

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету)
Кафедра комп'ютерних систем та мереж
(повна назва кафедри)

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи)

магістра

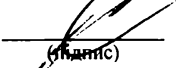
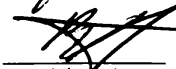
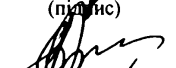
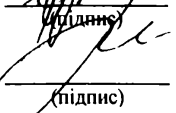
(освітній ступінь (освітньо-кваліфікаційний рівень))

на тему: **Методи і засоби збору, аналізу та відображення
даних в комп'ютерних системах «Розумний цінник»**

Виконав: студент (ка) 6 курсу, групи СІМ-61
спеціальності 123

«Комп'ютерна інженерія»

(шифр і назва спеціальності (напряму підготовки))

	 (підпис)	<u>Галан С.О.</u> (прізвище та ініціали)
Керівник	 (підпис)	<u>Яцишин В.В.</u> (прізвище та ініціали)
Нормоконтроль	 (підпис)	<u>Тиш Є.В.</u> (прізвище та ініціали)
Рецензент	 (підпис)	<u>Яцишин Н.Б.</u> (прізвище та ініціали)

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування введеного навчального закладу)

Факультет Комп'ютерна інженерія систем і програмне інжиніринг
Кафедра Комп'ютерних систем та мереж
Освітній ступінь Магістр
Напрямок підготовки _____
Спеціальність 123 "Комп'ютерна інженерія"
(цифр і назва) (цифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри К.С. Овчарук Р.П.
"30" 03 2019 р.

ЗАВДАННЯ
НА ДИПЛОМНИЙ ПРОЕКТ (РОБОТУ) СТУДЕНТУ

Талану Сергію Олександровичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Методи і засоби збору, аналізу та відображення даних в комп'ютерних системах "розумний будинок"

Керівник проекту (роботи) Яцишин В.В. доцент кафедри К.С.
(прізвище, ім'я, по батькові, науковий ступінь, вчений звання)

Затверджені наказом по університету від "27" 03 2019 року № 415-855

2. Термін подання студентом проекту (роботи) 23.12.19

3. Вихідні дані до проекту (роботи) Тема курсової роботи, шкелу колекторів, зовнішньої мережі, засоби відображення інформації

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

1. Збір
 2. Аналіз сучасного стану та вживаних процесів розробки розумних систем
 3. Побудова моделі архітектури системи "розумний будинок"
 4. Реалізація програмного забезпечення системи "розумний будинок"
 5. Аналіз впливу екологічних факторів
 6. Охорона праці та безпека в надзвичайних ситуаціях
 7. Екологія
 8. Висновки
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
 1. Розробка процесів збору даних у комп'ютерних системах
 2. Аналіз архітектури комп'ютерних систем
 3. Побудова моделі архітектури системи "розумний будинок"
 4. Реалізація програмного забезпечення системи "розумний будинок"
 5. Аналіз впливу екологічних факторів
 6. Охорона праці та безпека в надзвичайних ситуаціях
 7. Екологія
 8. Висновки

6. Консультанти розділів проекту (роботи)

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Структура еколог. ефект	Клирик Н.Б.		15.11
Безпека в НС	Ступчик В.С., ем. викл. кадр. ок.		16.11
Екологія	Маслова О.А. доц.		16.11
Оцінює праці	Суховста Н.В.		16.11
Спеціальні запити	Лущинко Н.Б.		16.11

7. Дата видачі завдання 30.03.19

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Термін виконання етапів проекту (роботи)	Примітки
1	Аналіз сучасного стану та особливостей процесу розробки "Торгівки", системи	7.10.19	Виконано
2	Побудова моделі архітектури системи "Торгівки" фізика	21.10.19	Виконано
3	Реалізація програмного забезпечення системи "Торгівки" фізика	4.11.19	Виконано
4	Обґрунтування екологічної спрямованості	11.11.19	Виконано
5	Охорона праці та безпека в надзвичайних ситуаціях	18.11.19 23.11.19	Виконано
6	Екологія		Виконано
	Попередній захист дипломної роботи	28.11.19	Виконано
	Захист дипломної роботи	23.12.19	

Студент

(підпис)

Телом С.О.

(прізвище та ініціали)

Керівник проекту (роботи)

(підпис)

Лущинко Н.Б.

(прізвище та ініціали)

АНОТАЦІЯ

Методи і засоби збору, аналізу та відображення даних в комп'ютерних системах «Розумний цінник» // Дипломна робота//Галан Сергій Олегович// Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем та програмної інженерії, група СІм-61 // Тернопіль, 2019 // с. – 140 , рис. – 46, табл. – 11, аркушів А1 – 10 , додат. – 2 , бібліогр. – 22.

КЛЮЧОВІ СЛОВА: МЕТОД, ЗАСІБ, РОЗУМНИЙ ЦІННИК, ЗБІР, АНАЛІЗ, КОМП'ЮТЕРНА СИСТЕМА, ІНТЕРНЕТ РЕЧЕЙ.

Мета роботи полягає у дослідженні методів та засобів збору, аналізу та відображення інформації при проектуванні комп'ютерних систем «розумний цінник».

У дипломній роботі поставлено та розв'язано наступні задачі: аналіз наукових публікацій та інтернет-джерел для визначення сучасних підходів до проектування архітектури комп'ютерних систем «розумний цінник»; аналіз методів штучного інтелекту для визначення конкурентної вартості товарів; побудова та обґрунтування моделі архітектури комп'ютерної системи «розумний цінник»; розробка концептуальної схеми комп'ютерної системи «розумний цінник» з інтеграцією інтелектуального сервісу визначення вартості товару; забезпечення організації взаємодії IoT, хмарних та інтелектуального сервісів.

У першому розділі дипломної роботи проаналізовано наукові публікації і практичні аспекти проектування комп'ютерних систем «Розумний цінник», визначено бізнес-процеси, автоматизацію яких необхідно провести та встановлено переваги використання пристроїв, які функціонують у мережі Internet. Широким полем до впровадження IoT технологій є застосування їх в домашніх умовах, зокрема для керування освітленням, медіа пристроями,

підключенням до автомобіля, відеонагляду та багатьох інших. Визначено стан розвитку систем «Розумний цінник» та виявлено проблеми і задачі, які необхідно вирішити, зокрема значна частина IoT пристроїв вимагає додаткових програмних чи апаратних шлюзів для взаємодії із іншими пристроями, необхідні величезні об'єми для збереження даних та потужності для їх обробки, відкритим залишається питання забезпечення захищеності даних, що вимагає додаткових досліджень щодо формалізації і концептуальної побудови апаратних, програмних та інтелектуальних сервісів при реалізації комп'ютерних систем «Розумний цінник».

У другому розділі дипломної одержано результати, які полягають у розробці та формалізації компонентів архітектури системи «розумний цінник», формалізації процесу збору даних про товари, обґрунтованому застосуванні інтелектуального сервісу аналізу товарів та формуванні цін з врахуванням відгуків користувачів, визначенні подібності товарів і споживачів, та конструюванні архітектури системи «розумний цінник». При цьому запропоновано організацію системи на основі взаємодії розумних пристроїв та хмарних сервісів. Спроектовано та запропоновано універсальну функціональну архітектуру системи «розумний цінник» на основі компонентного представлення та взаємодії кінцевих пристроїв, локального шлюзу, сервісів доступу та управління даними. На основі методу колаборативної фільтрації, формалізовано процес аналізу характеристик товарів, зокрема щодо формування ціни, визначення подібних товарів, статистики купівлі товарів і вподобань споживачів, що дало змогу враховувати ці критерії і підвищити ефективність управління бізнес-процесом з можливістю формування та миттєвої реакції в умовах зміни ринку.

У третьому розділі розроблено програмне забезпечення, що реалізує запропоновану архітектуру організації системи «розумний цінник» шляхом використання засобів NodeMCU, IoTHub, налаштуванням параметрів апаратного та програмного забезпечення при оновленні та візуалізації даних, а також протоколу MQTT, а також на основі платформи Onlizer реалізовано

сервіс збору даних про товари з можливістю запису у базу даних, яка спільно використовується із сервісом аналітики даних та відображення цін у системі «розумний цінник». За допомогою мови програмування Python та відкритих бібліотеки реалізовано сервіс аналізу даних про товари, що дає змогу формувати конкурентну ціну у системі «розумний цінник».

У четвертому розділі проведено розрахунки для обчислення економічної доцільності впровадження запропонованих рішень і одержано результати на основі яких підтверджено економічну ефективність результатів дипломної роботи магістра.

У п'ятому розділі проаналізовано вимоги з охорони праці при використанні комп'ютерних систем «Розумний цінник», шляхи дотримання вимог техніки безпеки при експлуатації таких систем, а також проведено оцінку стійкості роботи промислового об'єкту в умовах радіаційного забруднення при ядерному вибуху та забезпечення безпеки життєдіяльності населення в умовах надзвичайних ситуацій природного походження.

У шостому розділі розглянуто питання електромагнітного забруднення довкілля, його впливу на людину, шляхи забезпечення негативних факторів на життя біологічних об'єктів, а також проаналізовано джерела шуму і вібрацій, методи їх знешкодження.

ABSTRACT

Methods and tools of data collection, analysis and display in computer systems “Smart price list”// Master thesis// Galan Serhiy Olehovych// Ternopil Ivan Pul'uj National Technical University, Faculty of Computer Information Systems and software engineering, group CIm -61 // Ternopil, 2019// p. - 140, fig. – 46, table. – 11, Sheets A1 - 10, Add. - 2, Ref. - 22.

KEYWORDS: METHOD, TOOL, SMART LABEL, PARSING, ANALYSIS, COMPUTER SYSTEM, INTERNET OF THINGS.

The purpose of the master's thesis is to investigate the methods and means of collecting, analyzing and displaying information when designing a computer system "smart price label".

The thesis has set and solved the following tasks: analysis of scientific publications and Internet sources for defining modern approaches to the design of computer systems architecture «smart price label»; analysis of artificial intelligence methods to determine the competitive value of goods; construction and justification of the architecture model of the computer system «smart price label»; development of the conceptual scheme of the computer system «smart price label» with the integration of intelligent service of determining the value of goods; providing IoT, cloud and intelligent service organization.

The first chapter of the thesis analyzes the scientific publications and practical aspects of the design of computer systems "Smart price tag", identifies the business processes that need to be automate and the benefits of using devices that operate on the Internet. The broad field of application of IoT technologies is their application at home, in particular for lighting control, media devices, car connection, CCTV and many others. The state of the Smart Price System designing, problems and tasks has been identified, in particular a large number of IoT devices require additional software or hardware gateways to interact with other devices, huge volumes are

required to store data and processing power open the issue of data security remains, which requires additional research on the formalization and conceptual construction of hardware, software and intellectual services in the implementation of computer systems the price tag.

In the second chapter of the diploma the results are obtained that consist in the development and formalization of the components of the architecture of the system «smart price label», the formalization of the process of collecting data on goods, reasonable use of intelligent service analysis of goods and pricing taking into account user feedback, determining the similarity of goods and consumers, and design architecture of the system «smart price label». At the same time it is proposed to organize the system based on the interaction of smart devices and cloud services. The universal functional architecture of the "smart price tag" system is designed and proposed based on component representation and interaction of end devices, local gateway, access services and data management. Collaborative filtration method formalized the process of analyzing the characteristics of goods, including pricing, determining such products, statistics of purchases of goods and consumer preferences, which made it possible to take into account these criteria and improve the efficiency of business process management with the ability to form and instant response in the conditions of change market.

In the third chapter, the software is implemented that implements the proposed architecture of the organization of the system "smart price tag" by using NodeMCU, IoT Hub, configuring hardware and software parameters for updating and visualization of data, as well as MQTT protocol, as well as using the Onlizer platform service product data with the ability to write to a database that is shared with the data analytics service and to display prices in a smart price tag system. Python programming language and open libraries have a product data analysis service that allows you to set a competitive price in the "smart price tag" system.

In the fourth chapter, calculations were made to calculate the economic feasibility of implementing the proposed solutions and the results were obtained on

the basis of which the economic efficiency of the results of the master's thesis was confirmed.

The fifth chapter analyzes the requirements for occupational safety when using computer systems "Smart price tag", ways to comply with safety requirements during the operation of such systems, as well as assesses the stability of the industrial facility in the conditions of radiation contamination during nuclear explosion and providing safety of life of the population in the conditions of natural emergency.

The sixth chapter deals with the issue of electromagnetic pollution, its impact on humans, ways of providing negative factors on the life of biological objects, as well as the sources of noise and vibration, methods of their neutralization.

ЗМІСТ

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ І СКОРОЧЕНЬ	12
ВСТУП	13
РОЗДІЛ 1 АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОСОБЛИВОСТЕЙ ПРОЦЕСУ РОЗРОБКИ «РОЗУМНИХ» СИСТЕМ.....	17
1.1. Аналіз бізнес-процесів та процесів створення «розумних» систем.....	17
1.2. Аналіз базових понять при розробці систем «розумний цінник».....	24
1.3. Застосування розумних пристроїв при автоматизації бізнес-процесів.	29
1.4. Особливості взаємодії «розумних пристроїв» і хмарних сервісів	31
1.5. Висновки до розділу	32
РОЗДІЛ 2 ПОБУДОВА МОДЕЛІ АРХІТЕКТУРИ СИСТЕМИ «РОЗУМНИЙ ЦІННИК»	34
2.1. Аналіз принципів організації «розумних систем».....	34
2.1.1. Кінцеві пристрої.....	34
2.1.2. Шаблони взаємодії.....	36
2.1.3. Канали взаємодії (протоколи).....	43
2.2. Проектування та формалізація компонентів архітектури системи «розумний цінник» на рівні апаратного забезпечення і каналів зв'язку.....	51
2.3. Математичне забезпечення процесу збору даних у системі «розумний цінник»	54
2.4. Формалізація процесу аналізу даних про товари у системі «розумний цінник»	57
2.5. Конструювання архітектури системи «розумний цінник» на структурному рівні.....	61
2.6. Висновки до розділу	70

	10
РОЗДІЛ 3 РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ «РОЗУМНИЙ ЦІННИК»	72
3.1. Програмна реалізація взаємодії IoT і хмарного сервісу	72
3.2. Завантаження програмної реалізації в пристрій	77
3.3. Тестування роботи системи	82
3.4. Створення сервісу збору інформації про товари з використанням платформи Onlizer	84
3.5. Сервіс формування ціни товарів на основі подібності товарів та споживачів	90
3.6. Висновки до розділу	97
РОЗДІЛ 4 ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ	98
4.1. Розрахунок норм часу на виконання науково-дослідної роботи.....	98
4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи	101
4.3. Розрахунок витрат на електроенергію	104
4.4. Розрахунок витрат на матеріали	104
4.5. Розрахунок суми амортизаційних відрахувань	105
4.6. Обчислення накладних витрат.....	106
4.7. Складання кошторису витрат та визначення собівартості науково-дослідних робіт.....	106
4.8. Розрахунок ціни науково-дослідних робіт	107
4.9. Визначення економічної ефективності і терміну окупності капітальних вкладень.....	108
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ	110
5.1. Охорона праці.....	110

5.2. Оцінка стійкості роботи промислового цеху в умовах радіаційного забруднення при ядерному вибуху. Оцінка інженерного захисту робітників і службовців промислового об'єкту	113
---	-----

РОЗДІЛ 6 ЕКОЛОГІЯ.....	119
------------------------	-----

6.1. Електромагнітне забруднення довкілля, його вплив на людину. Шляхи його зменшення	119
---	-----

6.2. Джерела шуму і вібрацій та методи їх знешкодження.....	121
---	-----

ВИСНОВКИ.....	125
---------------	-----

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	128
---------------------------------	-----

Додаток А Тексти наукових публікацій.....	130
---	-----

Додаток Б Лістинг коду взаємодії IoT та хмарних сервісів на мові Lua	136
--	-----

ПЕРЕЛІК ОСНОВНИХ УМОВНИХ ПОЗНАЧЕНЬ,
СИМВОЛІВ І СКОРОЧЕНЬ

БД	База Даних
КС	Комп'ютерна Система
ПС	Програмні системи
CASE	Computer Aided Software Engineering
ІоТ	Internet of Things
MQTT	Message Queue Telemetry Transport
UML	Unified Modeling Language
UMM	Usability Maturity Model
XML	Extended Markup Language

ВСТУП

Актуальність теми. Автоматизація є одним із найбільш ефективних шляхів розвитку підприємств торгівлі, що дає змогу мінімізувати вплив людського фактору та швидко реагувати на зміни ринку. За допомогою програмно-апаратних комплексів можна значно підвищити продуктивність виконання задач і водночас зменшити витрати на обслуговування бізнес-процесів.

Враховуючи тенденції до міграції та одночасної роботи магазинів, супермаркетів та інших закладів торгівлі, як онлайн, так і офлайн, актуальним є впровадження комп'ютерних систем «розумний цінник». Перевагами таких систем є узгодженість цін на товари, динамічність реакції на зміну вартості товарів, економія витрат на обслуговування цінників.

Одним із шляхів реалізації систем «розумний цінник» є застосування підходу, що передбачає використання хмарних сервісів при зборі, аналізі та збереженні даних. Перевагою систем, побудованих на хмарних сервісах, є забезпечення ефективності, надійності, гнучкості і масштабованості архітектурних рішень без необхідності купівлі, налаштування та обслуговування власних центрів обробки інформації. Окрім цього, хмарні сервіси дають змогу проводити аналітику даних шляхом використання інтелектуальних сервісів при формуванні вартості товарів на основі аналізу цін компаній-конкурентів.

Дослідженню методів, засобів і технологій проектування «розумних сервісів» з використанням IoT компонентів присвячено ряд робіт як вітчизняних, так і закордонних вчених. Серед українських вчених даної галузі варто виділити праці О. Якуніна, Д. Кулешова, М. Дзюби, Т. Верного, А. Семакіної, а серед закордонних – Д. Нормана, П. Лукаса, Е. Брінджолфсона, Д. Абадовські, П. Вахера, С. Джамте.

Оскільки, сфера побудови «розумних сервісів» є доволі новою та активно розвивається, не вирішеними ще залишаються ряд задач, зокрема це

стосується протоколів взаємодії між «розумними пристроями», стандартизації способів інтеграції з існуючими системами, забезпечення надійності передачі даних та ін. Тому актуальним є дослідження методів і засобів збору, інтелектуального аналізу та відображення даних в комп'ютерних системах "розумний цінник".

Мета роботи полягає у дослідженні методів та засобів збору, аналізу та відображення інформації при проектуванні комп'ютерних систем «розумний цінник».

У дипломній роботі поставлено та розв'язано наступні **задачі**:

- аналіз наукових публікацій та інтернет-джерел для визначення сучасних підходів до проектування архітектури комп'ютерних систем «розумний цінник»;
- аналіз методів штучного інтелекту для визначення конкурентної вартості товарів;
- побудова та обґрунтування моделі архітектури комп'ютерної системи «розумний цінник»;
- розробка концептуальної схеми комп'ютерної системи «розумний цінник» з інтеграцією інтелектуального сервісу визначення вартості товару;
- забезпечення організації взаємодії IoT, хмарних та інтелектуального сервісів.

Об'єкт дослідження – процес проектування комп'ютерних систем «розумний цінник».

Предмет дослідження – моделі, методи і засоби проектування комп'ютерних систем «розумний цінник».

Наукова новизна одержаних результатів:

- уперше формалізовано компоненти архітектурного рішення щодо побудови систем «розумний цінник» на основі теоретико-множинних нотацій, що дало змогу забезпечити однозначне трактування компонентів архітектури, зв'язків між ними та забезпечити альтернативний вибір апаратної і програмної реалізації складових таких систем;

– уперше, на основі мультиагентного підходу до опису програмних систем, запропоновано та формалізовано сервіс збору даних про товари, що інтегрується у систему «розумний цінник» на рівні хмарного сервісу і дає змогу накопичувати дані про вартість, характеристики і відгуки про товари, що є базою для подальшого аналізу та ціноутворення.

– набули подальшого розвитку методи конструювання архітектури системи «розумний цінник», які включають апаратне та програмне забезпечення «розумних» пристроїв, визначені протоколи передачі даних, хмарні сервіси збору та аналізу даних про товари, підтримує модель брокер-підписник на основі протоколу MQTT і дає змогу забезпечити конкурентоспроможність підприємства на ринку торгівлі.

Методи дослідження: аналіз, узагальнення та порівняння – при проведенні аналізу існуючих методів, моделей та засобів проектування комп’ютерних систем «розумний цінник»; формалізації та моделювання – при описі та проектуванні архітектури комп’ютерних систем «розумний цінник»; методи штучного інтелекту – при розробці методу формування конкурентної ціни на товари; проектування та програмування – при створенні програмної реалізації запропонованої архітектури комп’ютерної системи «розумний цінник»; тестування і вимірювання – для перевірки правильності функціонування системи.

Практичне значення одержаних результатів. Впровадження запропонованих архітектурних, апаратних і програмних рішень апробовано при побудові комп’ютерної системи «Розумний цінник» шляхом реалізації розробленої моделі взаємодії IoT з використанням NodeMCU та хмарного сервісу IoTHub на основі шаблону підписник-видавець із використанням каналу передачі MQTTv3.1.

Публікації. Результати дослідження апробовано на VIII міжнародній науково - технічній конференції молодих учених і студентів «Актуальні задачі сучасних технологій» (27-28 листопада 2019 р.) Тернопільського національного технічного університету імені Івана Пулюя та на VII науково-

технічній конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (11-12 грудня 2019 року) у вигляді тез конференцій.

1. Галан С.О., Яцишин В.В. Формалізація підсистеми збору даних в системах «Розумний цінник. Матеріали VII міжнародній науково - технічній конференції молодих учених і студентів «Актуальні задачі сучасних технологій» (27-28 листопада 2019 р.) Тернопільського національного технічного університету імені Івана Пулюя. Тернопіль: ТНТУ. 2019. с. 17.

2. Яцишин В.В., Галан С.О. Особливості проектування «розумних систем» з можливістю взаємодії з хмарними сервісами. Матеріали VII науково-технічної конференції Тернопільського національного технічного університету імені Івана Пулюя «Інформаційні моделі, системи та технології» (11-12 грудня 2019 року). Тернопіль: ТНТУ. 2019. с. 29.

РОЗДІЛ 1

АНАЛІЗ СУЧАСНОГО СТАНУ ТА ОСОБЛИВОСТЕЙ ПРОЦЕСУ РОЗРОБКИ «РОЗУМНИХ» СИСТЕМ

1.1. Аналіз бізнес-процесів та процесів створення «розумних» систем

Під розробкою «розумних» систем розуміють процес інтеграції та композиції смарт-пристроїв з використанням готових програмних рішень, або створенням нових, що об'єднуються в єдину систему для задоволення потреб користувачів [1].

За інформацією Новосибірського державного технічного університету [1], їхні студенти розробили і впровадили розумний цінник, що представляє собою систему, до складу якої входить сукупність пристроїв і програмного забезпечення для відображення актуальної ціни товарів та аналізу поведінки споживачів.

Реалізацію електронного цінника, згідно [1], виконано за допомогою електронного паперу, при цьому мікросхема керування та wifi-модуль інтегровано у каркас полиці. Даний цінник з'єднується за допомогою безпроводної мережі з сервером бази даних і автоматично змінює ціну при її оновленні в БД.

Інтелект розумного цінника формує ціну на продукцію в автоматичному режимі з врахуванням багатьох факторів, зокрема, залишку товарів на складах, сезонної складової, попиту споживачів та інших критеріїв ціноутворення.

Згідно [1], особливістю розроблених розумних цінників є здатність збору статистики, обчислення потоку покупців з врахуванням статі, віку і пересування по торговому залу. Розроблене програмне забезпечення розраховує відстань до wifi-модуля смартфона, що дає змогу визначити геолокацію покупця, а при використанні системи знижок торгового підприємства, можна визначити інші критерії покупця.

При розробці систем «розумний цінник» необхідно враховувати специфіку бізнес процесів конкретного продавця чи надавача послуг. Так, для великих супермаркетів та гіпермаркетів, при впровадженні та розробці систем «розумний цінник», необхідно враховувати існуючу інформаційно-технічну інфраструктуру, розміщення товарів, типи товарів, розташування цінників, частота зміни обладнання при розміщенні товарів, здійснювати порівняння цін з конкурентами та ряд інших.

При розширенні торгових площ, кількості відділів та збільшенні працівників інформаційна інфраструктура підприємства повинна бути масштабованою і забезпечувати ріст інформаційних потоків з одночасним його контролем. Такі фактори є важливими при проектуванні смарт-систем, оскільки впливають на здатність персоналу адаптуватись до нових умов праці і ефективного виконання задач.

У випадку зростання об'ємів інформації, які необхідно контролювати та аналізувати, зокрема щодо формування каталогів товарів, продукції чи послуг, значно зростає навантаження на персонал та виникають ризики, пов'язані з повнотою наборів товарів, адекватності та конкурентоспроможності формування ціни на продукцію, своєчасного виконання поставлених задач.

Такі фактори мають значний вплив на бізнес-процеси, їх суть та зміни у систематизації. У випадку невеликого обсягу товарів і послуг, персонал здатний керувати бізнес-процесами в «ручному» режимі, однак при їх розширенні і зміні, виникає проблема збереження контролю над даними, їх узгодженістю та ефективністю процесів управління.

Одним із шляхів вирішення такої проблеми є впровадження автоматизованих рішень для підвищення ефективності бізнес-процесів та зменшення навантаження на менеджерів із постачання та збуту товарів. Автоматизація бізнес-процесів у будь-якій сфері діяльності вимагає детального дослідження об'єктів предметної області, їх атрибутів та зв'язків, і як результат вимагає перш за все вироблення єдиної уніфікованої стратегії представлення усіх процесів, що автоматизуються. При цілеспрямованому

впровадженні програмно-апаратних комплексів можна досягти високої ефективності реалізації бізнес-процесів за рахунок підвищення швидкості і точності виконання задач, мінімізації помилкових дій і практично нейтралізації «людського фактору». Автоматизація бізнес-процесів дає змогу забезпечити оптимальний баланс та раціональність розподілу функцій між інформаційно-технічною системою та працівником в залежності від сфери діяльності.

Основними вимогами до автоматизації бізнес-процесів підприємств є:

- відповідність моделі управління підприємством стратегії автоматизації бізнес-процесів;
- відповідність моделі бізнес-процесів технічним критеріям автоматизації.

Для виконання першої вимоги необхідно поточну модель управління підприємством трансформувати і представити у вигляді процесної моделі – сукупності процесів управління, їх критеріїв і взаємозв'язків. При цьому передбачається залучення фахівців або експертів в галузі управління та інформаційних технологій для забезпечення ефективності переходу від однієї моделі до іншої та можливості автоматизації бізнес-процесів.

Другою важливою вимогою є відповідність поточної моделі бізнес-процесів підприємства технічним критеріям, що використовується при їх автоматизації.

Процес автоматизації бізнес-процесів практично завжди супроводжується реструктуризацією та декомпозицією бізнес-процесів. Це дозволяє деталізувати та знайти неефективно використовувані ресурси, підвищити фінансову спроможність підприємства, модифікувати та запровадити нову загальну стратегію розвитку. Витрати на автоматизацію бізнес-процесів є доволі великими, тому спочатку необхідно провести детальний аналіз існуючої технічної інфраструктури, розробити нову бізнес-модель, яка б забезпечувала і враховувала способи автоматизації з мінімальними затратами.

На даний час використовується ряд програмних продуктів, які дають змогу автоматизувати роботу підрозділів підприємства, що в свою чергу впливає на організацію структури підприємства, описує бізнес-процеси та процеси управління.

Сучасний рівень розвитку програмних та апаратних засобів забезпечує здатність до автоматизації основних бізнес-процесів і відповідно структурних підрозділів підприємства. У табл. 1.1 наведено характеристику загальних бізнес-процесів та програмного забезпечення, що дає змогу їх автоматизувати.

Таблиця 1.1

Програмні рішення при автоматизації бізнес-процесів

№ з/п	Сфера застосування	Бізнес-процеси	Програмні рішення
1	Організаційна і адміністративна діяльність	<ul style="list-style-type: none"> – управління персоналом; – документообіг; – управління та контроль за виконанням наказів; – секретаріат. 	<ul style="list-style-type: none"> – ProcessMaker Enterprise; – Onit Corporate Secretary; – Oracle Business Process Management Suite
2	Управлінська діяльність	<ul style="list-style-type: none"> – економічні бізнес-процеси; – юриспруденція; – облік та аудит; – планування. 	<ul style="list-style-type: none"> – QPR Metrics, – Onit Contract Administration; – 1С Підприємство; – SAP R /3; – EpicorEPM Canvas Planning

№ з/п	Сфера застосування	Бізнес-процеси	Програмні рішення
3	Маркетингова діяльність	<ul style="list-style-type: none"> – керування маркетингом стратегією; – управління рекламними заходами; – керування продажами; – управління відносинами з клієнтами; 	<ul style="list-style-type: none"> – Improsy's Net. ERP; – SAS CRM; – Oracle CRM
4	Фінанси підприємства	<ul style="list-style-type: none"> – бухгалтерський податковий облік; – планування фінансів; – управління розрахунками; – нарахування заробітної плати 	<ul style="list-style-type: none"> – 1С Бухгалтерія; – Інфо-бухгалтер 10; – Бухгалтерія; – ПАРУС
5	Виробництво продукції	<ul style="list-style-type: none"> – управління виробництвом; – управління запасами; – постачання; – контроль управління обладнанням 	<ul style="list-style-type: none"> – SAP R /3; – Oracle JD Edwards Enterprise One; – BAAN

№ з/п	Сфера застосування	Бізнес-процеси	Програмні рішення
6	Управління якістю	<ul style="list-style-type: none"> – контроль якості продукції; – підвищення кваліфікації персоналу; – управління якістю бізнес-процесів 	<ul style="list-style-type: none"> – SAP R /3; – Sky mark PathMaker; – QPR Metrics
7	Зовнішні комунікації	<ul style="list-style-type: none"> – управління присутністю в Інтернеті; – залучення клієнтів; – взаємодія із структурними підрозділами; 	<ul style="list-style-type: none"> – Improsy's Net ERP; – Vocus PR Suite; – Media Grip Media Magnet

Виходячи з наведених в табл. 1.1 областей застосування і можливостей автоматизації із застосуванням програмних рішень формується інформаційно-технологічна і технічна стратегія підприємства. Під ІТ-стратегією, зазвичай, розуміють комплексний план розвитку інформаційної інфраструктури підприємства для забезпечення ефективності управління бізнес-процесами, їх моніторингу та контролю з врахуванням ризиків та невизначених ситуацій.

При проектуванні «розумних» систем виникає багато труднощів, які пов'язані з неоднорідністю апаратного і програмного забезпечення смарт-пристроїв, програмного забезпечення для керування ними, відсутністю або недосконалістю стандартів даної області та ряд інших.

Неоднорідність смарт-пристроїв призводить до того, що розробникам, доводиться інтегрувати багато пристроїв, кожен з яких має власні API [2].

При цьому, розробники повинні розуміти API кожного пристрою та знати як до них звертатись, які значення можна одержувати з конкретного пристрою.

