://sofa.uqam.ca/resources/antipatterns.php#Breaking%20Self-descriptiveness>]
9. N. Moha, F. Palma, M. Nayrolles, B. J. Conseil, Y.-G. Gúeheneuc, B. Baudry, J.-M. Jézequel. Specification and Detection of SOA Antipatterns. In International Conference on Service-Oriented Computing (ICSOC). Pp. 1–16. 2012
10. A. Rotem-Gal-Oz, SOA Patterns, 1st ed. Manning Publications, 2012.
11. J. Kr' al and M. Zemlicka, "The most important service-oriented antipatterns," in ICSEA, 2007, p. 29.
12. L. Cherbakov, M. Ibrahim, and J. Ang, "Soa antipatterns: the obstacles to the adoption and successful realization of service-oriented architecture".
13. B. Dudley, J. Krozak, K. Wittkopf, S. Asbury, and D. Osborne, J2EE Antipatterns, 1st ed. New York, NY, USA: John Wiley & Sons, Inc., 2002.

УДК 004.67

ПІДХІД ДО ОЦІНКИ ЮЗАБІЛІТІ ІНТЕРФЕЙСУ САЙТІВ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

Кошицький Н.В.

Західноукраїнський національний університет, магістрант

І. Актуальність проблеми

Популярність сайту залежить від низки параметрів. І щодня в Інтернеті з'являються нові майданчики, що згодом закриваються. Причина такого результату зазвичай криється в низькій відвідуваності сайту. Навіть за ідеального маркетингу немає гарантії, що відвідувачі заходять на вашу веб-сторінку. Тому потрібно докласти максимум зусиль, щоб сайт був зручним для користувачів. За зручність якраз і відповідає юзабіліті [1]. Основна частина цільової аудиторії заходить за

інформацією, якщо виникнуть труднощі в пошуку потрібного контенту, то користувач піде до конкурентів.

Проектування людино-машинного інтерфейсу є невід'ємним і важливим завданням життєвого циклу розробки програмного забезпечення. Від якості результатів проектування залежить рівень задоволеності користувачів під час використання цього продукту. Поняття юзабіліті поєднує у собі всі аспекти, що стосуються зручності інтерфейсу. Насамперед про юзабіліті говорять, коли розробляють веб-додатки та веб-сайти. Потенційний клієнт повинен затриматися на веб-ресурсі якнайдовше, підвищуючи ймовірність конверсії.

На даний момент існують різні підходи до оцінки юзабіліті сайтів з погляду наповнення сторінок, їх оформлення, розташування елементів керування тощо [2]. Але комплексного програмного продукту, який оцінював би юзабіліті будь-якого запропонованого сайту, поки немає.

Зручність використання веб-сайту безпосередньо впливає на кількість його постійних користувачів, продажів, якщо така функціональність передбачається на сайті, на ступінь задоволення користувачів від роботи з сайтом. Підвищення юзабіліті має ґрунтуватися на формальних критеріях оцінки, які ув'язуються в єдиній системі правил нечіткої логіки [5].

Таким чином, завдання розробки автоматизованої системи для оцінки юзабіліті веб-сайтів є актуальним.

II. Мета дослідження

Метою цієї роботи є розробка методу та алгоритму нечіткого висновку оцінки юзабіліті інтерфейсу, який базується на сформульованих правилах нечіткого висновку для лінгвістичних змінних, які характеризують інтерфейс сайту з точки зору його юзабіліті та дозволяють отримувати числову інтерпретацію показника юзабіліті на основі функцій належності.

III. Стратегії підвищення юзабіліті сайтів

Передусім необхідно перевірити швидкість завантаження сторінок. Оптимальним варіантом вважається 1-2 секунди, якщо довше, то 50% користувачів залишать сторінку, не дочекавшись її завантаження.

Щоб виправити ситуацію та прискорити свій сайт, слід оптимізувати код, а саме, прибрати зайві прогалини в коді, об'єднати стилі, те саме стосується і JavaScript. Слід керувати `htaccess` і додати кешування сторінок. Під час заходу веб-ресурс зберігатиметься в кеші, при повторному відвідуванні сторінка підвантажуватиметься з кешу, що заощаджує час на завантаження.

Ось ще кілька кроків, які слід виконати [3, 4]:

1. Виправлення помилок у HTML – необхідно переконатися, що всі теги закриті. Перевірити атрибути в тегах, наприклад, `class` у документі може траплятися кілька разів з однаковим ім'ям. Ідентифікатор `id` повинен зустрічатися з одним ім'ям без дублікату.

2. Кроссбраузерність та кроссплатформність – це важливий показник сайту, який має однаково добре відображатися у всіх браузерах та мобільних телефонах. Наприклад, як це виглядає на мобільних пристроях.

Коли справа стосується переносних гаджетів, то тут блоки сайту повинні вміщуватися у дисплей без смуг прокручування. Цього можна досягти, прописавши стилі у файлі CSS. Щоб зробити налаштування для смартфонів, необхідно додати в HTML мета-тег з атрибутом `name` і вписати туди `viewport`. Атрибут `content` теж потрібно заповнити значенням "`width = device - width, initial-scale = 1.0`". Для вказівки стилів смартфона необхідно використовувати команду `media`, і у фігурних дужках додавати стилі смартфону, приклад нижче.

3. Необхідно переконатися, що всі елементи клікабельності працюють і виконують необхідні функції. Іноді зустрічаються блоки, які виглядають як кнопки, але виявляється, що це є частиною дизайну. Краще для об'єктів дизайну використовувати окремий стиль, який відрізняється від кнопок.

4. Необхідно дотримуватися єдиного стилю. Коли користувач шукає інформацію, він переглядає кілька сторінок, треба зробити так, щоб у них був єдиний дизайн. Якщо вони відрізнятимуться, клієнт вирішить, що вже знаходиться на іншому сайті. Заздалегідь необхідно продумати, яких тонів буде сайт, темних чи світлих. Темний варіант відмінно підійде для використання у нічний час доби.

5. Необхідно уникати елементів, що перекриваються, і спливаючих вікон. Бувають ситуації, коли відвідувач шукає кнопку, а вона захована за контейнером. Основна риса спливаючих вікон (`popup`), вони з'являються в невідповідний момент і нерідко бувають ситуації, коли хрестик відсутній, а закрити можна, виконавши дію, наприклад, підписатися. Краще так не робити, а прив'язати `popup` до закриття

сторінки. Наприклад, користувач прочитав статтю і зібрався йти, в цей момент відкривається вікно з повідомленням: “Зібралися йти? Заберіть навчальний матеріал, залиште email”.

6. Доступність інформації. Стаття має бути читабельною та повністю відображатися на пристрої будь-якої роздільної здатності.

7. Логіка сайту. У ресурсу має бути певна структура, щоб клієнт міг швидко досягти мети і знайти інформацію, що цікавить.

IV. Метод оцінки юзабіліті інтерфейсу сайтів на основі нечіткого висновку

Основною проведення операції нечіткого логічного висновку є база правил, що містить нечіткі висловлювання у формі «якщо» і функція належності для відповідних лінгвістичних термів [5, 6].

Базою нечіткого висновку є чотири лінгвістичні змінні:

- Q_C – кількість кольорів на веб-сторінці;
- Q_O – кількість об'єктів;
- Q_P – кількість зображень;
- U – юзабіліті веб-сторінки загалом.

Множина термів для перших трьох змінних включає такі терми:

$Q_O, Q_P, Q_C = \{\text{«мало»}, \text{«нормально»}, \text{«багато»}\}.$

Змінна юзабіліті описується термами «погане», «середнє» та «хороше»:

$U = \{\text{«погане»}, \text{«середнє»}, \text{«хороше»}\}.$

Лінгвістичні змінні ранжуються за важливістю, їх впливом на кінцеве юзабіліті сайту. Кількість кольорів в першу чергу впливає на сприйняття і зручність роботи користувача на сайті. Кількість об'єктів має велику значущість у порівнянні з кількістю зображень. Ранжування зроблено нами на основі суб'єктивної оцінки різних сайтів.

Висновки

У даній роботі розглянуто питання необхідності автоматизованого оцінювання юзабіліті веб-сайтів. Для цього пропонується розробка додатку для оцінки юзабіліті інтерфейсу веб-сайтів, за допомогою якого можна здійснити оцінку та забезпечити підвищення юзабіліті ґрунтуючись на формальних критеріях оцінки, які ув'язуються в єдиній системі правил нечіткої логіки.

Список використаних джерел

1. Nielsen J, Pernice K. Eyetracking Web Usability. Berkeley, CA: New Riders; 2009.
2. Olmsted-Hawala E, Holland TH, Quach V. Usability testing. In: Bergstrom J, Schall A, editors. Eye Tracking in User Experience Design. Waltham, MA: Morgan Kaufmann; 2014:49-80.
3. Acton T, Golden W, Gudee S, Scott M. Usability and acceptance in small-screen information systems. In: Proceedings of 9th European Collaborative Electronic Commerce Technology and Research Conference. 2004 Presented at: The 9th European Collaborative Electronic Commerce Technology and Research Conference; January 1, 2004; Surrey, UK URL: <http://www.pmn.co.uk/smallscreenusability.pdf>
4. Abran A, Khelifi A, Witold S, Seffah A. Usability meanings and interpretations in ISO standards. Software Quality Journal 2003;11(4):325-338.
5. Priyanka Kausha, Neeraj Mohan, Parvinder Sandhu S. Relevanc y of Fuzzy Concept in Mathematics. Internationa Journal of Innovation, Management and Techno logy. 2010; 1(3):312-315.
6. Poonam Gupta. Application of Fuzzy Logic in Daily Life. International Journal of Advanced Research in Computer Science. 2017; 8(5):1795-1800.
7. Hayward, Davidson. Fuzzy Logic Application, Analyst. 2003; 128:1304-1306.

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ПОШУКУ КОНТЕНТУ НА ОСНОВІ АЛГОРИТМУ А*

Тимчишин Б.С.¹⁾, Шпінталь М.Я.²⁾

Західноукраїнський національний університет

¹⁾ магістрант; ²⁾ к.т.н., доцент

I. Постановка проблеми

Спільною проблемою більшості веб-сайтів та додатків є велика тривалість пошуку різного контенту, що призводить до проблем з оперативністю та продуктивністю роботи з системою. Працювати з електронними документами на репозиторії набагато зручніше, оскільки можна працювати дистанційно з будь-якого кутка нашої планети, не прив'язуючись до конкретного місця роботи. Створення навчального контенту, а також його резервної копії в репозиторії дозволяє таким архівам бути невразливими, а отже, захищеними від втрат інформації. Такий підхід до зберігання документів "застрахує" від одночасної втрати усіх архівів. До того ж, інформація може бути швидко відновлена.

II. Мета роботи

Основною метою системи є забезпечення способу централізованого і довготривалого зберігання в електронному вигляді різного навчального контенту, що забезпечує змістовне наповнення навчальних програм університету, включення до структури навчального процесу, формування методичних систем навчання університету, надання доступу до них викладачам і студентам.

