

Центральноукраїнський національний технічний університет
Механіко-технологічний факультет
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“___” _____ 2026 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за першим (бакалаврським) рівнем вищої освіти
на тему
“Програмне забезпечення системи комунікаційної платформи
як сервісу CPaaS”

Виконав здобувач вищої освіти
IV курсу, групи KI-22-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Вакка Н.С.
«___» _____ 2026 р.

Керівник проекту
кандидат технічних наук, доцент
_____ Буравченко К.О.
«___» _____ 2026 р.
Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
Кафедра Кібербезпеки та програмного забезпечення
Освітній ступінь бакалавр
Галузь знань 12 "Інформаційні технології"
Спеціальність 123 "Комп'ютерна інженерія"
Освітньо-професійна (освітньо-наукова) програма "Комп'ютерна інженерія"

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.т.н., проф.
_____ Олексій СМІРНОВ
« 27 » лютого 2026 року

ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА ПЕРШИМ (БАКАЛАВРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ ЗДОБУВАЧА ВИЩОЇ ОСВІТИ

Ваккі Назару Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи комунікаційної платформи як сервісу CPaaS*

2. Керівник роботи *Буравченко Костянтин Олегович, канд. техн. наук, доцент*
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 136-02 від 27.02.2026 року

3. Строк подання студентом роботи до захисту *23.05.2026 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи комунікаційної платформи як сервісу CPaaS*

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

1. Призначення та область використання.

2. Перегляд аналогічних існуючих систем.

3. Опис і обґрунтування проектних рішень.

4. Етапи програмування системи.

5. Впровадження системи в промислову експлуатацію.

6. Висновки

6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

Структурна схема системи *1 аркуш*

Функціональна схема системи *1 аркуш*

Діаграма процесів *1 аркуш*

Блок-схема алгоритму роботи додатку *2 аркуша*

7. Дата видачі завдання « 27 » лютого 2026 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2026 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2026 р.	
3.	Розробка моделі компонента	20.03.2026 р.	
4.	Розробка структур даних	25.03.2026 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2026 р.	
6.	Програмування алгоритмів	10.04.2026 р.	
7.	Оформлення ПЗ	17.04.2026 р.	
8.	Попередній захист роботи	23.05.2026 р.	

Дата видачі завдання
« 27 » лютого 2026 р.

Підпис керівника

Буравченко К.О.
(прізвище та ініціали)

Завдання прийнято до виконання
« 27 » лютого 2026 р.

Підпис здобувача

Вакка Н.С.
(прізвище та ініціали)

АНОТАЦІЯ

Вакка Н.С. Програмне забезпечення системи комунікаційної платформи як сервісу CPaaS. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2026.

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи комунікаційної платформи як сервісу CPaaS.

Метою розробки є програмне забезпечення системи комунікаційної платформи як сервісу CPaaS.

Результат роботи – програмна реалізація системи комунікаційної платформи як сервісу CPaaS.

В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

Розроблено зручний інтерфейс користувача. Наведені інструкції по роботі з програмними засобами.

Програма може використовуватися на ПЕОМ з ОС Windows 10/11.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, CPaaS

ABSTRACT

Vakka N.S. Software for the communication platform system as a service CPaaS. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2026.

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the communication platform system as a service CPaaS.

The purpose of the development is the software for the communication platform system as a service CPaaS.

The result of the work is the software implementation of the communication platform system as a service CPaaS.

In the process of working on the software model, an analysis of existing hardware and software was performed. All components of the developed software are fully described.

A convenient user interface has been developed. Instructions for working with software are provided.

The program can be used on a PC with Windows 10/11 OS.

The program was developed in the Python environment.

Keywords: computer engineering, CPaaS

3MIST

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	19
2.3 Розгорнута постановка завдання	21
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	23
3.1 Опис функціонування системи	23
3.2 Розробка структурної схеми.....	30
3.3 Розробка функціональної схеми	34
3.4 Розробка діаграми процесів.....	41
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	43
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	43
4.2 Захист розробленого програмного забезпечення.....	74
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	76
6 ОСНОВНІ ВИСНОВКИ.....	83
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	85

					ВКРБ-123.26.0003.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи комунікаційної платформи як сервісу CPaaS	Літ.	Аркуш	Аркушів
Розроб.	Вакка Н.С.					Б		
Перев.	Буравченко К.О.						1	92
						ЦНТУ КІ-22-2		
Н.контр.	Коваленко А.С.							
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АТМУ	—	автоматизована телефона мережа установи
БД	—	база даних
КМЗ	—	корпоративна мережа зв'язку
ЛОМ	—	локальна обчислювальна мережа
ОЗП	—	оперативно-запам'ятовувальний пристрій
ПАТМ	—	пристрій автоматизованої телефонної мережі
ПЗ	—	програмне забезпечення
ПП	—	програмний продукт
СПД	—	система передачі даних
СУБД	—	система управління БД
ТфОП	—	телефонний оператор
AVVID	—	архітектурна модель фірми Cisco Systems
CPaaS	—	комунікаційна платформа як послуга
CNG	—	Comfort Noise Generator. Генератор комфортного шуму
DSP	—	Digital Signal Processor. Процесор цифрової обробки сигналів
DTMF	—	Dual Tone Multi-Frequency. Багаточастотна система кодування
	—	цифр номера
GK	—	Gatekeeper. Гейткіпер . Виконує функції керування зоною
	—	мережі H.323
GW	—	Gateway. Шлюз. Апаратно-програмний комплекс, що забезпечує
	—	обмін даними між мережами різних типів
H.323	—	Рекомендація ITU-T, що визначає системи мультимедійного
	—	зв'язку в мережах з пакетною комутацією
H.248	—	Протокол керування транспортним шлюзом
IP	—	Internet Protocol. Протокол міжмережної взаємодії
LPC	—	Linear Prediction Coding. Кодування з лінійним проорокуванням
MCU	—	Multipoint Control Unit. Пристрій керування конференцією
MG	—	Media Gateway. Транспортний шлюз