Ще одна складність при проектуванні «розумних» систем пов'язана з великою кількістю однотипних пристроїв з різними методами доступу до них.

Одним із шляхів вирішення таких задач є формування абстрактного рівня над «розумними» пристроями і побудова так званої моделі виконання. Модель виконання дозволяє і забезпечує обробку запитів від і до неоднорідних пристроїв. Наприклад, розробники можуть визначити уніфіковані поняття для конкретної предметної області, які можна описати за допомогою абстрактного класу. Абстрактний клас немає реалізації, але володіє набором властивостей та операцій, характерних для опису об'єктів на концептуальному рівні. Реалізація властивості наслідування дозволяє згенерувати похідні класи, які вже будуть описувати конкретний екземпляр об'єкту і мати власну реалізацію. Наприклад при абстрактному описі об'єкта «лампа» можна розуміти об'єкт для професійного освітлення рослини, а також «розумну» лампу.

При використанні моделі виконання, програми та послуги також розглядаються як пристрої, які розробники можуть використовувати їх послідовно, якщо вони надають однакові функції.

По-друге, модель виконання забезпечує простий шлях розвитку «розумної» системи. Програмування на рівні моделі на основі існуючих технологій таких як OCL та ATL може полегшити розробку системи, знизити витрати на розробку та потенційно скоротити час виходу на ринок.

У даній роботі з автоматизацією пов'язаний бізнес-процес управління продажами, а саме контроль, аналіз та відображення актуальних цін на товари, який формується на основі інтелектуальних методів і враховує ціни конкурентів на ринку. Передбачається, що для автоматизації цього процесу буде застосовуватись комплекс програмних і апаратних рішень.

1.2. Аналіз базових понять при розробці систем «розумний цінник»

Проектування та реалізація «розумних систем» невід’ємно пов’язана із технологією Internet of Things (IoT). IoT – це об’єднання множини речей: пристроїв з датчиками, приводів і контролерів. Ідея IoT вперше виникла ще в 1999 році у Кевіна Ештона – дослідника з Массачусетського технологічного інституту. Тоді він запропонував концепцію системи управління промисловими об’єктами через інтернет і вендинговий апарат підключили до цієї мережі [9].

Протягом наступних п’ятнадцяти років в IoT стрімкого розвитку не спостерігалось, хоча деякі перспективні рішення з’являлись. За період інтенсивного розвитку інформаційних технологій, зокрема, апаратного забезпечення, значно зменшилась вартість комплектуючих, а вибір став дедалі більшим. Засоби взаємодії теж за цей час розвивались і зменшили вартість використання. Робота з даними також стала доступнішою. Ключове нововведення – поява хмарних технологій, які дають практично безмежні можливості для зберігання і обробки даних.

Всі пристрої Інтернету речей здатні підключатися до всесвітньої мережі і можуть мати найрізноманітніші форми та розміри. Ці пристрої (речі) підключаються до мережі та взаємодіють між собою через мережі на основі технологій Bluetooth, Zigbee, WiFi, мережі стільникового зв’язку, супутникової мережі і т.д. Кожним пристроєм управляє контролер (мікроконтролер) або операційна система [8].

У промисловості йдеться в першу чергу про програмовані логічні контролери (ПЛК), які вже кілька десятиліть є основою для промислової автоматизації, а також про системи диспетчерського управління та збору даних (SCADA-системи). Говорячи про пристрій, крім логіки управління слід розглядати і пов’язані з ним послуги, такі як збір даних, фільтрація подій, управління режимом енергоспоживання, вбудоване програмне забезпечення,

управління безпекою та ідентифікацією, а також людино-машинний інтерфейс.

Для виробників пристроїв концепція IoT відкриває нові можливості для творчого підходу, а реалізація в пристроях накопиченого галузевого досвіду може істотно сприяти їх поширенню.

Важливим аспектом розробки «розумних пристроїв» є вибір технології мережевого підключення. Стосовно IoT можна прийняти концепцію "тонкого" або "товстого" клієнта (периферії), з огляду на рівень складності і функціональні можливості, необхідні для підтримки функціонування кінцевих точок, таких як пристрій, мережа пристроїв зі шлюзом, або складна система.

Інтелектуальні пристрої стають «розумними» вузлами, що поєднують в собі як цифрові, так і фізичні аспекти і здатні по новому підійти до того, хто, які дії, де і коли повинен виконувати. Завдяки присутності фізичних аспектів IoT з'являється суміш комп'ютерної техніки з іншими дисциплінами: речі отримують можливість відчувати, рухатися, літати, нагріватися і охолоджуватися, прикладати зусилля, перекачувати рідину, взаємодіяти з людьми і т. д. Різноманітність "речей" величезна: від мобільних пристроїв і засобів домашньої автоматизації в споживчому секторі до промислових турбін і бурових установок.

На базовому рівні подібні пристрої вже мають засоби підключення до мережі або переобладнуються для оснащення їх мережевими можливостями. Крім цього, вже з'являються горизонтальні категорії пристроїв, що застосовуються в різних секторах бізнесу – мобільні пристрої, роботи, дрони (безпілотні літальні апарати), самокеровані транспортні засоби.

На даний момент IoT складається з слабо пов'язаних між собою мереж, кожна з яких призначена для вирішення своїх специфічних завдань. Наприклад, в автомобілях встановлені одразу декілька мереж: для управління роботою двигуна, для систем безпеки, для підтримки зв'язку і т.д. У будівлях також розташована велика кількість мереж для керування опаленням, вентиляцією, кондиціонуванням, телефонним зв'язком, безпекою, освітленням.

У міру розвитку IoT ці та багато інших мереж будуть підключатися один до одного і набувати все більш широкі можливості в сфері безпеки, аналітики та управління.

Інтернет речей – концепція, яка передбачає більш широке застосування технології M2M (machine-to-machine). Сегмент M2M багато в чому є основою концепції IoT, а на початковому етапі розвитку IoT був фактично синонімом M2M. До початку 2016 року не було чітких критеріїв поділу на IoT і M2M. Сьогодні до інтернету вже підключено 10 мільярдів пристроїв. Очікується, що до 2020 року їх кількість становитиме 50 – 60 мільярдів [6].

Технічні можливості для цього вже є. Новий інтернет-протокол – IPv6 надає практично нескінченну кількість IP-адрес. Британська компанія ARM Holdings створила мікро чіп, який маючи розмір 1 мм² і наднизьким споживання енергії дозволяє забезпечити доступ до мережі будь-якого розумного пристрою.

Стартап Supermechanical розробив бездротовий модуль Twine, здатний підключити до мережі практично будь-яку річ і навчити її відправляти SMS-повідомлення, електронні листи або «твіти». Великі IT-корпорації, зокрема Microsoft, надають платформи, на базі яких інші компанії можуть створювати рішення для кінцевих користувачів або власні платформи. Це один з найбільш перспективних напрямків у IoT: за прогнозами McKinsey, до 2020 року 70% ринку будуть генерувати b2b-рішення. А сам ринок IoT, за оцінкою Gartner, досягне \$ 1,7 трлн. [6].

Розробка проектів з використанням IoT поділяється на виробниче та невиробниче застосування.

Невиробниче застосування передбачає, що пристрої належать до таких сфер як «розумні» будинки, годинники, ферми, міста, лікарні і т.д.

Виробниче застосування «розумних пристроїв» дозволяє реалізувати більш розвинуті, складніші рішення, що представляють собою платформи для бізнесу або керування виробничими процесами.

Немає єдиної думки, який із напрямків буде розвиватись швидше, але найпоширеніші напрямки розвитку включають: безпеку даних, аналітику, управління, розробку пристроїв.

Безпека даних – це одна з найважливіших сфер розвитку. Сьогодні 85% розробників виводять на ринок пристрої з явними проблемами в системі безпеки. При цьому кожен п'ятий користувач застосовує свої персональні дані при роботі з IoT-рішеннями [6], тому необхідно забезпечити конфіденційність персональних даних. Вирішення проблем безпеки можна реалізувати не тільки з використанням криптопроцесорів, а й шляхом розвитку надійної інфраструктури або розробкою «безпечного» програмного забезпечення.

Аналітика – це важливий напрямок для складних виробничих систем, де в кожен момент часу отримуються величезні набори даних.

Управління – забезпечує керування пристроєм, залежно від отримуваних даних. Сюди відноситься і автоматизація процесів.

Розробка пристроїв включає сферу розвитку ОС та процесорів. Ринок процесорів для інтернету речей практично вільний, адже серед наявних екземплярів дуже малодоступних та одночасно якісних пристроїв. У даний момент найбільш використовувани 8-бітні процесори, хоча за попередніми прогнозами після 2019 року найпоширенішими будуть 32-бітні. А от операційні системи, то їх випускають компанії-гіганти, зокрема Microsoft із своєю Windows 10 IoT. Проте на ринку з'являються й інші виробники, які не є такими великими та відомими.

Важливо не тільки придумати платформу і створити прототип пристрою. Головне – знайти працюючу бізнес-модель. Не так складно створити розумні кухонні прилади. Складно придумати спосіб її інтеграції із споживчими брендами, щоб монетизувати продукт. Наприклад, системи «розумного будинку» можна продавати через Smart TV. Опитування споживачів платного телебачення показують, що 28% з них готові встановити Smart Home. Мабуть, тому що нова технологія вбудовується в уже знайомий інтерфейс. Ті ж смартфони отримали популярність завдяки правильно обраному інтерфейсу –

сенсорному екрану. Для систем Smart Home таким інтерфейсом може стати голосове управління. Інтерфейс – важливий драйвер зростання ринку IoT [6].

За результатами глобальних міжнародних досліджень, лідером ринку розумних рішень для будинку може стати компанія ІКЕА, поступово вбудовуючи нові пристрої у звичні побутові системи. Спочатку замість звичайної лампочки покупець вибере «розумну», потім підключить її до телевізора, і поступово весь будинок використовуватиме IoT-пристрої. Такий розвиток ринку «від споживача», в даному випадку, є кращим рішенням, ніж нав'язування корпораціями готових і поки незрозумілих кінцевому користувачу систем.

Інтернет речей стрімко розвивається та потроху переростає в «Інтернет всього» (Internet of Everything, IoE). IoE – це поєднання не тільки пристроїв та передачі, обробки даних, але і людей, процесів і даних.

IoT вже прийшов в повсякденне життя через RFID теги, штрих-коди. У найближчому майбутньому все більша кількість предметів зможуть виходити в інтернет. Фахівці стверджують, що це значно полегшить і забезпечить наше життя.

Cisco, світовий лідер в області інформаційних технологій, допомагає компаніям використовувати можливості майбутнього і власним прикладом доводить, що, підключаючи IoT, можна досягти вражаючих результатів.

«З'явившись близько двадцяти років тому, Інтернет змінив все, що нас оточує. Але ці зміни – ніщо в порівнянні з тим, що нам ще належить, - заявив голова ради директорів, головний виконавчий директор компанії Cisco Джон Чемберс (JohnChambers). – Всеохоплюючий Інтернет призведе до грандіозних змін в усіх галузях економіки. Це означає нові перспективи, нові компанії, новий досвід, нові послуги, а також величезні перспективи перед людьми, компаніями та цілими країнами».

Для Інтернету потрібні розподілені, зорієнтовані на додатки мережі, а також обчислювальні системи і платформи для зберігання даних. Зростання

кількості підключень неможливе без IP-пристроїв широкосмугового доступу і протоколу IPv6.

Проаналізувавши потенційний ефект впливу Інтернету на економіку, фахівці Cisco прийшли до висновку, що протягом 10 наступних років Інтернет обіцяє приватному бізнесу в усьому світі 14,4 трильйона доларів потенційної економічної вигоди. За їхніми підрахунками, в найближче десятиліття ІоЕ здатний збільшити сукупний прибуток світових корпорацій приблизно на 21%.

1.3. Застосування розумних пристроїв при автоматизації бізнес-процесів

На початок 2016 року у використанні технологій ІоТ компанії орієнтуються в першу чергу на масові сегменти, такі як:

- «Розумний будинок», включаючи: рішення для створення інтелектуальних сервісів безпеки; рішення для створення інтелектуальних сервісів оптимізації використання ресурсів домогосподарствами.

- «Розумний транспорт», включаючи: сервіси класу fleet management для індивідуальних перевізників (такий собі аналог Uber для вантажного транспорту); сервіси UBI-страхування; сервіси технічного обслуговування по фактичному стану.

- торгівля і фінансові послуги: рішення для автоматичної передачі та аналізу даних з POS-терміналів, включаючи віртуальні управління запасами домогосподарств як сервіс.

- промисловий сегмент – перенесення АСКТП на принципи ІоТ.

Vodafone опитав понад 1,1 тисячі керівників підприємств з різних сфер (ритейл, промисловості, енергетики, машинобудування та ін). З них 28% відповіли, що вони вже використовують ІоТ. Ще 35% заявили, що збираються впровадити подібні проекти протягом року.

Компанії, які вже використовують технологію інтернету речей в своїх бізнесах, витрачають на неї приблизно 24% ІТ-бюджету. Решта приблизно в рівних частинах йде на хмарні технології та хостинг, аналітику і підтримку мобільності.

63% опитаних менеджерів заявили, що вони бачать значне збільшення повернення інвестицій при впровадженні ІоТ. Кожен четвертий бізнес, який вже використовує ІоТ, заявив, що у них підключено більше 100 пристроїв до інтернету речей. Кожен десятий бізнес вже має більше 10 000 підключених пристроїв. У побуті ІоТ можна зобразити як показано на рис. 1.1.



Рис.1.1. Застосування ІоТ у побуті

Для чого використовують Інтернет речей:

- 51% компаній використовують ІоТ для оптимізації використання активів;
- 49% використовують ІоТ для скорочення операційних витрат;
- 46% використовують ІоТ для збільшення рівня безпеки та безпеки громадян і предметів;

- 42% за допомогою IoT оптимізують ланцюжок поставок товарів і послуг від виробника споживачу;
- 41% створюють нові підключення продукти і послуги;
- 40% проектів пов'язано з поліпшенням ефективності і безпеки публічних місць: розумні сміттєві баки, розумне вуличне освітлення, розумні міста.

Тобто, широким полем до впровадження IoT технологій є застосування їх в домашніх умовах, зокрема для керування освітленням, медіа пристроями, підключенням до автомобіля, відеонагляду та багатьох інших.

1.4. Особливості взаємодії «розумних пристроїв» і хмарних сервісів

Хмарні сервіси призначені для виконання найрізноманітніших завдань та взаємодії з надзвичайно великою кількістю пристроїв та програм. Однак для повноцінної взаємодії необхідна підтримка наперед визначених стандартів взаємодії, протоколів, каналів зв'язку, шифрування, форматування даних. Але значна частина IoT пристроїв, датчиків, елементів управління, плат для проектування в свою чергу, призначені для виконання величезного набору специфічних завдань і мають обмежені ресурси в апаратному і програмному аспекті, що призводить до виникнення ряду проблем при їх взаємодії.

Головне обмеження – це об'єм пам'яті для виконання програм. Його обмеження впливають на можливості обробки, зберігання, передачі даних та шифрування. Що викликає серйозні труднощі, адже хмарні технології дозволяють взаємодіяти лише по захищеному каналу зв'язку.

Не будучи в стані з'єднати кінцеві пристрої через брандмауери і безпечно передавати дані для хмарних додатків, організації не зможуть досягти всіх переваг аналітичного управління виробничими процесами, які їм обіцяли при переході на використання хмарних обчислювальних потужностей.

У висококласних IoT-шлюзів повинна бути можливість взаємодіяти з серверами Microsoft Azure або подібними системами інших виробників,

використовуючи найбільш популярні транспортні протоколи. Наприклад, після підключення до мережі IoT-шлюз, застосувавши безпечний протокол Advanced Message Queuing Protocol, може зареєструватися через платформу Microsoft IoT Hub на хмарному сервері Azure [2]. Таким чином, у нього з'явиться можливість відправляти і приймати дані, а також перевіряти справжність отриманої інформації. Виступаючи в ролі посередника при безпечній відправці повідомлень, платформа Azure IoT Hub дозволяє виробляти дистанційне конфігурування і керування пристроями [5].

Хмарні сервіси мають готові реалізації SDK для взаємодії з ними з різноманітних платформ, використовуючи широкий набір засобів розробки та мов програмування [4]. Але такі рішення для невеликих плат відсутні, що створює ще одну проблему в процесі створення IoT системи.

1.5. Висновки до розділу

У даному розділі проведено аналіз сучасного стану розвитку та особливостей проектування комп'ютерних систем, які відносяться до класу «розумних систем». Встановлено, що розвиток даної галузі перебуває в інтенсивному розвитку і вимагає додаткових досліджень в контексті проектування та конструювання як апаратного, так і програмного забезпечення, а також формалізації процесу автоматизації, структурного проектування та взаємодії компонентів. Особливістю процесів автоматизації бізнес-процесів у торгівлі є необхідність їх реструктуризації при мінімальних часових і фінансових витратах.

Основні наукові і практичні результати даного розділу полягають в наступному:

- 1 Проаналізовано бізнес-процеси, характерні для підприємств торгівлі, визначено можливі шляхи і рівень їх автоматизації, обґрунтовано актуальність побудови систем «розумний цінник» у результаті якого встановлено необхідність подальшого дослідження та оптимізації процесів проектування

архітектури, збору та аналізу даних щодо товарів, актуального відображення цін за допомогою електронних цінників, що дало змогу визначити шляхи підвищення ефективності функціонування підприємств торгівлі.

2 На основі аналізу наукових публікацій і практичних досліджень побудови «розумних систем» встановлено, що ефективним способом організації таких систем є використання технології Internet of Things, що дає змогу забезпечити гнучкість та масштабованість рішень при розробці систем «розумний цінник». Перевага «розумних речей» над іншими підходами полягає у можливості одночасного підключення великої кількості пристроїв, застосування стандартизованих і користувацьких протоколів передачі даних через мережу Інтернет, забезпечення ефективного моніторингу та управління бізнес-процесами.

3 Обґрунтовано організацію систем «розумний цінник» з використанням «розумних пристроїв» та хмарних сервісів, що дає змогу забезпечити одержання або передачу актуальних даних із сховищ даних, інтегрувати засоби аналітики даних при формуванні ціни товарів, однак відкритими ще залишаються питання безпеки передачі даних, побудови ефективної архітектури та формалізації таких систем.

РОЗДІЛ 2

ПОБУДОВА МОДЕЛІ АРХІТЕКТУРИ СИСТЕМИ «РОЗУМНИЙ ЦІННИК»

2.1. Аналіз принципів організації «розумних систем»

Функціонування «розумних систем» – це складний процес взаємодії та обміну даними між низкою найрізноманітніших компонентів. До них належать кінцеві пристрої, шлюзи, канали та моделі взаємодії. Оскільки для дослідження вибраний аспект роботи напряму з пристроями, без використання шлюзів, їх не буде розглянуто, а як кінцеві пристрої будуть виступати плати, без вбудованої ОС та підтримки готових рішень.

2.1.1. Кінцеві пристрої

Найпоширенішими кінцевими пристроями, які не мають операційної системи та достатніх можливостей для взаємодії з хмарними технологіями шляхом використання готових SDK, є: Arduino UNO та NodeMCU на базі ESP8266.

ArduinoUno – це плата макетування, що характеризується:

- 14 цифровими виводами;
- 6 аналоговими входами;
- напруга живлення 5 В;
- 32 Кб флеш-пам'яті;
- 2 Кб ОЗП;
- 1 Кб EEPROM;
- частота 16 МГц.

Не зважаючи на це, дана плата є найбільш поширеною та використовуваною для найрізноманітніших цілей: від простих проектів з декількома складовими, наприклад керування освітленням до роботизованих систем.

Плата ArduinoUno зображена на рис. 2.1.

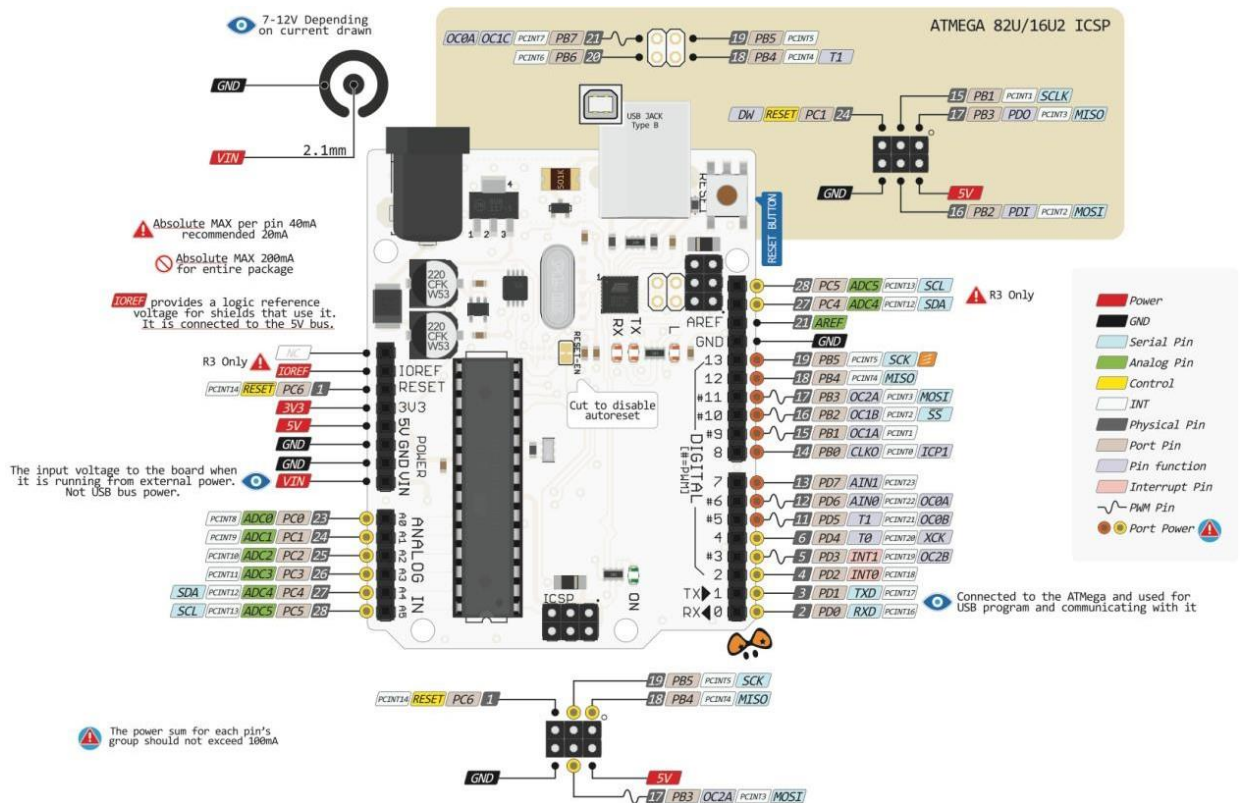


Рис. 2.1. Схема ArduinoUno

NodeMCU – плата, з вбудованим модулем WiFi (ESP8266-12E), з наступними характеристиками:

- підтримка стандарту IEEE 802.11 b/g/n;
- напруга живлення 3,5 В;
- 14 виводів;
- частота роботи 80 МГц [3].

NodeMCU – це одна з найкращих та найдешевших плат з вбудованою підтримкою WiFi, зображена на рис. 2.2.

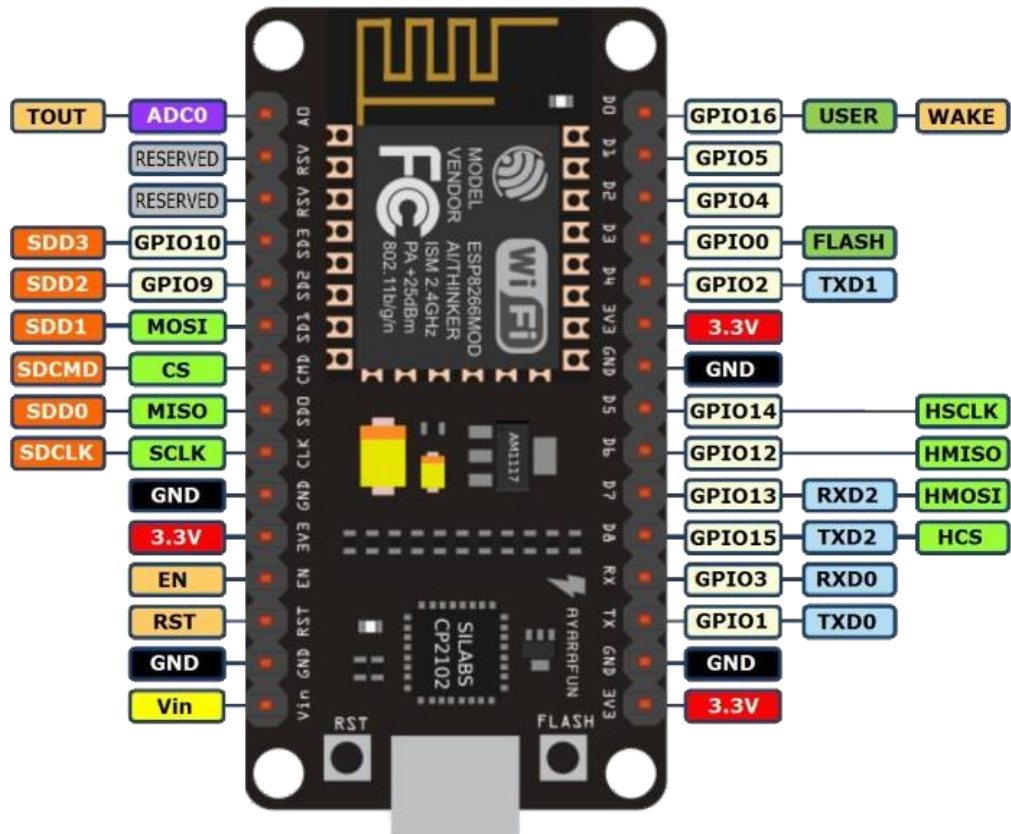


Рис. 2.2. Схема NodeMCU

Модуль має вбудований процесор та пам'ять, і може бути використаний не тільки як шлюз для підключення до мережі, але і як самостійний повноцінний пристрій, що розширює можливості застосування.

2.1.2. Шаблиони взаємодії

Шаблиони інтеграції процесів визначають набір шаблонів і методології використання їх у рамках інтеграції додатків в моделі «розумної системи». Методологія визначає процес аналізу вимог взаємодії та інтеграції з метою визначення вимог до сервісної інфраструктури.

У загальному випадку, взаємодія визначається як активність N-до-N між підсистемами в рамках розподіленої системи. Комплексна взаємодія між підсистемами може бути розподілена на базові взаємодії. Взаємодія базується на 1-до-1 або 1-до-N процесах, що походять з однієї підсистеми. У такий

спосіб, комплексні спільні процеси за участю багатьох підсистем можна розкласти на більш прості взаємодії, які простіше аналізувати.

Шаблони інтеграції процесів надають набір шаблонів для спільної роботи, які використовуються для опису організаційних зав'язків між прикладними додатками, а також набір шаблонів взаємодії, які використовуються для опису очікуваної поведінки в деталях.

Шаблони для спільної роботи можуть бути класифіковані наступним чином:

- шаблон композитного сервісу (composed service pattern): це додатки, які агрегують функції з інших додатків.
- зональний шаблон (zone pattern): широко застосовується і може представляти мережеві зони, такі як Інтранет або нейтральні зони, контейнери сервери додатків.

Шаблон концентратора – представляє спеціальну інтеграційну інфраструктуру, що забезпечує або основні послуги з підключення (мережевий концентратор), або більш розширені послуги, такі як управління процесами, або послуги з інтеграції даних (концентратор для спільної роботи).

Процес інтеграції – шаблони інтеграції визначають чотири стилі взаємодії і декілька варіацій, які перераховані тут в порядку зростання гнучкості та архітектурної оптимізації. Оскільки шаблони додатків використовують один одного, їхні можливості і залежність від проміжного програмного забезпечення зростає, і вони вимагають меншої кількості зусиль для розробки додатків:

Шаблон «Пряме підключення» - найпростіший тип взаємодії, який заснований на топології 1-до-1. Це дозволяє парі додатків всередині організації безпосередньо спілкуватися один з одним.

Різновид «відправка повідомлень». Відноситься до рішень, де бізнес-процес не вимагає відповіді від цільового додатку в межах взаємодії.

Різновид «підключені виклики». Відноситься до рішень, де бізнес-процес залежить від обробки запиту цільовим додатком і передбачає повернення відповіді в межах взаємодії.

Шаблон «Брокер» побудований на основі топології 1-до-N, яка відділяє правила розподілу від додатків. Це дозволяє передавати виклики від джерела до декількох цільових програм одночасно. Цей шаблон додатка знижує кількість з'єднань точка-точка.

Різновид «маршрутизатор». Відноситься до рішень, де основна програма ініціює взаємодію, яка направлена тільки до одної з декількох цільових програм.

Різновид «Pub/Sub». У різновиді «Pub/Sub», брокер надає додаткову взаємодію прямого підключення. Це забезпечує можливість динамічного оновлення правил розподілу, які регулює брокер. При виконанні цього типу взаємодії, зацікавлені сторони можуть додавати себе в список отримувачів для конкретного брокера.

Шаблон додатків «Послідовний процес» розширює топологію 1-до-N, представлену шаблоном Брокер, полегшуючи послідовне виконання бізнес-сервісів, які виконуються в декількох цільових програмах. Таким чином, вона дозволяє запуск послідовного бізнес-процесу, у відповідь на взаємодію, ініційовану основним додатком.

Різновид «Послідовний робочий процес». Розширює базові можливості послідовного процесу, підтримуючи людську взаємодію для завершення певних стадій процесу.

Шаблон «Паралельний процес» розширює базові можливості керування послідовним процесом, які надає шаблон додатків послідовний процес, за допомогою підтримки паралельного (одночасного) виконання підпроцесів.

Різновид «Паралельний робочий процес». Розширює базову функціональність паралельного процесу, підтримуючи людську взаємодію для завершення певних стадій процесу.

Профілі шаблонів інтеграції процесів - це розширення для базових шаблонів інтеграції процесів. Наприклад, «Розширений Enterprise профіль» розширює область дії шаблонів, щоб охопити сценарії Enterprise-to-Enterprise, в тому числі в Інтернеті. Моделі описані в цьому розділі є складовою частиною нового профілю SOA.

Шаблони для моделей інтеграції бізнес-процесів описують зону, як область, в якій доступний певний набір сервісів. Зона доступна через один або декілька шлюзів, які також згадуються як порти в профілі Services Oriented Architecture (SOA).

Прикладами зони та шлюзу може бути:

- Інтранет з доступом через брандмауер і проксі сервер.
- J2EE Web-контейнер з доступом через HTTP listener.
- Брокер повідомлень з доступом через черги.

Шаблон Enterprise Service Bus (ESB) є частиною профілю SOA в бізнес-моделях інтеграції процесів. Він включає набір композитних моделей інтеграції процесів, які є особливо корисними в SOA профілях.