Розроблений програмний продукт дозволить надійно зберігати та раціонально використовувати інтелектуальний ресурс – навчально-методичні комплекси дисциплін, та підручники, які необхідні для підвищення якості освіти.

III. Проектування програмної системи

При проектуванні архітектури системи, що розробляється було обрано клієнт-серверну архітектуру, оскільки саме вона найкраще підходить для вирішення зазначеної задачі. Для відображення бізнес-логіки системи, що розробляється, спроектовано UML - діаграму класів. Діаграма класів зображена відображає класи спроектованої системи і відношення між ними.

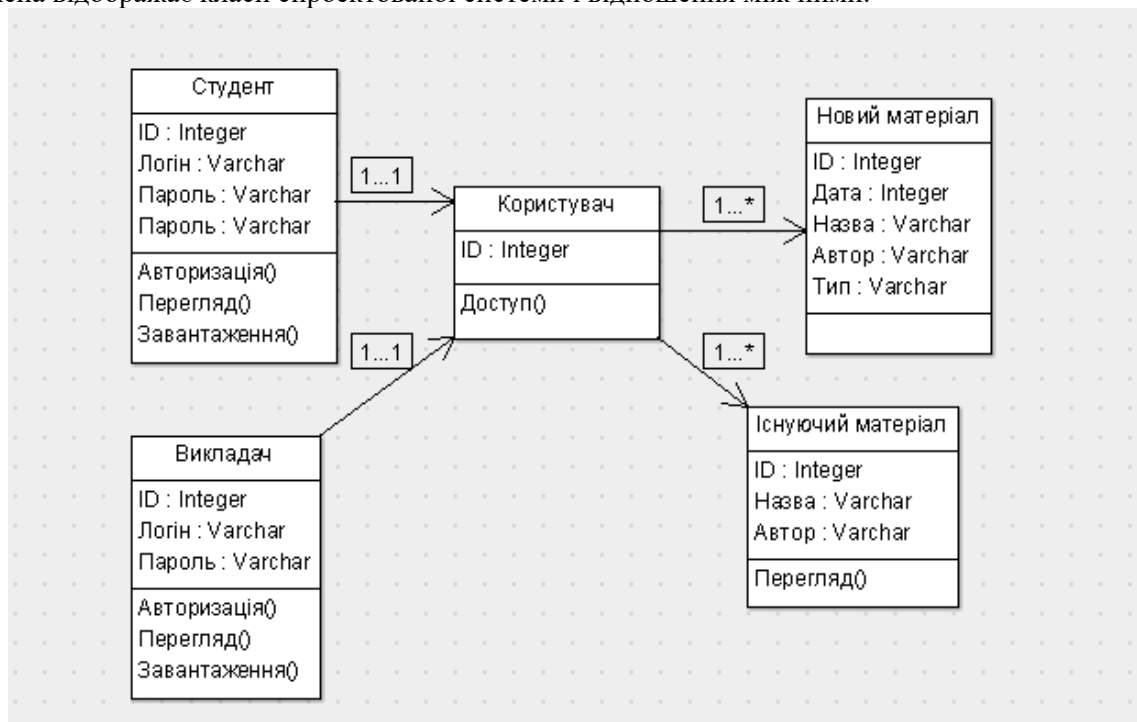


Рисунок 1 – Діаграма класів

Діаграма дозволяє створювати логічне представлення на основі якого створюється вихідний код описаних класів. Значки діаграми дозволяють відображати ієрархію системи, взаємозв'язки класів та інтерфейсів. Система яка проектується складається з наступних класів:

- User;
- NewMatirial;
- ExistMatirial.

Висновок

У даній роботі розроблено програмне забезпечення навчального репозиторію та системи пошуку контенту на основі алгоритму A*, яке дозволяє водночас користуватися системою багатьом користувачам, що в свою чергу полегшить роботу та пошук необхідної інформації.

Список використаних джерел

1. w3schools.com – the world's largest development site. [Електронний ресурс]. — Режим доступу: <http://w3schools.com/>, для доступу до інформаційних ресурсів авторизація не потрібна. — Назва з екрану. W3Schools – Educate Yourself.
2. HTML specification Version 4.01 [Електронний ресурс]: офіційна специфікація мови розмітки гіпертекстових посилань / W3C. Режим доступу: <http://www.w3.org/TR/REC-html40/>, для доступу до інформаційних ресурсів авторизація не потрібна. — Назва з екрану. World Wide Web Consortium (W3C).
3. uk.wikipedia.org – Вільна енциклопедія [Електронний ресурс]. – Режим доступу: <http://uk.wikipedia.org> для доступу до інформаційних ресурсів авторизація не потрібна — Назва з екрану. Вікіпедія. Вільна енциклопедія.

УДК 004.67

АНАЛІЗ І ПІДТРИМКА ОСВІТНЬОЇ ТА НАУКОВОЇ ДІЯЛЬНОСТІ ПРАЦІВНИКІВ ЗВО НА ОСНОВІ ХМАРНИХ СЕРВІСІВ

Колода М.К.¹⁾ Манжула В.В.²⁾ Коваль Д.В.³⁾

Західноукраїнський національний університет

^{1,3)} магістрант; ²⁾ к.т.н., доцент

I. Актуальність проблеми

Особливістю роботи викладачів вищих навчальних закладів, до яких належать університети, є те, що в основі їх діяльності лежить постійна взаємодія науково-педагогічної складових.

Важливим моментом характеристики науково-педагогічної діяльності викладача, яка визначається структурою знань, що становить зміст навчально-виховного матеріалу, є необхідність постійного оновлення цього змісту. Складність цієї структури і динаміка оновлення її змісту змушують викладача звертатися до наукових знань, які перейнялися на різних стадіях засвоєння з дисциплінарними знаннями. Саме тут діяльність викладача стає науковою.

Використання існуючих методів ведення обліку показників освітньої та наукової діяльності на сьогоднішній момент забирають багато часу, часто не є точними.

Зважаючи на вищенаведене, розробка хмарного сервісу [1] ведення звітності з наукової діяльності кафедри університету є актуальною задачею.

II. Мета дослідження

Мета і задачі дослідження є розробка хмарного сервісу для ведення звітності з наукової діяльності кафедри університету. Для досягнення цієї мети необхідно: провести огляд та аналіз існуючих систем ведення звітності з наукової діяльності, спроектувати хмарний сервіс для ведення звітності з наукової діяльності кафедри університету та створити програмне забезпечення для його реалізації.

III. Збір та аналіз даних про освітню та наукову діяльність працівників ЗВО

Веб-парсинг є процесом автоматичного добування даних або збору інформації зі всесвітньої мережі [2]. Це сфера з активними розробками, які мають спільну мету з семантичним веб-зором, амбітна ініціатива, яка все ще потребує проривів у обробці тексту, семантичному розумінні, штучному інтелекті та взаємодії людини та комп'ютера. Парсинг веб-сторінки включає її отримання та

витагування даних з неї. Отримання — це завантаження сторінки (що робить браузер, коли користувач переглядає сторінку). Тому веб-сканування є основним компонентом веб-парсингу, щоб отримати сторінки для подальшої обробки. Після отримання може відбутися вилучення. Вміст сторінки можна проаналізувати, шукати, переформатувати, її дані копіювати в електронну таблицю або завантажувати в базу даних. Веб-парсери зазвичай добувають щось зі сторінки, щоб використати це для інших цілей десь в іншому місці. Прикладом може бути пошук і копіювання імен і номерів телефонів, компаній та їх URL-адрес, або адрес електронної пошти до списку (збирання контактів).

Парсери веб-сайтів виконують роботу в кілька етапів [3]:

- пошук необхідної інформації в оригінальному вигляді: доступ до коду Інтернет-ресурсу, завантаження;
- добування значень з коду веб-сторінки, відокремлюючи потрібний матеріал від коду сторінки;
- формування звіту відповідно до встановлених вимог (запис інформації безпосередньо в бази даних, текстові файли).

Парсери мають ряд певних переваг при роботі з великими даними [4]:

- висока швидкість обробки (за хвилину кілька сотень / тисяч сторінок);
- великих обсягів даних;
- автоматизація процесу відбору (точний відбір і відокремлення необхідної інформації).

Сучасні рішення для веб-парсингу варіюються від тимчасових, що вимагають людських зусиль, до повністю автоматизованих систем, які здатні перетворювати цілі веб-сайти в структуровану інформацію з обмеженнями. В даній роботі було вирішено використати веб-парсинг для пошуку наукових видань України категорії А, Б. Це потрібно для перевірки на правильність введених даних користувачем: чи насправді дана наукова робота була опублікована в журналі. Для цього використовувався сайт <http://nfv.ukrintei.ua/> - реєстр наукових видань України.

IV. Реалізація хмарного сервісу

Для реалізації хмарного сервісу для ведення звітності з наукової діяльності було вибрано архітектуру «клієнт-сервер». Дана модель архітектури є поширеною серед програмних систем, які працюють з віддаленим доступом. Архітектура хмарного сервісу для ведення звітності з наукової діяльності складається з трьох основних модулів: рівень клієнта, рівень сервера та рівень даних. Клієнтська частина була реалізована засобами мови програмування TypeScript і фреймворку Angular. Для розробки клієнтської частини застосовано Visual Studio Code - це інтегроване середовище розробки, найпопулярніший інструмент розробки серед. Функції включають підтримку налагодження, виділення синтаксису, інтелектуальне завершення коду, фрагменти, рефакторинг коду та вбудований Git. Серверна частина реалізована за допомогою інструментів фреймворку ASP.NET Core. ASP.NET Core — це крос платформна платформа з відкритим вихідним кодом для створення сучасних хмарних додатків, підключених до Інтернету, таких як веб-сервіси, програми Інтернету речей та мобільні серверні програми. Для побудови серверного додатку використовується опіон-архітектура. Опіон-архітектура є формою багатопланової архітектури, і ми можемо уявити ці шари у вигляді концентричних кіл [5].

Клієнтська частина системи поділена на декілька основних модулів. Кожен з модулів відповідає за окремий URL шлях. AuthModule містить компоненти входу та реєстрації. TeacherModule містить вкладки публікацій, дисертацій, конференцій та наукової роботи студентів. Тут розміщуються лише ті дані, які створені викладачем. Компоненти DepartmentModule показують дані всіх викладачів вибраної кафедри. Модуль FacultyModule дозволяє переглянути дані всіх кафедр. Рис. 1 демонструє діаграму компонентів додатку.