MGCP	– Media Gateway Control Protocol. Протокол керування шлюзами
MP	Multipoint processor. Процесор для обробки інформації користувачів при централізованих конференціях
OSI	– Open System Interconnection. Взаємодія відкритих систем
PPP	– Point-to-Point Protocol. Протокол двостороннього зв'язку
RADIUS	Remote Authentication Dial-In User Service. Протокол автентифікації й авторизації абонентів, а також обліку обсягу наданих їм послуг
RAS	Registration Admission and Status. Протокол взаємодії термінального встаткування з gatekeeper. Входить у сімейство протоколів H.323
RSVP	– Resource Reservation Protocol. Протокол резервування ресурсів
RTCP	Real-time Transport Control Protocol. Протокол контролю транспортування інформації в реальному часі
SIP	– Session Initiation Protocol. Протокол ініціювання сеансів зв'язку
TAPI	Telephony Applications Programming Interface. Інтерфейс для програмування телефонних застосунків
TCP	Transmission Control Protocol. Протокол керування передачею (даних) Основний транспортний протокол у стеці протоколів TCP/IP.
TCP/IP	– Transmission Control Protocol/Internet Protocol. Стек протоколів, що забезпечують організацію зв'язку між комп'ютерами в мережі Інтернет
UDP	– User Datagram Protocol. Протокол передачі дейтаграмм користувача. Подібно TCP, використовує для доставки даних протокол IP. На відміну від TCP/IP, передбачає обмін дейтаграммами без підтвердження
VoIP	– Voice over Internet Protocol. Технологія, що дозволяє використовувати IP-мережу для передачі мовної інформації

ВСТУП

Актуальність теми. Комунікаційна платформа як послуга (CPaaS) – це хмарна модель надання сервісів, що дозволяє організаціям у реальному часі додавати комунікаційні можливості у використовувані бізнес-застосунки. Розроблювачі використовують технологію CPaaS для вбудовування комунікацій у сторонні бізнес-застосунки для створення інтегрованого користувацького інтерфейсу. Комунікаційні можливості, які надаються через програмний інтерфейс застосунку (API), включають службу коротких повідомлень, службу мультимедійних повідомлень, телефонію й відео.

Комунікаційні послуги можуть бути убудовані в такі бізнес-застосунки як програмне забезпечення для продажів, для додавання таких функцій як push-повідомлення й дзвінок по кличі. Найпоширеніший варіант використання CPaaS сьогодні – це інтеграція SMS-повідомлень у застосунок планування роботи організації. Наприклад, у кабінеті лікаря приватної клініки може бути встановлене програмне забезпечення для керування медичною практикою, що автоматично нагадує пацієнтам про призначений час візиту або про необхідність одержання нового рецепта на препарати. Клієнти можуть відповісти на текстове повідомлення для підтвердження одержання повідомлення, а застосунок може зафіксувати їхню відповідь.

Сьогодні багато постачальників CPaaS пропонують пакети, які надають попередньо впаковані, повністю функціональні функції зв'язку, які можуть бути додані в застосунки. Деякі постачальники CPaaS також надають функції забезпечення безпеки, такі як багатофакторна автентифікація. CPaaS звичайно оплачується щомісяця по фіксованій ставці за блоки повідомлень або хвилин. Деякі провайдери CPaaS пропонують модель білінгу з оплатою за дзвінок, за відео або за повідомлення.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи комунікаційної платформи як сервісу CPaaS.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем комунікаційної платформи як сервісу CPaaS.
- Дослідження системи комунікаційної платформи як сервісу CPaaS.
- Програмна реалізація системи комунікаційної платформи як сервісу CPaaS.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі комунікаційної платформи як сервісу CPaaS.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи комунікаційної платформи як сервісу CPaaS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Терміни комунікаційна платформа як послуга (CPaaS) і уніфікована комунікація як послуга (UCaaS) можуть вводити в оману, оскільки як CPaaS, так і UCaaS пропонують безліч однакових базових комунікаційних послуг. Однак вони відрізняються тим, як саме вони надають ці сервіси.

UCaaS нагадує традиційну локальну платформу, всі функції й застосунки якої перебувають на платформі постачальника послуг. На відміну від цього, CPaaS просто надає розроблювачам необхідні інструменти для налаштування й вбудовування комунікацій в існуючі бізнес-застосунки.

Теоретично, відносини між UCaaS і CPaaS є взаємодоповнюючими, а не конкурентними. Багато постачальників платформ UCaaS всі частіше займають проміжну позицію в суперечках UCaaS проти CPaaS, пропонуючи як API, так і платформи CPaaS. Коли платформа CPaaS перебуває поверх платформи UCaaS, вона може розширити можливості організації взаємодіяти з більше широким колом клієнтів без повного розриву й заміни комунікаційної інфраструктури організації.

1.2 Область застосування

CPaaS розширює принцип «Платформа як послуга» спеціально для комунікацій. Незважаючи на зростаюче зрушення у бік уніфікованих комунікацій як послуги (UCaaS), відмінність від CPaaS полягає в тому, що він прагне інтегрувати комунікації прямо в ключові бізнес-застосунки.

Це стало можливим, оскільки комп'ютерні гіганти всі частіше надають відкритий доступ до своєї IT-архітектури й хмарним сервісам. Використовуючи

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

API, бізнес-системи можна налаштовувати й налаштовувати відповідно до потреб, і ще краще те, що CPaaS може працювати поверх будь-яких існуючих успадкованих систем.

Як і у всіх PaaS, CPaaS пропонує перевага масштабованості, коли компанії платять тільки за ті послуги, які вони фактично використовують, що може зробити його набагато більше економічним, ніж інвестиції в локальне встаткування.

Крім того, мова йде не просто про підключення до хмари, а про віртуалізації комунікацій і інтеграцію їх із сервісами відкритої архітектури, і багато великих постачальників тепер легко справляються із цим.

Незважаючи на те, що між CPaaS і UCaaS і CCaaS – контакт-центром як послугою неминуче існує деякий збіг, насправді CPaaS – це скоріше створення механізму, що стоїть за службами зв'язку, а не просто їхнє надання.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи комунікаційної платформи як сервісу CPaaS, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

2.1 Огляд існуючих систем, технологій, архітектур, програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти

Twilio Flex

Лідер галузі. Широкий спектр застосунків. Голосування, чат, відео й SIPSome-кодування.