Роль ESB визначається як забезпечення набору можливостей, в тому числі функції точки контролю над службою адресації та присвоєння імен. Клієнти отримують доступ до ESB шляхом виклику служб з конкретними адресами за допомогою конкретних протоколів. ESB інтегрується з постачальниками сервісів шляхом підтримки декількох механізмів інтеграції, але всі з них можна охарактеризувати як виклик служб за допомогою конкретних адрес і протоколів - навіть якщо в деяких випадках "адреса" є ім'ям CICS, а "протоколом" є адаптер ресурсів J2EE інтегрований із CICS Transaction Gateway.

Це дає нам можливість визначити ESB з точки зору зонального шаблону:

- Enterprise Service Bus є зоною, в якій можливості ESB доступні для зовнішньої взаємодії;

- ESB надає набір портів для обслуговування запитів. Кожен порт ідентифікується з конкретним протоколом і набором адрес, через які він забезпечує доступ до ESB;

- ESB використовує набір портів для інтеграції з постачальниками послуг. кожен порт підтримує певний протокол і набір адрес, які є специфічними для послуги, доступ до якої він надає;

- ESB буде містити блоки обробки (концентратори), які застосовуються функції ESB для обслуговування взаємодії між портами користувача сервісу і портами постачальника послуг.

Ці блоки обробки можуть бути реалізовані в будь-якому фізичному шаблоні, але всі мають загальну інфраструктуру адміністрування і конфігурації. Конфігурація може бути частиною розгорнутої інфраструктури ESB або може бути окремо від нього.

Вхідні та вихідні порти визначені наступним чином:

- протокол;
- одна або декілька адрес;
- конкретний спосіб підтримки характеристик взаємодії служб, таких як передача, безпека і так далі.

Вхідний порт може прослуховувати певну адресу за певним протоколом. Тим не менше, кілька вихідних портів можуть посилатися на один і той же адрес через один протокол.

Така модель ESB має ефект визначення області дії окремого ESB:

- окремий ESB забезпечує доступ до чітко визначеного набору послуг реалізованого одним або декількома постачальниками послуг.;

- окремий ESB керується окремою інфраструктурою адміністрування;

- окремий ESB може бути пов'язаний з будь-якою кількістю протоколів, доволі часто обговорюється ідея HTTP Service Bus, при цьому ESB пов'язаний з набором реалізацій послуг в першу чергу, а з набором протоколів в другу;

- інфраструктура ESB може бути фізично розгорнута в будь-якій центральній, кластерній, або розподіленій схемі;
- окрема інфраструктура адміністрування задається, щоб забезпечити єдину точку контролю.

Модель окремої інфраструктури зображує ESB як систему розташовану в межах підприємства (мається на увазі також назва Enterprise Service Bus). Застосовуючи можливості ESB для цієї моделі, можна визначити, що одиниці виконання Hub в рамках ESB повинні підтримувати взаємодії, які відповідають брокеру моделі інтеграції процесів та її варіації, маршрутизатору. Різниця між цими двома моделями втілена в можливості обробки повідомлень - агрегації повідомлень/послуг та кореляції. Дана можливість є корисною, якщо ESB потрібна для підтримки брокера взаємодії, на відміну від взаємодії через маршрутизатор.

Роль ESB як компонента інфраструктури, на відміну від такого компонента як Business Service Choreography, який обробляє бізнес-логіку, означає, що ESB є менш придатним для моделей послідовного процесу і паралельного процесу та їх різновидів робочого процесу; проте, як зазначено в "The Enterprise Service Bus є компонентом інфраструктури", такі відмінності не завжди зрозумілі, і зростання посередницьких можливостей в ESB технології може означати, що збільшення послідовного процесу і паралельних процесів взаємодії підтримуються, але для опорної функції інфраструктури, а не бізнес-логіки.

Технології для реалізації шаблону ESB:

- компонент Web Services Gateway WebSphere Application Server Network Deployment може бути використаний для реалізації Enterprise Service Bus для керування групою послуг, доступних через SOAP/HTTP або SOAP/JMS і виконуваних через SOAP/HTTP, SOAP/JMS, RMI/IIOP, J2C.
- WebSphere Business Integration Message Broker може бути використаний для реалізації в Enterprise Service Bus керування групою послуг, доступних через SOAP/HTTP або SOAP/JMS і виконуваних через SOAP/HTTP,

SOAP/JMS, RMI/IIOP, J2C, EOD, JDBC, забезпечуючи при цьому додаткову обробку повідомлень, моделювання та можливості рівня обслуговування.

Хоча більша частина була присвячена визначенню ESB щодо бізнесу та IT-драйверів для SOA і ESB, потрібно визначити задачі цих драйверів як частину формального визначення моделі ESB:

- забезпечити надійну, керовану, розподілену інфраструктуру інтеграції, відповідно до принципів SOA.

- забезпечити взаємодії з допомогою служб, які визначаються явно незалежними від реалізації інтерфейсами, слабо пов'язані і викликаються через комунікаційні протоколи, які забезпечують прозорість розташування, сумісність і інкапсуляцію повторно використовуваних бізнес-функцій.

- надавати можливості інтегрованій інфраструктурі для підтримки SOA, проміжного програмного забезпечення, що орієнтоване на обробку повідомлень, а також інтеграція керування подіями.

- підтримувати маршрутизації і підміни, перетворення протоколів та інших можливостей обробки повідомлень.

- підтримувати як веб-служби так і традиційні комунікаційні стандарти і технології EAI.

У багатьох сценаріях ESB інфраструктура повинна підтримувати стилі методології SOA для більш складних аспектів взаємодії служб, включаючи безпеку і транзакційність. Кожна можливість реалізована в шині це можливість, яка не повинна бути реалізована в додатках і бути узгодженою між ними.

При моделюванні додаткових компонентів SOA виконується поєднання моделі Enterprise Service Bus з іншими шаблонами інтеграції процесів. Щоб спростити зображення цих комбінацій, варто використовувати модель декомпозиції Enterprise Service Bus рівня 0. Enterprise Service Bus надає послуги обслуговування запитів від обох сторін, так само як це роблять компоненти Business Service Choreography. Основні можливості різних стилів

взаємодії (наприклад, запит/відповідь, публікація/підписка, запустили і забути, події) включають:

- користувачі сервісу можуть зв'язатися з постачальниками послуг з використанням конкретних імен послуг.

- постачальники послуг повинні або підтримувати контакт з користувачами, щоб надати відповідь на запит, або використовувати локальний обробник (для швидкої відповіді), або використовувати інфраструктурні рішення для доставки відповіді клієнту.

- моделі взаємодії, які включають відправлення повідомлень від постачальників послуг до користувачів (наприклад, публікації/підписки або моделі подій) вимагають, щоб постачальник послуг підтримував контакти з клієнтами або надавати форму зворотного зв'язку для клієнтів.

Таким чином, ролі користувача і постачальника послуг досить різні, і тому варто розрізняти їх, навіть коли обидві виконуються в одному додатку або компоненті. Наступним кроком у процесі проектування є вибір конкретної технології, стандартів та інші методи для реалізації обраної моделі.

2.1.3. Канали взаємодії (протоколи)

Для взаємодії між пристроями використовуються основні протоколи: I²C, SPI, UART.

Inter-Integrated Circuit (I²C) – це послідовна асиметрична шина для забезпечення комунікації між інтегральними схемами всередині електронних приладів. Використовує дві двонаправлені лінії зв'язку (SDA і SCL), застосовується для з'єднання низькошвидкісних периферійних компонентів з процесорами і мікроконтролерами.

Дані передаються по двох каналах – канал даних і канал тактів. Є ведучий (master) та ведений (slave), такти генерує master. Всього на одній двопровідниковій шині може бути до 127 пристроїв. Схема підключення - зображена на рис. 2.3.

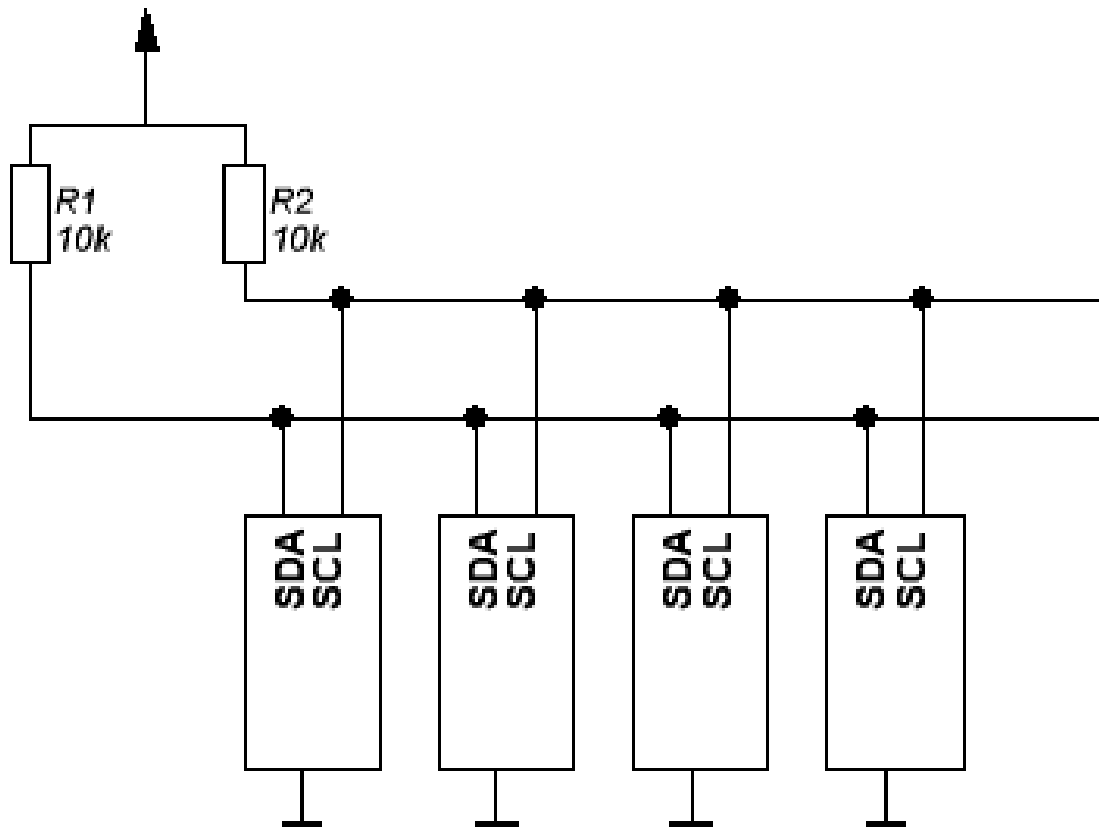


Рис. 2.3. Схема підключення по I²C

Передача даних розпочинається із Start послідовності – SDA встановлює значення 0, а SCL утримує значення 1. Якщо відбувається передача даних від master до slave, то ведучий пристрій керує тактами на лінії SCL і відправляє дані по SDA, які ведений пристрій зчитує.

Якщо передача даних здійснюється від веденого до ведучого пристрою, то master все ще керує каналом тактів, але зчитує дані з SDA. Slave, орієнтуючись по тому, коли SCL встановлено в 0, відправляє свої дані через SDA, які ведучий пристрій зчитає коли встановить SCL в 1. Передача завершується STOP послідовністю. Лінія тактів утримується на рівні 1, а лінія даних змінює низький рівень на високий. Отже, під час передачі даних змінювати наступний біт можна лише, коли лінія тактів утримується на низькому рівні, коли вона на високому – потрібно зчитувати дані. Службові команди, такі як старт, стоп виконуються коли тактова лінія утримується на високому рівні.

Дані надсилаються пакетами, кожен із яких містить у собі дев'ять бітів. Вісім із них займають дані, останній вказує на те, чи потрібно підтверджувати факт прийому. Перший пакет даних містить фізичну адресу пристрою, з яким буде виконуватись взаємодія. На рис. 2.4 показано приклад такого пакету.

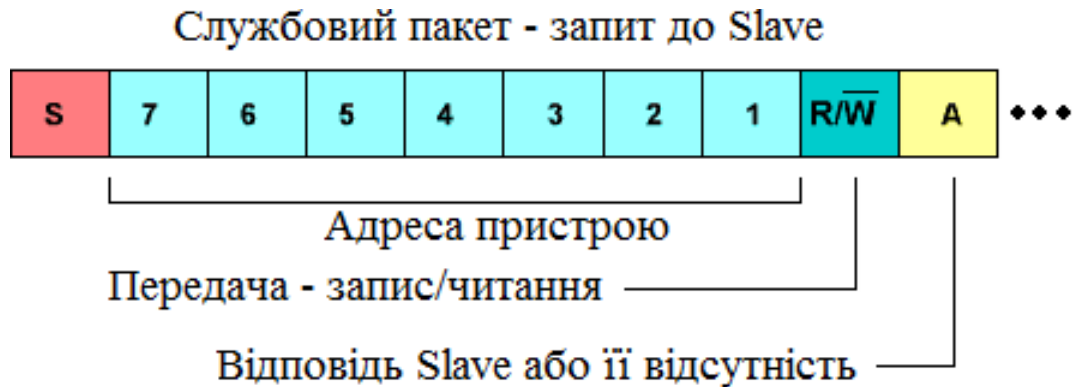


Рис. 2.4. Біти пакету початку передачі

Фізична адреса – це набір із семи бітів, комбінація яких дозволяє забезпечити унікальні адреси для 127 пристроїв на одній шині. Восьмий біт відповідає за команду передачі або зчитування для наступного байту. Дев'ятий біт відповідає за підтвердження або відсутність при отриманні даних (ACK). Якщо ведений пристрій отримує на лінії свою адресу, на 9-му такті він встановлює лінію даних в 0 і генерує відповідь, чим дає зрозуміти ведучому пристрою, що готовий до передачі. Якщо все виконано в такому порядку, ведучий пристрій розуміє, що все правильно і можна продовжити передачу. В іншому ж випадку, якщо відповіді не отримано через те, що ведений пристрій не встиг зчитати адресу, зробив це неправильно, зайнятий іншими процесами або просто відсутній пристрій з такою адресою на лінії, то лінія даних на дев'ятому такті не змінить свого рівня і буде отримано NACK.

На цьому робота з адресним пакетом завершується, наступними ідуть пакети даних між ведучим та веденим пристроєм, в залежності від вказаної команди зчитування чи передачі.

SerialPeripheralInterface (SPI) представляє собою протокол синхронізації лінійного передавання даних, що забезпечує комунікацію контролера з периферійними пристроями. Для SPI характерною ознакою є висока швидкість і невелика відстань передачі даних.

Концепція протоколу SerialPeripheralInterface передбачає наявність одного керуючого пристрою, що забезпечує взаємодію і контроль над іншими периферійними пристроями. При реалізації взаємодії пристроїв використовується три загальні канали зв'язку [7]:

- Master In Slave Out – канал, що забезпечує комунікацію між периферійним і керуючим пристроєм, при цьому подія ініціалізується периферійним (веденим) пристроєм.

- Master Out Slave In – канал, що забезпечує передавання даних від керуючого пристрою до керованого.

- Serial Clock – канал, що забезпечує синхронізацію процесу передачі даних шляхом генерування тактових імпульсів керуючим пристроєм.

Для активації периферійних пристроїв використовується канал зв'язку Slave Select за допомогою якого керуючий пристрій надсилає сервісне повідомлення. Низький рівень сигналу на виводі Slave Select периферійного пристрою сигналізує про звернення і необхідність сприйняття команди від керуючого пристрою. При високому рівні сигналу Slave Select – периферійний пристрій ігнорує звернення від керуючого пристрою.

Архітектура SPI забезпечує взаємодію декількох периферійних пристроїв, які підключеними до тієї самої шини.

Функція SPI.setBitOrder() забезпечує порядок передачі даних, а SPI.setDataMode() – контролює режим роботи. SPI.setClockDivider() представляє собою функцію контролю швидкості передачі даних.

Зазвичай, стандарт SPI передбачає використання 4-ох режимів передачі даних. Ці режими характеризуються умовами зсуву даних та рівнем сигналу. Режими передачі даних у протоколі SPI, які враховують полярність і фазу наведені у вигляді табл. 2.1.

Режими передачі даних в протоколі SPI

№з/п	Режим	Полярність	Фаза
1	SPI_MODE0	0	0
2	SPI_MODE1	0	1
3	SPI_MODE2	1	0
4	SPI_MODE3	1	1

Machine2Machine(M2M) – це технологія, яка забезпечує надійну, просту та одночасно вигідну передачу даних між пристроями. Рішення на основі мобільного M2M з'явилося близько 25 років тому. За оцінками Harbor Research на кінець 2010 року у світі налічувалося кілька десятків мільярдів так званих «розумних» пристроїв (smart devices) – електронних машин, здатних взаємодіяти між собою.

Стаціонарний M2M дозволяє застосовувати використання пристроїв для найрізноманітніших проектів, таких як охоронний моніторинг, керування технологічними процесами, використання лічильників, платіжних терміналів, автоматів роздрібної торгівлі.

До варіантів реалізації M2Mрішень належать:

- Dual-Tone Multi-Frequency (DTMF) – це реалізація, яка надає можливості обміну кодованими повідомленнями через звичайні голосові канали передачі даних. Таке рішення дуже ефективно для забезпечення потреб охорони, простих реалізацій дистанційного керування, реалізацій управління персональними даними. Хорошим прикладом є сервіс голосової пошти, для входу в який потрібно ввести пароль. Більшість мереж використовують такі можливості.

- SMS – це передача коротких повідомлень. Використання цієї технології для користувачів забезпечує розширені можливості. Основною перевагою є те, що повідомлення надсилає за кілька мілісекунд велику кількість інформації.

– DATA – це модемний зв'язок. Дана реалізація передбачає використання комутованих каналів, які забезпечують низьку швидкість передачі інформації. Системи передачі пакетної інформації, такі як GPRS або CDPD на даний момент одними із найпоширеніших та дозволяють використовувати інтерактивні комунікації в режимі реального часу.

– AlwaysonDATA – це модемний зв'язок із використанням виділених каналів з вищою швидкістю передачі даних, ніж у DATA. Ця реалізація є ключовою у напрямку розвитку M2M.

– WAP-IP – це реалізація, яка розроблена для проектів на основі взаємодії людина-машина, і з часом вони все більше використовуються як стандарт для електронних замовлень, платежів та покупок. Реалізація об'єднує два елементи: інформаційний сервіс і додатки для управління та моніторингу. Також серед можливостей WAP-IP як торговельної мережі є можливість розробки платформи, яка б забезпечила розвиток антивандальних та охоронних систем для торговельних автоматів.

– 3G – це третє покоління мереж стільникового зв'язку, яке надає необмежені засоби в телекомунікаційному середовищі, але лише в той момент, коли абсолютно усе обладнання в мережі та пристрої підтримують та відповідають одному протоколу.

До проектів, на основі M2M належать:

– системи надають можливості розмежування доступу до ресурсів, базуючись на користувачах чи групах користувачів. За допомогою телефону користувач отримує доступ до ресурсів шляхом дзвінка або відправлення коду доступу;

– системи охорони приміщень – це M2M-рішення, яке дозволяє забезпечувати дистанційну охорону приміщень через стільникову мережу. Такі рішення забезпечують користувачам можливості керування увімкненням/вимкненням системи, моніторингу стану завдяки передачі зображень або відео через GSM мережу;

- системи безпеки. Бездротові системи безпеки до яких відносяться системи протипожежної, аварійна, персональна безпеки. M2M надає можливості дистанційно слідкувати за станом об'єктів та при необхідності надсилати інформацію про стан об'єктів або сигнали тривоги;

- дистанційний контроль та керування пристроями. За допомогою мобільного телефону можна дистанційно керувати домашньою технікою, здійснювати моніторинг її стану, підтримувати кліматичні умови в приміщеннях за допомогою кондиціонерів, обігрівачів та іншої кліматичної техніки.

- автотранспорт і моніторинг рухомих об'єктів. Застосування систем M2M для транспорту можна сказати практично необмежене. Реалізації включають в себе автоматичні повідомлення про виникнення аварій в службу порятунку, при цьому надаючи точні координати місця; системи захисту автомобіля від викрадення шляхом забезпечення інформації про місцезнаходження транспортного засобу в реальному часі, також інформацію про технічний стан транспортного засобу;

- використання в автоматах з продажу різної продукції, кавових автоматах, платіжних терміналах, банківських, касових автоматах та ін. M2M надає можливості дистанційного моніторингу та контролю таких апаратів, можливості забезпечення охорони, збирання інформації в процесі роботи про стан роботи та працездатність обладнання;

- охорона здоров'я. M2M надає можливості проведення віддаленої діагностики пацієнтів, надання допомоги людям з обмеженими фізичними можливостями, автоматизації викликів в екстрених ситуаціях.

В якості каналів комунікації між хмарним сервісом та кінцевими пристроями можуть бути використані протоколи AMQP, HTTP та MQTT. Для використання інших протоколів потрібно задіяти апаратний або програмний шлюз протоколу.

AdvancedMessageQueningProtocol (AMQP) – це відкритий протокол, який використовується для обміну повідомленнями між елементами системи.

Принцип роботи заснований на тому, що елементи здійснюють обмін повідомленнями через брокер, який відповідає за маршрутизацію, доставку, керування передачею.

AMQP протокол представляє собою двійковий протокол з сучасними функціями: багатоканальність, асинхронність, безпека, портативність та ефективність. AMQP доцільно розділити на функціональний та транспортний шар. Функціональний шар – це основні операції, обмін чергами повідомлень, транспортний – це упакування даних, зміст, представлення даних, обробка помилок, керування частотою каналів.

HyperTextTransferProtocol (HTTP) – це протокол обміну даними, який використовується в комп'ютерних мережах. Головною задачею цього протоколу є передача веб-сторінок, але його також використовують для передачі інших даних.

Структура даних, які передаються розділена на три частини:

- стартовий рядок, який визначає метод, адресу та версію протоколу;
- заголовки, які містять службову інформацію;
- тіло повідомлення, що містить інформацію, яку буде передано.

Наявність заголовків займає значну частину повідомлення, що збільшує розмір і, відповідно, трафік в мережі, через що такий поширений прокол може бути недоречним для використання у мобільних мережах, або для обміну невеликої кількістю даних між кінцевими пристроями. Але є набагато ефективніший M2M протокол MQTT.

MQTT – спрощений мережевий протокол, що працює над протоколом TCP/IP і забезпечує обмін повідомленнями між пристроями на основі технології видавець-передплатник.

До основних можливостей протоколу належить:

- простота у використанні – забезпечується мінімально-необхідним набором функцій та може легко інтегруватись у будь-яку систему;
- зручність використання – модель видавець-передплатник забезпечує можливість одержувати дані з давачів, які не були наперед визначеними.;

- простота адміністрування;
- мінімальне навантаження на канал зв'язку;
- функціонування при втраті зв'язку або інших проблем на лінії;
- відсутність обмежень на формат переданого контенту.

Але, як зазначалось раніше, хмара підтримує тільки захищене з'єднання, тому потрібно шифрувати дані, використовуючи SSL (TLS).

2.2. Проектування та формалізація компонентів архітектури системи «розумний цінник» на рівні апаратного забезпечення і каналів зв'язку

На основі аналізу принципів організації архітектури «розумних систем», проведеного у першому розділі дипломної роботи магістра, запропоновано універсальну архітектуру у вигляді, як показано на рис. 2.5.



Рис. 2.5. Універсальна архітектура системи «розумний цінник»

Як видно з рис. 2.5 до системи «розумний цінник» входять:

- Client devices – пристрої, які є кінцевими у комп'ютерній системі і можуть включати різні типи давачів або безпосередньо виконувати функції керування чи моніторингу, або пристрої контролю візуалізації даних;
- Field gateways – шлюзи, які забезпечують зв'язок з кінцевими пристроями і можуть концентрувати дані для подальшої передачі із визначеним ступенем захисту інформації;

– *Service access point* – сервіси, що розміщені у хмарі, основна функція яких полягає у забезпеченні контрольованого доступу до ресурсів та масштабованості, яка пов’язана із зростанням навантаження на сервіс.

– *Storages, services, runtime* – програмне забезпечення, що виконує логіку роботи комп’ютерної системи, залежно від призначення.

Запропоноване представлення такої архітектури системи «розумний цінник» дає змогу забезпечити захист даних, які генерують кінцеві пристрої, врахувати надійність при територіальному розподілі компонентів комп’ютерної системи, забезпечити гнучкість і масштабованість рішень у конкретно взятій предметній області.

Проведемо формалізацію компонентів архітектури системи «розумний цінник».

В загальному випадку архітектуру можна представити у вигляді кортежа

$$\text{SmartPriceArch} = \langle ED, FG, SA, SW \rangle \quad (2.1)$$

де ED – множина кінцевих пристроїв;

FG – множина шлюзів для з’єднання кінцевих пристроїв з cloud-сервісами;

SA – множина сервісів для забезпечення захищеності та масштабованості на стороні cloud інфраструктури;

SW – множина програмного забезпечення, що включає сервіси, сховища і додаткове програмне забезпечення для функціонування системи «розумний цінник».

Множину кінцевих пристроїв можна описати у вигляді структури, елементами якої є тип пристрою, його характеристики і назва

$$ED = \{ ED_i, EDChar_{ij}, EDType_i \} \quad (2.2)$$

де ED_i – назва i -го кінцевого пристрою в комп'ютерній системі, $i=1, N$, N – кількість кінцевих пристроїв;

$EDChar_{ij}$ – характеристики i -го кінцевого пристрою в комп'ютерній системі, $j=1, K$, K – кількість технічних характеристик i -го кінцевого пристрою;

$EDType_i$ – тип i -го кінцевого пристрою в комп'ютерній системі.

Для забезпечення збору, відображення і подальшого надсилання даних до або з cloud-сервісів використовуються локальні шлюзи, що можуть використовувати різні технології передачі даних. Серед технологій передачі даних можуть бути використані Bluetooth, Wi Fi, Ethernet та ін. Формально шлюз можна описати наступним чином

$$FG = \{ FG_k, FGchar_{ki}, ED_{kj}, FG Prot_{kj} \} \quad (2.3)$$

де FG_k – назва k -го шлюза, $k=1, M$, M – кількість шлюзів комп'ютерної системи;

$FGchar_{ki}$ – множина характеристик, $i=1, I$, I – кількість характеристик k -го шлюза;

ED_{kj} – множина кінцевих пристроїв, під'єднаних до k -го шлюза $j=1, J$, J – кількість кінцевих пристроїв;

$FG Prot_{km}$ – множина протоколів, що підтримуються, $m=1, M$, M – кількість підтримуваних протоколів k -им шлюзом.

Точки доступу на стороні cloud можна описати наступним чином

$$SA = \{ CL_c, SA_{ca}, FG_{af} \} \quad (2.4)$$

де CL_c – назва c -го cloud, $c=1, C$, C – кількість cloud на яких розгорнута комп'ютерна система;

SA_{ca} – множина сервісів, які забезпечують функціонування, перевірку і масштабованість IoT компонентів комп’ютерної системи на c -му cloud, $a = 1, A$, A – кількість сервісів, розміщених на c -му cloud;

FG_{af} – множина локальних шлюзів, що взаємодіють з a -им сервісом, $f = 1, F$, F – кількість локальних шлюзів, що взаємодіють з a -им сервісом;

Для опису програмного забезпечення чи сервісів, зокрема збору інформації про товари і їх ціни у конкурентів та інтелектуальний сервіс формування ціни, які виконують опрацювання і передачу даних до IoT компонентів, а також керування ними можна описати у вигляді

$$SW = \{ CL_c, SA_{ca}, SW_{cw} \} \quad (2.5)$$

де SW_{cw} – множина сервісів чи іншого програмного забезпечення, які розміщені на c -му cloud і взаємодіють a -им сервісом.

Таким чином, розроблено уніфіковану модель архітектури системи «розумний цінник» та проведено формалізацію її складових. Наступний крок полягає у формалізації процесу збору даних щодо товарів і їх вартості у конкурентів. Для вирішення цієї задачі запропоновано скористатись підходом, що передбачає використання парсерів з можливістю їх запуску за розкладом.

2.3. Математичне забезпечення процесу збору даних у системі «розумний цінник»

У загальному випадку система збору даних про товари представляє собою множину елементів, підсистем і комунікацій між ними. Процес збору даних P можна розглядати як деякий процес над яким виконується управління ззовні, тобто розклад запуску парсерів відносно конкретних ресурсів. При цьому, парсери та їх кількість, можна описати множиною агентів, над якими виконується управління і на які здійснюється вплив – A_1, \dots, A_n , n – кількість

парсерів. Розклад запуску парсерів представляє собою множину керуючих сигналів k , $\{k \in K\}$. Окрім сигналів керування, парсерам відомо про сигнали зовнішнього середовища ε , сигнали відповіді від ресурсів на запит z , $\{z \in Z\}$, та про взаємодію різних парсерів-агентів між собою r , $\{r \in R\}$. Дана сукупність сигналів формують множину виходів Y – результат виконання парсингу. Формально процес керування парсингом товарів можна представити у вигляді відображення:

$$C(P_i): K \times Z \times R \times \varepsilon \rightarrow Y \quad (2.6)$$

Оскільки, існує множина агентів-парсерів, то відповідно в асинхронному режимі можна виконувати сукупність процесів P_n . Відповідно, для запуску агентів існує множина керуючих сигналів K для кожного процесу P_i , що представляється у вигляді комбінації усіх можливих сигналів керування, тобто $K = K_1 \times \dots \times K_n$ – декартовий добуток множини керуючих сигналів.

Нехай користувач або система запуску подій має можливість визначати та обирати керуючий сигнал k , $\{k \in K\}$, спрямовувати його на виконання деякого процесу P_i , і викликати агент A_i . Як наслідок, результат виконання процесу виражається через вихідні сигнали агентів, які представляють реакції агента на подію.

При цьому, керування процесом P_i породжує реакцію:

$$C_i: K_i \times P_i \rightarrow Z_i \times R_i \quad (2.7)$$

Вихідну множину результатів виконання процесу, яка представлена множиною Y , можна записати у вигляді кортежу:

$$Y: \langle K_i \times Z_i \times R_i \rightarrow K_{i+1} \rangle \quad (2.8)$$

З цього випливає, що при взаємодії сигналів з результатами парсингу можна одержати нові керуючі сигнали.

Комп'ютерну систему можна представити у вигляді множини станів, які змінюються впродовж її життєвого циклу. Виходячи з цього, сам життєвий цикл можна представити у вигляді множини

$$I = (X, C, P, \Omega, \varphi) \quad (2.9)$$

де X – сукупність станів комп'ютерної системи;

C – множина керуючих сигналів, що переводять систему з одного стану в інший, тобто $\forall c \in C, c : X_i \rightarrow X_{i+1}$;

P – сукупність процесів щодо парсингу товарів;

Ω – множина факторів зовнішнього впливу, що формує сукупність відображень $\forall \omega \in \Omega, \omega : X_i \rightarrow X_{i+1}$;

$\varphi(x)$ – відображення, що описує розвиток системи протягом життєвого циклу

$$\varphi : C(P) \rightarrow X \quad (2.10)$$

У відповідності до представлення процесу парсингу, під керуванням даним процесом і життєвим циклом системи одержання даних про товари і ціни конкурентів, слід розуміти послідовність c_1, c_2, \dots, c_k , що переводить стан системи парсингу x у стан x_k . При цьому $c_1(x_0)=x_1, \dots, c_k(x_{k-1})=x_k$.