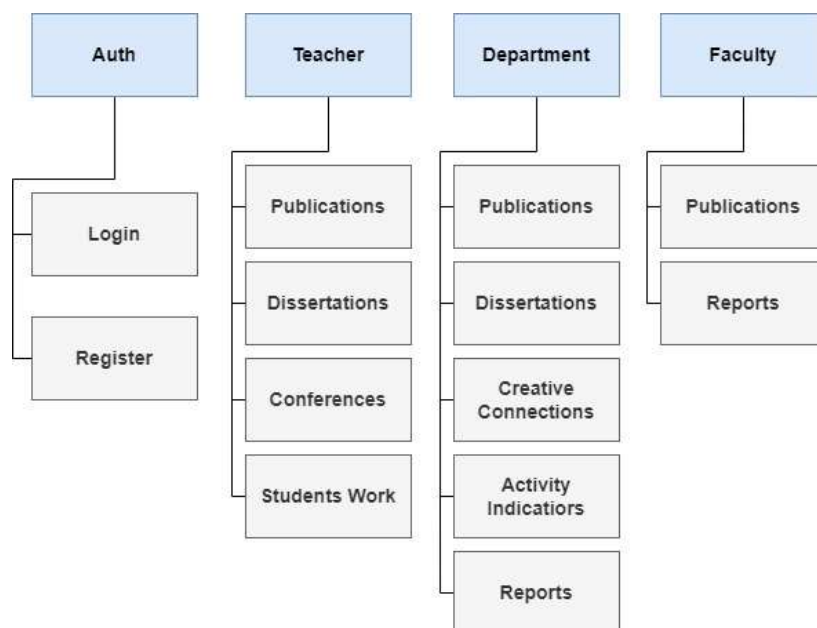


Рис. 1. Діаграма компонентів хмарного сервісу

Висновки

У даній роботі розглянуто питання необхідності автоматизованої підтримки діяльності підрозділу ЗВО. Описано архітектуру хмарного сервісу. Було обрано клієнт-серверну архітектуру для зв'язку між частинами системи. Для серверної частини було вибрано onion-архітектуру для вирішення проблем залежності модулів один від одного. Розміщено систему у хмарній платформі Microsoft Azure, яка добре поєднується з групою технологій .NET.

Список використаних джерел

1. Cloud Services [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://corporatefinanceinstitute.com/resources/knowledge/data-analysis/cloud-services>.
2. Sandeep Dhawan (2014): An Overview of Efficient Data Mining Techniques.
3. Mehmed K. (2011): Data Mining Concepts, Models, Methods, and Algorithms (Second Edition).
4. Shuyi Zheng, Ruihua Song, Ji-Rong Wen, and Di Wu. 2007. Joint optimization of wrapper generation and template detection. Association for Computing Machinery, New York, NY, USA, 894–902.
5. Distributed Application Architecture. Sun Microsystem. 2009.

УДК 004.02

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПОШУКУ ДОСТУПНИХ МАРШРУТІВ ДЛЯ ЛЮДЕЙ З ОСОБЛИВИМИ ПОТРЕБАМИ

Войтюк І.Ф.¹⁾, Бойчук Ю.Ю.²⁾

Західноукраїнський національний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

І. Опис проблеми

Мобільність, здатність самостійно пересуватися з одного місця в інше, є звичайною і рутинною діяльністю, яку виконують усі люди (наприклад, дім, лікарня, зупинка громадського транспорту чи місце для громади). Проте на сьогодні інфраструктура деяких українських міст не є сприятливою для людей з особливими потребами. Складність під час пересування по вулицях міста виникає через об'єкти, що перешкоджають досягненню кінцевих пунктів призначення. Бар'єрний простір перешкоджає руху не лише людям на візках, а й маломобільним групам населення – особам похилого віку, людям з травмами, мамам з візочками. Створення картографічного сервісу доступного середовища шляхом збору даних про

доступність дорожніх шляхів та надання відкритого доступу до інформації зможе частково розв'язати проблему повноцінного використання міських ресурсів людьми з обмеженими можливостями.

II. Мета роботи

Метою роботи є розробка математичного та програмного забезпечення для навігації в доступному міському середовищі для людей з особливими потребами, що дозволить планувати пересування з урахуванням обмежень людини.

III. Особливості математичної та програмної реалізації

Алгоритми пошуку маршруту для людей з особливими потребами складаються з трьох кроків:

- кількісна оцінка вартості сегмента маршруту;
- оцінка величин сегменту шляху;
- обчислення оптимального маршруту.

Програмний продукт розроблений у вигляді веб сервісу для пошуку доступних пішохідних маршрутів. Основними завданнями сервісу є: зображення інтерактивної карти [1]; надання користувачу можливості вибрати початкове місце та місце призначення; візуалізація знайденого маршруту на карті.

В якості вихідних даних сервіс використовує кластеризовані дані про перешкоди, шляхи користувачів та відкриті дані дорожньої мережі населеного пункту. Задача пошуку шляху вирішується за допомогою вдосконаленого алгоритму Дейкстри [2]. Він визначає найбільш доступні найкоротші шляхи від однієї вершини графу до іншої, що відповідають дорожній мережі. Кожна вершина графу характеризується чотирма величинами: довжина шляху з початкової точки до величини; довжина прохідних ділянок без перешкод; прохідність даних ділянок; критичність непрохідних ділянок.

На прохідність та критичність ділянок впливає доступність шляху дорожньої мережі, що визначається в наслідок оцінки можливого впливу об'єкту (точки доступності або перешкоди) на певний сегмент. При розрахунку доступності сегменту враховуються ступені серйозності перешкоди та доступності точки шляху. Пріоритетними для алгоритму будуть найкоротші шляхи серед шляхів з найменшою сумарною критичністю непрохідних ділянок, які мають найбільшу сумарну прохідність та найбільшу кількість ділянок, що проходять.

В якості основного засобу розробки програмного продукту вибрана платформа ASP.NET Core. Для завантаження даних дорожньої мережі населених пунктів було використано Overpass API – інструмент для вилучення даних із бази OSM на запит користувача.

Висновок

Розроблено алгоритм для обчислення оптимального маршруту дорожньої мережі з урахуванням заборони маршрутизації, що включає сегменти, які недоступні. Здійснено реалізацію програмного продукту для пошуку маршруту.

Список використаних джерел

1. Leaflet - a JavaScript library for interactive maps [Електронний ресурс]. Режим доступу - <https://leafletjs.com/>
2. Алгоритм Дейкстри – Вікіпедія [Електронний ресурс]. Режим доступу - https://uk.wikipedia.org/wiki/Алгоритм_Дейкстри

УДК 004.02

СИСТЕМА ОБМІНУ ПОВІДОМЛЕННЯМИ ДЛЯ ОС ANDROID

Беркищук В.Б.¹⁾, Шпінталь М.Я.²⁾

Західноукраїнський національний університет

¹⁾магістрант; ²⁾к.т.н., доцент

I. Постановка проблеми

Спільною проблемою більшості мобільних додатків є забезпечення інформаційної безпеки, на жаль, непереможних мобільних додатків не існує. Компанія, яка створила мобільний додаток повинна постійно інвестувати в його працездатність. Мова йде не тільки про функціонал, а й захист даних, як особистих так і платіжних. Для захисту додатку було застосовано авторизацію по номеру телефону та перевірку за допомогою “САРТСНА” (англ. «completely automated public turing test to tell computers and humans apart» — повністю автоматизований публічний тест Тюринга для розрізнення комп'ютерів і людей), що значно ускладнить крадіжку даних користувачів.

II. Мета роботи

Основна мета системи – це миттєвий обмін текстовими повідомленнями, фото, картинками, відео, документами з друзями, родичами, знайомими, колегами по роботі або по навчанню. Забезпечення високого рівня захисту даних користувачів.

Розроблений програмний продукт дозволить безпечний та швидкий обмін повідомленнями, при цьому забезпечуючи користувача позитивним досвідом через зручний та інтуїтивно-зрозумілий інтерфейс.

III. Проектування програмної системи

При проектуванні архітектури розроблюваної програмної системи було обрано клієнт-серверну архітектуру, оскільки саме вона найкраще підходить для вирішення зазначеної задачі. Для відображення відношення між акторами і прецедентами системи спроектовано діаграму варіантів використання. Актор представляє роль користувача в системі і на схемі відображається намальованим чоловічком. Прецеденти позначають дії, які виконує система і відображають до чого може призвести вибір актора. На схемі зображені в вигляді прямокутників.

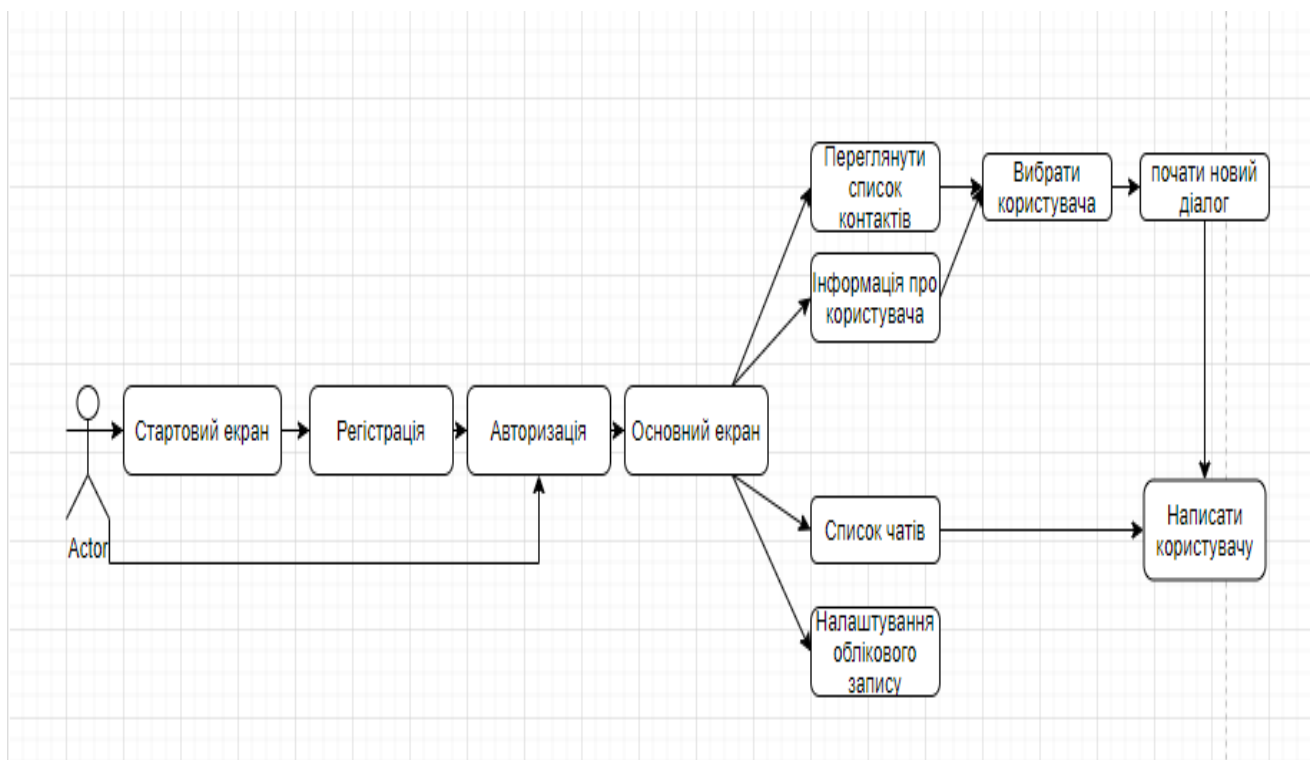


Рисунок 1 – Діаграма варіантів використання

Для користувача доступні наступні способи взаємодії з додатком:

Регістрація. Для користувачів, які не використовували додаток раніше, в першу чергу потрібно створити свій обліковий запис. Для цього достатньо ввести номер телефону, на який у подальшому прийде смс-повідомлення з шестизначним кодом. Номер телефону записується у базу даних і обліковий запис прив'язується до нього.