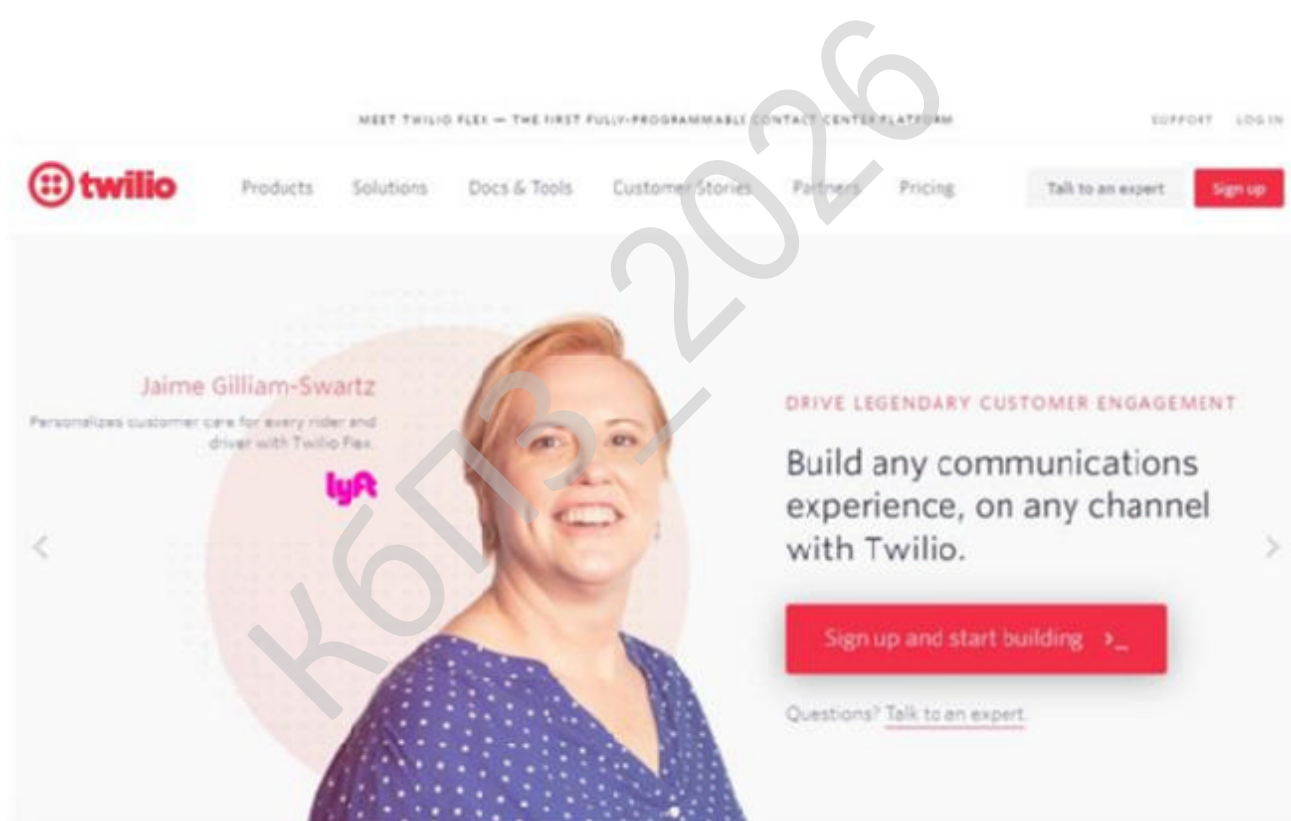


Рисунок 2.1 – Інтерфейс користувача Twilio Flex

Twilio Flex ефективно лідирує в області розробки послуг CPaaS із самого початку. Замість того щоб просто намагатися уніфікувати комунікації, вони надають кошти для повного налаштування роботи ваших комунікацій.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

Хоча це означає використання деякої форми кодування, це на самому ділі дуже простий і інтуїтивно зрозумілий у використанні, і робить неймовірно простим створення єдиного API, що може управляти вашими комунікаційними функціями.

Як альтернатива, якщо вам потрібно тільки програмувальне рішення для одного каналу, наприклад, голосові послуги, це теж можна зробити. Насправді Twilio пропонує окрему ціну для голосу, SMS, електронної пошти, чата й відео, а також спеціальні API для транкінгу WhatsApp і SIP.

Ціноутворення засноване на моделі з оплатою по факті використання, при цьому знижка за обсяг відкривається в міру використання.

Платформа Ringcentral Connect

Провідний постачальник CPaaS. Створено для хмарної бібліотеки APIsStrong функціональність.