У результаті виконаного формального представлення парсера щодо збору даних про товари у конкуруючих організацій, пропонується для його практичної реалізації використати платформи з можливістю налаштування параметрів пошуку даних, запуску агентів за розкладом та асинхронним їхнім функціонуванням.

2.4. Формалізація процесу аналізу даних про товари у системі «розумний цінник»

Для формування цін та рекомендацій товарів споживачам при проектуванні систем «розумний цінник» пропонується скористатись методами колаборативної фільтрації, введемо наступні поняття.

Нехай r_x буде вектором оцінок користувача x , N – множина k користувачів, які найбільше подібні до користувача x , що також мають оцінки товару (послуги) i . Необхідно здійснити прогнозування оцінок товарів на множині I для користувача x .

Перший варіант прогнозування оцінок товарів для нового користувача полягає у знаходженні середнього значення по кожному товару, тобто

$$r_{xi} = 1 / k \sum_{y \in N} r_{yi} \quad (2.11)$$

де k – кількість оцінок товару u_i ;

r_{yi} – оцінки товару u_i користувачами матриці крос-табуляції.

Приклад матриці крос-табуляції та обчислення прогнозованих значень оцінок товарів для нового користувача наведено на рис. 3.7. При цьому, обчислення самих значень оцінок товарів для користувача E проводилось за формулою (2.11)

$$r_{E1} = 1 / 2(4 + 5) = 4,5 ;$$

$$r_{E2} = 1 / 2(5 + 3) = 4 ;$$

$$r_{E3} = 4 ;$$

$$r_{E4} = 1 / 2(5 + 2) = 3,5 ;$$

$$r_{E5} = 1 / 2(1 + 4) = 2,5 ;$$

$$r_{E6} = 5 ;$$

$$r_{E7} = 3 .$$

Результат прогнозування оцінок товарів наведений на рис. 2.6.













							
	Product 1	Product 2	Product 3	Product 4	Product 5	Product 6	Product 7
A 	4			5	1		
B 	5	5	4				
C 				2	4	5	
D 		3					3
E 	4,5	2,67	4	3,5	2,5	5	3

Рис. 2.6. Приклад обчислення прогнозованих значень оцінок товарів для користувача *E*

У даному випадку, використовувався тип фільтрації «користувач-товар». Хоча аналогічним чином можна застосувати тип фільтрації «користувач-користувач». Перевагою прогнозування значень фільтрації за формулою (2.11) є вирішення питання «холодного старту» у випадку, коли з'явився новий товар або користувач і для нього необхідно визначити ціну чи сформувати рекомендації. Однак, даний підхід не є ефективним, коли користувачі уже мають сформовані свої корзини товари. У такому випадку використовується інший підхід до визначення прогнозованих оцінок. Розглянемо його більш детально з точки зору колаборативної фільтрації «користувач-користувач».

Формально постановки задачі можна задати наступним чином:

- для товару *i* знайти інші подібні товари;
- провести прогнозування значень оцінки для товару *i* на основі оцінок подібних товарів;
- існує можливість застосування таких самих метрик і функцій прогнозування як у моделі «користувач-користувач»;

- $sim(i,j)$ – подібність товару i та j ;
- r_{xi} – оцінки користувача x щодо товару j ;
- $N(i,x)$ – множина товарів, оцінених користувачем x і подібних до товару i .

У цьому випадку будемо використовувати та обчислювати метрику подібності за формулою

$$r_{xi} = \frac{\sum_{j \in N(i;x)} sim(i,j) \cdot r_{xj}}{\sum_{j \in N(i;x)} sim(i,j)} \quad (2.12)$$

Застосуємо на практиці наведені вище підходи до прогнозування значень оцінки вхідної матриці крос-табуляції, що наведена на рис. 2.7.

		<i>users</i>											
		1	2	3	4	5	6	7	8	9	10	11	12
Products	1	1		3			5			5		4	
	2			5	4			4			2	1	3
	3	2	4		1	2		3		4	3	5	
	4		2	4		5			4			2	
	5			4	3	4	2					2	5
	6	1		3		3			2			4	

Рис. 2.7. Вхідна матриця крос-табуляції, для якої необхідно обчислити значення невідомих оцінок

Жовтим кольором у матриці наведено відомі значення оцінок, а білим – невідомі. Для початку необхідно визначити товари, які подібні до товару 1, скориставшись метрикою подібності на основі коефіцієнту кореляції Пірсона.

Процедура визначення подібних товарів полягає в наступному:

– обчислюємо середнє значення оцїнок m_j для усїх рядкїв матрицї, тобто для кожного товару: $m_1 = (1 + 3 + 5 + 5 + 4)/5 = 3,6$. У результатї одержимо рядок – (-2,6;0;-0,6;0;0;1,4;0;0;1,4;0;0,4;0)

– обчислюємо косинусну мїру кута.

На рис. 2.8 наведено матрицю з обчисленими коефїциєнтами подїбностї (справа вїд матрицї).

		<i>users</i>												
		1	2	3	4	5	6	7	8	9	10	11	12	<i>sim (1,m)</i>
Products	1	1		3			5			5		4		1
	2			5	4			4			2	1	3	-0,18
	3	2	4		1	2		3		4	3	5		<u>0,41</u>
	4		2	4		5			4			2		-0,1
	5			4	3	4	2					2	5	-0,31
	6	1		3		3			2			4		<u>0,59</u>

Рис. 2.8. Матриця з обчисленими коефїциєнтами подїбностї товарїв

Наступна задача полягає у прогнозуваннї значення оцїнки для товару 1 користувачем 5. Їз обчислених значень коефїциєнтїв подїбностї видно, що товар 1 найбільш подїбний до товарїв 3 та 6. У випадку, коли взяти норму колаборативної фїльтрацїї 2, можна спрогнозувати значення оцїнки або цїни товару 1 користувачем 5 за формулою середнього зваженого (2.12).

У результатї отримуємо:

$$r_{15} = (0,41 * 2 + 0,59 * 3) / (0,41 + 0,59) = 2,6$$

На рис. 2.9 наведено матрицю їз визначеним прогнозованим значенням оцїнки для товару 1 користувачем 5.

		<i>users</i>												
		1	2	3	4	5	6	7	8	9	10	11	12	<i>sim (1,m)</i>
Products	1	1		3		2,6	5			5		4		1,00
	2			5	4			4			2	1	3	-0,18
	3	2	4		1	2		3		4	3	5		<u>0,41</u>
	4		2	4		5			4			2		-0,10
	5			4	3	4	2					2	5	-0,31
	6	1		3		3			2			4		<u>0,59</u>

Рис. 2.9. Обчислене значення оцінки для товару 1 користувачем 5

Таким чином, у результаті дослідження та аналізу методів колаборативної фільтрації, встановлено доцільність їх застосування для формування ціни товарів, шляхом створення відповідного сервісу, який може бути інтегрованим у систему «розумний цінник».

2.5. Конструювання архітектури системи «розумний цінник» на структурному рівні

Зі сторони хмарного сервісу, а саме Azure, для взаємодії можна використовувати сервіси:

- Event Hub;
- IoT Hub.

Концентратор подій Microsoft Azure - це служба керованої платформи, яка забезпечує основу для отримання великих обсягів даних в самих різних сценаріях. Стандартна роль концентраторів подій в архітектурі рішення - роль «двері» для конвеєра подій, який часто називають поглиначем подій. Приймач подій - це компонент або служба, що розташовуються між джерелами подій і адресатами подій і відокремлюють створення потоку подій від використання цих подій, що показано на рис. 2.10.

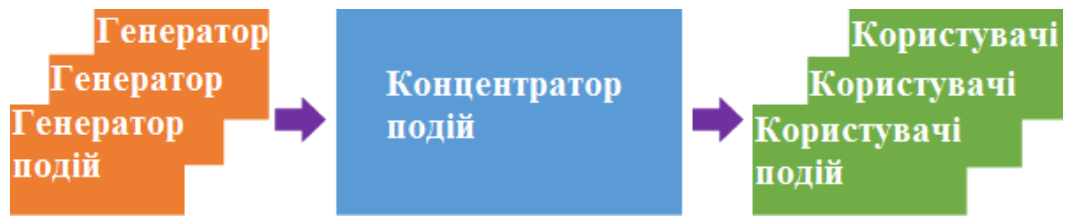


Рис. 2.10. Концентратор подій

Концентратори подій Microsoft Azure - це служба обробки подій, що використовується для великомасштабної передачі подій і даних телеметрії в хмару при низькій затримці і з високою надійністю. Ця служба при використанні разом з іншими підлеглими службами особливо корисна в сценаріях інструментування додатків, забезпечення зручності роботи користувача або функціонування робочих процесів, а також Інтернету речей (IoT). Концентратори подій надають можливість обробки потоку повідомлень з високою пропускнуою здатністю[10].

Концентратори подій використовують шаблон розподілених на секції споживачів, в якому кожен споживач зчитує тільки певну підмножину, або секцію, потоку повідомлень. Цей шаблон забезпечує горизонтальне масштабування для обробки подій. Секція - це впорядкована послідовність подій, що зберігається в концентраторі подій. У міру надходження нових подій вони додаються в кінець цієї послідовності, що показано на рис. 2.11.



Рис.2.11. Секція концентратора подій

Секції зберігають дані протягом заданого часу зберігання, яке задається на рівні концентратора подій.

Будь-яка сутність, яка відправляє події або дані в концентратор подій, є видавцем подій. Видавці подій можуть публікувати події за допомогою HTTPS або AMQP 1.0.

Будь-яка сутність, що зчитує дані з концентратора подій, є споживачем подій. Всі споживачі подій зчитують потік подій через секції в групі споживачів. Всі споживачі концентраторів подій підключаються через сеанс AMQP 1.0, в якому події доставляються, як тільки стають доступними. Клієнту не потрібно проводити опитування доступності даних.

Пропускна здатність концентраторів подій контролюється за допомогою одиниць пропускної здатності. Одиниці пропускної здатності - це попередньо придбані одиниці місткості. Одна одиниця пропускної здатності включає до 1 МБ в секунду або 1000 подій в секунду для вхідного потоку даних та до 2 МБ в секунду для вихідного потоку.

Концентратори подій Azure надають службу обробки великомасштабних подій і даних телеметрії, яку можна використовувати для моніторингу стандартних додатків і користувальницьких робочих процесів будь-якого масштабу. Завдяки можливості публікації і підписки з низькою затримкою і з необмеженим масштабуванням концентратори подій стають "трампліном" для великих обсягів даних. Завдяки посвідченням на базі видавця і списками відкликання ці можливості використовуються в поширених сценаріях Інтернету речей.

Azure IoT Hub представляє собою сервіс управління, що дає змогу забезпечити надійність та безпечність двосторонньої комунікації між багатьма розумними пристроями та програмними рішеннями. Засоби Azure IoT Hub дозволяють забезпечити наступні можливості:

- надійність передачі даних в напрямку device-cloud і cloud-device з одночасним масштабуванням кількості пристроїв;
- наявність засобів ідентифікації пристроїв шляхом перевірки облікових даних та ключів доступу для підвищення безпеки взаємодії між пристроями;

- моніторинг та управління підключеними пристроями;
- підтримка бібліотек мов програмування і платформ.

Для підключення, безпечного та надійного використання розумних пристроїв використовується IoT Hub і бібліотеки конкретних пристроїв, що дає змогу розробляти та впроваджувати засоби керування логікою виконання на рівні back-end рішень.

«Розумні пристрої», при використанні їх разом з cloud-сервісами, забезпечують :

- представлення пристроїв, як вбудованих системи без втручання користувача;
- фізичне розташування у віддалених, недоступних або небезпечних для людини місцях;
- доступ до керування пристроєм може бути організований лише через програмне рішення (back-end);

При використанні «розумних пристроїв» можуть виникати наступні ситуації:

- підключення до повільних або дорогих мереж передачі даних;
- споживання енергоресурсів не на належному рівні;
- застосування галузевих корпоративних стандартів і протоколів.
- неоднорідність апаратного і програмного забезпечення [22].

Крім наведених вище критеріїв, при реалізації розумних систем, необхідно забезпечити критерії надійності, продуктивності, безпеки, переносимості та гнучкого масштабування. Традиційні технології, по типу контейнерів чи брокерів обміну повідомлень, не дають в повному обсязі забезпечити виконання наведених вимог. Однак Azure IoT Hub забезпечує вирішення проблем з підключенням «розумних пристроїв» наступним чином:

- аутентифікація та безпечне підключення – для кожного пристрою передбачено власний ключ захисту, який дає змогу підключитися до IoT Hub, реєстр якого зберігає ідентифікаційні дані пристрою;

– моніторинг процесу підключення пристрою – забезпечує можливість одержання детальної інформації про операції управління ідентифікацією і подій підключення конкретного пристрою, що дозволяє визначити стан або проблеми з підключенням пристроїв. У результаті можна одержати інформацію про частоту звернень від пристрою до cloud-сервісу, тип даних ідентифікації, помилки при аутентифікації пристроїв.

– наявність набору бібліотек пристроїв – підтримка мов програмування і відповідних бібліотек, зокрема мови програмування C під дистрибутиви Linux, операційну систему Windows та інші. Бібліотеки також підтримуються для таких мов як C #, Java і JavaScript.

– IoT протоколи та розширення – у випадку відсутності стандартного рішення, IoT Hub забезпечує можливість використання протоколів MQTT v3.1.1, HTTP 1.1, AMQP. Існує можливість налаштування шлюзу протоколу Azure IoT, що забезпечує підтримку користувацьких протоколів для взаємодії в межах «розумних систем»;

– масштабованість. Azure IoT Hub масштабується для багатьох одночасно підключених пристроїв і великої кількості подій за одиницю часу.

Ці переваги є спільними для багатьох моделей комунікації. На рис. 2.12 зображена структура та можливі схеми підключення пристроїв до IoT Hub.

IoT Hub забезпечує підтримку наступних моделей комунікації:

Модель device-to-cloud на основі подій пристрою. IoT концентратор організований таким чином, що може одержувати мільйони подій в секунду з розумних пристроїв із визначеним рівнем надійності. Після одержання подій, можливе опрацювання даних з використанням відповідного процесора. Крім, того концентратор може зберігати дані про події до 7 днів, тим самим забезпечуючи високу надійність та здатність до гарантування обробки даних при пікових навантаженнях.

Модель cloud-to-device. Back-end рішення використовує IoT Hub для передачі даних до пристроїв і передбачає, що дані хоча б один раз будуть доставлені до адресата. Існує можливість індивідуального налаштування

параметрів повідомлень щодо терміну їх використання, хоча існує можливість створення запиту на перевірку існування повідомлення чи його доставку. Це формує відкритість життєвого циклу повідомлення у моделі cloud-to-device.

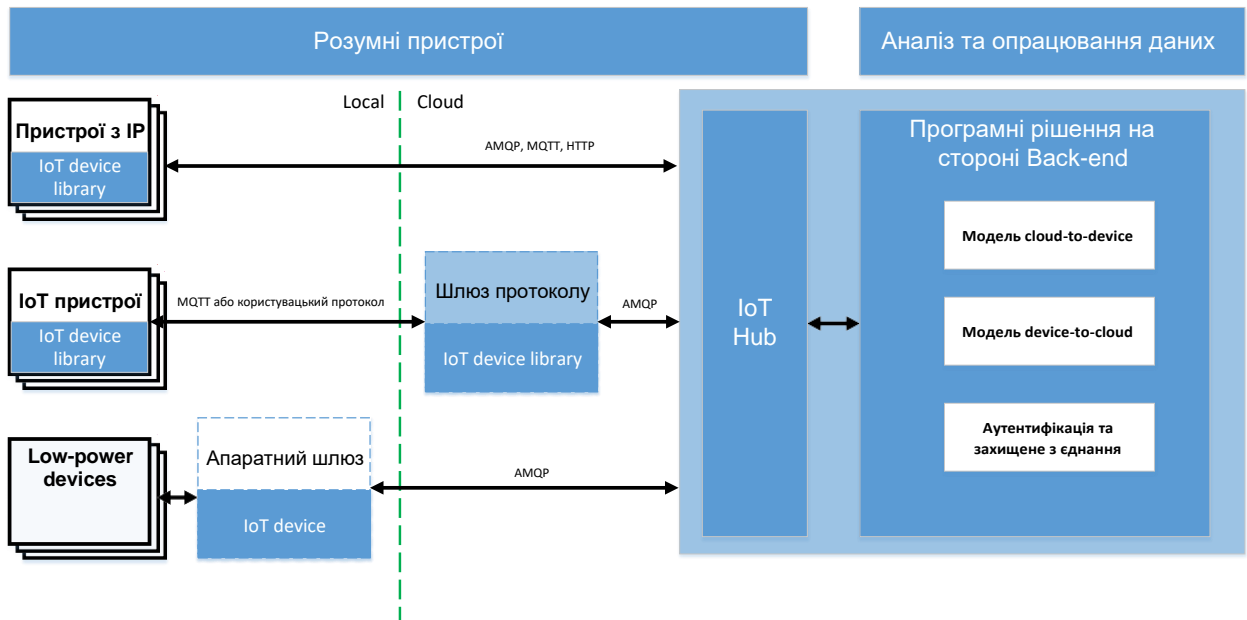


Рис. 2.12. Структура та можливі методи підключення

Крім того, існує можливість виконання інших функцій, таких як завантаження та скачування файлів. Ці функції включають в себе управління пристроями, моніторинг підключення і масштабування.

Для обміну даними можуть бути використані протоколи HTTP, MQTT, AMQP але тільки з захищеним з'єднанням за допомогою SSL(TLS). А за допомогою використання апаратного чи програмного шлюзу, можна використовувати практично будь-який протокол, який найкраще підходить для конкретного рішення.

Для IoT рішень корисними також будуть сервіси StreamAnalytics та PowerBI. Microsoft Power BI - набір засобів бізнес-аналітики, а Azure Stream Analytics надає можливості обробки потоків даних в реальному часі [16]. Йдеться про мільйони подій в секунду[17]. Це дозволяє порівнювати і погоджувати безліч потоків, працювати з даними, використовуючи SQL, створювати інформаційні панелі, налаштовувати нагадування.

Для реалізації комп'ютерної системи необхідно розробити її архітектуру. В якості архітектури вибрано модель з виділеним сервером прикладних додатків та сервером даних. На рис. 2.13 зображена розроблена архітектура системи автоматизації процесів на базі IoT та хмарних сервісів.

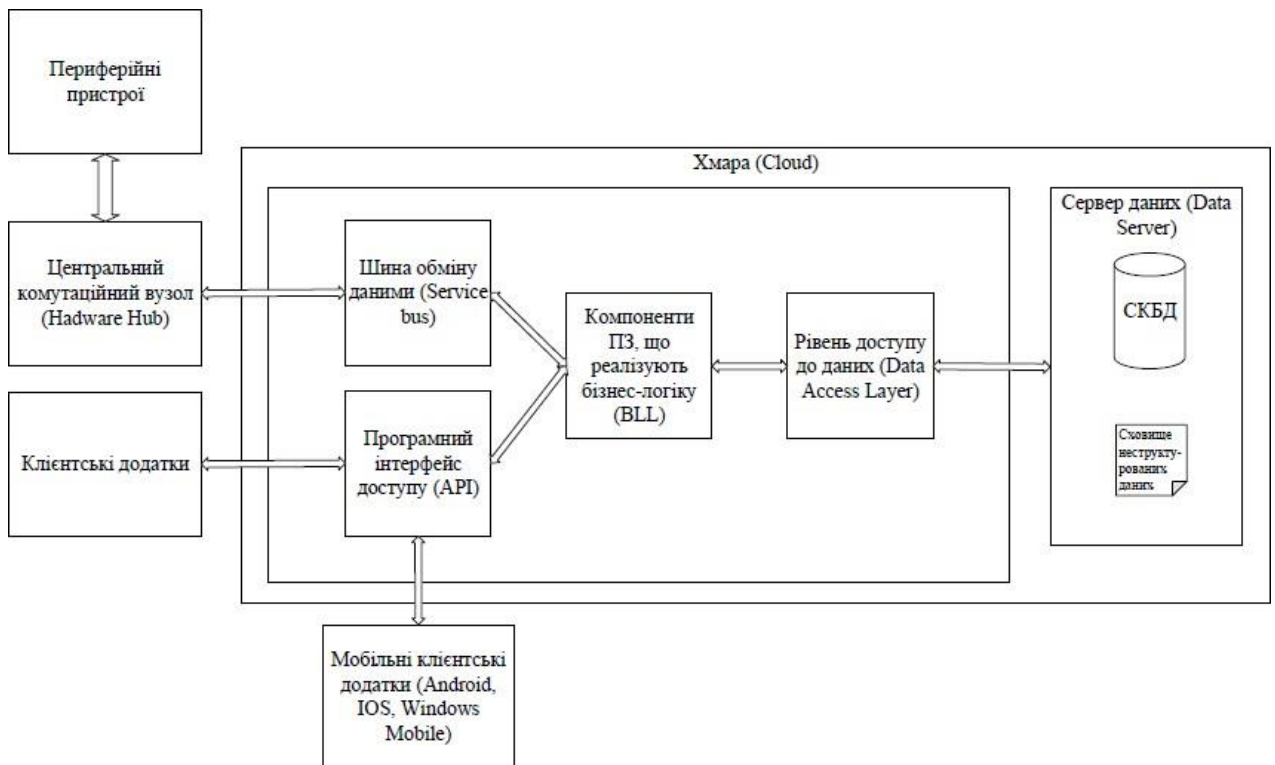


Рис. 2.13. Архітектура системи

Периферійні пристрої – набір електронних цінників та можливих додаткових апаратних модулів.

Центральний апаратний комутаційний вузол – одна або декілька NodeMCU.

Шина обміну даними – призначення для керування отриманими повідомленнями та чергами.

Компоненти ПЗ, що реалізують бізнес логіку – основа системи, яка займається обробкою усіх даних та керуванням.

Рівень доступу до даних – частина ПЗ, що надає доступ до даних, збережених в базах.

Сервер даних – являє собою СКБД та БД для збереження інформації. Окрім структурованої інформації також можуть бути і неструктуровані дані, по типу логів роботи системи. Також можливий доступ до існуючої БД обліку товару в торгових точках для розвитку існуючої інфраструктури без її повної заміни.

Програмний інтерфейс доступу – інтерфейс для розмежування доступу різних користувачів у системі.

Клієнтські додатки – додатки для адміністраторів, бухгалтерів та касирів для доступу до системи.

Мобільні клієнтські додатки – клієнтські додатки, реалізовані для доступу з мобільних платформ та/або через веб-інтерфейс, що дозволяє не прив'язувати користувача до конкретного місця чи ПК.

Серед найбільш поширених представників хмарних сервісів присутні AmazonWebServices, WindowsAzure, GoogleCloud, IBMCloudService.

Хмарне рішення дозволяє отримати ряд переваг над звичайним рішенням. Використання хмарних технологій дозволяє легко масштабувати кількість серверів прикладних додатків при зростанні кількості запитів і трафіку та серверів зберігання даних при зростанні об'ємів даних чи кількості складних вибірок. Також їх можна масштабувати незалежно один від одного без зниження продуктивності заздалегідь при досягненні критичних значень використання наявних ресурсів. Це дозволяє використовувати ресурси відповідно до рівня поточного навантаження на систему, навіть якщо потреби додатку в ресурсах часто змінюються або непередбачувані.

Хмара забезпечує безвідмовність роботи системи, що в основному становить 99,99% та необхідність обслуговування серверів переноситься від власника системи на постачальника послуг.

В системі дуже легко підключити аналітику по будь-яких даних, які наявні в системі. Це дозволяє в межах проекту легко реалізувати наступні сценарії:

- моніторинг використаних ресурсів;

- аналіз та контроль затрачених коштів на підтримку системи;
- аналіз трафіку даних від датчиків та їх стан;
- моніторинг зміни параметрів елементів протягом тривалих проміжків часу та їх візуалізація.

Але в невеликих об'ємах проектів використання хмарних технологій є нерентабельним і всі перераховані переваги ними не являються, оскільки в таких проектах не потрібна масштабованість, надійність забезпечується адміністратором, а об'єми даних занадто малі для реалізації системи аналітики та її підтримки в хмарі.

Модель системи з відображення даних з БД розробляється на основі уніфікованої архітектури, розробленої в попередньому пункті.

Периферійні пристрої – набір датчиків та пристроїв виведення інформації та можливих додаткових апаратних модулів.

Центральний апаратний комутаційний вузол – засіб зв'язку між кінцевими апаратними елементами та програмною частиною, керує потоком даних від і до кінцевих елементів.

Шина обміну даними – IoTHub.

Компоненти ПЗ, що реалізують бізнес логіку – можлива реалізація користувачем своєї логіки, або на основі StreamAnalytics.

Рівень доступу до даних – частина IoTHub, SA або клієнтського ПЗ для доступу до даних, збережених в базах.

Сервер даних – залежить від даних та необхідного виду збереження та доступу до них, найімовірніше SQL сервер.

Програмний інтерфейс доступу, клієнтські додатки, мобільні клієнтські додатки реалізується на основі вимог проекту.

На рис. 2.14 зображена архітектура системи з виводом інформації з БД.

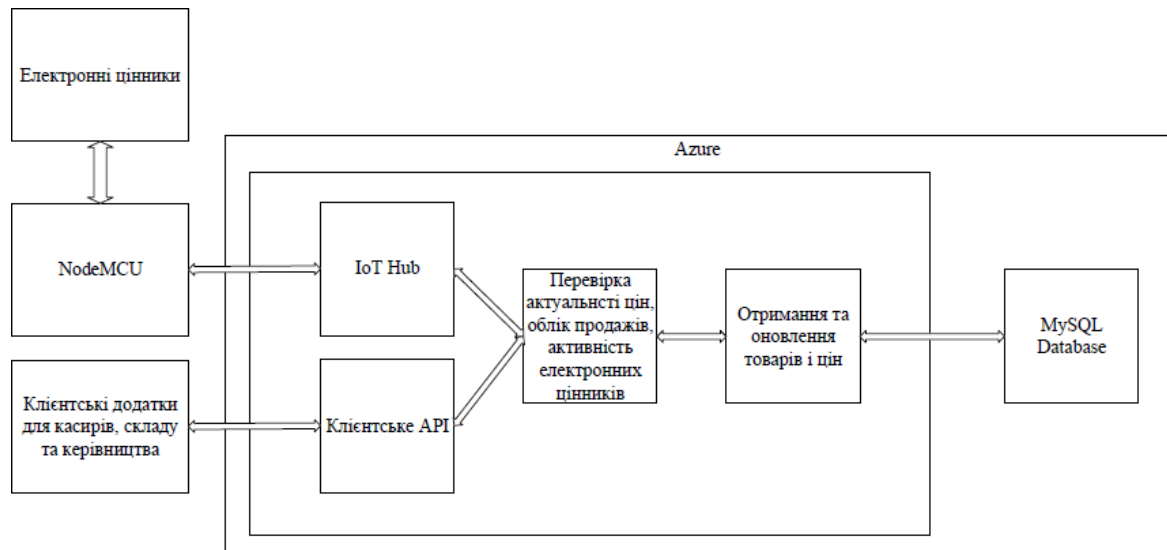


Рис. 2.14. Концептуальна модель системи роботи з БД

Наступний крок реалізації архітектури комп'ютерної системи з IoT компонентами пов'язаний з практичною реалізацією і передбачає обґрунтування та вибір технологій програмування, а також налаштування параметрів cloud сервісів.

2.6. Висновки до розділу

Основні результати, одержані в даному розділі, полягають у розробці та формалізації компонентів архітектури системи «розумний цінник», формалізації процесу збору даних про товари, обґрунтованому застосуванні інтелектуального сервісу аналізу товарів та формуванні цін з врахуванням відгуків користувачів, визначенні подібності товарів і споживачів, та конструюванні архітектури системи «розумний цінник». При цьому запропоновано організацію системи на основі взаємодії розумних пристроїв та хмарних сервісів.

Наукові та практичні результати, які одержано у цьому розділі, полягають в наступному:

1 У результаті аналізу принципів організації «розумних систем» встановлено, що доцільним є використання таких структурних компонентів як

кінцеві пристрої, шаблони взаємодії та протоколи передачі даних з опрацюванням та управлінням даними на стороні хмарних сервісів, що дало змогу забезпечити гнучкість реалізації архітектури таких систем, їх масштабованість та надійність експлуатації.

2 Спроектовано та запропоновано універсальну функціональну архітектуру системи «розумний цінник» на основі компонентного представлення та взаємодії кінцевих пристроїв, локального шлюзу, сервісів доступу та управління даними, що дало змогу забезпечити надійну та захищену комунікацію між складовими системи.

3 Формалізовано компоненти архітектурного рішення щодо побудови систем «розумний цінник» на основі теоретико-множинних нотацій, що дало змогу забезпечити однозначне трактування компонентів архітектури, зв'язків між ними та забезпечити альтернативний вибір апаратної і програмної реалізації складових таких систем.

4 На основі мультиагентного підходу до опису програмних систем, запропоновано та формалізовано сервіс збору даних про товари, що інтегрується у систему «розумний цінник» на рівні хмарного сервісу і дає змогу накопичувати дані про вартість, характеристики і відгуки про товари, що є базою для подальшого аналізу та ціноутворення.

5 На основі методу колаборативної фільтрації, формалізовано процес аналізу характеристик товарів, зокрема щодо формування ціни, визначення подібних товарів, статистики купівлі товарів і вподобань споживачів, що дало змогу враховувати ці критерії і підвищити ефективність управління бізнес-процесом з можливістю формування та миттєвої реакції в умовах зміни ринку.

6 Сконструйовано архітектуру системи «розумний цінник», що включає апаратне та програмне забезпечення «розумних» пристроїв, визначені протоколи передачі даних, хмарні сервіси збору та аналізу даних про товари, підтримує модель брокер-підписник на основі протоколу MQTT і дає змогу забезпечити конкурентоспроможність підприємства на ринку торгівлі.

РОЗДІЛ 3

РЕАЛІЗАЦІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ СИСТЕМИ «РОЗУМНИЙ ЦІННИК»

3.1. Програмна реалізація взаємодії IoT і хмарного сервісу

Реалізація комп'ютерної системи з IoT компонентами передбачає використання таких апаратних платформ, які не підтримують готові рішення від хмарного сервісу. Для реалізації даної частини використано мову Lua та протокол MQTT, що знову ж демонструє незалежність мов реалізації окремих частин рішення. Lua – скриптова мова програмування. Завдяки використанню в ролі вбудованої мови сценаріїв, яка інтегрується в інші реалізації, Lua отримала значне розповсюдження. Дана мова містить процедурний синтаксис, який є дуже простим та великі можливості для опису даних завдяки асоціативним масивам та семантиці мови. В ній немає статичної типізації, програма компілюється в байт-код та виконується, також є вбудований «збирач сміття». Програма – це набір файлів, кожен з яких містить свій функціонал та викликаються і виконуються один за одним. Це забезпечує оптимізацію використання доступних ресурсів, зокрема пам'яті.