Авторизація. Зареєстрованим користувачам не потрібно нічого вводити, адже авторизація відбувається автоматично за номером телефону.

Створення нового діалогу. На основній сторінці відображений список всіх наявних діалогів. Для створення нового потрібно знайти користувача, з яким хочете почати діалог і написати йому.

Спілкування з іншими користувачами. В списку контактів потрібно вибрати користувача. Після чого стане доступне поле для написання і відправки нового повідомлення.

Пошук нових контактів. Відбувається за допомогою телефонної книги автоматично, якщо користувач авторизувався у додатку.

Інформація користувача. Буде доступна інформація інших користувачів, які є наявні у телефонній книзі а саме: статус їхньої активності, інформація та ім'я які користувач задав сам.

Налаштування облікового запису. Після авторизації користувач може онивити тло профілю, встановити користувацьке ім'я та ввести короткі данні про себе.

За допомогою діаграми варіантів на рисунку 1, проілюстровані всі набори дій для кожного варіанта системи при діалозі з користувачем.

Висновок

У даній роботі розроблено програмне забезпечення системи обміну повідомленнями для Android OS, яке дозволяє безпечно листування між користувачами із зручним інтерфейсом, що в свою чергу покращить досвід користування системою.

Список використаних джерел

1. w3schools.com – the world's largest development site. [Електронний ресурс]. — Режим доступу: <http://w3schools.com> — Назва з екрану. W3Schools – Educate Yourself.
2. Developer.android.com [Електронний ресурс]: офіційна документація для розробників андроїд додатків. Режим доступу: <https://developer.android.com/docs> — Назва з екрану. Documentaation for app developers.
3. uk.wikipedia.org – Вільна енциклопедія [Електронний ресурс]. – Режим доступу: <http://uk.wikipedia.org> — Назва з екрану. Вікіпедія. Вільна енциклопедія.

УДК 004.02

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ МОДЕЛЮВАННЯ ГАЗООБМІНУ В ДИХАЛЬНИХ ШЛЯХАХ ЛЮДИНИ

Войтюк І.Ф.¹⁾, Панасюк К. А.²⁾

Західноукраїнський національний університет

¹⁾ к.т.н., доцент; ²⁾ магістрант

I. Опис проблеми

Моделювання процесів газообміну в дихальних шляхах людини під впливом гіпоксії може поліпшувати розуміння таких складних взаємозв'язків і забезпечити ефективний кількісний інструмент для аналізу та синтезу функціонування кардіо-легеневої системи. Дихальна недостатність – це стан, при якому порушення функції дихальної системи погіршують газообмін у легенях та призводять до гіпоксемії (зниження парціального тиску кисню у артеріальній крові) або гіперкапнії (підвищення парціального тиску вуглекислого газу) [1].

За останні кілька десятиліть завдяки прогресу в обчислювальних технологіях, наукова активність у цій галузі помітно зросла, і складність моделювання виросла до більш точного опису фізіологічних механізмів та їх динаміки. Однак більшість з цих моделей в основному або серцево-судинні, або легеневі і, отже, не дозволяють провести всебічне дослідження серцево-легеневої відповіді на різні гострі стани. У зв'язку з цими проблемами є необхідність у розробці системи моделювання процесів газообміну в дихальних шляхах людини в умовах гіпоксичного впливу та створенні програмного забезпечення, яке б за мінімумом загальнодоступної інформації дозволяло швидко та якісно оцінити стан дихальної системи людини.

Здійснено постановку задачі моделювання процесів газообміну в дихальних шляхах людини під впливом гіпоксії, що представляє собою розроблений механізм постійного спостереження за показниками кардіо-легеневої системи для оперативного виявлення недоліків та погіршення станів людини.

II. Мета роботи

Мета дослідження полягає в розробці математичного та програмного забезпечення, який дозволить моделювати процеси газообміну дихальної системи людини.

III. Особливості математичної та програмної реалізації

Огляд існуючих програмних засобів дозволив внести певні зміни до класичних підходів розв'язку даної задачі. А саме досліджено параметри, що описують модель газообміну та транспортування, які можна розділити на параметри, що відносяться до моделі газообміну в легенях, до моделі газообміну в тканинах, а також переносу крові [2]. Для відтворення реальних імітаційних обсягів легень, які спостерігаються в стані спокою людини, були внесені додаткові параметри моделі газообміну в дихальних шляхах людини. Для розв'язку задачі було прийнято використовувати диференціальні рівняння та методи їх вирішення. Для побудови складної математичної моделі, у програмній системі компоненти повинні вміти динамічно “обробляти” події і обмінюватися інформацією про свої властивості і уміння з іншими компонентами. MongoDB – найпопулярніша document-store технологія і сама використовувана NoSQL база даних, перевагою якої є наявність такої функціональності як реплікація і sharding [3].

Висновок

В якості основного засобу розробки програмного продукту для моделювання газообміну в дихальних шляхах використано платформу Node.js, документоорієнтовану систему управління базами даних MongoDB, яка не вимагає опису схеми таблиць, використовує JSON-подібні документи та схему бази даних.

Список використаних джерел

1. <https://empendium.com/ua/chapter/B27.II.3.1>.
2. Шулагін Ю.А., Дьяченко О.І., Єрмолаєв Е.С., Гончаров А.О. Розробка методу оцінки чутливості дихання людини до вуглекислого газу для застосування в гравітаційної фізіології // Технології живих систем, 2012. Том 9, №10, стр.14-22.
3. Eugene S. Ermolaev, Alexander I. Dyachenko, Yury A. Shulagin, Alexander O. Goncharov, Artem V. Demin. Effect of head-down human body position on chemoreflex control of Breathing // M. Long (Ed.): World Congress on Medical Physics and Biomedical Engineering, IFMBE Proceedings, vol.39, pp.2068-2071, 2012. ISSN: 1680-0737.
4. <https://dou.ua/forums/topic/33453/>

УДК 004.056.55

ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ МОДУЛЯ ШИФРУВАННЯ ДАНИХ МЕСЕНДЖЕРА

Шаповал Д.А.

Західноукраїнський національний університет, магістрант

I. Постановка проблеми

Обмін повідомленнями – важливий аспект сучасного життя. Ще з давніх часів люди передавали один одному інформацію різними способами. Писали листи, малювали власноруч карти, схеми тощо. І насправді вчасне отримання інформаційних даних (новин, оголошень, подій) дуже важливе та відіграє велику роль, наприклад при терміновій евакуації.

Наразі існує дуже багато методів обміну повідомленнями та дзвінками, але безпечність таких сервісів частіше бажає кращого. Оскільки приватні бесіди та чати не є стрічкою новин і не кожному адресовані, важливо бути впевненим, що ваші дані в безпеці і надходять тільки адресанту.

II. Мета роботи

Метою дослідження є розробка програмного забезпечення для модуля шифрування даних месенджера.

III. Обґрунтування отриманих результатів

Запропонований у роботі додаток [1-4] має такі функції: заміна наявних ключів для листування новими; можливість копіювання ключа з'єднання; підтвердження з'єднання повідомленням; створення прямого від браузера до браузера з'єднання (Peer-to-peer); перехід до листування; підтвердження входу повідомленням; створення текстового повідомлення; надсилання повідомлення іншому абоненту в чаті; надсилання запиту на відео дзвінок; можливість; відповісти на запит; завершення розмови; вихід з екрану відео дзвінка; вихід з чату.

На рисунку 1 наведено діаграму варіантів використання для розроблюваного програмного забезпечення.

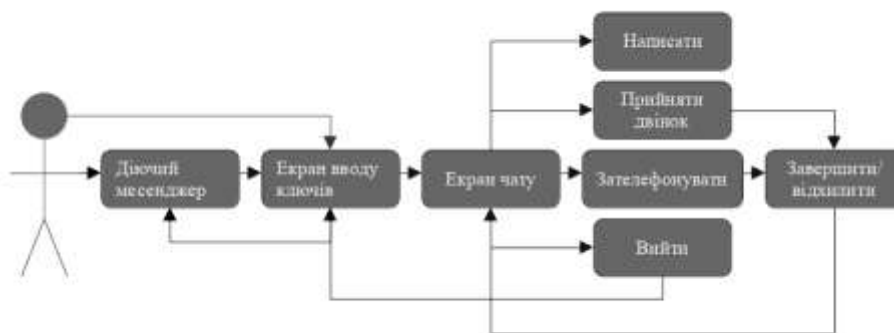


Рис. 1 Діаграма варіантів використання

Дана система рекомендується як додатковий рівень безпеки, загальним її призначенням є вбудованість в інші месенджери, за допомогою ботів отримання першої сторінки та надсилання ключа через будь-який месенджер співрозмовнику, з яким хочете розпочати безпечний чат.

Висновок

Систему рекомендовано до використання користувачам, які вагаються щодо безпеки їхніх повсякденних листувань в тому чи іншому месенджері.

Список використаної літератури

1. Чи варто вчити JavaScript [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://ru.hexlet.io/blog/posts/stoit-li-uchit-javascript-perspektivy-situatsiya-na-rynke-truda-mneniya-ekspertov>.
2. Навіщо вивчати Node.js [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://ru.hexlet.io/blog/posts/zachem-izuchat-node-js-ili-o-perspektivah-bekenda-na-javascript>.
3. Node.js® — це JavaScript-оточення побудоване на JavaScript-рушієві Chrome V8 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://nodejs.org/uk/>
4. Simple-peer [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: <https://www.npmjs.com/package/simple-peer>

УДК 004.056.5

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ МОДЕЛЮВАННЯ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ У СОЦІАЛЬНИХ МЕРЕЖАХ

Фатюк В.І., Міклашевський Е.Р., Сушко С.С., Гарчар Р.Р.