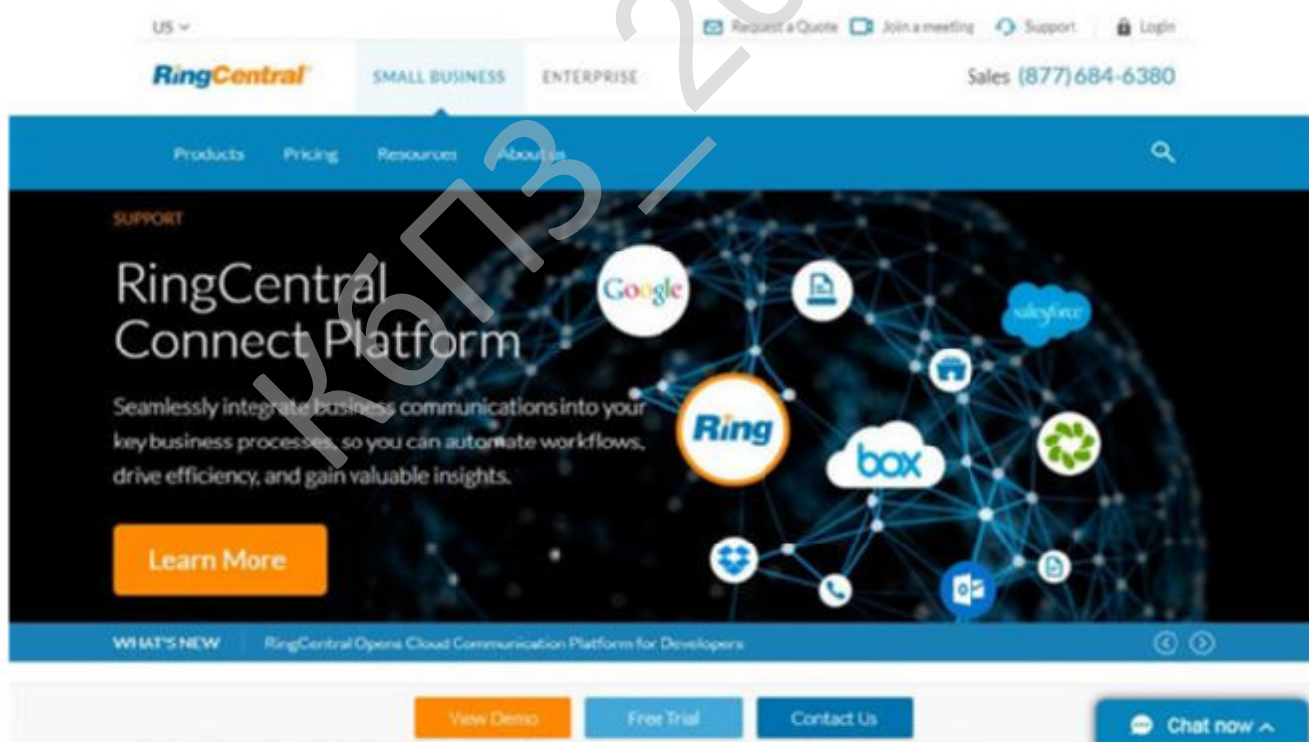


Рисунок 2.2 – Інтерфейс користувача RingCentral

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

RingCentral, можливо, не був першим сервісом, що запропонував CPaaS, але його платформа RingCentral Connect стала однією із провідних платформ для розроблювачів, коли мова заходить про хмарні комунікації.

На відміну від деяких відомих провайдерів, які створили свої системи для локального встаткування, RingCentral створював свої сервіси з нуля в хмарі. Це зробило їхніми лідерами галузі, і вони недавно підписали угоду про владу Avaya's UCaaS.

Платформа RingCentral Connect містить існуючу бібліотеку API-Інтерфейсів для налаштування й налаштування всіх аспектів зв'язку, не в останню чергу для інтеграції існуючих функцій RingCentral Office в інші застосунки.

Поки платформа не'стремлені до простоти, як роблять деякі інші провайдери CPaaS, на чому він сфальцьований, це функціональність, що робить цей потужний набір послуг для використання з його основним сервісом RingCentral Office.

Платформа Brightlink CPaaS

Видатне рішення CPaaS. Зосередьтеся на простоті використання пробної середи – платформа CPaaS



Рисунок 2.3 – Інтерфейс користувача Brightlink

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

Brightlink уже є відзначеним нагородами постачальником VoIP і UCaaS, але також пропонує платформу Brightlink CPaaS. Однак компанії вдається знайти гарний баланс між функціями й функціональністю, простотою й зручністю використання.

Результатом є дійсно потужна пропозиція CPaaS, що дозволяє інтегрувати функціональність зв'язку в усі види CRM і маркетингові інструменти, наприклад, з використанням різних API's.

Від Brightlink це означає для голосу, обміну повідомленнями й SIP, з додатковим “командний центр” яка функціонує як бібліотека для документації й ресурсів, а також як середовище тестування для ваших налаштувань API.

У результаті в підприємств з'являється можливість не тільки інтегрувати комунікації в рамках усього свого бізнесу, але й надавати доступ до мережних послуг «під ключ», не ускладнюючи процес.

Amazon Connect

Велика хмара CPaaS забезпечують Служби III, Шар безпеки, Візуальний редактор.

Ми вже згадували, що CPaaS дозволяє легко інтегруватися із хмарними сервісами й Amazon'я не квапився із цим, надаючи свої власні CPaaS-пропозиції через AWS за назвою Amazon Connect.

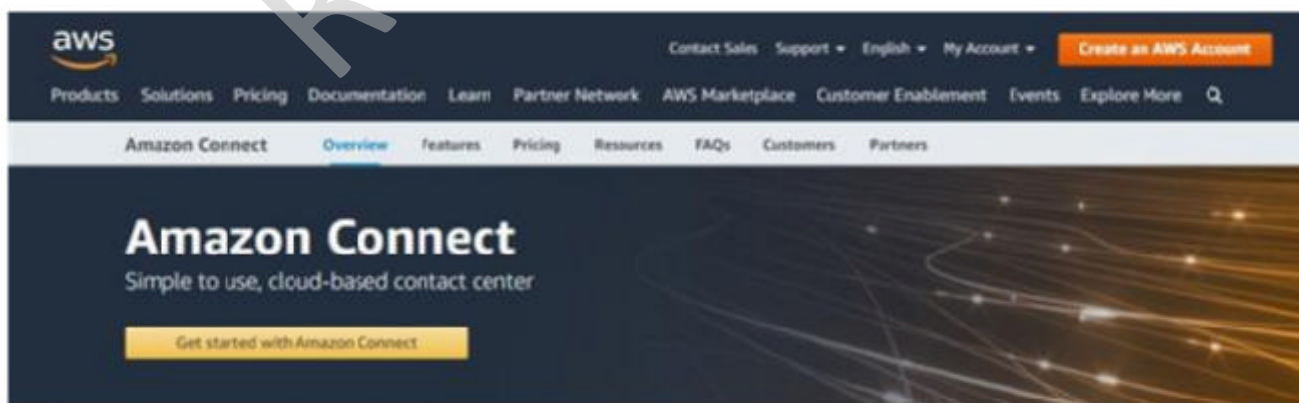


Рисунок 2.4 – Інтерфейс користувача Amazon Connect

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

В основі Amazon Connect лежить просте, масштабоване й надійне середовище для роботи з комунікаційними інтеграціями. Незважаючи на те, що ця реклама рекламується як рішення для хмарних контактів, вона недооцінює як інструменти, так і міць цієї платформи.