Підключення пристрою до Інтернету, шляхом WiFi з'єднання наведено на рис. 3.1. у вигляді лістингу.

```

wifi.setmode(wifi.STATION)
print('set mode=STATION (mode='..wifi.getmode()..'')')
print('MAC-Address: '..wifi.sta.getmac())
print('chip: '..node.chipid())
print('heap: '..node.heap())

SSID = "DreamsIT"
PASSWORD = "20_DreamsIT16"

wifi.sta.config(SSID, PASSWORD)

wifi_codes = {
    [0] = "Idle",
    [1] = "Wait",
    [2] = "Incorrectpassword",
    [3] = "Can't find AP",
    [4] = "Fail",
    [5] = "New IP-address"
}

tmr.alarm(1,1000, 1, function()
    if wifi.sta.getip()==nil then

print("Connecting..."..wifi_codes[wifi.sta.status()])
        else
            print("Got IP:"..wifi.sta.getip())
            tmr.stop(1)
        end
    end
)

```

Рис. 3.1. Лістинг «Підключення до WiFi»

Для створення з'єднання з сервером потрібно авторизуватись, використовуючи отримані для конкретного пристрою дані та дані користувача. Вони наведені на рис.3.2 у вигляді відповідного лістингу.

```

BOARD = "onlizer-arduino-board"
HUB = "onlizer-boards.azure-devices.net"
KEY = "j8CH2g+WsAOAUZmqa8q4xxy0p/xVuHrmdKl3NdqsVQ="
MQTT_PORT = 8883
USER_ID = IO|THUB.."../..DEVICE
PASSWORD = new_token(HUB..'../devices/'..BOARD, KEY, 7 * 24 *
60)
topic="devices/".. BOARD.."/messages/events/"
command="devices/"..BOARD.."/messages/devicebound/#"

```

Рис. 3.2. Лістинг «Оголошення змінних з параметрами ідентифікації користувача»

З наведених даних наступним кроком необхідно створити токен доступу, що показано в лістингу 3.3.

```

function new_token(URL, KEY, expire_time)
URL = string.lower(encodeURIComponent(string.lower(URL)));
time = rtctime.get()
expire_time = math.ceil(time + expire_time * 60)
URL = URL.."\\n"..expire_time
pass = encoder.fromBase64(KEY)
base64 = crypto.toBase64(crypto.hmac("SHA256", URL, pass))
base64URL = encodeURIComponent(base64)
new_token = "SharedAccessSignature
sr="..URL.."&sig="..base64URL.."&se="..expire_time;
new_token = new_token.."&skn="
return new_token
end

function encodeURIComponent(str)
return (str:gsub("[^%w%-_%.!~*%'%(%)]", char_to_pchar))
end

function char_to_pchar(c)
return string.format("%%%02X", c:byte(1,1))
end

```

Рис. 3.3. Лістинг «Функції кодування для створення токена»

В лістингу 3.4 наведена функція синхронізації з тайм-сервером, оскільки час завершення дії токена створюється, виходячи з поточної дати та часу.

```
sntp_codes = {
    [1] = "Fail DNS",
    [2] = "Fail memory",
    [3] = "Fail UDP",
    [4] = "Noreponse"
}

sntp.sync('pool.ntp.org',
    function(sec, usec, server)
        print("Clock Synced: "..sec..", "..usec..", "..server)
    end,
    function(error)
        print("SNTP sync Fail: "..sntp_codes[error])
    end)
end)
```

Рис. 3.4. Лістинг «Синхронізація з NTP сервером»

В лістингу 3.5 наведено підключення до сервісу, отримання та відправлення повідомлень через MQTT[7].

```
connection = mqtt.Client(BOARD, 240, USER_ID, PASSWORD)

print ("Please wait while connecting")

connection:connect(HUB, MQTT_PORT, 1, 0,
    function(client)
        print("Succesfull connection to:
"..HUB..":"..MQTT_PORT)
        setup()
```

Рис. 3.5. Лістинг «Підключення та відправлення/отримання даних»

```

connection = mqtt.Client(BOARD, 240, USER_ID, PASSWORD)

print ("Please wait while connecting")

connection:connect(HUB, MQTT_PORT, 1, 0,
  function(client)
    print("Succesfull connection to:
"..HUB..":"..MQTT_PORT)
    setup()
  end,
  function(client, reason)
    print("Error: "..reason)
  end
)

function setup()

connection:subscribe(command_topic, 1, function(client)
  print("Subscribed to command and control topic.")
end)

print("begin lab04")

tmr.alarm(2, 10000, tmr.ALARM_AUTO, publish_data)

connection:on("message", function(conn, topic, data)
  print(topic .. ":" )
  if (data ~= nil ) then
    print ( data )
  end
end )

connection:on("offline", function(client)
  print("Disconnected.")
  connected = false

```

Рис. 3.6. Продовження лістингу «Підключення та відправлення/отримання даних»

```

        end)
    end

    function publish_data()
        if connected == true then

            gpio.write(LEDPIN, gpio.LOW)

            TEMPERATURE = Thermistor(adc.read(0))
            TIME = rtctime.get();

            payload =
                "{  \"deviceId\"    :  \"\"..BOARD..\"\",  \"data\"    :
                {\"Temperature\": \"..TEMPERATURE..\",  \"Time\" : \"..TIME..\"}"

            connection:publish(topic, payload, 1, 0, function(client)
                print("Send message.")
            end)

            gpio.write(LEDPIN, gpio.HIGH)
        end
    end

    function Thermistor(RawADC)

        Temp = ((RawADC/1024)*5000)/10;
        return Temp;
    end

```

Рис. 3.7. Продовження лістингу «Підключення та відправлення/отримання даних»

3.2. Завантаження програмної реалізації в пристрій

Для роботи з Azure, використовуючи мову Lua потрібно виконати наступні пункти:

- створити акаунт Azure та налаштувати IoT Hub;
- встановити програми для прошивання плати на ПК;
- скачати прошивку або зробити власну;
- прошити плату;
- написати та завантажити файли з кодом.

Процес створення акаунту Azure, налаштування IoT Hub та додавання пристрою, використовуючи утиліту DeviceExplorer описаний в офіційному туторіалі Azure.

Для завантаження прошивки можна використати:

- ESP8266Flasher, який потрібно лише завантажити.
- ESPTool. Для неї потрібно встановити Python, потім встановити утиліту командою

```
pip install esptool;
```

Для роботи з Azure потрібно завантажити прошивку та сертифікат в пристрій.

Для роботи з Lua також можна зібрати свою прошивку. Для створення своєї прошивки можна використати <https://nodemcu-build.com/> , вибравши необхідні параметри. Обов'язкові модулі в прошивці:

- Net (захищене з'єднання use TLS1.1);
- WIFI (WiFi модуль);
- MQTT (MQTT протокол версії 3.1.1);
- Tmr (функції роботи з таймером);
- UART (протокол передачі даних даних через послідовний порт)
- Crypto (забезпечення функцій роботи з криптографічними алгоритмами);
- Encoder (забезпечення функцій для кодування та декодування байтів даних);
- File (доступ до файлової системи та файлів);
- Node (доступ до системних функцій);

- RtcTime (збереження часу та робота з ним);
- Sntp (реалізація протоколу SNTP);

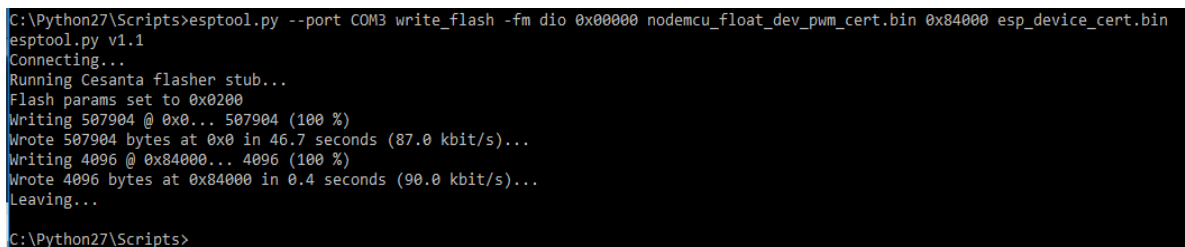
На електронну пошту прийде повідомлення про початок створення прошивки - «This is just to let you know that your NodeMCU custom build is now running.»[1]. Після закінчення розробки прошивки прийде повідомлення – «NodeMCUcustombuildfinished»[1], в якому будуть посилання на скачування своєї прошивки.

Прошивка плати за допомогою утиліти ESPTool. Щоб записати прошивку в плату потрібно виконати наступну команду, яка також наведена на рис. 3.8:

```
esptool.py -p Com3 write_flash 0x00000
nodemcu_float_dev_pwm_cert.bin 0x84000 esp_device_cert.bin.
```

-p - порт через який підключена плата (може відрізнятись від Com3)

0x00000 та 0x84000 адреси в пам'яті



```
C:\Python27\Scripts>esptool.py --port COM3 write_flash -fm dio 0x00000 nodemcu_float_dev_pwm_cert.bin 0x84000 esp_device_cert.bin
esptool.py v1.1
Connecting...
Running Cesanta flasher stub...
Flash params set to 0x0200
Writing 507904 @ 0x0... 507904 (100 %)
Wrote 507904 bytes at 0x0 in 46.7 seconds (87.0 kbit/s)...
Writing 4096 @ 0x84000... 4096 (100 %)
Wrote 4096 bytes at 0x84000 in 0.4 seconds (90.0 kbit/s)...
Leaving...
C:\Python27\Scripts>
```

Рис. 3.8. Запис прошивки

Для виконання цього ж процесу за допомогою утиліти ESP8266Flasher потрібно її відкрити, на вкладці Config вибрати створену прошивку та записати за адресою 0x00000, а сертифікат в 0x84000, що показано на рис.3.9.

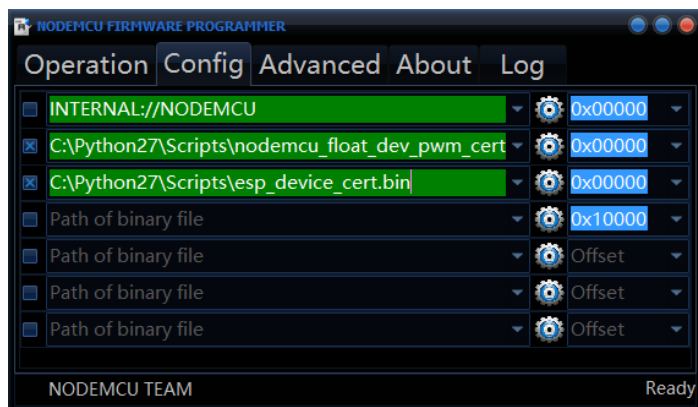


Рис.3.9. Вкладка Config утиліти ESP8266Flasher

На вкладці Advanced встановлюємо параметри плати: частоту, розмір флеш-пам'яті, частоту пам'яті та режим SPI, що показано на рис.3.10.



Рис.3.10. Вкладка Advanced

На вкладці Operation (див. рис. 3.11) запускаємо Flash та очікуємо завершення завантаження прошивки.

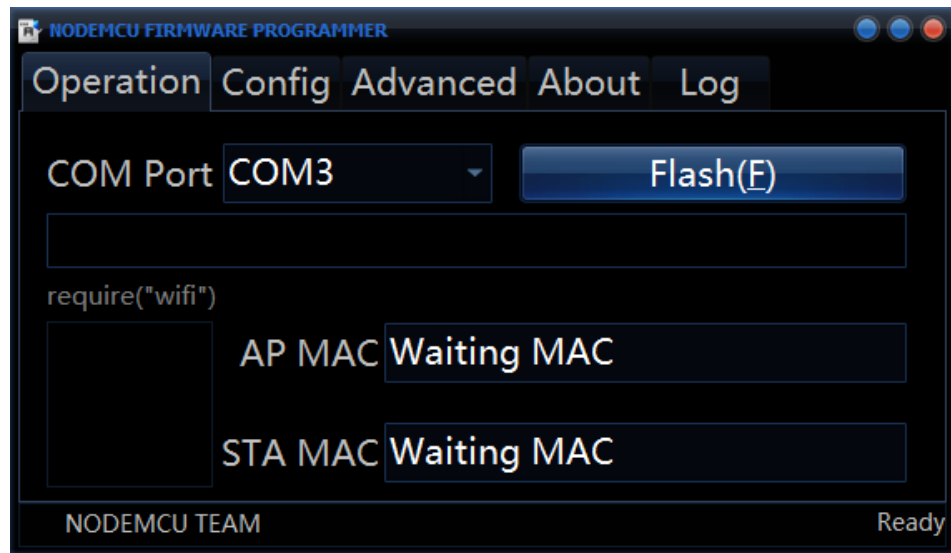


Рис.3.11. Вкладка Operation

Після цього потрібно перезавантажити плату.

Для подальшої роботи буде використано утиліту ESPLorer, що показана на рис.3.12.

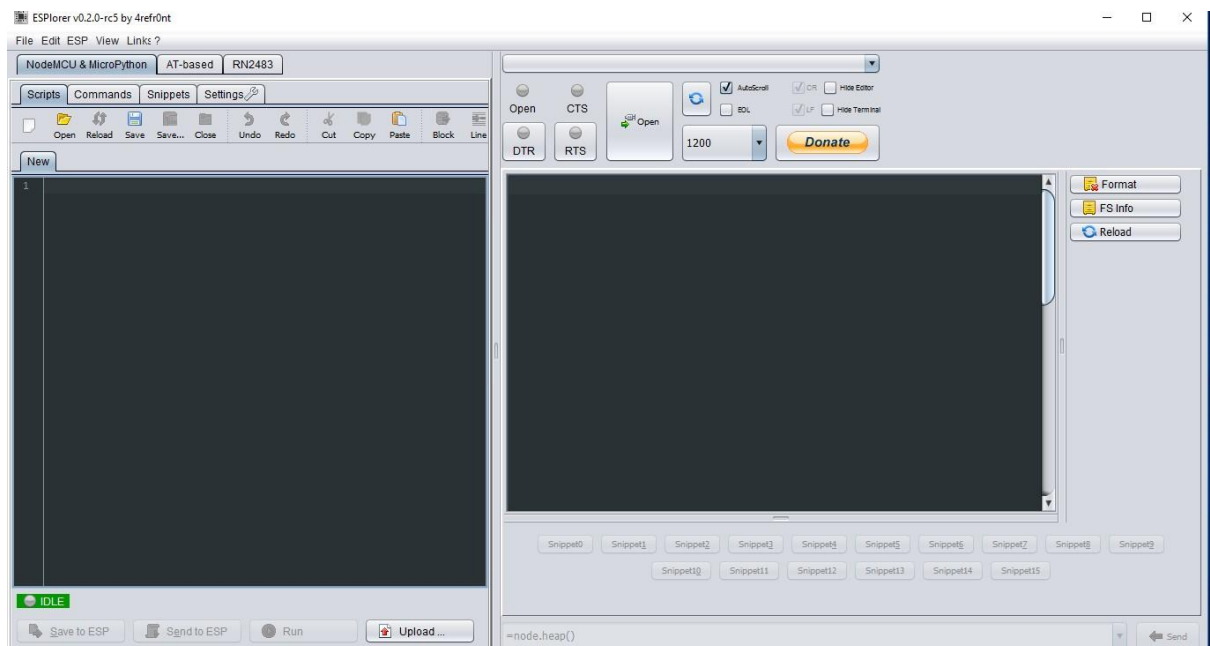


Рис. 3.12. Програма ESPLorer

Для початку відкриваємо (File->Open) файл init.lua, редагуємо, вказуємо SSID та PASS для точки доступу, зберігаємо (File->Save). Для завантаження файлів натискаємо клавішу Upload та чекаємо завершення процесу

завантаження в плату. Після завершення необхідно перезапустити плату (апаратно або за допомогою RTS в програмі) та натиснути клавішу Reload, нам покажуть список усіх файлів на платі.

Наступним завантажуюємо azure.lua. В ньому прописуємо отримані з DeviceExplorerId пристрою, IoTHubname та DeviceKey для доступу до хмарних сервісів.

Виконати файл можна командою doFile("file_name.lua"). Видалити файл можна командою os.remove(filename).

3.3. Тестування роботи системи

Згідно поставленої мети принцип роботи повинен бути наступним. Система підключається до мережі, синхронізується з NTP сервером для отримання поточного часу (завдяки вибору сервера можна синхронізуватись по UTC-0 або конкретній часовій зоні). Після цього із заданих користувачем даних створюється токен для авторизації на сервері. Відкривається з'єднання за допомогою протоколу MQTT та надсилаються тестові дані у вигляді Id пристрою та температури. В процесі тестування отримані дані виводяться в послідовний порт для підтвердження отримання та перевірки коректності даних.

Для тестового варіанту усі дані в процесі виводяться в послідовний порт. На рис.3.13 зображені дані в процесі підключення до IoTHub.

```
wifi_event_monitor_handle_event_cb is ca
Got IP:192.168.0.102
Clock Synced: 1487098641, 133386, 82.193.117
SharedAccessSignature sr=onlizer-boards.azure
enter mqtt_socket_client.
NodeMCU_8658f4
length username: 54
length password: 167
MQTT: Init info: onlizer-arduino-board, onli
leave mqtt_socket_client.
Connecting to MQTT broker. Please wait...
pm open,type:2 0
enter mqtt_socket_connect.
TCP ip is set: 255.255.255.255
TCP port is set: 8883.
leave mqtt_socket_connect.
enter socket_dns_found.
TCP ip is set: 40.118.27.192
```

Рис.3.13. Підключення пристрою до хмарного сервісу

Після успішного підключення пристрій починає з заданою періодичністю надсилати дані. За допомогою DeviceExplorer-а відстежуються дані, які надійшли від пристрою. На рис.3.14 зображені успішно отримані дані.

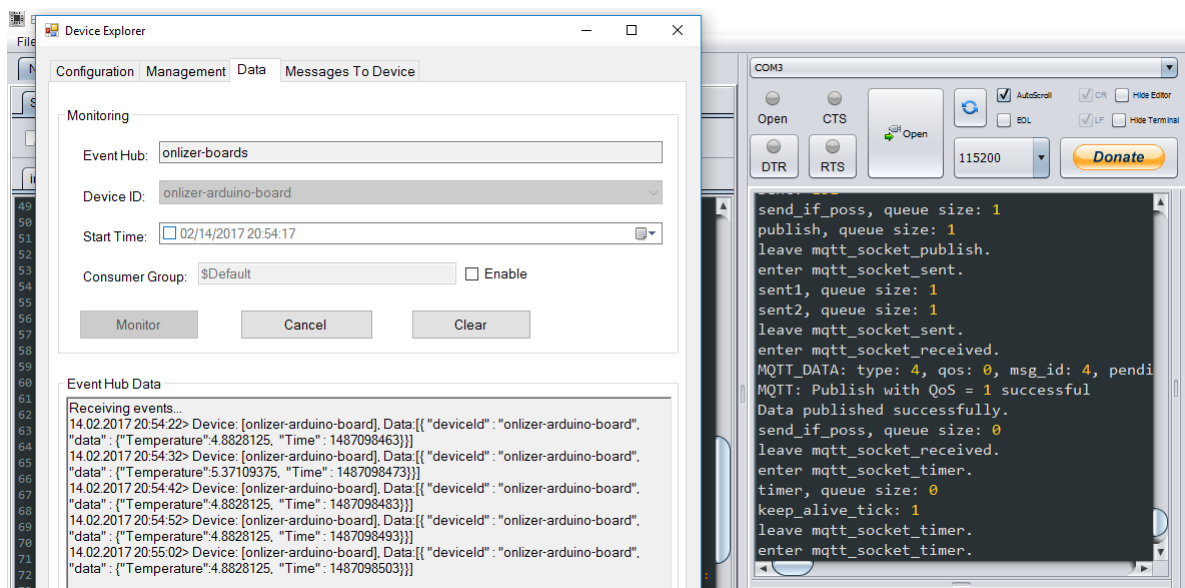


Рис.3.14. Моніторинг отриманих даних

Раніше згаданий DeviceExplorer використано для надсилання повідомлень C2D та перевірки отримання. На рис.3.15 зображено надіслані з DeviceExplorer дані та отримані дані на пристрої відповідно.

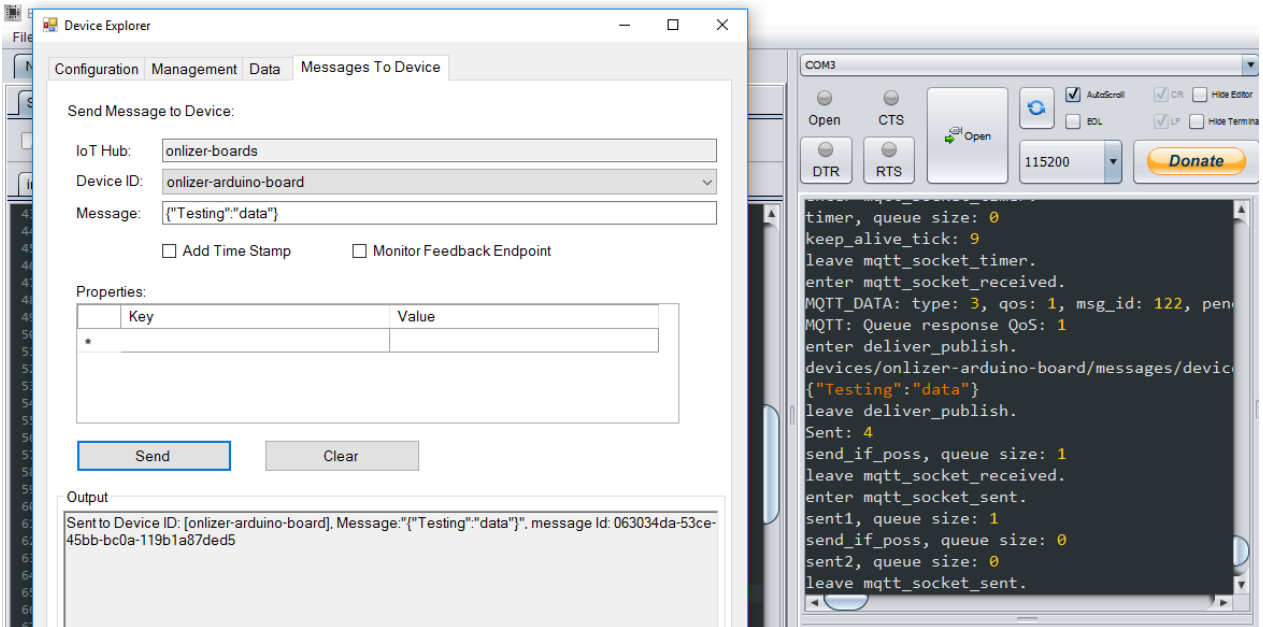


Рис.3.15. Відправлення даних з утиліти, отримані дані на пристрої

Таким чином ефективно реалізовано комп'ютерну систему, яка дає змогу візуалізувати дані з бази даних і виводити їх на кінцеві пристрої.

3.4. Створення сервісу збору інформації про товари з використанням платформи Onlizer

Для того, щоб розпочати роботу із системою Onlizer необхідно провести реєстрацію на сайті <http://portal.onlizer.com/>, після чого увійти у особистий кабінет використовуючи форму для входу. Якщо авторизація пройшла успішно, то буде відкрита головна сторінка системи(рис. 3.16), на якій можна побачити існуючі Application, а також які Workflows є в цьому Application та їхній статус.

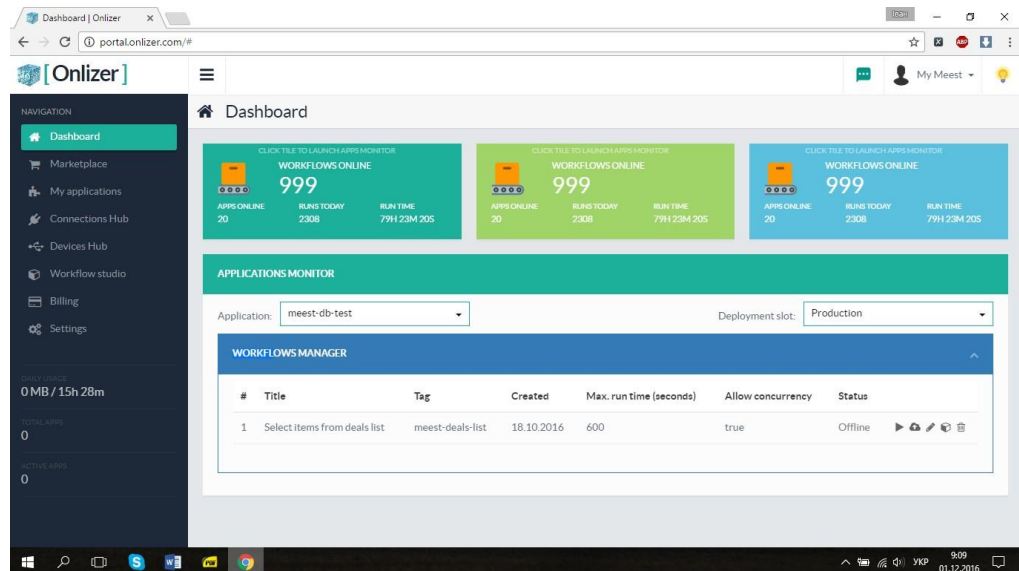


Рис. 3.16. Головна сторінка системи Onlizer

Application — це основна системна одиниця в системі Onlizer в середині якої розміщаються функціональні Workflows. Application створюються для виконання певної роботи, а Workflows в середині них забезпечують виконання цієї роботи.

Щоб створити Application потрібно в меню обрати пункт My applications, після чого буде здійснено перехід на відповідну сторінку де потрібно натиснути кнопку New application. З'явиться вікно(рис. 3.17) у яке потрібно ввести ім'я application, яке повинне бути унікальним для всієї системи.

Коли application буде створено здійсниться автоматичний перехід на сторінку з його настройками. На цій сторінці можна створити новий Workflow, або запустити вже існуючі.

Workflow — це компонент в середині, якого розміщуються інші компоненти, які разом призначені для виконання певного завдання.

Для створення workflow потрібно в налаштуваннях application натиснути кнопку Add workflow і у формі(рис. 3.18), що відкриється ввести його ім'я, тег для пришвидшення пошуку, максимальну затримку виконання в секундах і т.д.

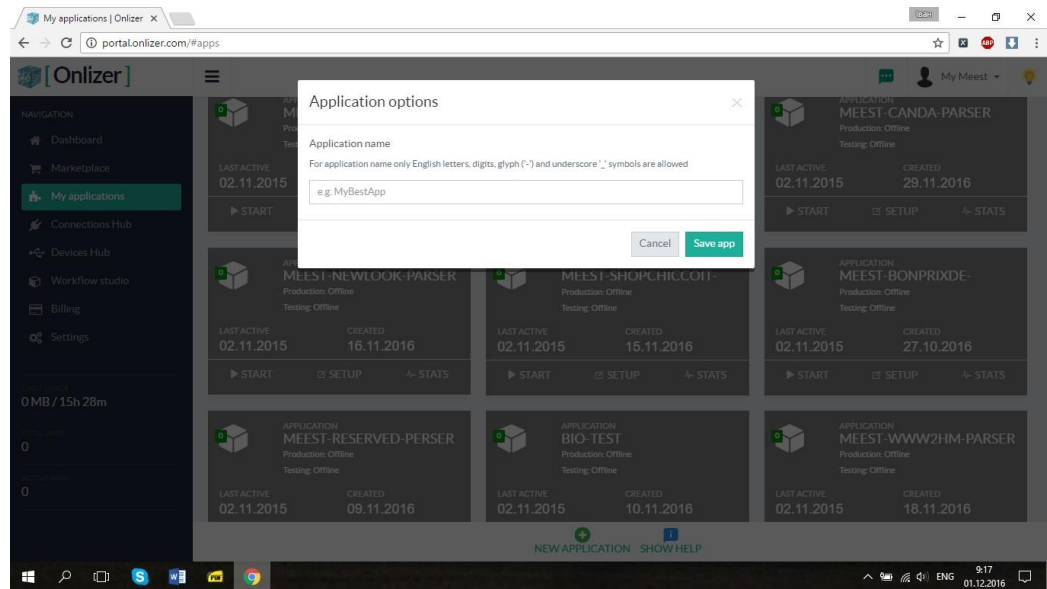


Рис. 3.17. Зовнішній вигляд вікна створення нового Application

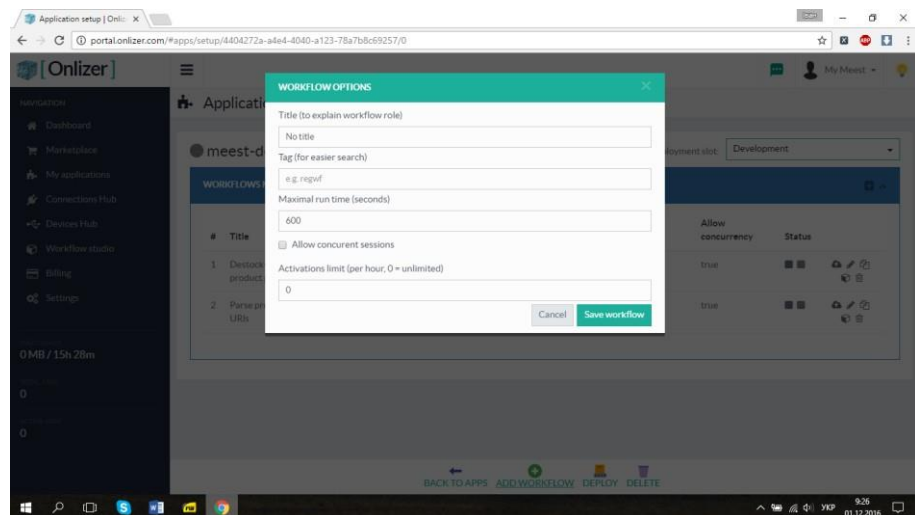


Рис. 3.18. Зовнішній вигляд вікна створення нового Workflow

Після того як пройде створення workflow відкриється вікно редактора Workflow studio(рис. 3.19), в якому можна шляхом поєднання графічних компонент(конекторів) реалізувати необхідну бізнес-логіку.

Розглянемо роботу із системою Onlizer на прикладі workflow, який буде збирати інформацію про товар із сайту <http://www.destock-sport-et-mode.com>.