Західноукраїнський національний університет, магістранти

I. Постановка проблеми

Моделювання та прогнозування інформаційних процесів у цифрових соціальних мережах дозволяє досліджувати, виявляти особливості перебігу поширення інформації в даний момент і на фіксований час вперед, виявляти незвичайні або підозрілі дії (поширення свідомо неправдивої інформації, зламування окремих сторінок). Крім цього, моделювання дозволяє досліджувати можливі сценарії розвитку подій, коли немає відомих даних щодо поведінки користувачів у таких сценаріях. Тим не менш, варто відзначити складність реалістичного відтворення таких процесів, оскільки

необхідно враховувати велику кількість факторів, тому що люди діють і взаємодіють один з одним різними методами. Необхідно враховувати, що поведінка людей змінюється з плином часу та відрізняється в різних групах користувачів. Це ускладнює передбачення реакцій окремої людини або цілої спільноти навіть на невелику зміну в своєму середовищі. Як наслідок, стають потрібними нові підходи до підтримки прийняття рішень, засновані на моніторингу, прогнозуванні та оптимізації як самих інформаційних процесів, і результатів їх взаємодії з використанням передбачуваного моделювання.

Таким чином, для дослідження процесів у соціальних мережах необхідно розробляти методи моделювання та прогнозування поширення інформації на різних рівнях. Різні рівні взаємопов'язані, визначають багатомасштабність і представлені на мікро-масштабі окремими користувачами, що створюють та сприймають інформаційні повідомлення, на мезо-масштабі – комплексними мережами, що задають топологію можливих шляхів передачі інформації, на макромасштабі – динамікою популярності рівня онлайн-спільнот.

Незважаючи на те, що у реальному світі існує велика кількість процесів представлених у вигляді мереж, можливості їх спостереження та аналізу загалом обмежені через швидкості розрахунків на великих мережах. Більше того, швидкість розрахунку прогнозу протікання процесів є критичною, тому що інформаційні процеси мають високу швидкість поширення та згасання, і високий ризик старіння прогнозу. Щоб отримати результат прогнозу для мереж із мільйонами вузлів за час, поки прогноз не втратить свою актуальність, необхідно розробляти обчислювально ефективні алгоритми, зокрема багатопроекторних обчислювальних систем. Можливість запуску процесу моделювання в паралельному режимі може скоротити час отримання результатів.

II. Мета роботи

Метою дослідження є розробка методів та алгоритмів для моделювання та прогнозування характерних сценаріїв для різних масштабів соціальних мереж.

III. Метод моделювання інформаційних процесів в соціальних мережах

Об'єктом моделювання є мережі тематичних онлайн-спільнот (рисунки 2.1). Під тематичною спільнотою розуміється контекст кіберпростору, що генерує інформаційні повідомлення на задану тематику (набір тематик). При цьому набору тематик у соціальній мережі може відповідати кілька спільнот з аудиторією, що частково перетинається.

Спільнота – це інформаційний канал, керований одним чи кількома власниками (користувачами системи). Спільнота містить кілька контекстів поширення інформаційних повідомлень (стіна, повідомлення, коментарі, обговорення). З точки зору мережі, вона є вузол мережі (джерелом інформації), що транслює повідомлення користувачів.

Користувач є вузлом мережі сприйнятливим для інформації. Соціальні системи, зокрема, тематичні онлайн-спільноти представлені у вигляді мережі, де вузли замінюються агентами. На відміну від вузла агент (актор) – це елемент, що має поведінку, тобто можливість здійснювати в рамках системи різні дії. У цілому нині такий підхід реалізує парадигму комплексні агентні мережі. Подання мережі як статичної комбінації вузлів та ребер дозволяє задати структуру та атрибути елементів системи.

Однак такого представлення не достатньо, коли топологія та властивості вузлів мережі схильні до змін. Для дослідження таких процесів використовують темпоральні (динамічні) агентні мережі. У випадку темпоральну мережу можна зобразити як набір графів (1):

$$G = \{G(1), \dots, G(t), \dots, G(T)\} = \{(G(t))\}_{t=1\dots T} \quad (1)$$

де $G(t) = \langle V(t), E(t) \rangle$ – граф заданий набором вершин $V(t)$ та ребер $E(t)$ в момент часу t

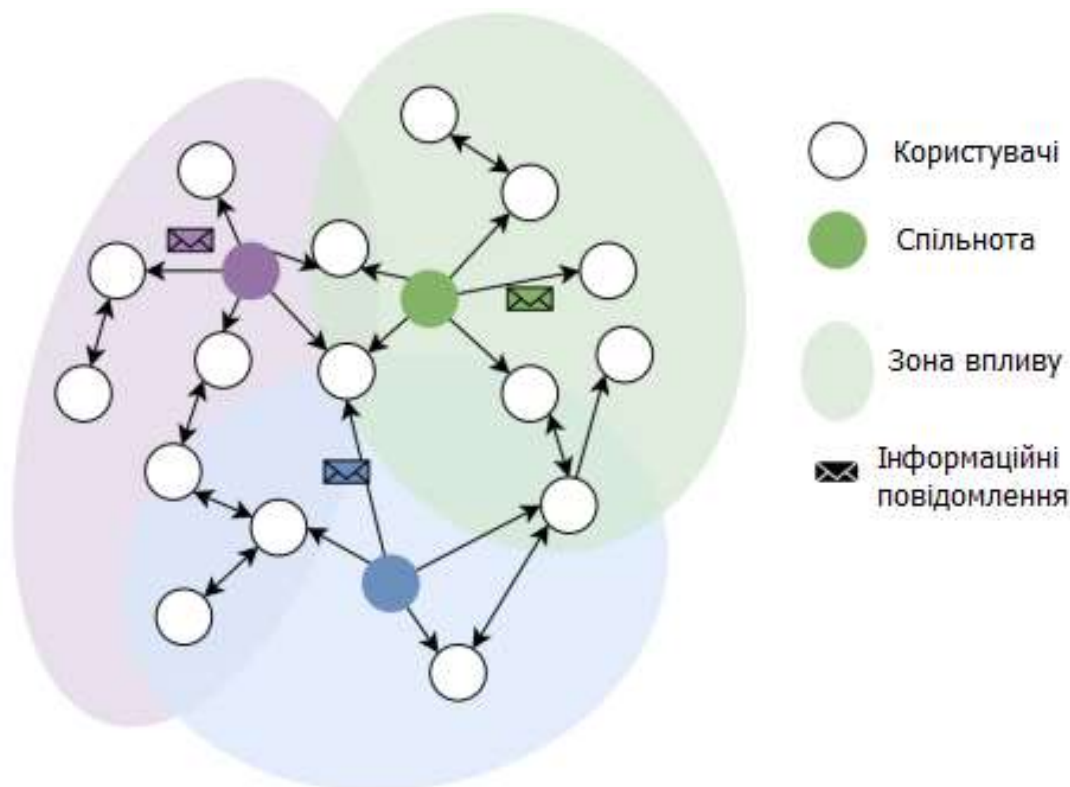


Рисунок 1 – Схема мережі тематичних онлайн-спільнот

Для моделювання та прогнозування інформаційних процесів було реалізовано агентну модель поширення інформації в мережі. Модель дискретна: кожна ітерація відповідає певному моменту часу моделі. Джерелом та приймачем інформації можуть бути агенти, що мають як однаковий, так і різні типи. Наприклад, фізична особа може бути джерелом, коли публікує лайк до посту на стіні спільноти, а тоді спільнота є приймачем (бо також може реагувати з використанням механізму модерации).

В результаті використання інтернету та соціальних мереж, зокрема, користувачі залишають набір дій, повідомлень та інших активностей, що простежуються, які можуть бути використані для ідентифікації розглянутих вище моделей різного масштабу. Такі цифрові сліди окремих користувачів найчастіше містять невелику кількість подій, тому виникає потреба агрегування даних окремих користувачів.

Висновок

Розроблено метод багатомасштабного моделювання динаміки поширення інформаційних повідомлень у цифрових соціальних мережах на основі взаємопов'язаних моделей інформаційних процесів у мережах тематичних онлайн-спільнот з використанням комплексних агентних мереж.

Список використаних джерел

1. Buasilovsky, P. Adaptive and intelligent Web-based educational systems / P. Buasilovsky, C Peylo // International Journal of Artificial Intelligence in Education. Special Issue on Adaptive and Intelligent Web-based Educational Systems -2003. -№13 (2-4). -P. 159-172.
2. Murray, T. Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art / T. Murray. // International Journal of Artificial Intelligence in Education. -1999.-№10-P. 98-129
3. De Meo, P.; Ferrara, E.; Fiumara, G.; Proveti, A. Mixing local and global information for community detection in large networks. J. Comput. Syst. Sci. 2014, 80, 72–87.
4. Karimi, N.; Kondrood, R.R.; Alizadeh, T. An intelligent system for quality measurement of Golden Bleached raisins using two comparative machine learning algorithms. Measurement 2017, 107, 68–76.

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ПРОЦЕСУ РОЗПОДІЛУ РЕСУРСІВ В ХМАРНИХ ОБЧИСЛЕННЯХ

Гончар Л.І.¹⁾, Порученко В.Ю.²⁾, Мельниченко В.В.³⁾, Лукьянчук В.А.⁴⁾

Західноукраїнський національний університет

¹⁾ к.е.н., доцент; ²⁻⁴⁾ магістрант

І. Постановка проблеми

В даний час хмарні обчислення є новою концепцією, яка ефективно поєднує багато сфер обчислювальної техніки. Хмарні обчислення пропонують користувачам кілька онлайн-сервісів таких, як зберігання, потужність процесів, бази даних, віртуальні сервери з мінімальними витратами та мінімальним споживанням енергії.

Користувачі можуть легко отримати доступ до цих послуг з будь-якого місця та будь-який час через інтернет. Користувачі платять лише за ресурси, які вони використовували. Хмарні обчислення зберігають дані та розподілені ресурси відкритому середовищі і обсяг зберігання даних дуже швидко збільшується. У зв'язку зі збільшенням кількості додатків, навантаження на хмару також збільшується. Балансування навантаження є основною проблемою серед хмарних мереж.

Основною метою балансування навантаження є ефективне використання ресурсів та підвищення продуктивності. Поряд із цим вона видаляє вузли, які містять велике навантаження, а також вузли, які не працюють належним чином чи виконують невелике завдання.

У більшості алгоритмів розподілу навантаження задачі розподіляються за віртуальним машинам до призначення хостам. Тому більшість методів балансування навантаження розробляються таким чином, що використовуються два послідовних рівня.

II. Мета роботи

Мета дослідження полягає у підвищенні якості балансування навантаження у хмарних системах, прискоренні часу виконання завдань та скорочення витрат обробки на основі розробки математичних моделей та алгоритмів оптимізації розподілу ресурсів та планування завдань додатків у реальному часі.

III. Програмна реалізація системи

Для реалізації нашої моделі було внесено кілька змін до класів симулятора CloudAnalyst та створено комплексне програмне забезпечення процесу балансування навантаження, щоб настроїти їх конфігурацію для зазначеної проблеми.

Тому існуючі класи були змінені та додані нові класи для розподілу ресурсів та планування завдань на віртуальні машини. Програмна реалізація модулів комплексного програмного забезпечення балансування навантаження створене за допомогою об'єктно-орієнтованої мови програмування JAVA, що працює на платформі NetBeansIDE 8.2.