Для початку, Amazon Connect використовує ті ж AI і динамічні процеси, що й Alexa, що робить його не тільки підключеною платформою, але й розумною. Крім того, Amazon намагається спростити налаштування й використання робочих процесів і інтеграцій, надаючи візуальний редактор, щоб групи могли легко налаштовувати те, що їм потрібно, нічого не програмуючи.

Amazon може зробити це за допомогою убудованих у своє ядро API, але вони однаково можуть бути настроєні в міру необхідності. Крім того, інтеграції легко додати для інших платформ відкритого доступу, таких як CRM.

У результаті ви одержуєте простий спосіб створення власних правил і інтеграцій з найпростішою голосовою платформою, а оскільки він працює через AWS, це означає, що є додаткові переваги використання інших частин цього сервісу, таких як безпека й AI, а також тільки платити за ресурси, які ви використовуєте.

Messagebird

Багатоканальний CPaaS-провайдер. Omnichannel APIs. Глобальна операторська мережа. Просте налаштування.

MessageBird – це ще один постачальник CPaaS, на який варто звернути увагу, тому що є кілька причин: по-перше, вони намагаються зробити весь процес максимально простим і легким у використанні, надаючи візуальний редактор для розробки ваших API; і, по-друге, через те, що вони зосереджені на наданні посправжньому багатоканального досвіду.

Таким чином, у той час як інші служби CPaaS, як правило, фокусуються на голосовому зв'язку, текстах і електронній пошті, MessageBird також прагне інтегрувати широкий спектр застосунків і опцій чата поверх цих основних служб.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

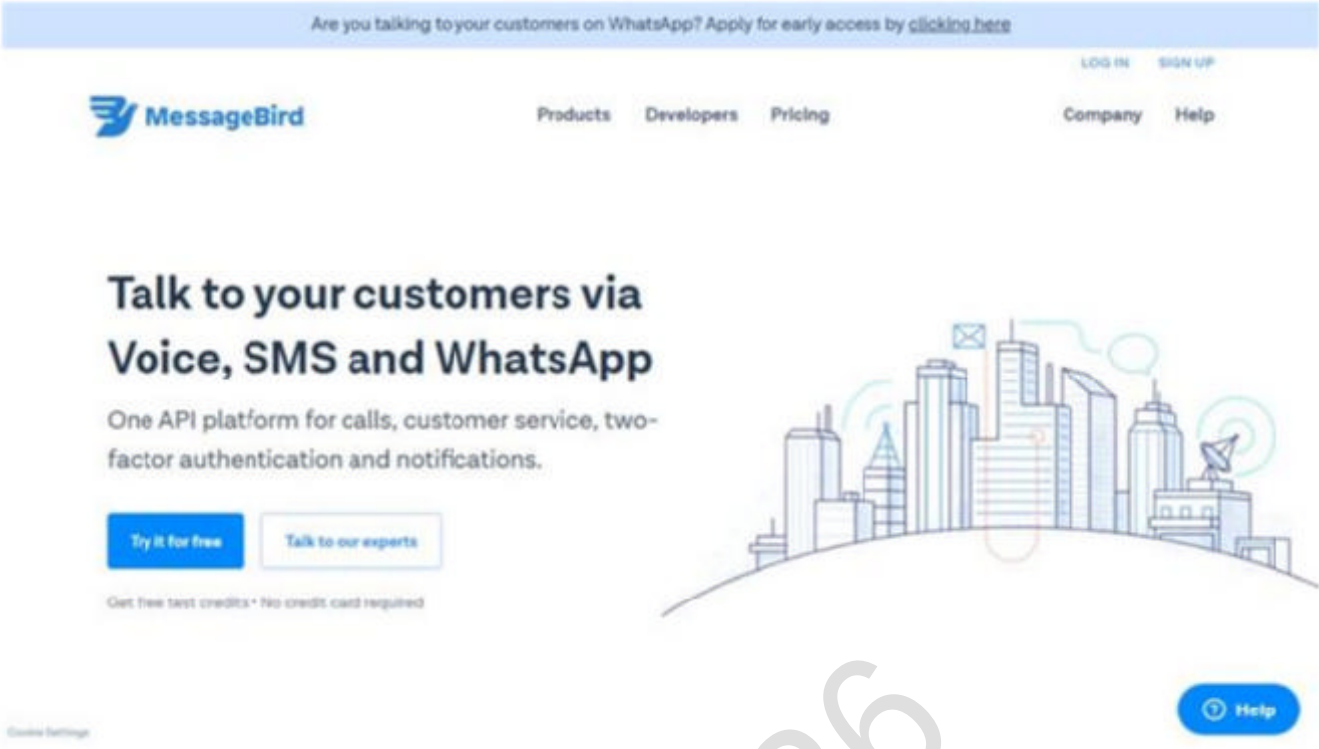


Рисунок 2.5 – Інтерфейс користувача MessageBird

Ще одним плюсом є те, що Messagebird також має величезну інфраструктуру прямого доступу до операторів зв'язку, що дозволяє йому не тільки надавати високоякісну голосовий й відео зв'язок, але й надавати цю послугу глобально.

Таке сполучення функцій дійсно робить MessageBird видатним провайдером CPaaS. Прості інструменти інструментальної панелі, які він надає, можуть зробити його привабливим тільки для широкого кола компаній, оскільки їм не потрібно наймати або відволікати ІТ-персонал просто для перенесення з їхньої інтеграції.

Дивно, але їхні ціни також конкурентоспроможні в багатьох областях, особливо в порівнянні з іншими провайдерами UCaaS або CCaaS.

Avaya IX Workplace

Підніміть спільну роботу й продуктивність на новий якісний рівень. Дайте співробітникам єдиний застосунок для телефонії, чата, нарад і спільної роботи. І нехай ці завдання вирішуються так само легко, як у звичних особистих застосунках.

Чим менше складностей, тим менше ризики й втрати продуктивності, що виникають через безліч різнорідних застосунків. Співробітники одержать відмінний зв'язок і відмінні інструменти для роботи. Вони зможуть краще захищати інтереси клієнтів і підвищувати престиж вашого бренда.

Avaya – лідер в області уніфікованих комунікацій. Ми можемо впроваджувати їх у будь-якому зручному для вас виконанні: у приватній, публічній або гібридній хмарі.