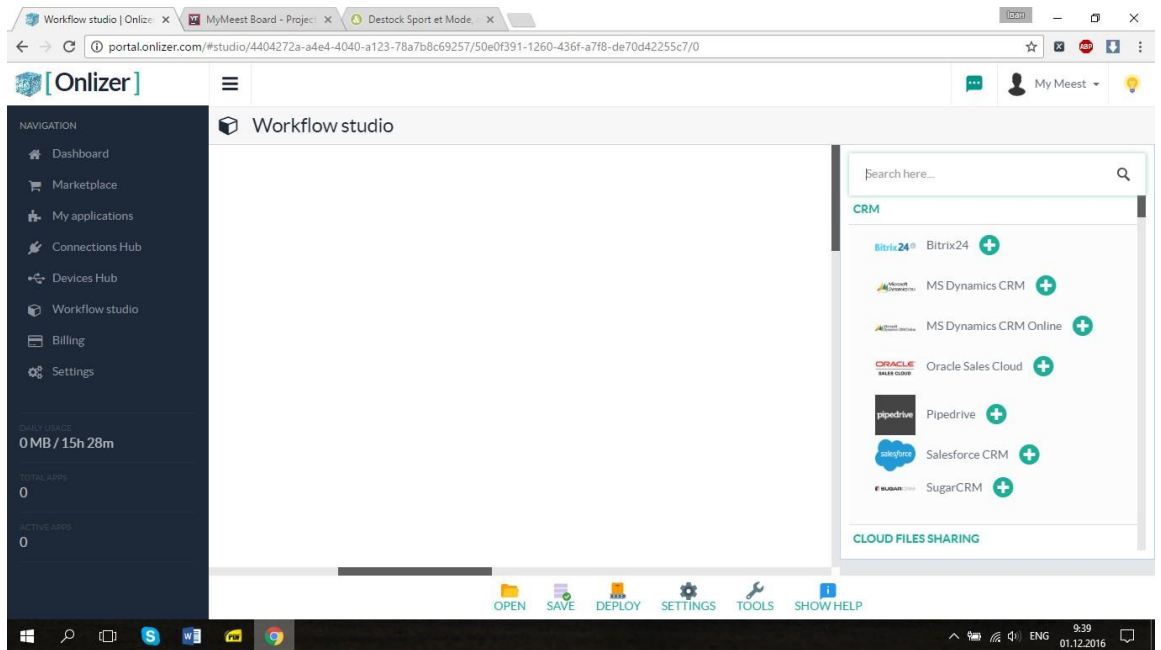


Рис. 3.19. Зовнішній вигляд Workflow studio

Щоб це зробити, спочатку потрібно створити application та workflow, відповідно до кроків описаних вище. Коли вони будуть створені у Workflow studio, на панелі з права потрібно буде знайти конектор HTTP Endpoint та HTML Parser. HTTP Endpoint буде використовуватись для переходу на сторінку товару та відправлення результату своєї роботи, а HTML Parser для збору інформації про товар. Щоб можна було використовувати конектор потрібно натиснути кнопку «+» біля його імені і він буде показаний в полі Workflow studio.

Зовнішній вигляд Workflow studio із доданими конекторами показано на рис. 3.20.

Після того як конектори були додані до workflow потрібно налаштувати зв'язки між ними. Для цього треба натиснути на кнопці у вигляді трикутника, яка знаходить з правого верхнього краю іконки конектора і у меню (рис. 3.21), що з'явиться обрати пункт Connection, після цього натиснути на конектор з яким потрібно встановити зв'язок.

Щоб виконати поставлену задачу потрібно з'єднати HTTP Endpoint з HTML Parser, а потім HTML Parser з іншим HTTP Endpoint.

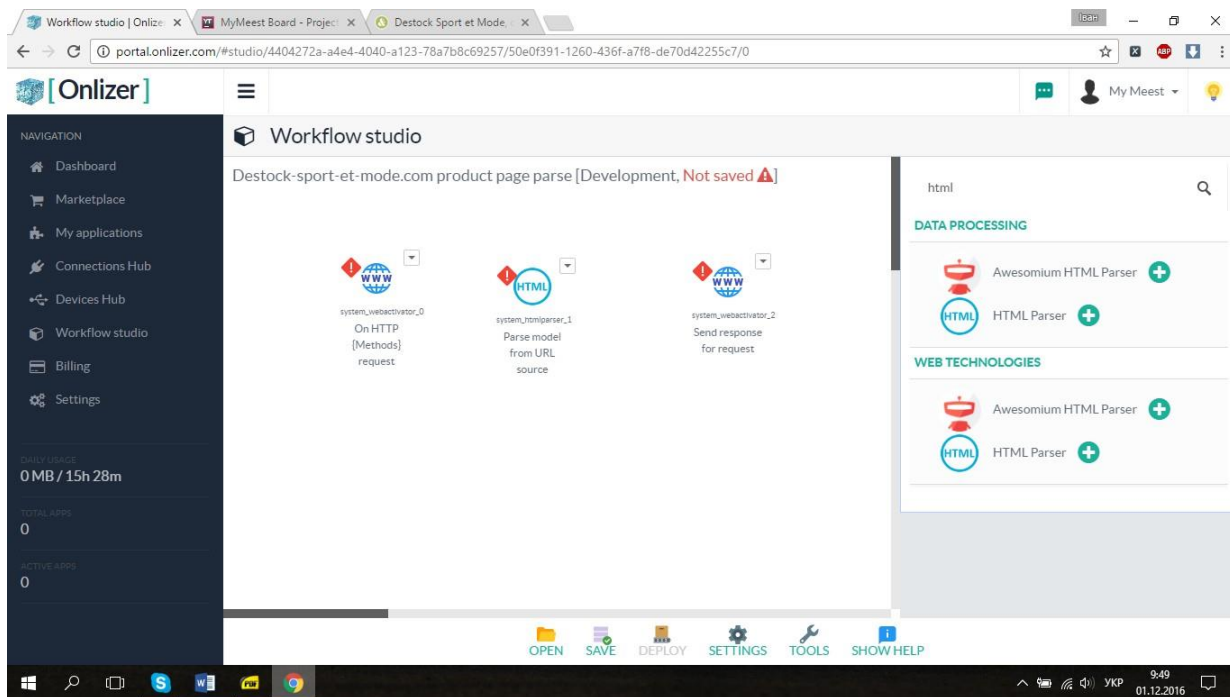


Рис. 3.20. Вигляд конекторів у Workflow studio

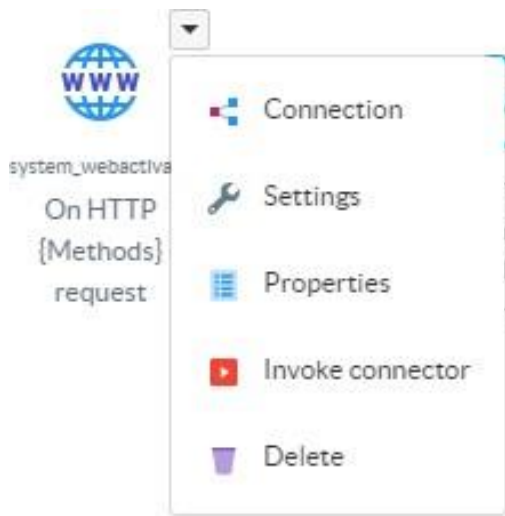


Рис. 3.21. Контекстне меню конектора

Коли всі компоненти будуть з'єднані можна приступити до їхніх налаштувань, щоб це зробити потрібно в контекстному меню конектора обрати пункт Settings. Під час першої настройки потрібно буде обрати метод, який буде використовувати конектор, після чого відкриється вікно налаштувань(рис. 3.22) де потрібно буде ввести відповідні параметри методу.

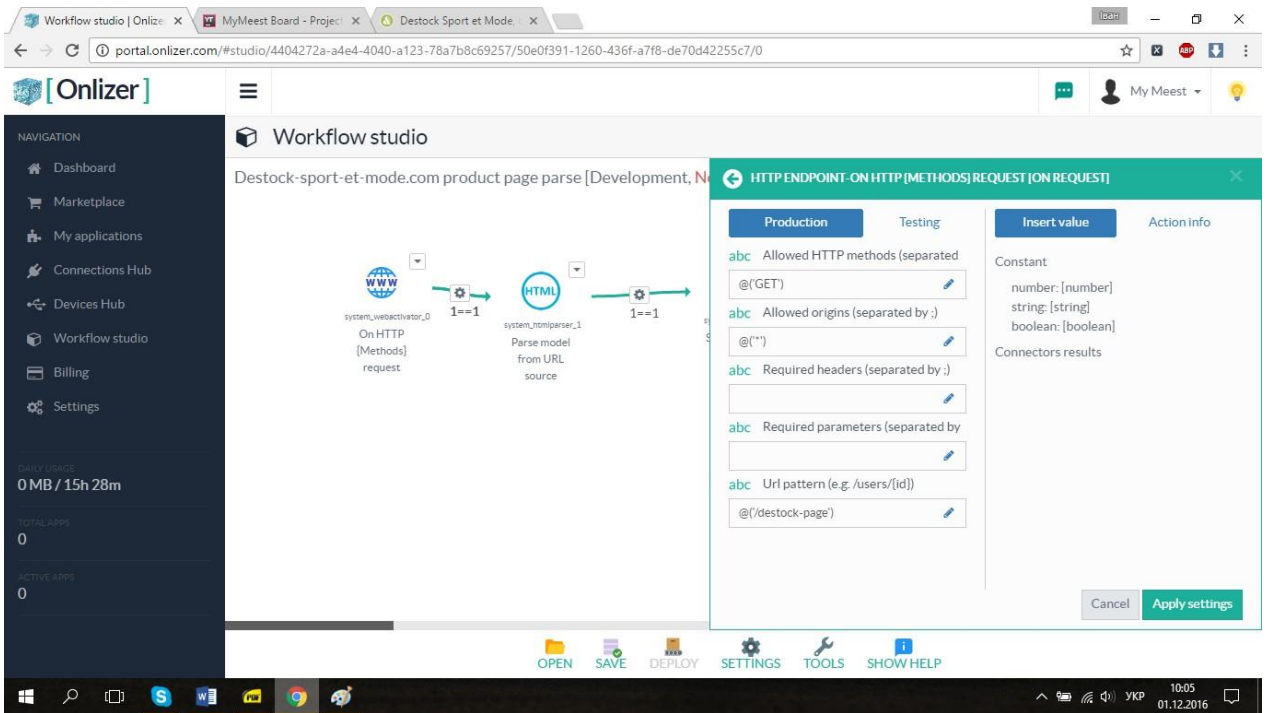


Рис. 3.22. Вікно налаштувань конектора

Коли всі компоненти будуть налаштовані потрібно натиснути кнопку Save для збереження workflow. Збережений workflow можна запустити натиснувши кнопку Start у вікні настройок application.

Приклад результату збору інформації про товар <http://www.destock-sport-et-mode.com/converse-ctas-ox-chaussure-unisex,27348310,143802395,p.html> показано на рис. 3.23.



Рис. 3.23. Результат роботи workflow із збору інформації про товар

За допомогою системи Onlizer можна створювати різноманітні системи при цьому не маючи ніяких знань у програмуванні та будувати ефективні програмні інтерфейси для інтеграції існуючих програмних продуктів.

3.5. Сервіс формування ціни товарів на основі подібності товарів та споживачів

В якості вхідних даних, на основі яких буде проводиться розробка сервісу аналізу цін системи «розумний цінник», обрано джерело, що містить 500000 записів в форматі csv. На рис. 3.24 наведено структуру цього файлу.

DATA_OF_ONLINE_RETAIL.csv - Блокнот

Файл Редагування Формат Виліза Довірка

InvoiceNo,StockCode,Description,Quantity,InvoiceDate,UnitPrice,CustomerID,Country

536365.85123A.WHITE HANGING HEART T-LIGHT HOLDER.6,12/1/2010 8:26.2.55.17850.United Kingdom

536365.71053.WHITE METAL LANTERN.6,12/1/2010 8:26.3.39.17850.United Kingdom

536365.84406B.CREAM CUPID HEARTS COAT HANGER.8,12/1/2010 8:26.2.75.17850.United Kingdom

536365.84029G.KNITTED UNION FLAG HOT WATER BOTTLE.6,12/1/2010 8:26.3.39.17850.United Kingdom

536365.84029E.RED WOOLLY HOTTIE WHITE HEART.6,12/1/2010 8:26.3.39.17850.United Kingdom

536365.22752.SET 7 BABUSHKA NESTING BOXES.2,12/1/2010 8:26.7.65.17850.United Kingdom

536365.21730.GLASS STAR FROSTED T-LIGHT HOLDER.6,12/1/2010 8:26.4.25.17850.United Kingdom

536366.22633.HAND WARMER UNION JACK.6,12/1/2010 8:28.1.85.17850.United Kingdom

536366.22632.HAND WARMER RED POLKA DOT.6,12/1/2010 8:28.1.85.17850.United Kingdom

536367.84879.ASSORTED COLOUR BIRD ORNAMENT.32,12/1/2010 8:34.1.69.13047.United Kingdom

536367.22745.POPPY'S PLAYHOUSE BEDROOM.6,12/1/2010 8:34.2.1.13047.United Kingdom

536367.22748.POPPY'S PLAYHOUSE KITCHEN.6,12/1/2010 8:34.2.1.13047.United Kingdom

536367.22749.FELTCRAFT PRINCESS CHARLOTTE DOLL.8,12/1/2010 8:34.3.75.13047.United Kingdom

536367.22310.IVORY KNITTED MUG COSY.6,12/1/2010 8:34.1.65.13047.United Kingdom

536367.84969.BOX OF 6 ASSORTED COLOUR TEASPOONS.6,12/1/2010 8:34.4.25.13047.United Kingdom

536367.22623.BOX OF VINTAGE JIGSAW BLOCKS.3,12/1/2010 8:34.4.95.13047.United Kingdom

536367.22622.BOX OF VINTAGE ALPHABET BLOCKS.2,12/1/2010 8:34.9.95.13047.United Kingdom

536367.21754.HOME BUILDING BLOCK WORD.3,12/1/2010 8:34.5.95.13047.United Kingdom

536367.21755.LOVE BUILDING BLOCK WORD.3,12/1/2010 8:34.5.95.13047.United Kingdom

536367.21777.RECIPE BOX WITH METAL HEART.4,12/1/2010 8:34.7.95.13047.United Kingdom

536367.48187.DOORMAT NEW ENGLAND.4,12/1/2010 8:34.7.95.13047.United Kingdom

536368.22960.JAM MAKING SET WITH JARS.6,12/1/2010 8:34.4.25.13047.United Kingdom

536368.22913.RED COAT RACK PARIS FASHION.3,12/1/2010 8:34.4.95.13047.United Kingdom

536368.22912.YELLOW COAT RACK PARIS FASHION.3,12/1/2010 8:34.4.95.13047.United Kingdom

536368.22914.BLUE COAT RACK PARIS FASHION.3,12/1/2010 8:34.4.95.13047.United Kingdom

536369.21756.BATH BUILDING BLOCK WORD.3,12/1/2010 8:35.5.95.13047.United Kingdom

536370.22728.ALARM CLOCK BAKELIKE PINK.24,12/1/2010 8:45.3.75.12583.France

536370.22727.ALARM CLOCK BAKELIKE RED.24,12/1/2010 8:45.3.75.12583.France

536370.22726.ALARM CLOCK BAKELIKE GREEN.12,12/1/2010 8:45.3.75.12583.France

536370.21724.PANDA AND BUNNIES STICKER SHEET.12,12/1/2010 8:45.0.85.12583.France

536370.21883.STARS GIFT TAPE.24,12/1/2010 8:45.0.65.12583.France

536370.10902.INFLATABLE POLITICAL GLOBE.48,12/1/2010 8:45.0.85.12583.France

536370.21791.VINTAGE HEADS AND TAILS CARD GAME.24,12/1/2010 8:45.1.25.12583.France

536370.21035.SET 2 RED RETROSPOT TEA TOWELS.18,12/1/2010 8:45.2.95.12583.France

536370.22326.ROUND SNACK BOXES SET OF 4 WOODLAND.24,12/1/2010 8:45.2.95.12583.France

536370.22629.SPACEMAN LUNCH BOX.24,12/1/2010 8:45.1.95.12583.France

536370.22659.LUNCH BOX I LOVE LONDON.24,12/1/2010 8:45.1.95.12583.France

536370.22631.CIRCUS PARADE LUNCH BOX.24,12/1/2010 8:45.1.95.12583.France

536370.22661.CHARLOTTE BAG DOLLY GIRL DESIGN.20,12/1/2010 8:45.0.85.12583.France

536370.21731.RED TOADSTOOL LED NIGHT LIGHT.24,12/1/2010 8:45.1.65.12583.France

536370.22900.SET 2 TEA TOWELS I LOVE LONDON.24,12/1/2010 8:45.2.95.12583.France

536370.21913.VINTAGE SEASIDE JIGSAW PUZZLES.12,12/1/2010 8:45.3.75.12583.France

536370.22540.MINI JIGSAW CIRCUS PARADE.24,12/1/2010 8:45.0.42.12583.France

536370.22544.MINI JIGSAW SPACEBOY.24,12/1/2010 8:45.0.42.12583.France

536370.22492.MINI PAINT SET VINTAGE.36,12/1/2010 8:45.0.65.12583.France

536370.POST.POSTAGE.3,12/1/2010 8:45.18.12583.France

Рис. 3.24. Структура вхідних даних

Для побудови сервісу аналізу цін і товарів та перевірки його роботи реалізовано код мовою Python у середовищі GraphLab Create, який наведено у лістингу на рис. 3.25.

```
import graphlab as gl

data_set =
gl.SFrame.read_csv('D:\DATA_OF_ONLINE_RETAIL.csv')

data_set.dropna()

data_set = data_set[data_set['Quantity']>=0]

print(data_set)
```

Рис. 3.25. Лістинг «Інтелектуальний сервіс формування цін на товари»

```

train_data_set,
test_data_set=gl.recommender.util.random_split_by_user
(data_set,user_id='CustomerID',item_id='StockCode',item_test_pro
portion=0.6)

popularity_model =
gl.popularity_recommender.create(train_data_set,
user_id='CustomerID', item_id='StockCode', target='Quantity')
popularity_recomm = popularity_model.recommend(k=5)
popularity_recomm.print_rows(num_rows=25)
popularity_recomm.show()

item_sim_model =
gl.item_similarity_recommender.create(train_data_set,
user_id='CustomerID', item_id='StockCode', target='Quantity',
similarity_type='cosine')

item_sim_recomm = item_sim_model.recommend(k=5)
item_sim_recomm.print_rows(num_rows=25)
item_sim_recomm.show()

model_performance = gl.compare(test_data_set,
[popularity_model, item_sim_model])

gl.show_comparison(model_performance,[popularity_model,
item_sim_model])

```

Рис. 3.26. Продовження лістингу «Інтелектуальний сервіс формування цін на товари»

Проведемо пострічковий аналіз коду.

`import graphlab as gl` – підключення бібліотеки Graph Lab Create і створення псевдоніму для подальшого використання об'єктів з бібліотеки.

`data_set = gl.SFrame.read_csv ('D:\DATA_OF_ONLINE_RETAIL.csv')` – зчитування даних з файлу `D:\DATA_OF_ONLINE_RETAIL.csv` і створення джерела даних у вигляді `SFrame`.

У результаті зчитування, одержуємо структуру даних, яку показано на рис. 3.27. у вигляді реляційної структури даних.

`data_set.dropna()` – видалення пропущених, невизначених або некоректних даних.

`data_set = data_set[data_set['Quantity']>=0]` – вибір даних, де кількість придбаних товарів більша нуля.

`print(data_set)` – вивід даних на екран.

```
train_data_set, test_data_set =
gl_recommender.util.random_split_by_user
(data_set, user_id='CustomerID',
item_id='StockCode',
```

`item_test_proportion=0.6)` – формування навчальних і тестових вибірок у відношенні 60%/40%. При цьому у вибірках беруть участь лише стовпці з ідентифікатором користувача та ідентифікатором товару.

InvoiceNo	StockCode	Description	Quantity
536365	85123A	WHITE HANGING HEART T-LIGH...	6
536365	71053	WHITE METAL LANTERN	6
536365	84406B	CREAM CUPID HEARTS COAT HANGER	8
536365	84029G	KNITTED UNION FLAG HOT WAT...	6
536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6
536365	22752	SET 7 BABUSHKA NESTING BOXES	2
536365	21730	GLASS STAR FROSTED T-LIGHT...	6
536366	22633	HAND WARMER UNION JACK	6
536366	22632	HAND WARMER RED POLKA DOT	6
536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32

InvoiceDate	UnitPrice	CustomerID	Country
12/1/2010 8:26	2.55	17850	United Kingdom
12/1/2010 8:26	3.39	17850	United Kingdom
12/1/2010 8:26	2.75	17850	United Kingdom
12/1/2010 8:26	3.39	17850	United Kingdom
12/1/2010 8:26	3.39	17850	United Kingdom
12/1/2010 8:26	7.65	17850	United Kingdom
12/1/2010 8:26	4.25	17850	United Kingdom
12/1/2010 8:28	1.85	17850	United Kingdom
12/1/2010 8:28	1.85	17850	United Kingdom
12/1/2010 8:34	1.69	13047	United Kingdom

Рис. 3.27. Структура вхідних даних

`popularity_model = gl.popularity_recommender.create`
`(train_data_set, user_id='CustomerID', item_id='StockCode',`
`target='Quantity')` – створення моделі на основі найбільш
популярних товарів у вибірці з цільовим атрибутом «Кількість товару».

`gl.popularity_recommender.create` – створює модель, яка формує
рекомендації товарів на основі їх популярності. Якщо не вказано атрибут-ціль,
популярність визначається наявною кількістю кожного товару. Коли ціль
визначена, популярність обчислюється, використовуючи середнє значення
цільового елемента. Наприклад, якщо в атрибуті-цілі містяться оцінки, модель
обчислює середню оцінку для кожного товару та використовує її, щоб
оцінювати позиції для формування рекомендацій ціни та користувачів.

`train_data_set` – множина даних для навчання моделі. Вона повинна
містити стовпець ідентифікаторів користувача та стовпець ідентифікатора
товару. Кожний рядок представляє собою відношення між користувачем і
товаром. Пари (користувач, товар) зберігаються у моделі, щоб потім їх можна
було виключити з рекомендацій, якщо це потрібно. Дана множина може
містити стовпець рейтингу цілей, зокрема ціни товару. Всі інші стовпці
інтерпретуються базовою моделлю як сторонні функції для аналізу. Стовпці
ідентифікатора користувача та ідентифікатора елементів повинні бути типу
"int" або "str". Цільовий стовпець повинен бути типу "int" або "float".

`popularity_recomm = popularity_model.recommend(k=5)` –
формування 5 рекомендацій для кожного користувача.

`popularity_recomm.print_rows(num_rows=25)` – вибрати
випадковим чином 5 користувачів із сформованими 5 рекомендаціями товарів
(рис. 3.28).

`popularity_recomm.show()` – вивести параметри моделі та
візуалізувати результати побудови моделі за допомогою утиліти GraphLab
Canvas. Візуалізовані дані та їх аналіз наведено нижче.

`item_sim_model = gl.item_similarity_recommender.create`
`(train_data_set, user_id='CustomerID', item_id='StockCode',`

`target='Quantity', similarity_type='cosine')` – створення моделі навчання з використання метрик подібності. Вона повинна містити стовпець ідентифікаторів користувача та стовпець ідентифікатора елемента. Кожний рядок представляє собою відношення між користувачем та елементом. Пари (користувач, об'єкт) зберігаються у моделі, щоб потім їх можна було виключити з рекомендацій, якщо це потрібно. Дана множина може містити стовпець рейтингу цілей. Всі інші стовпці інтерпретуються базовою моделлю як сторонні функції для аналізу. Стовпці ідентифікатора користувача та ідентифікатора елементів повинні бути типу "int" або "str". Цільовий стовпець повинен бути типу "int" або "float".

CustomerID	StockCode	score	rank
17850	23843	80995.0	1
17850	47556B	1300.0	2
17850	84611B	1287.0	3
17850	84826	399.382352941	4
17850	23166	368.233333333	5
13047	23843	80995.0	1
13047	47556B	1300.0	2
13047	84611B	1287.0	3
13047	84826	399.382352941	4
13047	23166	368.233333333	5
12583	23843	80995.0	1
12583	47556B	1300.0	2
12583	84611B	1287.0	3
12583	84826	399.382352941	4
12583	23166	368.233333333	5
13748	23843	80995.0	1
13748	47556B	1300.0	2
13748	84611B	1287.0	3
13748	84826	399.382352941	4
13748	23166	368.233333333	5
15100	23843	80995.0	1
15100	47556B	1300.0	2
15100	84611B	1287.0	3
15100	84826	399.382352941	4
15100	23166	368.233333333	5

Рис. 3.28. Ранжовані рекомендації для 5 користувачів на основі популярності товарів

`similarity_type='cosine'` – параметр, що визначає міру подібності користувачів і товарів на основі косинуса кута. Крім цього можливе застосування `similarity_type='pearson'` та `similarity_type='jaccard'`.

`item_sim_recomm = item_sim_model.recommend(k=5)` – формування п'яти рекомендацій товарів на основі косинусної міри кута для кожного користувача. На рис. 3.29 наведено рекомендації 5 товарів для 5 користувачів, одержані із застосування косинусної міри подібності.

CustomerID	StockCode	score	rank
17850	23173	5.18414549601	1
17850	22230	4.5320960709	2
17850	82484	4.08754424822	3
17850	22811	3.31181599696	4
17850	17096	3.08755435262	5
13047	23173	11.1098671378	1
13047	23560	7.00286057401	2
13047	22230	6.73160998745	3
13047	82484	5.97540065963	4
13047	20749	5.1889593449	5
12583	23173	22.2744030667	1
12583	20749	19.1016506558	2
12583	84584	18.0050833738	3
12583	22811	15.6402904339	4
12583	22797	12.5542704681	5
13748	23173	21.3480120947	1
13748	22230	15.5828570103	2
13748	82484	13.0571364785	3
13748	20749	11.4242457673	4
13748	23560	9.41903278232	5
15100	23560	55.2875471115	1
15100	85034C	36.2483263016	2
15100	84708B	30.9180577397	3
15100	22740	30.7177098393	4
15100	21190	30.3642892838	5

Рис. 3.29. Рекомендації товарів, сформовані на основі косинусної міри

Для порівняння точності моделей і одержаних результатів використовуються метрики точності і повноти. За порівняння відповідає фрагмент коду

```
model_performance = gl.compare(test_data_set,
                               [popularity_model, item_sim_model])
```

Результати порівняння моделей щодо рекомендацій товару на основі популярності та косинусної міри подібності наведено на рис. 3.30 та рис. 3.31. відповідно.

cutoff	mean_precision	mean_recall
1	0.0	0.0
2	0.000504032258065	0.000168010752688
3	0.000336021505376	0.000168010752688
4	0.00100806451613	0.000231287529675
5	0.00342741935484	0.000500396869832
6	0.00319220430108	0.000510438788312
7	0.0028801843318	0.000518634434785
8	0.00264616935484	0.000522960891936
9	0.00246415770609	0.000523176014538
10	0.00241935483871	0.000624197588754

Рис. 3.30. Характеристики моделі на основі популярності товарів

cutoff	mean_precision	mean_recall
1	0.023185483871	0.000965947104656
2	0.0196572580645	0.00156078775368
3	0.0208333333333	0.00221665505856
4	0.0176411290323	0.00227683593647
5	0.0171370967742	0.00287254188633
6	0.0171370967742	0.00328125648271
7	0.0167050691244	0.00383389673699
8	0.0170110887097	0.00479389976498
9	0.0164650537634	0.00557899651582
10	0.0165322580645	0.00597273023285

Рис. 3.31. Характеристики моделі на основі метрики подібності «косинус кута»

Таким чином на основі інтелектуального аналізу подібності товарів та вподобань користувачів, які можна одержати з електронних магазинів шляхом застосування розробленого парсера, можна сформувати набори товарів для користувачів, встановити ціни подібні до тих, які є у конкурентів.

3.6. Висновки до розділу

1. Розроблено програмне забезпечення, що реалізує запропоновану архітектуру організації системи «розумний цінник» шляхом використання засобів NodeMCU, IoTHub, налаштуванням параметрів апаратного та програмного забезпечення при оновленні та візуалізації даних, а також протоколу MQTT, що дало змогу підтвердити ефективність застосування такої організації системи.

2. Проведено тестування двосторонньої взаємодії між компонентами архітектури системи «розумний цінник», у результаті чого підтверджено можливість керування кінцевими пристроями (електронними цінниками) з хмарного сервісу Azure.

3. На основі платформи Onlizer реалізовано сервіс збору даних про товари з можливістю запису у базу даних, яка спільно використовується із сервісом аналітики даних та відображення цін у системі «розумний цінник».

4. За допомогою мови програмування Python та відкритих бібліотеки реалізовано сервіс аналізу даних про товари, що дає змогу формувати конкурентну ціну у системі «розумний цінник».

РОЗДІЛ 4

ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

У дипломній роботі магістра розроблено методи і засоби збору, аналізу та відображення даних у комп'ютерних системах «Розумний цінник». Для обґрунтування економічної ефективності результатів дослідження необхідно провести розрахунки щодо економічної ефективності і терміну окупності запропонованих рішень, що є однією з важливих стадій розробки.

4.1. Розрахунок норм часу на виконання науково-дослідної роботи

Основні етапи при виконанні НДР можна визначити наступним чином:

- обґрунтування актуальності теми дипломної роботи магістра;
- аналіз наукових публікацій та інтернет-джерел для визначення сучасних підходів до проектування архітектури комп'ютерних систем «розумний цінник»;
- аналіз методів штучного інтелекту для визначення конкурентної вартості товарів;
- побудова та обґрунтування моделі архітектури комп'ютерної системи «розумний цінник»;
- розробка концептуальної схеми комп'ютерної системи «розумний цінник» з інтеграцією інтелектуального сервісу визначення вартості товару;
- забезпечення організації взаємодії IoT, хмарних та інтелектуального сервісів;
- створення інструкції з інсталяції комп'ютерної системи з IoT компонентами;
- оформлення інструкцій.

При оцінюванні тривалості виконання окремих робіт використовують нормативи часу або попередній досвід. До таких нормативів відносять тривалість написання операцій (команд), які для окремих підприємств

становлять: для однієї операції – 0,5-1,6 год та 8 годин для п'яти операцій (тривалість зміни).

У разі їх відсутності звертаються до експертних оцінок по встановленню тривалості кожного етапу, яка при трьох оцінках обчислюється за формулою [31]

$$T_{ec} = (t_{min} + 4t_{n.i} + t_{max}) / 6, \quad (4.1)$$

При двох оцінках, експертна оцінка обчислюється за формулою:

$$T_{ec} = (3t_{min} + 2t_{max}) / 5, \quad (4.2)$$

де T_{ec} – очікуване (середнє) значення тривалості виконання етапу (стадії);

t_{min} – мінімальна оцінка тривалості виконання етапу;

$t_{n.i}$ – найбільш імовірна оцінка тривалості виконання етапу;

t_{max} – максимальна оцінка тривалості виконання етапу.

Дані витрат часу на виконання окремих стадій (етапів) можна звести у табл. 4.1.

Таблиця 4.1

Основні етапи виконання НДР

№ та назва етапу	Середній час виконання стадії (етапу) інженером, год.	
	Інженер	Керівник
1. Обґрунтування актуальності теми дипломної роботи магістра	6	12

№ та назва етапу	Середній час виконання стадії, год.	
	Інженер	Керівник
2. Аналіз наукових публікацій та інтернет-джерел для визначення сучасних підходів до проектування архітектури комп'ютерних систем «розумний цінник»	14	8
3. Аналіз методів штучного інтелекту для визначення конкурентної вартості товарів	16	4
4. Розробка концептуальної схеми комп'ютерної системи «розумний цінник» з інтеграцією інтелектуального сервісу визначення вартості товару	24	6
5. Забезпечення організації взаємодії IoT, хмарних та інтелектуального сервісів	32	4
5 Реалізація прототипу системи відображення даних з «хмарного сервісу»	32	2
6 Створення інструкції з інсталяції комп'ютерної системи з IoT компонентами	12	2
7 Оформлення інструкцій	6	2
Разом	142	40

Витрати часу керівника на виконання окремих стадій (етапів) при недостатній кількості інформації доцільно приймати в межах 5% сумарних витрат часу інженерів на виконання цих стадій (етапів).