Платформа NetBeans надає надійну та гнучку архітектуру додатків. Програма не обов'язково має виглядати так само, як IDE. Це допоможе нам заощадити час, що витрачається на розробку. Платформа NetBeans надає перевірену часом архітектуру безкоштовно. Ця архітектура допоможе забезпечити надійність процесів розробки. Оскільки архітектура платформи NetBeans є модульною, створювати надійні та додатки, що розширюються, буде дуже просто.

Регіон: комплексна програма поділить весь світ на шість регіонів. Ці регіони співпадають із шістьма основними континентами світу. Центри обробки даних та бази користувачів, які є іншими основними об'єктами, належать до деяких із цих регіонів. Ця екологічно обґрунтована категоризація дуже корисна для підтримки рівня реальної та простої моделі для моделювання з великою тривалістю.

Інтернет-абстрактне представлення інтернету в реальному світі. Він моделює шлях передачі даних через мережу і додає затримку через дроти передачі. Користувач може настроїти затримку передачі та всю існуючу смугу пропускання серед шести регіонів.

Брокер обслуговування: керує всім трафіком, який переміщується між центрами обробки даних та базами користувачів. Черезброкер обслуговування приймається рішення про обслуговування центру обробки даних для кожної бази користувачів. В даний час Cloudanalyst надає три типи брокерів обслуговування, всі ці три типи реалізують різні політики маршрутизації.

База користувачів: представляє велику групу користувачів, які потрапляють у схожу географічну область та розглядаються як один користувач. У нашій комплексній програмі для ефективності моделювання усі користувачі в одному регіоні розглядаються як єдина база користувачів.

База користувачів відповідає за створення потоків даних та завдань, які необхідно виконати, і відправляє їх у центри обробки даних через Інтернет. Завдання - колекція всіх запитів користувачів та обробки у вигляді одного запиту. Це групування корисне для представлення великої кількості запитів та надсилання їх в одному пакеті. Кожне завдання має низку параметрів, таких як номер завдання, розмір завдання, ідентифікатор користувача, якому належить. Графічний інтерфейс користувача (ГІ): реалізований у вигляді набору екранів, які дозволяють користувачеві докладно визначити характеристики центру обробки даних, включаючи детальну специфікацію обладнання фірми серверів; визначити бази користувачів та їх характеристики, такі як кількість користувачів, години пікового та непікового навантаження та частота генерації трафіку; переглянути та зберегти результати моделювання з графічними висновками.



Рисунок 1 – Головне вікно програми

Розподіл віртуальних машин включає пошук підходящих віртуальних машин для хостів, щоб хмарні ресурси ефективно використовувалися для зниження навантаження на систему та збільшення прибутку.

Оптимізація розподілу віртуальних машин потребує розробки методології міграції віртуальних машин, що розподіляє навантаження між хостами, щоб задовольнити цілі балансування навантаження та зменшити витрати шляхом відключення хостів, що не використовуються.

Висновок

Запропоновано структуру програмного комплексу розподілу ресурсів на основі моделювання процесів хмарних обчислень, контролю мережевих змінних та центрів обробки даних.

Отримані результати можуть знайти застосування у сфері управління системами зберігання даних та системами розподіленої обробки інформації. Напрями подальших досліджень можуть полягати у розробці нових підходів до дослідження станів навантажень вузлів системи з використанням кількох центрів обробки даних, аналізу можливості реалізації розподілу в реальному часі при використанні запропонованих алгоритмів.

Список використаних джерел

1. De Meo, P.; Ferrara, E.; Fiumara, G.; Provetti, A. Mixing local and global information for community detection in large networks. *J. Comput. Syst. Sci.* 2014, 80, 72–87.
2. Karimi, N.; Kondrood, R.R.; Alizadeh, T. An intelligent system for quality measurement of Golden Bleached raisins using two comparative machine learning algorithms. *Measurement* 2017, 107, 68–76.

УДК 004.056.5

МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ СИСТЕМИ ФОРМУВАННЯ ПОРТФЕЛЯ ЦІННИХ ПАПЕРІВ В УМОВАХ ОБМЕЖЕНОЇ ВИБІРКИ

Магера С.Р.¹⁾, Костик Б.П.²⁾, Романюк М.С.³⁾, Кушнірчук Р.М.⁴⁾

Західноукраїнський національний університет

¹⁾ магістрант; ²⁾ аспірант; ³⁾ магістрант; ⁴⁾ магістрант;

І. Постановка проблеми

Фінансовий ринок – складна система, що вимагає для свого вдосконалення прийняття множини рішень, що стосуються всіх її складових. В даний час щодо фінансового ринку діє ряд стратегій, прийнятих у протягом останніх 10 років, метою яких є оздоровлення фінансової сектори країни. Введення нових регуляторних вимог у частині інвестиційної діяльності банків та інших фінансових інститутів, а також оптимізація регулятивного та податкового навантаження на учасників фінансового ринку призвели до підвищення конкурентоспроможності цього сектора підвищення доступності фінансових послуг для суб'єктів економіки, особливо приватних інвесторів.

В умовах активної державної політики щодо розвитку фінансового сектора в країні, необхідність у розробці нових та модифікації старих математичних моделей, що використовуються для формування та управління інвестиційним портфелем цінних паперів з урахуванням специфіки внутрішнього ринку, є однією з актуальних проблем, що стоять перед дослідниками.

II. Мета роботи

Метою дослідження є полягає у покращенні якості процедури формування інвестиційного портфеля та підбору моделі динаміки базових активів шляхом розробки нової математичної моделі та відповідного програмного забезпечення..

III. Програмна реалізація системи

Комплекс програм реалізований мовою програмування Python 3.6 в середовищі розробки Spyder. Для створення інтерфейсу був використаний набір розширень графічного фреймворку Qt для мови програмування Python, виконаний як розширення Python.

На рисунку 1 зображено головне вікно програми.

Для ручного введення даних необхідно встановити прапорець «Вручну» та вибрати існуючі файли формату .csv або .txt за допомогою натискання кнопки "Вибрати файли", позначеної під номером 1 на рисунку 3.6.

Для завантаження даних із зовнішніх джерел необхідно встановити флажок «З сайту» та заповнити необхідну інформацію. Наприклад, для завантаження даних про вартість акцій компанії Нафтогаз із сайту необхідно вибрати період, що цікавить, вказати ринок «Український фондовий ринок», клієнта і частоту даних, що цікавить (від точкових даних до місячних). Додаток також дозволяє за необхідності відразу заповнювати пропуски даних кількома способами, а саме:

1. Заповнення попереднім значенням;
2. Заповнення наступним значенням;
3. Заповнення за допомогою лінійної інтерполяції.

Після чого дані можна використовувати для подальшого аналізу або зберегти на диск.

У розділі, позначеному цифрою 3 рисунку 3.6, проводиться підбір одновимірного розподілу ймовірностей. Якщо на вхід подано більше одного часового ряду, то розподіл підбирається до кожного. В фінансовому аналізі аналізують збільшення цін активів (або прибутковості), то додатком також передбачена опція розрахунку дохідностей безпосередньо перед підбором розподілу. Підтримуються

три види дохідностей: абсолютні (або прирощення), відносні та логарифмічні. Для зручності роботи з даними, а також для можливості візуально порівнювати щільності розподілу дохідностей різних активів передбачено опцію нормування даних. Процедура підбору розподілу включає оцінку параметрів кожного аналізованого розподілу, перевірку статистичних гіпотез, розрахунок метрики L2, а також візуалізацію густини розподілу.

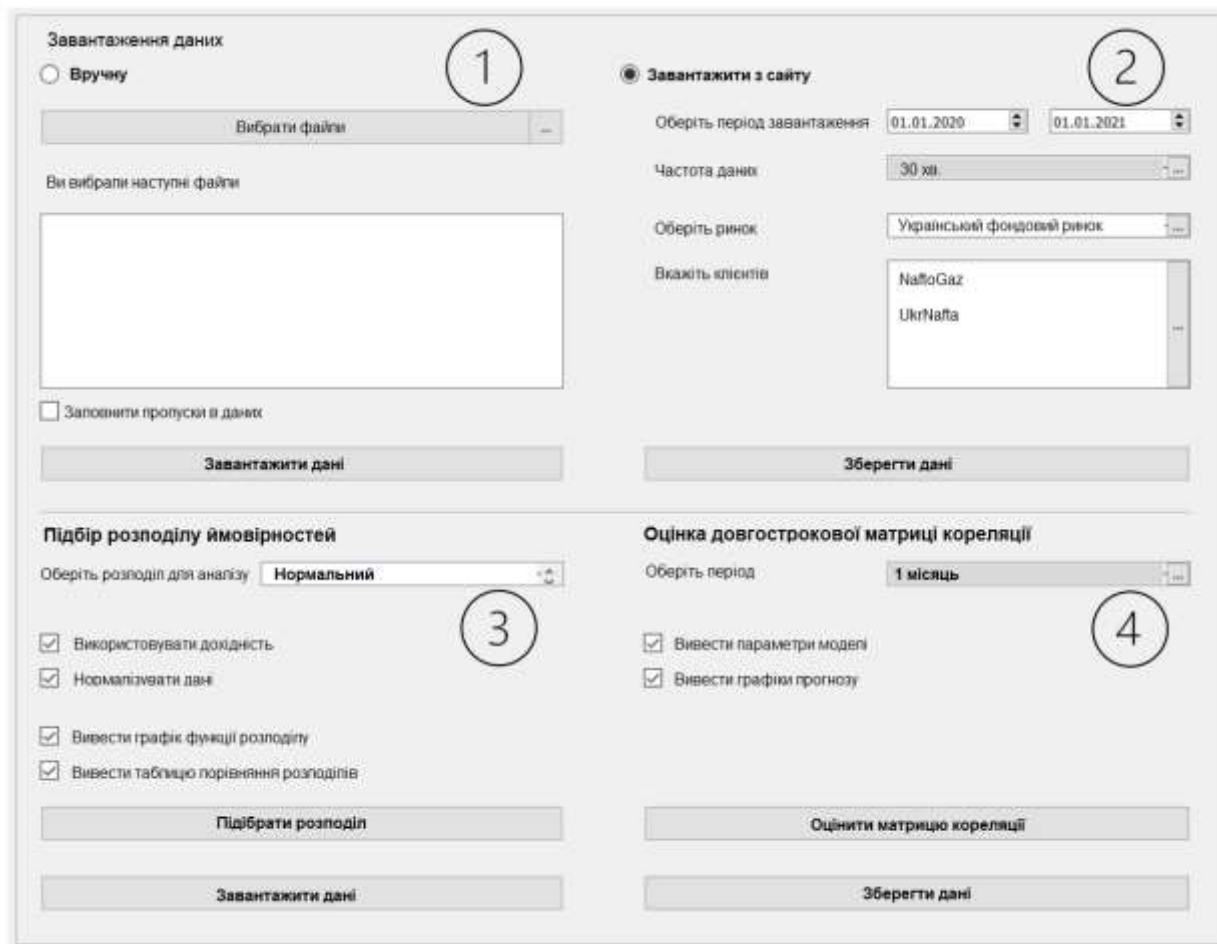


Рисунок 1 – Головне вікно програми

Ключовою перевагою розробленого програмного комплексу на відміну від існуючих альтернатив є можливість підбору спільного ймовірнісного розподілу до даних, а також:

- Вільний доступ та можливість завантажувати дані безпосередньо з сайтів провайдерів;
- Простий і зрозумілий інтерфейс користувача;
- Наявність норми L2 як метрики порівняння розподілів під час добору та можливість розширення кількості підтримуваних розподілів;
- Можливість моделювання кореляції інтервальних часових рядів.