Голосовий зв'язок

Голосовий зв'язок залишається найважливішим компонентом комунікацій. Якщо в співробітників є система, що підтримує мобільність, обмін повідомленнями, організацію зустрічей і спільну роботу, вони можуть швидко й легко вишиковувати відносини із клієнтами й усередині компанії.

Спільна робота в групі

Рішення Team Collaboration від компанії Avaya дозволяє створити віртуальні кімнати для продуктивної роботи, незалежно від організаційних і географічних границь. Працюйте разом за допомогою аудіо, відео й групових чатов з функціями зберігання документів і керування завданнями.

Зустрічі

Розвивайте ділові відносини на зустрічах віч-на-віч за допомогою HD відео, аудіо й веб-застосунків, доступних вашим співробітникам, замовникам і партнерам практично на будь-якому пристрої.

UCaaS

Будьте впевнені, що рішення UCaaS (уніфіковані комунікації як послуга) – це самі останні версії ПЗ, висока безпека, надійне задоволення нормативних

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

вимог, тобто те, що необхідно для комунікацій і спільної роботи. Установка рішення у вашім ЦОД, у хмарі, або сполучаючи ці варіанти.

Пристрої

Правильний вибір пристроїв впливає на весь ваш бізнес: на обслуговування клієнтів, спільну роботу й продуктивність праці. Виберіть правильне рішення, що надасть необхідні інструменти кожному співробітникові.

Послуги

Досвідчені фахівці Avaya допоможуть вам перебороти труднощі, прискорити впровадження й домогтися найкращих результатів. Наша загальна мета – зробити ваш бізнес самим продуктивним і конкурентоспроможним.

Чому Avaya?

Всеосяжний досвід

Лідируючі позиції Avaya в області уніфікованих комунікацій визнані в Gartner Magic Quadrant, у дослідженнях Aragon Research Globe™, а також IDC MarketScape: Worldwide Unified Communications and Collaboration Vendor Assessment

Інновації й інвестиції

Автоматизуйте процеси, використовуйте штучний інтелект і можливості Інтернету Речей. Підвищуйте ефективність за допомогою машинного навчання. Володіючи більш ніж 4800 патентами, наші інновації здатні підтримати ваш бізнес і сьогодні, і в майбутньому.

Відкрита екосистема

Інтегруйте й кастомізуйте своє рішення за допомогою відкритих інтерфейсів API, комплектів розроблювача SDK і широких можливостей інтеграції на базі стандартів.

Широкий портфель продуктів

Рішення Avaya IX Workplace можна точно настроїти на ваші потреби. Ми працюємо з усіма компаніями, від самих більших до самих маленьких. Avaya здатна розгорнути системи уніфікованих комунікацій як у ЦОД замовника або хмарі, так і в гібридному варіанті.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Avaya IX Contact Center

Якщо клієнт щораз буде одержувати позитивне враження від спілкування з вами, ви заслужите довгострокову лояльність. Контакт-центр Avaya Intelligent Xperiences™ допоможе вирішити це завдання. Клієнти будуть користуватися найшвидшими й ефективними методами самообслуговування по найбільш зручним для себе каналам. При цьому ваш фахівець завжди буде готів допомогти їм якщо буде потреба. Ваша взаємодія завжди буде безперешкодною і контекстно-обумовленою. Фахівці контакт-центра будуть мати можливості працювати узгоджено й відповідати очікуванням клієнтів. Маючи потрібну інформацію, вони зможуть попереджати очікування клієнтів і швидше домагатися потрібних результатів.

Чим краще ви будете розуміти потреби клієнтів, тим більше ефективно стануть працювати ваші фахівці. Avaya – визнаний лідер в області рішень для контакт-центрів. Ми можемо розгорнути будь-який контакт-центр по вашім виборі: у хмарі – публічному або частці – або у вашім ЦОД.

Культура Customer-First

Ваші команди взаємодіють із загальною метою: обслуговувати ваших клієнтів. Вибудовування всіх ваших процесів, людей і застосунків у єдиному ключі сприяє ефективній спільній роботі, оперативному обміну інформацією й ефективним прийняттям рішень, що забезпечить клієнтоорієнтованість вашого бізнесу.

Результат у реальному часі

Ви збираєте кращі дані із усього вашого підприємства. Але чи витягаєте ви максимум користі із цього? Ви можете швидко перетворити ці дані в корисні ідеї в режимі реального часу, які може використовувати ваше керівництво, щоб забезпечити більше комфортні умови роботи як для клієнтів, так і для співробітників.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Природні комунікації

Ваші клієнти очікують простого обслуговування при кожній взаємодії з вашим підприємством. Забезпечуйте природні комунікації по всіх каналах для всіх сервісів: людських або автоматизованих, цифрових або фізичних, простих або комплексних. У результаті ваше взаємодії із клієнтами стає легким і зручним для всіх.

Avaya працює зі своїми клієнтами для оперативного рішення проблем, пов'язаних з бізнесом, і використання сценаріїв використання, які є ключовими для успіху їхнього бізнесу. Завдяки таким інноваціям Avaya пропонує можливості, які містять у собі використання ШІ, виявляють знання в режимі реального часу й інтегрують допоміжні функції протягом усього взаємодії із клієнтом для підвищення якості обслуговування клієнтів і співробітників.

Голосовий зв'язок

Діалог із клієнтом – це найважливіша частина вашої роботи. Визначите моменти, коли клієнтам потрібний контакт із живим співробітником, і автоматизуйте всі інші вхідні й вихідні взаємодії в каналах голосової й відеозв'язку, в електронній пошті й чатах.

Цифрові канали

Підключайте клієнта до самого кваліфікованого співробітника, що краще інших може вирішити його проблему. Нехай співробітники управляють всіма каналами зв'язку через єдиний користувальницький інтерфейс. Наші рішення враховують досвід взаємодії із клієнтом по всіх каналах і при кожному контакті пропонують клієнтові персоналізовану підтримку.