4.2. Визначення витрат на оплату праці та відрахувань на соціальні заходи

Відповідно до Закону України «Про оплату праці» заробітна плата – це «винагорода, обчислена, як правило, у грошовому виразі, яку власник або уповноважений ним орган виплачує працівникові за виконану ним роботу».

Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійно-ділових якостей працівника, результатів його праці та господарської діяльності підприємства. Заробітна плата складається з основної та додаткової оплати праці.

Основна заробітна плата нараховується на виконану роботу за тарифними ставками, відрядними розцінками чи посадовими окладами і не залежить від результатів господарської діяльності підприємства.

Додаткова заробітна плата – це складова заробітної плати працівників, до якої включають витрати на оплату праці, не пов'язані з виплатами за фактично відпрацьований час. Нараховують додаткову заробітну плату залежно від досягнутих і запланованих показників, умов виробництва, кваліфікації виконавців. Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна з/п складається із прямої з/п і доплати, яка при укрупнених розрахунках становить 25% – 35% від прямої з/п. При розрахунку з/п кількість робочих днів в місяці слід приймати – 21 дні/міс., що відповідає 168 год./міс. Розмір місячних окладів керівника та інженерів слід приймати згідно існуючих на даний час норм. Основна заробітна плата розраховується за формулою:

$$Z_{осн.} = T_c \cdot K_r \quad (4.3)$$

де T_c – тарифна ставка, грн.;

K_r – кількість відпрацьованих годин.

Посадові оклади (тарифні ставки) за розрядами Єдиної тарифної сітки визначаються шляхом множення окладу (ставки) працівника 1 тарифного розряду на відповідний тарифний коефіцієнт. У разі коли посадовий оклад (тарифна ставка) визначені у гривнях з копійками, цифри до 0,5 відкидаються, від 0,5 і вище – заокруглюються до однієї гривні. У 2019 році посадові оклади (тарифні ставки) розраховуються згідно з Законом України «Про Державний бюджет України на 2019 рік».

Мінімальна зарплата у 2019 р. складає 4173,00 грн., в погодинному розмірі 25,13 грн., приймемо 90,00 грн. для інженера, для керівника – 140,00 грн.

Для інженера: $Z_{осн.} = 90,00 \cdot 142 = 12780,00$ грн.

Для керівника: $Z_{осн.} = 140,00 \cdot 40 = 5600,00$ грн.

Додаткова заробітна плата становить 10 – 15% від суми основної заробітної плати і обчислюється за формулою

$$Z_{дод.} = Z_{осн.} \cdot K_{дод.} \quad (4.4)$$

де $K_{дод.}$ – коефіцієнт додаткових виплат (0,1).

Для інженера: $Z_{дод.} = 12780,00 \cdot 0,1 = 1278,00$ грн.

Для керівника: $Z_{дод.} = 5600,00 \cdot 0,1 = 560,00$ грн.

Звідси загальні витрати на оплату праці ($B_{оп.}$) визначаються за формулою

$$B_{оп.} = Z_{осн.} + Z_{дод.} \quad (4.5)$$

Для інженера: $B_{оп.} = 12780,00 + 1278,00 = 14058,00$ грн.

Для керівника: $B_{оп.} = 5600,00 + 560,00 = 6160,00$ грн.

Таким чином загальна сума становить 20218,00 грн. Крім того, необхідно визначити відрахування на соціальні заходи:

- податок на доходи фізичних осіб: 18% – 3639,24 грн.;
- військовий збір: 1,5% – 303,27 грн.;
- єдиний внесок: 22% – 4447,96 грн.

У сумі зазначені відрахування становлять 41,5%. Отже, загальна сума відрахувань на соціальні заходи становитиме:

$$B_{c.з.} = \Phi ОП \cdot 0,415 \quad (4.6)$$

де $\Phi ОП$ – фонд оплати праці, грн.

У даному випадку сума відрахувань становить:
 $B_{c.з.} = 20218,00 \cdot 0,415 = 8390,47$ грн.

Проведені розрахунки витрат на оплату праці зведемо у табл. 4.2.

Таблиця 4.2

Зведені витрати на заробітну плату

Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахування на ФОП, грн.	Всього витрати на оплату праці, грн.
	Тарифна ставка, грн.	К-сть відпрацьованих	Фактично нарах. з/пл., грн.			
Інженер	90	142	12780,00	1278,00	5834,07	19892,07
Керівник проекту	140	40	5600,00	560,00	2556,40	8716,40
Разом			18380,00	1838,00	8390,47	28608,47

4.3. Розрахунок витрат на електроенергію

Затрати на електроенергію при використанні обладнання визначаються за формулою:

$$Z_e = W \cdot T \cdot S \quad (4.7)$$

де W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Згідно з постановою НКРЕКП України від 05.10.2018 р. № 1177 вартість електроенергії становить 243,71 коп./кВт.год.

Потужність комп'ютера – 600 Вт, а кількість годин роботи обладнання згідно табл. 4.1 – 182 години.

Затрати на електроенергію становлять: $Z_e = 0,6 \cdot 182 \cdot 2,4371 = 266,13$ грн.

4.4. Розрахунок витрат на матеріали

Результати розрахунку затрат на матеріали наведено у табл. 4.3.

Таблиця 4.3.

Визначення величини затрат на матеріал

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю, грн.	Затрати матеріалів, грн	Транспортно – заготівельні витрати, грн.	Загальна сума витрат на матеріали, грн
Пакет MS Office	шт.	1	6125	6125	-	6125

Найменування матеріальних ресурсів	Одиниця виміру	Норма витрат	Ціна за одиницю, грн.	Затрати матеріалів, грн	Транспортно – заготівельні витрати, грн.	Загальна сума витрат на матеріали, грн
Visual Studio 2013 SNGL OLP NL	шт.	1	8520	8520	-	8520
Компакт диски	шт.	2	6	12	-	12
Разом						14657

4.5. Розрахунок суми амортизаційних відрахувань

Характерною особливістю застосування основних фондів у процесі виробництва є їх відновлення. Для відновлення засобів праці у натуральному виразі необхідне їх відшкодування у вартісній формі, яке здійснюється шляхом амортизації.

Амортизація – це процес перенесення вартості основних фондів на вартість новоствореної продукції з метою їх повного відновлення.

Комп'ютери та оргтехніка належать до четвертої групи основних фондів. Для цієї групи річна норма амортизації дорівнює 60 % (квартальна – 15 %).

Для визначення амортизаційних відрахувань застосовуємо формулу:

$$A = \frac{B_6 \cdot H_A}{100\%} \quad (4.8)$$

де A – амортизаційні відрахування за звітний період, грн.,

B_6 – балансова вартість комп'ютера, на початок звітного періоду, грн..

H_A – норма амортизації, яку приймемо на рівні 15%.

Амортизаційні відрахування при балансовій вартості ПК у 20000 грн. та нормі амортизації на рівні 15%, амортизаційні відрахування становитимуть:

$$A = \frac{20000 \cdot 15\%}{100\%} = 3000,00 \text{ грн.}$$

4.6. Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

Накладні витрати можна встановити на рівні 20% від суми основної та додаткової заробітної плати працівників:

$$H_B = B_{оп} \cdot 0,2 \quad (4.9)$$

де H_B – накладні витрати, грн.,

$B_{оп}$ – суми основної та додаткової заробітної плати працівників, грн..

У даному випадку накладні витрати становитимуть:

$$H_B = 20218,00 \cdot 0,2 = 4043,60 \text{ грн.}$$

4.7. Складання кошторису витрат та визначення собівартості науково-дослідних робіт

Собівартість (C_B) науково-дослідних робіт розрахуємо за формулою:

$$C_B = B_{о.п.} + B_{с.з.} + Z_{м.в.} + Z_e + T_g + A + H_g \quad (4.10)$$

У даному випадку собівартість (СВ) науково-дослідних робіт розраховуємо за формулою: $C_B = 20218,00 + 8390,47 + 14657,00 + 266,13 + 3000,00 + 4043,60 = 50575,20$ грн.грн.

Результати проведених вище розрахунків зведемо у табл. 4.4.

Таблиця 4.4.

Кошторис витрат на науково-дослідних робіт

Зміст витрат	Сума, грн.	В % до загальної суми
Витрати на оплату праці (основну і додаткову заробітну плату)	20218,00	39,98%
Відрахування на соціальні заходи	8390,47	16,59%
Матеріальні витрати	14657	28,98%
Витрати на електроенергію	266,13	0,53%
Амортизаційні відрахування	3000,00	5,93%
Накладні витрати	4043,60	8,00%
Собівартість	50575,20	100,00%

4.8. Розрахунок ціни науково-дослідних робіт

Ціну науково-дослідних робіт можна визначити за формулою:

$$Ц = \frac{C_v \cdot (1 + P_{рен.}) + K \cdot B_{н.і.}}{K} \cdot (1 + ПДВ), \quad (4.11)$$

де $P_{рен.}$ – рівень рентабельності, 20 %;

K – кількість замовлень, од. (встановлюється лише при розробці програмного продукту та мікропроцесорних систем);

$B_{н.і.}$ – вартість носія інформації, грн. (встановлюється лише при розробці програмного продукту);

$ПДВ$ – ставка податку на додану вартість, (20 %).

Оскільки розробка є прикладною, і використовуватиметься тільки для одного підприємства, то для розрахунку ціни не потрібно вказувати коефіцієнти K та $B_{н.і.}$, оскільки їх в даному випадку не потрібно.

Тоді, формула для обчислення ціни розробки буде мати вигляд:

$$Ц = C_г \cdot (1 + P_{рен.}) \cdot (1 + ПДВ) \quad (4.12)$$

Ціна НДР становитиме: $Ц = 50575,20 \cdot (1 + 0,2) \cdot (1 + 0,2) = 72828,29$ грн.

Визначимо величину прибутку:

$$П = Ц - C_г \quad (4.13)$$

Прибуток буде становити: $П = 72828,29 - 50575,20 = 22253,09$ грн.

4.9. Визначення економічної ефективності і терміну окупності капітальних вкладень

Ефективність виробництва – це узагальнене і повне відображення кінцевих результатів використання робочої сили, засобів та предметів праці на підприємстві за певний проміжок часу.

Економічна ефективність (E_p) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E = \frac{П}{C_г} \quad (4.14)$$

де Π – прибуток;

C_B – собівартість.

Економічна ефективність становить:

$$E_p = \frac{22253,09}{50575,20} = 0,44.$$

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

$$T_p = \frac{1}{E_p} \quad (4.15)$$

В даному випадку термін окупності становить: $T_p = \frac{1}{0,44} = 2,27$ року.

Про доцільність розробки програми можна сказати при врахуванні критеріїв, які наведені у табл. 4.5.

Таблиця 4.5.

Техніко-економічні показники НДР

№ з/п	Показник	Значення
1	Собівартість, грн	50575,20
2	Плановий прибуток, грн	22253,09
3	Ціна, грн	72828,29
4	Економічна ефективність	0,44
5	Термін окупності, рік	2,27

У результаті проведених розрахунків встановлено, що собівартість розроблених методів і засобів збору, управління та відображення даних у комп'ютерних системах «розумний цінник» становить 50575,20 грн., а термін окупності 2,27 року. Тому доцільність такої розробки є економічно обґрунтованою.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

5.1. Охорона праці

При дослідженні методів і засобів збору, аналізу та відображення даних у комп'ютерних системах «Розумний цінник» усі процеси повинні виконуватись із врахуванням вимог техніки безпеки на робочому місці, пожежної безпеки, відповідно з діючими нормативно-правовими актами та встановленими нормами. Дослідження передбачають пошук та аналіз інформації, розробку методів збору, аналізу та відображення даних у комп'ютерних системах «Розумний цінник», реалізацію взаємодії між хмарними сервісами та платами для розробки кінцевих пристроїв або шлюзів, а також розробка моделі автоматизації системи на основі IoT компонентів, що передбачає використання ЕОМ.

Перед початком усіх робіт проведено інструктаж з техніки безпеки на робочому місці відповідно до вимог Типового положення про інструктажі, спеціальне навчання та перевірку знань з питань пожежної безпеки на підприємствах, в установах та організаціях України, затвердженого постановою Кабінету міністрів України про порядок здійснення навчання населення діям у надзвичайних ситуаціях від 26 червня 2013 р. № 444.

Розробка програмного забезпечення та проведення досліджень відбувались відповідно до Закону України "Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями" затвердженого наказом міністерства соціальної політики України 14.02.2018 N207 Зареєстровано в Міністерстві юстиції України 25 квітня 2018р. за N508/31960.

Робоче місце відповідає ергономічним, організаційним вимогам, вказаних у вищеперерахованих документах. Електробезпека на робочому місці, повинна відповідати вимогам Правил безпечної експлуатації

електроустановок споживачів, затверджених наказом Держнаглядохоронпраці від 09.01.98 N 4, зареєстрованих у Міністерстві юстиції України 10.02.98 за N 93/2533 (НПАОП 40.1-1.21-98) [20].

Для забезпечення електробезпеки при роботі з ЕОМ під час проведення досліджень електромережа зроблена відповідно до правил:

- лінія електромережі живлення виконана як окрема групова трипровідна мережа, шляхом прокладання фазового, нульового робочого та нульового захисного провідників;
- нульовий захисний провідник використовується для заземлення (занулення) електроприймачів;
- електричне устаткування, електропроводи та кабелі мають апаратуру захисту від струму короткого замикання та інших аварійних режимів;
- під час монтажу та експлуатації електромережі необхідно повністю унеможливити виникнення електричного джерела загоряння внаслідок короткого замикання та перевантаження проводів, обмежено застосування проводів з легкозаймистою ізоляцією та застосовано негорючу ізоляцію;
- електричне устаткування підключено до мережі лише за допомогою справних штепсельних з'єднань і розеток заводського виготовлення;
- у штепсельних з'єднаннях та електророзетках є спеціальні контакти для підключення нульового захисного провідника [20].

Електробезпека під час розробки компонентів комп'ютерної системи «Розумний цінник» забезпечувалась завдяки використанню безпровідних технологій та напруги живлення в діапазоні 3.3 – 9 В, що зменшує шанс ураження струмом та його вплив при виникненні контакту з мережею чи аварійних ситуацій.

Пожежна безпека будівель та приміщень, де розміщені робочі місця, обладнані ЕОМ з ВДТ і ПП відповідає вимогам, встановленим державними будівельними нормами "Пожежна безпека об'єктів будівництва", затвердженими наказом Держбуду України від 03.12.2002 N 88 [19], а також правилами пожежної безпеки України, затвердженими наказом Міністерства

України з питань надзвичайних ситуацій від 19.10.2004 N 126, зареєстрованими в Міністерстві юстиції України 04.11.2004 за N 1410/10009.

Будівлі та приміщення, де розміщені робочі місця операторів, також повинні відповідати вимогам чинних санітарних норм, санітарних норм і правил, вказаних у ДСанПіН 3.3.2-007-98 [12].

На робочих місцях з ВДТ забезпечені оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря, відповідно до вимог нормативних документів.

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення відповідно до ДБН В.2.5-28-2018. Природне освітлення забезпечується через світлові прорізи, орієнтовані переважно на північ чи північний схід і коефіцієнт природною освітленості (КПО) не нижче ніж 1,5%.

Розраховується КПО за методикою, викладеною в ДБН В.2.5-28-2018. Штучне освітлення в приміщенні з робочими місцями, обладнаними ВДТ ЕОМ та ПЕОМ, забезпечувалось системою загального рівномірного освітлення.

В якості джерела світла для штучного освітлення використані люмінесцентні лампи типу ЛБ. Також під час дослідження методів та засобів проектування архітектури комп'ютерних систем «Розумний цінник» не було порушено наступних вимог:

- самостійного ремонту ЕОМ, ВДТ чи ПП;
- внесення змін до конструкції ЕОМ;
- не торкатись задньої панелі системного блоку при включеному живленні;
- вживання напоїв та їжі на робочому місці;
- нагромадження біля ЕОМ, ВДТ, ПП предмети, які не використовуються для поточної роботи.

Дослідження методів і засобів збору, аналізу та відображення даних у комп'ютерних системах «Розумний цінник» проведено з дотриманням вимог техніки безпеки та охорони праці, електробезпеки, пожежної безпеки та санітарних норм при роботі з ЕОМ та ВДТ.

5.2. Оцінка стійкості роботи промислового цеху в умовах радіаційного забруднення при ядерному вибуху. Оцінка інженерного захисту робітників і службовців промислового об'єкту

Уражаючими факторами ядерного вибуху є: ударна хвиля, світлове випромінювання, проникаюча радіація, радіоактивне зараження, електромагнітний імпульс. Для забезпечення стійкості промислового об'єкту до впливу електромагнітних імпульсів ядерного вибуху необхідно провести відповідні розрахунки. При цьому вихідними даними для визначення міри впливу електромагнітних імпульсів і вжиття заходів щодо зменшення їх негативного впливу на промисловий об'єкт є наступні: промисловий цех розташовується на відстані $R = 10 \text{ км}$ км від ймовірного ядерного вибуху.

Очікувана потужність ядерного боєзапасу $q = 1500 \text{ кт}$, вибух наземний.

Елементи системи, схильні до дії ЕМІ:

1 Живлення апаратних: напруга 380 В по наземних неекраниваних кабелях $L_1 = 80 \text{ м}$. Кабелі мають вертикальне відхилення заввишки $l_1 = 1 \text{ м}$. Допустимі коливання напруги мережі $\pm 5\%$, коефіцієнт екранування кабелю $\eta = 2$.

2 Система організаційного обліку та керування підприємства складається з 35 ПК та активного і периферійного обладнання. Згадані пристрої виконані на мікросхемах, що мають струмопровідні елементи заввишки $l_3 = 0,05 \text{ м}$. Робоча напруга мікросхем 5 В. Живлення від загальної мережі напруги 220 В через трансформатор. Допустимі коливання напруги мережі $\pm 5\%$. Комп'ютерна мережа управління, на якій функціонує система має

горизонтальну лінію $L_2 = 500$ м і вертикальні відгалуження заввишки $l_2 = 2$ м .
Робоча мінімальна напруга живлення 4В. Допустимі коливання напруги мережі $\pm 5\%$, коефіцієнт екранування комп'ютерної мережі $\eta = 2$.

Таблиця 5.1

Вхідні дані для оцінки дії ЕМП на стійкість промислового об'єкту

Відстань, км.	Потужність, кт	Довжина, м		Допуск
		L1	L2	%
10	1500	80	500	5

Розрахуємо очікувані на об'єкті промислового цеху максимальні значення вертикальної (ЕВ) і горизонтальної (ЕГ) складових напруженості електричного поля [27]:

$$E_B = 5 \cdot 10^3 \cdot \frac{(1+2 \cdot R)}{R^3} \cdot \lg(14,5 \cdot q), \text{ В/м} \quad (5.1)$$

$$E_G = 10 \cdot \frac{(1+2 \cdot R)}{R^3} \cdot \lg(14,5 \cdot q), \text{ В/м} \quad (5.2)$$

де R — відстань об'єкту від вірогідного ядерного вибуху;

q — очікувана потужність ядерного боєзапасу.

$$E_B = 5 \cdot 10^3 \cdot \frac{(1+2 \cdot 10)}{10^3} \cdot \lg(14,5 \cdot 1500) = 455,43 \text{ В/м}$$

$$E_G = 10 \cdot \frac{(1+2 \cdot 10)}{10^3} \cdot \lg(14,5 \cdot 1500) = 0,91 \text{ В/м.}$$

Визначимо максимальну очікувану напругу наводок [17]:

а) у системі електроживлення:

$$U_B = \frac{E_B \cdot l_1}{\eta}, \quad (5.3)$$

$$U_r = \frac{E_r \cdot L_1}{\eta}, \quad (5.4)$$

де l_1 – висота вертикального відхилення кабелю до верстатів;

L_1 – довжина екранованого кабелю;

η – коефіцієнт екранування кабелю.

$$U = \frac{455,43 \cdot 1}{1} = 227,72 \text{ В}$$

$$U = \frac{0,91 \cdot 80}{2} = 36,43 \text{ В}$$

б) у комп'ютерній мережі промислового об'єкту:

$$U = \frac{E_B \cdot l_2}{\eta}, \quad (5.5)$$

$$U = \frac{E \cdot L}{\eta}, \quad (5.6)$$

де l_2 – висота вертикального відгалуження комп'ютерної мережі управління;

L_2 – довжина горизонтальної лінії комп'ютерної мережі управління;

η – коефіцієнт екранування кабелю.

$$U = \frac{455,43 \cdot 2}{1} = 455,43 \text{ В}$$

$$U = \frac{0,91 \cdot 500}{2} = 227,72 \text{ В}$$

в) в комп'ютерах та периферійних пристроях:

$$U = \frac{E_B \cdot l_3}{\eta}, \quad (5.7)$$

де l_3 – висота струмопровідних елементів;

η – коефіцієнт екранування кабелю.

$$U = \frac{455,43 \cdot 0,05}{2} = 11,39 \text{ В}$$

Визначимо допустиму максимальну напругу наводок [16]:

а) у мережі живлення промислового об'єкту:

$$U_d = U + U(5\%), \quad (5.8)$$

де U — напруга живлення щитових;

$$U_{д1} = 380 + \frac{380 \cdot 5}{100} = 399 \text{ В}$$

б) у комп'ютерній мережі:

$$U_{д2} = 4 + \frac{4 \cdot 5}{100} = 4,2 \text{ В}$$

в) у комп'ютерах та периферійних пристроях:

$$U_{д4} = 5 + \frac{5 \cdot 5}{100} = 5,25 \text{ В}$$

Розрахуємо коефіцієнт безпеки [16]:

$$K = 20 \cdot \lg \left(\frac{U_d}{U_e} \right), \text{ дБ} \quad (5.9)$$

де U_d — допустима максимальна напруга наведень в пристрої введення, ЕОМ, блоці управління

U_e — очікувана максимальна напруга наведень в комп'ютерах та периферійних пристроях.

$$K = 20 \cdot \lg \left(\frac{5,25}{11,39} \right) = -6,72 \text{ дБ}$$

Зведемо отримані дані в таблицю (табл. 4.3).

Таблиця 5.2

Результати оцінки стійкості промислового об'єкту до дії ЕМІ

Елементи системи	Допустима напруга мережі, В	Напруженість електричного поля, В/м		Наводки напруги в струмопровідних елементах, В	
		Е _В	Е _Г	U _В	U _Г
Електропостачання апаратної	399	455,43	0,91	227,72	36,43
Комп'ютери, периферійні пристрої	5,25	455,43	0,91	11,39	—
Комп'ютерна мережа, активне обладнання	4,2	455,43	0,91	455,43	227,72
Коефіцієнт безпеки $K = -6,72 \text{ дБ} \ll 40 \text{ дБ}$.					

Локальна комп'ютерна мережа та обладнання може опинитися в зоні дії ЕМІ наземного ядерного вибуху. Нестійкими елементами об'єкту є: комп'ютери та периферійне обладнання, а також локальна комп'ютерна мережа.

Об'єкт не стійкий до дії ЕМІ, оскільки коефіцієнт безпеки значно менше задовільного значення, що становить до 40 дБ. Для підвищення стійкості роботи об'єкту до дії ЕМІ ядерного вибуху необхідно провести наступні заходи:

- кабель живлення щитових екранувати;
- комп'ютери і периферійні пристрої заземлити;
- для комп'ютерів встановити відключаючі пристрої.

Такі заходи дозволяють забезпечити працівників і службовців від уражаючої дії ядерного вибуху та радіаційного забруднення при збереженні працездатності інформаційної інфраструктури підприємства.

РОЗДІЛ 6

ЕКОЛОГІЯ

6.1. Електромагнітне забруднення довкілля, його вплив на людину.

Шляхи його зменшення

Електромагнітну енергію використовують у радіо-, радіорелейному і космічному зв'язках, радіолокації, радіонавігації, на телебаченні, у металургії та металообробній промисловості для індукційного плавлення, зварювання, напилювання металів, у деревообробній, текстильній, легкій та харчовій промисловості, у радіоспектроскопії, сучасній обчислювальній техніці, медицині тощо.

У виробничих приміщеннях з комп'ютерною технікою джерелами електромагнітного випромінювання є неекрановані робочі елементи високочастотних установок (індуктори, конденсатори, високочастотні трансформатори, фідерні лінії, батареї конденсаторів, котушки коливальних контурів тощо). При експлуатації ВЧ-, ДВЧ-, УВЧ-передачів на радіо- та телецентрах джерелами електромагнітного випромінювання є високочастотні генератори, антенні комутатори, пристрої складання потужностей електромагнітного поля, комунікації (від генератора до антенного пристрою), антени.

Ступінь опромінення працюючих залежить від кількості розміщуваних у приміщенні передавачів (в окремих зонах, на радіо- та телецентрах їх може бути до 20), їх потужності, ступеня екранування, розміщення окремих блоків всередині приміщення і поза його межами.

Для всіх видів зв'язку джерелом електромагнітного випромінювання є радіолокаційні станції, зокрема генератори, фідерні лінії, антени, окремі блоки енергії електромагнітного поля ЗВЧ- та НВЧ-діапазонів.

Для захисту людини від шкідливого впливу електромагнітних полів приймаються нормативи та стандарти. Треба зазначити, що будь-які норми та

стандарти, пов'язані із захистом людини від небезпечного впливу, завжди є компромісом між перевагами використання нових технологій та нової техніки і можливим ризиком, спричиненим цим використанням.

ДСТУ “Електромагнітні поля радіочастот” охоплює діапазон частот 60 кГц-300МГц. Він встановлює, що оцінка ЕМП в діапазоні 60 кГц-300МГц проводиться окремо з електричних і магнітних складових поля. Допустимі рівні протягом робочого дня по електричній складовій не повинні перевищувати 50 В/м знижуючись ступенями 5 В/м на міру підвищення частоти. По магнітній складовій встановлені рівні тільки для окремих ділянок діапазону: 5 А/м для частот 60 кГц-1.5 МГц та 0,3 А для частот 30-50 МГц. Допускається перевищення цих стандартів, але не більше ніж двократно, при скороченні робочого дня не менш як на 50%.

Для частот 300 МГц-30 ГГц гранично допустимі значення щільності визначаються як результат ділення нормованої величини енергетичного навантаження за робочий день на час впливу. Енергетичне навантаження протягом робочого дня не повинно перевищувати 200 мкВтЧгод/ см².

Ми бачимо, що електромагнітні поля дуже сильно впливають на людський організм. Вони негативно впливають майже на усі системи організму. Тому треба створювати певні методи захисту від їх дії. Найпоширенішими з таких методів є такі:

- зменшення щільності потоку енергії, якщо дозволяє даний технологічний процес або обладнання;
- захист часом (тобто обмеження часу знаходження у зоні джерела ЕМП);
- захист відстанню;
- екранування робочого місця чи джерела;
- раціональне планування робочого місця;
- застосування засобів попереджувальної сигналізації;
- застосування засобів особистого захисту.

Для зменшення впливу електромагнітних полів на персонал, який знаходиться у зоні дії радіоелектронних засобів необхідним є ряд захисних заходів: організаційні, інженерно-технічні та лікувально-профілактичні.

Слід сказати, що ще на етапі проектування взаємне розміщення об'єктів має бути забезпечено таким чином, щоб інтенсивність опромінення була мінімальною. Також треба заздалегідь забезпечити зменшення часу перебування персоналу у зоні опромінення. Потужність джерел випромінювання повинна бути найменшою з можливих.

Отож є досить багато методів захисту свого здоров'я від небезпеки на робочому місці з підвищеним електромагнітним фоном. Крім того треба вимагати від керуючих органів дотримування державних стандартів України та не порушувати їх норм.

6.2. Джерела шуму і вібрацій та методи їх знешкодження

Джерелами шуму при роботі з ПК є жорсткий диск, вентилятор блока живлення мережі, вентилятор, розташований на процесорі, швидкісні CD-ROM, механічні сканери, пересувні механічні частини принтера. При роботі вентиляційної системи ПК, яка забезпечує оптимальний температурний режим електронних блоків, створюється аеродинамічний шум. Окрім того, діють і інші зовнішні джерела шуму, не пов'язані з роботою ПК.

Шум, що створюється працюючими ПК, є широко смужним, постійним з аперіодичним посиленням при роботі принтерів. Тому шум повинен оцінюватися загальним рівнем звукового тиску по частотному коригуванню «А» та вимірюватися в дБА.

Параметрами постійного шуму, що підлягають нормуванню, є рівні звукового тиску в октавних смугах частот з середньо геометричними частотами 16; 31,5; 63; 125; 250; 500; 1000; 2000; 4000; 8000 Гц. Допустимі значення октавних рівнів звукового тиску, рівнів звуку на робочих місцях в

приміщеннях кабінетів комп'ютерної техніки необхідно приймати згідно з ДСанПіН 5.5.6.009-98 (табл. 6.1).

Таблиця 6.1

Гранично допустимі рівні шуму на робочому місці

Призначення приміщення та умови	Рівні звукового тиску, (дБ), в октавних смугах частот з середньгеометричними частотами, (Гц)										Рівні звуку, (дБ)
	16	31,5	63	125	250	500	1000	2000	4000	8000	
Кабінети (без роботи ПК)	-	-	63	52	45	39	35	32	30	28	40
Кабінети при роботі ПК	85	75	67	57	49	44	40	37	35	33	45

Кабінети з комп'ютерною технікою рекомендується обладнувати ПК, корегований рівень звукової потужності яких не перевищує 45 дБ.

Зниження рівня шуму можна здійснити таким чином:

- використанням блоків живлення ПК з вентиляторами на гумових підвісках;
- використанням ПК, в яких термодатчики вмонтовані в блоці живлення та в критичних точках материнської плати (процесор, мікросхеми чіпсету), які дозволяють програмним шляхом регулювати як моменти ввімкнення вентиляторів, так і їх швидкість обертання;
- переведення жорсткого диска в режим «сну» (Standby), якщо комп'ютер не працює на протязі визначеного часу. Цей час встановлюється в опціях керування напругою в операційних системах Windows, Linux, MacOS. Якщо в режимі Standby немає необхідності, його можна вимкнути в BIOS материнської плати;
- використанням ПК, в яких вентилятор на процесорі встановлено виробником (BOX - процесор);

- застосуванням материнських плат формату ATX та ATX-корпусів, що дозволяє регулювати автономну швидкість та моменти часу відмикання вентилятора блока живлення від електромережі;
- використанням 24-32-х швидкісних CD-ROM для застосувань, які створюють менше шуму, аніж швидкісні 48-50-х CD-ROM, або ж застосовувати привід з одночасним зчитуванням декількох доріжок CD;
- застосуванням принтерів колективного користування, розташованих на значній відстані від більшості робочих місць користувачів ПК;
- зменшенням шуму на шляху його розповсюдження через розміщенням звукоізолюючого відгородження у вигляді стін, перетинок, кабін;
- акустичною обробкою приміщень – зменшення енергії відбитих звукових хвиль шляхом збільшення площі звукопоглинання (розміщення на поверхнях приміщення облицювань, що поглинають звук, розміщенням в приміщеннях штучних поглиначів звуку).