Висновок

Проведено апробацію розробленої моделі кореляції інтервальних часових рядів на історичних даних українського фондового ринку, а саме було сформовано три оптимальні портфелі відповідно до теорії Марковіца з використанням матриць кореляцій, отриманих трьома можливими способами, у тому числі за допомогою запропонованої моделі.

Як апробація методики оцінки впливу порушень припущень моделі динаміки базових активів на процес її побудови і на одержувані за її допомогою цільові показники портфеля була розглянуто практичне завдання оцінки метрики VaR портфеля параметричним способом.

Список використаних джерел

1. Buasilovsky, P. Adaptive and intelligent Web-based educational systems / P. Buasilovsky, C Peylo // International Journal of Artificial Intelligence in Education. Special Issue on Adaptive and Intelligent Web-based Educational Systems -2003. -№13 (2-4). -P. 159-172.

- Murray, T. Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art / T. Murray. // International Journal of Artificial Intelligence in Education. -1999.-№10-Р. 98-129
- De Meo, P.; Ferrara, E.; Fiumara, G.; Provetti, A. Mixing local and global information for community detection in large networks. J. Comput. Syst. Sci. 2014, 80, 72–87.
- Karimi, N.; Kondrood, R.R.; Alizadeh, T. An intelligent system for quality measurement of Golden Bleached raisins using two comparative machine learning algorithms. Measurement 2017, 107, 68–76.

УДК 004.056.5

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ВИЯВЛЕННЯ НЕДОСТОВІРНОЇ ІНФОРМАЦІЇ НА ВЕБ-РЕСУРСАХ

Сидорчук І.А.¹⁾, Пасічник М.П.²⁾, Карпишин Ю.В.³⁾, Костик Б.П.⁴⁾

Західноукраїнський національний університет

¹⁻³⁾ магістрант; ⁴⁾ аспірант

І. Постановка проблеми

Достовірність інформації – властивість інформації, яка показує ступінь відповідності знань про об'єкт реальному стану об'єкту. Оцінка достовірності перевіряється за наступними параметрами: наявність підтвердження з інших джерел; довіра до джерела даних; стикування з іншою інформацією; відсутність прихованих помилок. Своєчасність – здатність інформації відповідати запитам споживача в потрібний момент часу. Оцінка достовірності інформації відіграє дуже важливу роль і є актуальним напрямком наукових досліджень.

II. Мета роботи

Мета дослідження полягає у створенні системи оцінки достовірності та актуальності інформації, яка отримана із соціальних мереж.

III. Програмна реалізація системи

На клієнті застосований Angular, на бекенді(сервер) використано .Net Core (Web API Core), для реалізації БД - MS SQL. На стороні користувача було задіяно Angular (JavaScript фреймворк). До цього проекту не працював із даним фреймворком, проте швидко оволодів ним, сподобалося те, що він використовує TypeScript, а для мене його переваги над JavaScript були відчутні, одні з них:

- Можливість явного визначення типів(статична типізація);
- Підтримка використання повноцінних класів (як в традиційних об'єктно орієнтованих мовах);
- Підтримка підключення різноманітних модулів та можливість створення власних та формування логічних одиниць в окремі підсистеми незалежні від інших;

Крім цього, підтримка RxJS, та й самий стиль побудови компонентів припав до душі. Звичайно виникали і багато питань, однак спільнота Angular досить велика і є в кого, що та й запитати, для мене це теж плюс. Ще одна очевидна перевага – WebPack ("бандлер модулів", він приймає модулі JavaScript, аналізує їх залежності один від одного, а потім з'єднує їх найефективнішим способом, випускаючи у кінці лише один JavaScript файл, більше того Webpack розуміє, що модуль JavaScript може потребувати CSS файл, а цей CSS файл може потребувати зображення, результат роботи Webpack буде містити тільки те, що потрібно), котрий вже відразу вбудований в Angular.

Звісно ж на клієнтській частині використано багато сторонніх бібліотек, проте, на мою думку, такі деталі несуттєві.

На стороні сервера було задіяно платформу .Net Core, додаток створений на основі Web API Core із архітектурним підходом REST(Representational State Transfer, підхід до архітектури мережових протоколів, які забезпечують доступ до інформаційних ресурсів), також були задіяні патерни UnitOfWork та Repository, здебільшого для маніпуляції із даними. В основному на сервері додаток поділений на 2 підпроекти:

- Business logic (Бізнес логіка).
- DAL (Data Access Layer, рівень доступу до даних).

На рівні доступу до даних, визначено головні моделі моєї програми, які забезпечують роботу із БД. Для створення та зв'язку БД із нашими сутностями в додатку було використано Entity Framework Core, використав підхід Code First, котрий спершу дозволяє із легкістю оголосити (створити, як класи) головні сутності програми, а згодом завдяки міграціям побудувати базу даних.

На рівні бізнес логіки оголошені головні сервіси для маніпулювання із даними котрі приходять від бази даних або від користувача із клієнтської частини. Більше того саме тут розміщені сервіси, які комунікують із зовнішніми апі. Для цього вони були вбудовані в залежності, через інтерфейсні конектори, які забезпечуть безболісну заміну при необхідності у майбутньому.

Хотів би окремо виділити .Net Core, технологія невпинно росте, власне в open-source напрямі, завдяки, цьому розробники нового функціоналу прислухаються до користувачів, та й кожен може стати автором чогось нового. При використанні даної технології можу виділити наступні позитивні аспекти, які стали несподіванкою для мене, а саме кросплатформеність та наявність підключення сторонніх пакетів через nuget manager. Завдяки цьому доволі багато пакетів було підключено до мого застосунку, для прикладу головні з них:

HtmlAgilityPack для парсінгу даних контенту сайту,
 Microsoft.AspNetCore.Cryptography.KeyDerivation для криптування паролів користувачів,
 Microsoft.EntityFrameworkCore – для мапінгу та підняття БД.

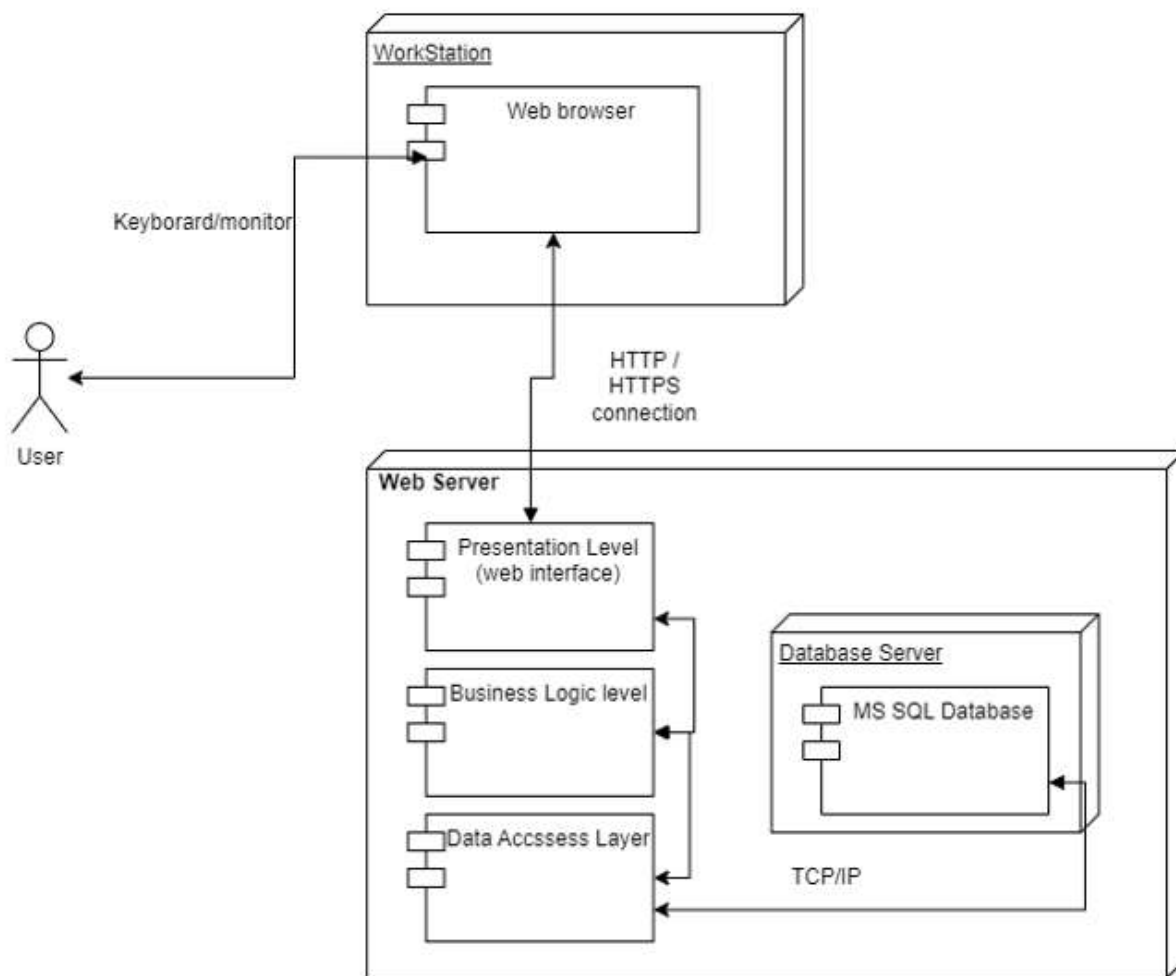


Рисунок 1 – Діаграма компонентів системи

У рамках цієї роботи створено новий додаток, який спрощує облік фейкових вебсайтів та дає змогу отримувати детальну інформацію про вебсайт. Користувачі з легкістю зможуть переглядати дані про недостовірні вебсайти та бачити докази(пости), більше того вони мають змогу додавати власноруч дані, котрі згодом будуть перевірені адміністратором.

З іншого боку можна виділити функціонал, який повертає детальну інформацію про веб сайт та будує на основі цього графіки, таблиці, які несуть інформативний характер, посилаючись на отримані джерела.