Робоче місце агента

Змініте до кращого роботу операторів і спілкування із замовниками. Оператор, що має необхідні інструменти, одержує всю інформацію про клієнта й може управляти всіма цифровими каналами. Якщо кожний оператор бачить загальну картину, йому легко передати завдання своїм колегам.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Керування персоналом

Підвищуйте продуктивність. Дотримуйте законодавчих і нормативних вимог. Використовуйте поточні й історичні дані для виявлення передового досвіду й удосконалювання роботи. Записуйте взаємодії, відслідковуйте виклики, аналізуйте голос клієнта, управляйте якістю, балансуйте навантаження, автоматизуйте навчання.

Штучний інтелект

Нехай ваші взаємодії стануть більше інтуїтивними. Це дозволить збільшити продуктивність співробітників і задоволеність клієнтів. Наші рішення з функціями штучного інтелекту допомагають приймати обґрунтовані рішення, спрощують роботу й автоматизують процеси.

Послуги

Досягайте результатів швидше. Покладетесь на наш багатий досвід в області контакт-центрів, інноваційних рішень і галузевих послуг, які допоможуть вам трансформувати досвід клієнтів і одержати реальні відчутні результати.

Чому Avaya?

Всеосяжний досвід

Покладетесь на лідера в області контакт-центрів і станьте однією з безлічі організацій в 170 країнах, які довіряють самої досвідченій компанії в нашій галузі.

Відкрита екосистема

Відкритість – найважливіша умова успішної взаємодії. Відкрита архітектура, інтерфейси API і комплекти розроблювача SDK дозволять вам легко й надійно налаштовувати наші рішення на свої потреби й потреби ваших замовників.

Інновації й інвестиції

Підвищуйте ефективність і одержуйте чудові результати за допомогою штучного інтелекту й машинного навчання. Ми допоможемо впровадити інновації на вашім підприємстві й побудувати цей "міст у майбутнє".

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Широкий портфель продуктів

Неважливо, на якому етапі розвитку клієнтського досвіду ви перебуваєте. Avaaya завжди допоможе вам. Що б ви не встановлювали – рішення для самообслуговування, послуги з підключенням до "живого" фахівцеві, нові канали взаємодії або нові пристрої для підвищення продуктивності – ми надамо вам кращі рішення в галузі.

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Як мова програмування обрана Python. Python – високорівнева мова програмування загального призначення з акцентом на продуктивність розроблювача й читаність коду. Синтаксис ядра Python мінімалістичний. У той же час стандартна бібліотека включає великий обсяг корисних функцій.

Python підтримує кілька парадигм програмування, у тому числі структурне, об'єктно-орієнтоване, функціональне, імперативне й аспектно-орієнтоване. Основні архітектурні риси – динамічна типізація, автоматичне керування пам'яттю, повна інтроспекція, механізм обробки виключень, підтримка багатопоточні обчислень і зручні високорівневі структури даних. Код у Python організовується у функції й класи, які можуть поєднуватися в модулі (які у свою чергу можуть бути об'єднані в пакети).

Еталонною реалізацією Python є інтерпретатор CPython, що підтримує більшість активно використовуваних платформ. Він поширюється вільно під дуже ліберальною ліцензією, що дозволяє використовувати його без обмежень у будь-яких застосунках, включаючи пропрієтарні. Є реалізації інтерпретаторів для JVM (з можливістю компіляції), MSIL (з можливістю компіляції), LLVM і інших. Проект PyPy пропонує реалізацію Python на самому Python, що зменшує витрати на зміни мови й постановку експериментів над новими можливостями.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Python – мова програмування, що активно розвивається, нові версії (з додаванням/змінюю язових властивостей) виходять приблизно раз у два з половиною року. Внаслідок цього й деяких інших причин на Python відсутні ANSI, ISO або інші офіційні стандарти, їхня роль виконує CPython.

Python портований і працює майже на всіх відомих платформах – від КПК до мейнфреймів. Існують порти під Microsoft Windows, практично всі варіанти UNIX (включаючи FreeBSD і Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і вище, Palm OS, OS/2, Amiga, AS/400 і навіть OS/390, Symbian і Android.

При цьому, на відміну від багатьох портуємих систем, для всіх основних платформ Python має підтримку характерних для даної платформи технологій (наприклад, Microsoft COM/DCOM). Більше того, існує спеціальна версія Python для віртуальної машини Java – Jython, що дозволяє інтерпретаторові виконуватися на будь-якій системі, що підтримує Java, при цьому класи Java можуть безпосередньо використовуватися з Python й навіть бути написаними на Python. Також кілька проектів забезпечують інтеграцію із платформою Microsoft .NET, основні з яких – IronPython і Python.Net.

Python підтримує динамічну типізацію, тобто тип змінної визначається тільки під час виконання. Тому замість «присвоювання значення змінної» краще говорити про «зв'язування значення з деяким ім'ям». У Python є убудовані типи: бульові, рядки, Unicode-рядки, цілі числа довільної точності, числа із плаваючою коми, комплексні числа й деякі інші. З колекцій Python підтримує кортежі (*tuples*), списки, словники (асоціативні масиви) і, починаючи з версії 2.4, безлічі. Всі значення в Python є об'єктами, у тому числі функції, методи, модулі, класи.

Додати новий тип можна або написавши клас (*class*), або визначивши новий тип у модулі розширення (наприклад, написаному мовою C). Система класів підтримує спадкування (одиначне й множинне) і метапрограмування. Можливе спадкування від більшості убудованих типів і типів розширень.

Всі об'єкти діляться на посиальні й атомарні. До атомарного ставляться *int*, *long*, *complex* і деякі інші. При присвоюванні атомарних об'єктів копіюється

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

їхнє значення, у той час як для посилальних копіюється тільки покажчик на об'єкт, таким чином, обидві змінні після присвоювання використовують те саме значення. Посилальні об'єкти бувають змінювані й незмінні. Наприклад, рядки й кортежі є незмінними, а списки, словники й багато інших об'єктів – змінюваними. Кортеж у Python є, по суті, незмінним списком. У багатьох випадках кортежі працюють швидше списків, тому якщо ви не плануєте змінювати послідовність, то краще використовувати саме їх.

Мова має чіткий і послідовний синтаксис, продуману модульність й масштабованість, завдяки чому вихідний код написаних на Python програм легко читаємий.