Вібрація на робочих місцях, що створюється ПК, не повинна бути вище значень, визначених ДСанПіН5.5.6.009-98 (табл. 6.2). Рівні, що перевищують, наведені у табл. 6.2, гранично-допустимі рівні вібрацій негативно позначаються на здоров'ї людини, а при тривалому впливі провокують професійні захворювання.

Таблиця 6.2

Гранично допустимі рівні вібрації на робочому місці

Нормований параметр	Середньо геометричні частоти октавних смуг, (Гц)						Кориговані та еквівалентні кориговані рівні, (дБ)
	2	4	8	16	31,5	63	
Віброшвидкість	79	73	67	67	67	67	72
Віброприскорення	25	25	25	31	37	43	30

Основним джерелом шуму комп'ютерної техніки при виконанні дипломної роботи магістра є вентилятори блоків живлення комп'ютерів та накопичувачі. Згідно ДСанПіН 3.3.2.007-98 рівень шуму на робочому місці не повинен перевищувати 50 дБА. У приміщеннях з обчислювальною технікою рівні вібрацій не повинні перевищувати допустимих норм за ДСН 3.3.6.037-99. Допустимий рівень вібрації $L_{доп}$ – 115 дБА.

ВИСНОВКИ

Основні наукові та практичні результати роботи полягають у наступному:

1. Проаналізовано бізнес-процеси, характерні для підприємств торгівлі, визначено можливі шляхи і рівень їх автоматизації, обґрунтовано актуальність побудови систем «розумний цінник» у результаті якого встановлено необхідність подальшого дослідження та оптимізації процесів проектування архітектури, збору та аналізу даних щодо товарів, актуального відображення цін за допомогою електронних цінників, що дало змогу визначити шляхи підвищення ефективності функціонування підприємств торгівлі.

2. На основі аналізу наукових публікацій і практичних досліджень побудови «розумних систем» встановлено, що ефективним способом організації таких систем є використання технології Internet of Things, що дає змогу забезпечити гнучкість та масштабованість рішень при розробці систем «розумний цінник». Перевага «розумних речей» над іншими підходами полягає у можливості одночасного підключення великої кількості пристроїв, застосування стандартизованих і користувацьких протоколів передачі даних через мережу Інтернет, забезпечення ефективного моніторингу та управління бізнес-процесами.

3. Обґрунтовано організацію систем «розумний цінник» з використанням «розумних пристроїв» та хмарних сервісів, що дає змогу забезпечити одержання або передачу актуальних даних із сховищ даних, інтегрувати засоби аналітики даних при формуванні ціни товарів, однак відкритими ще залишаються питання безпеки передачі даних, побудови ефективної архітектури та формалізації таких систем.

4. У результаті аналізу принципів організації «розумних систем» встановлено, що доцільним є використання таких структурних компонентів як кінцеві пристрої, шаблони взаємодії та протоколи передачі даних з опрацюванням та управлінням даними на стороні хмарних сервісів, що дало

зможу забезпечити гнучкість реалізації архітектури таких систем, їх масштабованість та надійність експлуатації.

5. Спроектовано та запропоновано універсальну функціональну архітектуру системи «розумний цінник» на основі компонентного представлення та взаємодії кінцевих пристроїв, локального шлюзу, сервісів доступу та управління даними, що дало змогу забезпечити надійну та захищену комунікацію між складовими системи.

6. Формалізовано компоненти архітектурного рішення щодо побудови систем «розумний цінник» на основі теоретико-множинних нотацій, що дало змогу забезпечити однозначне трактування компонентів архітектури, зв'язків між ними та забезпечити альтернативний вибір апаратної і програмної реалізації складових таких систем.

7. На основі мультиагентного підходу до опису програмних систем, запропоновано та формалізовано сервіс збору даних про товари, що інтегрується у систему «розумний цінник» на рівні хмарного сервісу і дає змогу накопичувати дані про вартість, характеристики і відгуки про товари, що є базою для подальшого аналізу та ціноутворення.

8. На основі методу колаборативної фільтрації, формалізовано процес аналізу характеристик товарів, зокрема щодо формування ціни, визначення подібних товарів, статистики купівлі товарів і вподобань споживачів, що дало змогу враховувати ці критерії і підвищити ефективність управління бізнес-процесом з можливістю формування та миттєвої реакції в умовах зміни ринку.

9. Розроблено програмне забезпечення, що реалізує запропоновану архітектуру організації системи «розумний цінник» шляхом використання засобів NodeMCU, IoTHub, налаштуванням параметрів апаратного та програмного забезпечення при оновленні та візуалізації даних, а також протоколу MQTT, що дало змогу підтвердити ефективність застосування такої організації системи.

10. Проведено тестування двосторонньої взаємодії між компонентами архітектури системи «розумний цінник», у результаті чого підтверджено

можливість керування кінцевими пристроями (електронними цінниками) з хмарного сервісу Azure.

11. На основі платформи Onlizer реалізовано сервіс збору даних про товари з можливістю запису у базу даних, яка спільно використовується із сервісом аналітики даних та відображення цін у системі «розумний цінник».

12. За допомогою мови програмування Python та відкритих бібліотеки реалізовано сервіс аналізу даних про товари, що дає змогу формувати конкурентну ціну у системі «розумний цінник».

13. Проведено розрахунки для обчислення економічної доцільності впровадження запропонованих рішень і одержано результати на основі яких підтверджено економічну ефективність результатів дипломної роботи магістра.

14. Проведено аналіз вимог з охорони праці при використанні комп'ютерних систем «Розумний цінник», що дало змогу визначити шляхи дотримання вимог техніки безпеки при експлуатації систем, що передбачають використання комп'ютерної техніки.

15. Проведено оцінку стійкості роботи промислового об'єкту в умовах радіаційного забруднення при ядерному вибуху та забезпечення безпеки життєдіяльності населення в умовах надзвичайних ситуацій природного походження.

16. Розглянуто питання електромагнітного забруднення довкілля, його впливу на людину, шляхи забезпечення негативних факторів на життя біологічних об'єктів, а також проаналізовано джерела шуму і вібрацій, методи їх знешкодження.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Building the firmware - NodeMCU Documentation. URL: <https://nodemcu.readthedocs.io/en/dev/en/build/> (дата звернення 10.10.2019 р.).
2. Familiar B. Microservices, IoT, and Azure . 2015. 212 с.
3. Jamthe S. IoT Disruptions: The Internet of Things. Innovation & Jobs. 58 p.
4. IoT Hub REST | Microsoft Docs [Електронний ресурс] URL: <https://docs.microsoft.com/en-us/rest/api/iothub/?redirectedfrom=MSDN> (дата звернення 16.10.2019 р.).
5. IoT-шлюзи: автоматизація виробництва на рівні Індустрії 4.0 - Control Engineering Russia URL: http://www.controlengrussia.com/internet-veshhej/iot_gateways/ (дата звернення 15.11.2019 р.).
6. IoT: від «розумних» лампочок до передових технологій виробництва / Новини / IT українською URL: <http://it-ua.info/news/2016/06/21/iot-vd-rozumnih-lampochok-do-peredovih-tehnology-virobnictva.html> (дата звернення 18.11.2019 р.).
7. MQTT with NodeMCU – IoT Bytes URL:: <https://iotbytes.wordpress.com/mqtt-with-nodemcu/> (дата звернення 18.11.2019 р.).
8. Pfister C. Getting started with Internet of Things. Maker Media, Inc. 2011. 322 p.
9. Waher P. Learning Internet of Things. Packt Publishing. 2015. 286 p.
10. What is Azure Event Hubs and why use it | Microsoft Docs URL: <https://docs.microsoft.com/en-us/azure/event-hubs/event-hubs-what-is-event-hubs>. (дата звернення 19.11.2019 р.).
11. Автоматизація бізнес-процесів підприємства. URL: http://stud.com.ua/37085/ekonomika/avtomatizatsiya_biznes_protseviv_pidpriyemstva (дата звернення 20.11.2019 р.).

12. Державні санітарні правила і норми роботи з візуальни... | від 10.12.1998. № 7. URL: <http://zakon0.rada.gov.ua/rada/show/v0007282-98> (дата звернення 24.11.2019 р.).

13. Бобильов Ю. П., Бригадиренко В.В., Булахов В. Л. Екологія: базовий підручник для студентів вищих навчальних закладів. 2014. 672 с.

14. Екологія та соціальна реклама URL: <http://ukrbukva.net/page,5,77699-Ekologiya-i-social-naya-reklama.html>. (дата звернення 18.11.2019 р.).

15. Кляuze В. П. Санітарія і гігієна праці. 2011. 182 с.

16. Основные сведения о Stream Analytics | Microsoft Docs. URL: <https://docs.microsoft.com/ru-ru/azure/stream-analytics/stream-analytics-introduction> (дата звернення 01.12.2019 р.).

17. Подключаем устройства Intel для интернета вещей к Microsoft Azure IoT Suite. Блог компании Intel / Хабрахабр. URL: <https://habrahabr.ru/company/intel/blog/267815/> (дата звернення 01.12.2019 р.).

18. Пономаренко В. С., Мінухін С. В., Знахур С. В. Теорія та практика моделювання бізнес – процесів. Харків: ХНЕУ. 2013. 244 с/

19. Про затвердження ДБН «Пожежна безпека об'єктів бу... | від 03.12.2002. № 88. URL: <http://zakon0.rada.gov.ua/rada/show/va088509-02> (дата звернення 04.12.2019 р.).

20. Про затвердження Правил безпечної експлуатації електр... | від 09.01.1998. № 4. URL: <http://zakon0.rada.gov.ua/laws/show/z0093-98> (дата звернення 04.12.2019 р.).

21. Про затвердження Правил охорони праці під час експлуатації... | від 26.03.2010. № 65. URL: <http://zakon5.rada.gov.ua/laws/show/z0293-10> (дата звернення 04.12.2019 р.).

22. Экологическая реклама – ReclamBox/ URL: <https://sites.google.com/site/reclambox/ekologiceski-prodvinutaa-reklama/ekologiceskaa-reklama> (дата звернення 27.11.2019 р.).

Додаток А

Тексти наукових публікацій

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
 Тернопільський національний технічний університет імені Івана Пулюя (Україна)
 Національна академія наук України
 Університет імені П'єра і Марії Кюрі (Франція)
 Маріборський університет (Словенія)
 Технічний університет у Кошице (Словаччина)
 Вільнюський технічний університет ім. Гедимінаса (Литва)
 Шяуляйська державна колегія (Литва)
 Жешувський політехнічний університет ім. Лукасевича (Польща)
 Білоруський національний технічний університет (Республіка Білорусь)
 Міжнародний університет цивільної авіації (Марокко)
 Національний університет біоресурсів і природокористування України (Україна)
 Наукове товариство ім. Шевченка
 ГО «Асоціація випускників Тернопільського національного технічного університету імені
 Івана Пулюя»

АКТУАЛЬНІ ЗАДАЧІ СУЧАСНИХ ТЕХНОЛОГІЙ

Збірник

тез доповідей

Том II

**VIII Міжнародної науково-технічної
конференції молодих учених та студентів**

27-28 листопада 2019 року



**УКРАЇНА
ТЕРНОПІЛЬ – 2019**

12.	С.О. Галап, В.В. Яцишин ФОРМАЛІЗАЦІЯ ПІДСИСТЕМИ ЗБОРУ ДАНИХ В СИСТЕМАХ «РОЗУМНИЙ ЦІННИК»	17
13.	І.О. Гарасимів, Д.В. Дмитрів ІНФОРМАТИЗАЦІЯ ОБЩИН	18
14.	Ю.І. Голояд КОМП'ЮТЕРНА СИСТЕМА РОСПІЗНАВАННЯ КНИГ НА ФОТОГРАФІЯХ	19
15.	Н.В.Грабовський, С.М.Квач, О.Б. Назаревич АНАЛІЗ ЗАСОБІВ АВТОМАТИЗАЦІЇ ТЕХНОЛОГІЧНОГО ПРОЦЕСУ ВИРОБНИЦТВА ПИВА	20
16.	Д.О. Гракова ЕФЕКТИВНІСТЬ ВИКОРИСТАННЯ ПОВІТРЯНИХ БАЗОВИХ СТАНЦІЙ В МОБІЛЬНІЙ МЕРЕЖІ	21
17.	Є.І. Григчук, П.П. Данів, Д.П. Стухляк ДОСЛІДЖЕННЯ СИСТЕМ ЗАБЕЗПЕЧЕННЯ КОМФОРТУ ТА ЕНЕРГОЕФЕКТИВНОСТІ ЖИТЛОВИХ ПРИМІЩЕНЬ	23
18.	Р.А. Склярів, Губич І.В. АНАЛІЗ МЕТОДІВ ПОДІЛУ ПРУТКІВ НА ШТУЧНІ ЗАГОТОВКИ	24
19.	Р.А. Склярів, І.В. Гуцалюк ВИМОГИ ДО ТЕХНОЛОГІЧНОГО ОСНАЩЕННЯ ЯКЕ ВИКОРИСТОВУЄТЬСЯ ДЛЯ ЗАТИСКУ ПРИЗМАТИЧНИХ ЗАГОТОВОК	26
20.	В.О. Дармограй А. М. Луцків АНАЛІЗ БІБЛІОТЕК ДЛЯ РЕАЛІЗАЦІЇ BLOCKCHAIN-ІНФРАСТРУКТУРИ ДЛЯ СИСТЕМ ІОТ	27
21.	М.І. Паламар, А.З. Джинджиристий ЗАСТОСУВАННЯ МЕТРИКИ КОСИНУСА КУТА ПРИ ПІДБОРІ КОМАНДИ РОЗРОБНИКІВ КОМП'ЮТЕРНИХ СИСТЕМ	29
22.	О.А. Дідуник, М.В. Дрозд, А.П. Заблоцький, А.М. Курко ДОСЛІДЖЕННЯ СИСТЕМ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ АДМІНІСТРАТИВНИХ ПРИМІЩЕНЬ	30
23.	Л.Р. Цюка, В.І. Довганич ВДОСКОНАЛЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ КОНТРОЛЮ ТА КЕРУВАННЯ КВАДРОКОПТЕРАМИ ТА ДРОНАМИ	31
24.	М.М. Долік ІНФОРМАЦІЙНА СИСТЕМА ОЦИФРУВАННЯ ДОКУМЕНТІВ ДЛЯ ЗБЕРЕЖЕННЯ ІСТОРИКО-КУЛЬТУРНОЇ СПАДЩИНИ УКРАЇНИ	32

УДК 004.021

С.О. Галай, В.В. Яцишин канд. техн. наук

Тернопільський національний технічний університет імені Івана Пулюя, Україна

**ФОРМАЛІЗАЦІЯ ПІДСИСТЕМИ ЗБОРУ ДАНИХ В СИСТЕМАХ
«РОЗУМНИЙ ЦІННИК»**

S.O. Galan , V.V. Yatsyshyn PhD, Assoc. Prof.

**FORMALIZATION OF DATA COLLECTION SUBSYSTEM IN
“SMART PRICE LABEL” SYSTEMS**

Автоматизація є одним із найбільш ефективних шляхів розвитку підприємств торгівлі, що дає змогу мінімізувати вплив людського фактору та швидко реагувати на зміни ринку. За допомогою програмно-апаратних комплексів можна значно підвищити продуктивність виконання задач і водночас зменшити витрати на обслуговування бізнес-процесів.

Враховуючи тенденції до міграції та одночасної роботи магазинів, супермаркетів та інших закладів торгівлі, як онлайн, так і офлайн, актуальним є впровадження комп'ютерних систем «розумний цітник». Перевагами таких систем є узгодженість цін на товари, динамічність реакції на зміну вартості товарів, економія витрат на обслуговування цітників.

Одним із шляхів реалізації систем «розумний цітник» є застосування підходу, що передбачає використання хмарних сервісів при зборі, аналізі та збереженні даних. Перевагою систем, побудованих на хмарних сервісах, є забезпечення ефективності, надійності, гнучкості і масштабованості архітектурних рішень без необхідності купівлі, налаштування та обслуговування власних центрів обробки інформації.

У загальному випадку система збору даних про товари представляє собою множину елементів, підсистем і комунікацій між ними. Процес збору даних P можна розглядати як деякий процес над яким виконується управління ззовні, тобто розклад запуску парсерів відносно конкретних ресурсів. При цьому, парсери та їх кількість, можна описати множиною агентів, над якими виконується управління і на які здійснюється вплив – A_1, \dots, A_n , n – кількість парсерів. Розклад запуску парсерів представляє собою множину керуючих сигналів k , $\{k \in K\}$. Окрім сигналів керування, парсерам відомо про сигнали зовнішнього середовища ε , сигнали відповіді від ресурсів на запит z , $\{z \in Z\}$, та про взаємодію різних парсерів-агентів між собою r , $\{r \in R\}$. Дана сукупність сигналів формують множину виходів Y – результат виконання парсингу. Формально процес керування парсингом товарів можна представити у вигляді відображення:

$$C(P_i) : K \times Z \times R \times \varepsilon \rightarrow Y \quad (1)$$

Оскільки, існує множина агентів-парсерів, то відповідно в асинхронному режимі можна виконувати сукупність процесів P_n . Відповідно, для запуску агентів існує множина керуючих сигналів K для кожного процесу P_i , що представляється у вигляді комбінації усіх можливих сигналів керування, тобто $K = K_1 \times \dots \times K_n$ – декартовий добуток множини керуючих сигналів. У результаті виконаного формального представлення парсера щодо збору даних про товари у конкуруючих організацій, пропонується для його практичної реалізації використати платформи з можливістю налаштування параметрів пошуку даних, запуску агентів за розкладом та асинхронним їхнім функціонуванням.

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ**

МАТЕРІАЛИ

VII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ

**«ІНФОРМАЦІЙНІ МОДЕЛІ,
СИСТЕМИ ТА ТЕХНОЛОГІЇ»**



11–12 грудня 2019 року

**ТЕРНОПІЛЬ
2019**

СЕКЦІЯ 3. КОМП'ЮТЕРНІ СИСТЕМИ ТА МЕРЕЖІ

В. Владика, Д. Величко, Г. Осухівська ТЕХНОЛОГІЇ ПЕРЕДАВАННЯ ДАНИХ В СИСТЕМІ «ЦИФРОВА ЛІКАРНЯ»	109
В. Барбарич, Ю. Івануса ШКІДЛИВИЙ ВПЛИВ WI-FI НА ОРГАНІЗМ ЛЮДИНИ	110
М. Бедрійчук РЕАЛІЗАЦІЯ МЕТОДУ АДАПТИВНОЇ СЕЛЕКЦІЇ КАНАЛІВ ЗВ'ЯЗКУ ДЛЯ СИСТЕМИ «РОЗУМНИЙ БУДИНОК»	112
Д. Войтина, В. Яцишин АНАЛІЗ ОСОБЛИВОСТЕЙ КОМП'ЮТЕРНИХ СИСТЕМ КРИТИЧНОГО ПРИЗНАЧЕННЯ	113
Р. Гаван, В. Яцишин ОСОБЛИВОСТІ АНАЛІЗУ ТОНАЛЬНОСТІ ТЕКСТУ НА ОСНОВІ АСПЕКТІВ ПРИ ВДОСКОНАЛЕННІ КОМП'ЮТЕРНИХ СИСТЕМ	114
С. Галан, В. Яцишин ОСОБЛИВОСТІ ПРОЕКТУВАННЯ «РОЗУМНИХ СИСТЕМ» З МОЖЛИВІСТЮ ВЗАЄМОДІЇ З ХМАРНИМИ СЕРВІСАМИ	115
Р. Гайдук, Д. Михалик РОЗРОБКА СОЦІАЛЬНОЇ МЕРЕЖІ ДЛЯ БІБЛІОТЕК З ВИКОРИСТАННЯМ .NET ТЕХНОЛОГІЙ	116
Ю. Голояд МЕТОДИ ТА АЛГОРИТМИ РОЗПІЗНАВАННЯ ОБРАЗІВ	117
І. Голуб, О. Яснії МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ МЕРЕЖЕВИХ КОМУТАТОРІВ ЗПІДТРИМКОЮ ТЕХНОЛОГІЙ GERON ТА LTE	118
А. Джинджиристий, М. Паламар АНАЛІЗ МЕТОДІВ КОЛАБОРАТИВНОЇ ФІЛЬТРАЦІЇ ПРИ ПІДБОРІ КОМАНДИ РОЗРОБНИКІВ КОМП'ЮТЕРНИХ СИСТЕМ	119
П. Євтух, В. Храпа МЕТОДИ ТА ЗАСОБИ УЗГОДЖЕННЯ ЦИФРОВИХ ПОТОКІВ ПРИ ЗАСТОСУВАННІ ІОТ ДО МЕДИЧНИХ ПРИЛАДІВ	120
А. Жуйвода МЕТОД ЗАБЕЗПЕЧЕННЯ ЯКОСТІ ПЕРЕДАЧІ ГОЛОСОВИХ СИГНАЛІВ В КОМП'ЮТЕРНИХ МЕРЕЖАХ	121
А. Жуйвода ПІДВИЩЕННЯ ЯКОСТІ ПЕРЕДАЧІ ПОТОКОВОГО ТРАФІКУ	122
О. Зимницький ВРАЗЛИВОСТІ РЕАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ МЕТОДІВ ЗАХИСТУ ПРОТОКОЛУ SSL/TLS	123
Б. Калиниченко, І. Грод ДОСЛІДЖЕННЯ ВРАЗЛИВОСТЕЙ МЕРЕЖІ ОФІСУ "ZoomSupport" ТА МЕТОДІВ ЇХ УСУНЕННЯ	124
В. Ковальов, С. Лупенко МЕТОДИ ТА ЗАСОБИ ПОБУДОВИ КОМП'ЮТЕРИЗОВАНИХ ДІАЛОГОВИХ СИСТЕМ ТОРГОВОГО ЦЕНТРУ	125
І. Купратний НЕЙРОМЕРЕЖІ У СИСТЕМАХ БІОМЕТРИЧНОЇ АУТЕНТИФІКАЦІЇ ЗА КЛАВІАТУРНИМ ПОЧЕРКОМ	126
О. Ліщук, Є. Тиш ПЕРЕВАГИ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ АГРЕГАЦІЇ КАНАЛІВ КОМП'ЮТЕРНОЇ МЕРЕЖІ	127

УДК 004.021

С. Галан, В. Яцишин

Тернопільський національний технічний університет імені Івана Пулюя

**ОСОБЛИВОСТІ ПРОЕКТУВАННЯ «РОЗУМНИХ СИСТЕМ» З
МОЖЛИВІСТЮ ВЗАЄМОДІЇ З ХМАРНИМИ СЕРВІСАМИ**

UDC 004.021

S. Galan, V. Yatsyshyn

(Teropil Ivan Puluj National Technical University, Ukraine)

**FEATURES OF "SMART SYSTEMS" DESIGN WITH THE ABILITY TO
INTERACT WITH CLOUD SERVICES**

Під розробкою «розумних» систем розуміють процес інтеграції та композиції смарт-пристроїв з використанням готових програмних рішень, або створенням нових, що об'єднуються в єдину систему для задоволення потреб користувачів. За інформацією Новосибірського державного технічного університету, їхні студенти розробили і впровадили розумний цінник, що представляє собою систему, до складу якої входить сукупність пристроїв і програмного забезпечення для відображення актуальної ціни товарів та аналізу поведінки споживачів.

Реалізацію електронного цінника виконано за допомогою електронного паперу, при цьому мікросхема керування та wifi-модуль інтегровано у каркас полиці. Даний цінник з'єднується за допомогою безпроводної мережі з сервером бази даних і автоматично змінює ціну при її оновленні в БД. Альтернативою такого підходу є використання хмарних сервісів та смарт-пристроїв. Хмарні сервіси дають змогу проводити аналітику даних шляхом використання інтелектуальних сервісів при формуванні вартості товарів на основі аналізу цін компаній-конкурентів. Хмарні сервіси призначені для виконання найрізноманітніших завдань та взаємодії з надзвичайно великою кількістю пристроїв та програм. Однак для повноцінної взаємодії необхідна підтримка наперед визначених стандартів взаємодії, протоколів, каналів зв'язку, шифрування, форматування даних. Але значна частина IoT пристроїв, давачів, елементів управління, плат для проектування в свою чергу, призначені для виконання величезного набору специфічних завдань і мають обмежені ресурси в апаратному і програмному аспекті, що призводить до виникнення ряду проблем при їх взаємодії.

Головне обмеження – це об'єм пам'яті для виконання програм. Його обмеження впливають на можливості обробки, зберігання, передачі даних та шифрування. Що викликає серйозні труднощі, адже хмарні технології дозволяють взаємодіяти лише по захищеному каналу зв'язку. Не будучи в стані з'єднати кінцеві пристрої через брандмауери і безпечно передавати дані для хмарних додатків, організації не зможуть досягти всіх переваг аналітичного управління виробничими процесами, які їм обіцяли при переході на використання хмарних обчислювальних потужностей. У висококласних IoT-шлюзів повинна бути можливість взаємодіяти з серверами Microsoft Azure або подібними системами інших виробників, використовуючи найбільш популярні транспортні протоколи. Наприклад, після підключення до мережі IoT-шлюз, застосувавши безпечний протокол Advanced Message Queuing Protocol, може зареєструватися через платформу Microsoft IoT Hub на хмарному сервері Azure. Таким чином, у нього з'явиться можливість відправляти і приймати дані, а також перевіряти справжність отриманої інформації. Виступаючи в ролі посередника при безпечній відправці повідомлень, платформа Azure IoT Hub дозволяє виробляти дистанційне конфігурування і керування пристроями. Хмарні сервіси мають готові реалізації SDK для взаємодії з ними з різноманітних платформ, використовуючи широкий набір засобів розробки та мов програмування. Але такі рішення для невеликих плат відсутні, що створює ще одну проблему в процесі створення IoT системи.

Додаток Б

Лістинг коду взаємодії IoT та хмарних сервісів на мові Lua

```

-- Інформація
print('init.luaver 1.2')
wifi.setmode(wifi.STATION)
print('set mode=STATION (mode='..wifi.getmode()..'')')
print('MAC: '..wifi.sta.getmac())
print('chip: '..node.chipid())
print('heap: '..node.heap())

SSID = "*****"
PASSWORD = "*****"

-- Підключення до WiFi
wifi.sta.config(SSID, PASSWORD)

wifi_codes = {
    [0] = "Idle",
    [1] = "Wait",
    [2] = "Incorrect password",
    [3] = "Can't find AP",
    [4] = "Fail",
    [5] = "New IP address"
}

-- Очікування отримання адреси
tmr.alarm(1,1000, 1, function()
    if wifi.sta.getip()==nil then

print("Connectig..."..wifi_codes[wifi.sta.status()].."")
        else
            print("Got IP:"..wifi.sta.getip())
            tmr.stop(1)
        end
    end)

```

```

    end
  end
)

-- Запустити наступний файл
tmr.alarm(0, 5000, 0, function() -- Zero as third parameter.
Call once the file.
    dofile('credential.lua')
end)

--Credential.lua
function char_to_pchar(c)
    return string.format("%%%02X", c:byte(1,1))
end
function encodeURI(str)
    return (str:gsub("[^%;/?:@&=+;%$w%-
_%!.~*%'(%)#]", char_to_pchar))
end
function encodeURIComponent(str)
    return (str:gsub("[^w%_%.!~*%'(%)]", char_to_pchar))
end

function _(hex) return string.char(tonumber(hex, 16)) end

function decodeURI(str)
    str = string.gsub(str, '%%(%x%x)', _)
    return str
end

--Генерація токєну доступу
function new_token(URL, KEY, expire_time)
    URL = string.lower(encodeURIComponent(string.lower(URL)));
    time = rtctime.get()
    expire_time = math.ceil(time + expire_time * 60)
    URL = URL.."\\n"..expire_time
    pass = encoder.fromBase64(KEY)

```

```

    base64 = crypto.toBase64(crypto.hmac("SHA256", URL, pass))
    base64URL = encodeURIComponent(base64)
    new_token = "SharedAccessSignature
sr="..URL.."&sig="..base64URL.."&se="..expire_time;
    new_token = new_token.."&skn="
    return new_token
end

sntp_codes = {
    [1] = "Fail DNS",
    [2] = "Fail memory",
    [3] = "Fail UDP",
[4] = "Noreponse"
}

-- Синхронізувати час системи з сервером
sntp.sync('pool.ntp.org',
    function(sec, usec, server)
        print("Clock Synced: "..sec..", "..usec..", "..server)
        dofile('azure.lua')
    end,
    function(error)
        print("SNTP sync Fail: "..sntp_codes[error])
    end)

--Azure.lua
BOARD = "onlizer-arduino-board"
HUB = "onlizer-boards.azure-devices.net"
KEY = "j8CH2g+WsAOAUZmqa8q4xxy0p/xVuHrmdK13NdqsVQ="
MQTT_PORT = 8883
USER_ID = HUB.."../"..BOARD

PASSWORD = new_token(HUB..'../devices/'..BOARD, KEY, 7 * 24 * 60)

topic="devices/"..BOARD.."/messages/events/"
command="devices/"..BOARD.."/messages/devicebound/#"

```

```

connected = false

esp8266 = mqtt.Client(BOARD, 240, USER_ID, PASSWORD)

print ("Connecting to MQTT broker. Please wait...")

esp8266:connect(HUB, MQTT_PORT, 1, 0,
  function(client)
    print("Connected to MQTT: "..HUB..":"..MQTT_PORT.." as
"..BOARD)
    setup()
  end,
  function(client, reason)
    print("Error Connecting: "..reason)
  end
)

function setup()

esp8266:subscribe(command_topic, 1, function(client)
  print("Subscribed to command and control topic.")
end)

tmr.alarm(2, 10000, tmr.ALARM_AUTO, publish_data)

esp8266:on("message", function(conn, topic, data)
  print(topic .. ":" )
  if (data ~= nil ) then
    print ( data )
  end
end )

esp8266:on("offline", function(client)
  print("MQTT Disconnected.")
  connected = false
end)

```

end

```
function publish_data()
```

```
    if connected == true then
```

```
        gpio.write(LEDPIN, gpio.LOW)
```

```
        TEMPERATURE = Thermistor(adc.read(0))
```

```
        --TEMPERATURE = adc.read(0)
```

```
        TIME = rtctime.get();
```

```
        print("Light: 100 Temperature: 20 Humidity: 55%")
```

```
        payload =
```

```
        "{ \"deviceId\" : \"\"..DEVICE..\"\", \"data\" :  
        {\"Temperature\":\"..TEMPERATURE..\", \"Time\" : \"..TIME..\"}}"
```

```
        esp8266:publish(telemetry_topic, payload, 1, 0,
```

```
function(client)
```

```
    print("Data published successfully.")
```

```
end)
```

```
        gpio.write(LEDPIN, gpio.HIGH)
```

```
    end
```

end

```
function Thermistor(RawADC)
```

```
    Temp = ((RawADC/1024)*5000)/10;
```

```
    return Temp;
```

end