Висновок

У даній системі було обрано Agile модель розробки програмного забезпечення та методологію Scrum. Отже, завдяки детальній проробці кожного етапу, було створено програмний продукт, якому поки немає аналогів. Даний продукт із легкістю можна буде розширювати у майбутньому.

Список використаних джерел

1. Dyvak, M. P., Kovbasisty, A. V., Melnyk, A. M., Turchyn, L. Y., & Martsenyuk Y. O. (2018). System for web resources content structuring and recognizing with the machine learning elements. *Radio Electronics, Computer Science, Control*, (3). <https://doi.org/10.15588/1607-3274-2018-3-14>
2. Дивак, М.П., Мельник, А.М., Ковбасистий, А.В. Папа, О.А. 2020. Підхід до математичного моделювання ефективності web-ресурсів. *Оптико-електронні інформаційно-енергетичні технології*. 38, 2 (Бер 2020), 29–37. DOI:<https://doi.org/10.31649/1681-7893-2019-38-2-29-37>.

УДК 004.056.5

МАТЕМАТИЧНЕ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ АНАЛІЗУ КОДІВ ПРОГРАМ НА ОСНОВІ СЕМАНТИЧНИХ МОДЕЛЕЙ

Тарасюк В., Кравчук А., Матвєєв В.А.

Західноукраїнський національний університет, магістранти

I. Постановка проблеми

У галузі розробки програм та програмних систем велике значення має інструментальне програмне забезпечення, призначене для підтримки процесу розробки. Основним інструментом є інтегроване середовище розробки (Integrated Development Environment, IDE) – комплекс програм, що надає розробнику засобу редагування вихідного коду, контролю версій, налагодження, трансляції, організації командної роботи та багато інших можливостей.

Однією з основних функцій інтегрованих середовищ розробки є редагування вихідного коду програм. Сучасні текстові редактори здатні частково автоматизувати процес розробки, що суттєво підвищує продуктивність праці програміста. Можливості текстових редакторів дозволяють виділяти кольором синтаксичні елементи ("підсвічування синтаксису" від англ. "syntax highlight"), повідомляти програміста про помилки, автоматично доповнювати синтаксичні конструкції при наборі, виконувати навігацію за вихідним кодом, візуалізувати структурну організацію програми.

Особливістю сучасної індустрії розробки програмного забезпечення є застосування кількох мов програмування в один програмний проект. Ця особливість породжує проблему перевірки узгодженості компонентів, реалізованих різними мовами, що застосовуються у складі однієї програмної системи. Як правило, неузгодженість компонентів відстежується тільки на етапі налагодження або тестування. Наприклад, якщо у програмі мовою C# використовується функція, реалізована в окремому модулі мовою Cі, та сигнатура цієї функції змінилася, то відстежити помилку, що виникла внаслідок цього в програмній системі можлива лише під час виконання. Такий стан справ суттєво знижує ефективність процесу розробки.

II. Мета роботи

Метою дослідження є виявлення помилок у програмному коді, написаному з використанням декількох мов програмування на етапі редагування вихідних текстів програм.

III. Метод предметно-орієнтованого аналізу вихідних текстів програм

Ключовими поняттями у роботі є семантична модель програми та предметно-орієнтований аналіз.

Модель – це форма відображення певного фрагмента вихідної системи, що містить її суттєві властивості та є об'єктом дослідження чи аналізу. Предметна область – це система взаємопов'язаних термінів або понять. Наприклад, система понять мови програмування, система понять прикладної області або спеціальна система понять, що вводиться для розв'язання певного завдання.

Семантична модель програми – це модель, яка описує елементи програми, їх властивості, характеристики та зв'язки у термінах мови програмування чи іншої предметної області. На рівні

семантичних моделей виконується смислове поєднання між частинами тексту програмної системи, представленими в термінах різних предметних областей.

У даній роботі вихідною системою є текст програми, а предметом аналізу – елементи семантичної моделі, є смисловими одиницями, що формують модель програми. Аналізована інформація представлена на рівні вихідного тексту, рівні дерева синтаксичного розбору (синтаксичної моделі програми) та на рівні семантичної моделі програми.

Для представлення інформації про елементи моделі та взаємозв'язки між ними можуть використовуватись семантичні мережі. Формальним представленням семантичної мережі є граф, у якому кожна вершина відповідає елементу множини сутностей чи понять, а дуги відповідають відношенням між цими сутностями. Формальні мови, мови програмування та моделювання даних, розробляються за певним правилам із опорою на суворо організовану систему понять та визначень. Це дозволяє будувати семантичні моделі вихідних текстів програм у термінах, визначених семантичними специфікаціями, що розробляються для кожного з використовуваних у програмних проєктах мов програмування. Такий набір термінів буде метамоделлю для екземпляра семантичної мережі.

Предметно-орієнтований аналіз – аналіз властивостей семантичної моделі, специфічні для способу її застосування. Предметно-орієнтований аналіз тексту програми – аналіз властивостей тексту програми, специфічні для способу застосування синтаксичних конструкцій мови, якою написана програма. Такою мовою може бути мовою загального призначення, предметно-орієнтованою мовою (Domain Specific Language, DSL) або мовою запитів, наприклад мовою структурованих запитів (Structured Query Language, SQL) та інші. На рисунку зображено представлення семантичної моделі M, метамоделлю якої є специфікація S.

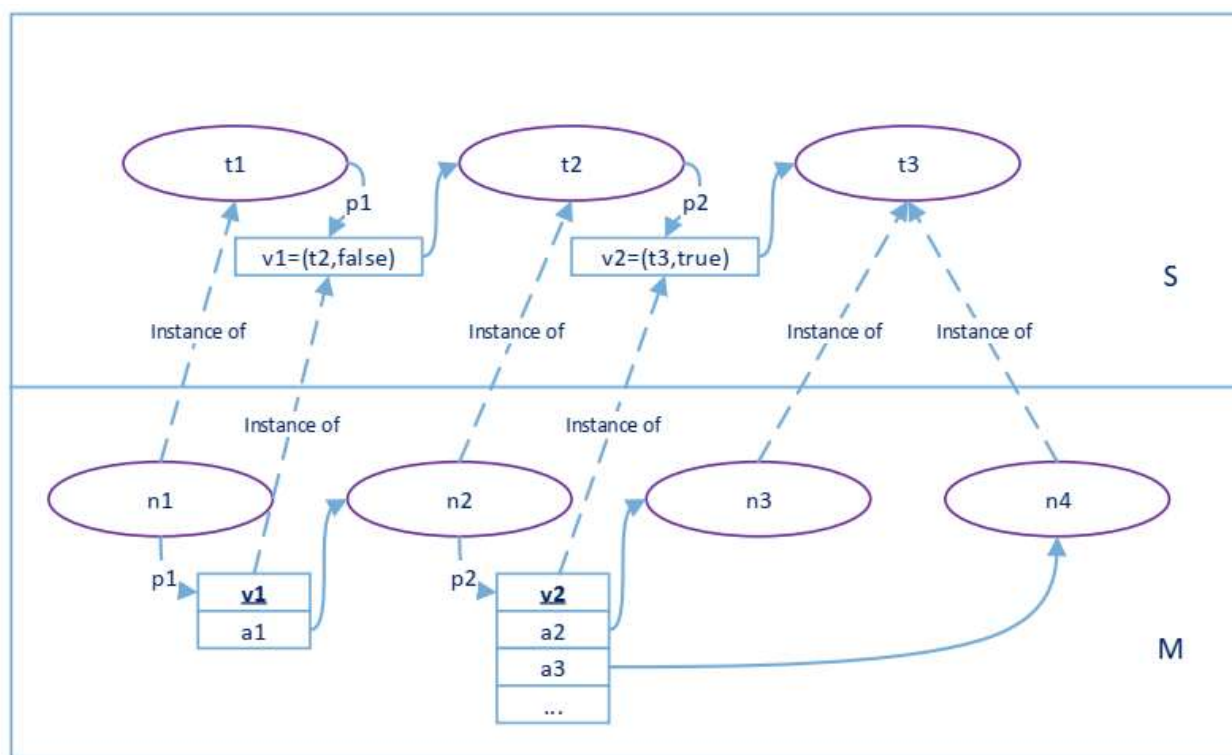


Рисунок 1 – Ілюстрація формального представлення семантичної моделі M зі специфікацією S

Сформовані моделі зберігають взаємозв'язки з текстовим редактором у складі середовища програмування та продовжують використовуватись під час редагування тексту. Врахування залежностей між елементами даних, описаних відображеннями у складі специфікації семантичної трансляції, дозволяє оновлювати лише ті частини моделей, дані яких були порушені у процесі редагування раніше проаналізованого тексту. Так забезпечується гранулярність та інкрементальність аналізу.

Висновок

Подання елементів текстових даних у складі семантичної мережі може використовувати елементи даних із складу внутрішньої структури даних текстового редактора. Це виключає дублювання даних та спрощує виявлення частин семантичних моделей, що торкаються змінами тексту.

Таким чином, відкривається можливість підтримки гранулярного та ітеративного аналізу на основі запропонованого методу.

Список використаних джерел

1. Buasilovsky, P. Adaptive and intelligent Web-based educational systems / P. Buasilovsky, C Peylo // International Journal of Artificial Intelligence in Education. Special Issue on Adaptive and Intelligent Web-based Educational Systems -2003. -№13 (2-4). -P. 159-172.
2. Murray, T. Authoring Intelligent Tutoring Systems: An Analysis of the State of the Art / T. Murray. // International Journal of Artificial Intelligence in Education. -1999.-№10-Р. 98-129
3. Андон, Ф. И. Логические модели интеллектуальных информационных систем / Ф. И. Андон, А. Е. Яшунин, В. А. Резниченко. - К: Наукова думка.-1999.-397 с.
4. Краковецкий А. Кластеризация: алгоритмы k-means и c-means [Электронный ресурс] / Александр Краковецкий // Habrahabr. – 2009. – Режим доступа до ресурсу: <http://habrahabr.ua/post/67078/>.
5. Marshall, B. Convergence of Knowledge Deputatgement and E-Learning: the GetSmart Experience [Электронный ресурс] / Marshall, B., et al. // JC DL. -Houston, 2003. - Режим доступа: <http://ai.bpa.arizona.edu/go/intranet/Publication/JC DL-2003-Marshall.pdf>.

Наукове видання

Комп'ютерні інформаційні технології

Матеріали
школи-семінару молодих вчених і студентів
СІТ'2021

Відповідальний за випуск:

Дивак М. П., д.т.н., професор,
декан факультету комп'ютерних інформаційних технологій

Підписано до друку 16.12.2021 р.
Формат 60x84/16. Папір офсетний.
Друк офсетний. Зам. № 6-165
Тираж 50 прим.

Віддруковано ФО-П Шпак В. Б.
Свідоцтво про державну реєстрацію В02 № 924434 від 11.12.2006 р.
Свідоцтво платника податку: Серія Е № 897220
м. Тернопіль, вул. Просвіти, 6.
тел. 8 097 299 38 99
E-mail: tooums@ukr.net