Python – стабільна й розповсюджена мова. Він використовується в багатьох проектах і в різних якостях: як основна мова програмування або для створення розширень і інтеграції застосунків. На Python реалізоване велика кількість проектів, також він активно використовується для створення прототипів майбутніх програм. Python використовується в багатьох великих компаніях.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи комунікаційної платформи як сервісу CPaaS.

В процесі розробки випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти необхідно виконати наступний обсяг роботи:

а) провести аналіз існуючих систем-аналогів для виявлення їх позитивних і негативних якостей. Результати аналізу врахувати в подальших розробках;

б) вибрати та обґрунтувати методику побудови системи контролю роботи технологічного обладнання на виробництві в автоматизованому режимі. Розробити функціональну та структурну схеми системи;

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

в) розробити програмне забезпечення системи, що дозволить реалізувати поставлену технічним завданням задачу. Побудувати блок-схеми алгоритмів програми та підпрограми;

г) організувати інтерфейс користувача з метою формування та виводу на екран ЕОМ повідомлень про некоректні дії користувача та нестандартні ситуації в роботі технологічного обладнання;

д) розробити рекомендації по організаційних та методичних заходах, які забезпечать впровадження системи в промислову експлуатацію та її подальшу успішну експлуатацію;

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

ж) розробити заходи по охороні праці при впровадженні та експлуатації системи, а також розробити заходи з цивільного захисту;

з) сформулювати висновки про виконаний обсяг робіт та одержані результати.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Даний момент характеризується розквітом контекстуальних комунікацій, тобто комунікацій, у яких обидві сторони відразу й повністю розуміють контекст спілкування. Звучить абстрактно, тому пари прикладів: вам дзвонять із незнайомого номера. Відповідно, ви не знаєте, хто дзвонить, звідки й з якою метою. І навпроти, якщо ви робите якусь операцію в додатку інтернет-банкінгу й на якомусь кроці щось пішло не так, ви можете прямо із цього кроку подзвонити в підтримку – контекст ситуації з ходу буде ясний як вам, так і операторові. Щоб забезпечувати таку поінформованість, бізнеси використовують комунікаційні платформи (CPaaS, Communications Platform as a Service), а ті, у свою чергу, використовують AI і Machine Learning.

Успіх CPaaS

CPaaS – це швидкоростучий бізнес. Чому? В успіху концепції CPaaS є кілька причин.

По-перше, розквіт CPaaS трапився багато в чому завдяки розквіту «нового ентєрпрайза» – коли свою життєздатність довели компанії начебто Uber і Lyft, усім раптом стало зрозуміло, що всі ці вчорашні стартапи використовують хмарні комунікаційні платформи. Коли ринок почав це розуміти, попит на CPaaS став рости, тому що хмарні рішення дозволяють збирати на своїй основі готові «коробкові рішення» дуже швидко, щоб починати заробляти гроші.

По-друге, треба пам'ятати, що CPaaS-платформи завжди були націлені на розроблювачів. А в кожного сучасного стартапа завжди є розроблювачі, для яких не становить праці використовувати CPaaS.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

По-третє, хмари – є хмари, що означає доступність до сервісу по усьому світі, масштабованість і збільшення потужностей по запиті. І все це без головного болю для того, хто використовує CPaaS.

І, нарешті, більшість платформ пропонують принцип оплати as-you-go, коли треба платити тільки за те, що використовуєш: є розпізнавання мови й переклад її в текст – ці функції біляться, а немає розпізнавання – ну, ви зрозуміли. Це досить гнучко й прозоро.

Нове в індустрії

Отут першою справою треба згадати про Serverless, що підняв зручність CPaaS на новий рівень. Один раз ми вже докладно писали на цю тему, зараз же обмежимося головною тезою: Serverless означає не відсутність серверів взагалі, а їхня відсутність на стороні клієнта. З погляду використовуваних обчислювальних ресурсів це такий же as-you-go, тому що плата стягується згідно навантаженню на computing provider'a. Інший важливий момент serverless – це те, що клієнтам можна давати доступ до райнтайму платформи, а це веде до зниження затримок і збільшенню надійності.

Інший тренд – WYSIWYG-редактори. Це один із кроків назустріч бізнес-аудиторії, що (найчастіше) не вміє кодити, але при цьому може зібрати логіку бота / колл-центра у візуальному редакторі. Підходи до реалізації небагато відрізняються (див. Smartcalls від Voximplant, Studio від Twilio, FlowBuilder від MessageBird і т.д.), але суть схожа – користувач використовує не код, а візуальні блоки, варіюючи їхнє розташування й зв'язки між ними. До речі, деякі такі редактори все-таки дозволяють використовувати код як просунута можливість, наприклад, наш Smartcalls, але вже небагато інша історія.

Нарешті, хмарні IDE. Звичайно, поки вони навряд чи можуть зрівнятися з умовної IDEA, але от з VS Code – легко. Якщо CPaaS дає розроблювачеві потужний інструмент для роботи з кодом, то такий розроблювач буде, швидше за все, дуже задоволений. Нормальний відладник, розумний автокомпліт, підсвічування коду, кастомні стилі, вкладки й т.д. – коли це є у веб-інтерфейсі й

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

швидко працюють, то платформа одержує додаткові окуляри в карму за свою гнучкість.

Але наша радість була б не повної якби не AI. Машинне навчання дає нові ступені волі комунікаційним платформам, а саме:

Розпізнавання

Розпізнавання й синтез мови – хтось розробляє їх самостійно, але це досить трудомістко. Можна звернутися за цим до великих гравців начебто Google, Amazon – їхньої моделі вже дуже добре розпізнають людську мову, так само як і імітують її (кивок убік WaveNet).

Автоматизація NLU/NLP

Natural Language Understanding (Processing) – обробка природної мови – зараз сама гаряча тема у світі комунікацій. І якщо бізнес-рішення опирається на NLU, те, як варіант, там відбувається синтез мови, потім людина щось відповідає, його мова транслітерується, цей текст віддається назад роботіві й він, щоб зреагувати, підбирає текст відповіді, які знов-таки треба синтезувати. Звучить не як rocket science, але все-таки тут розумно використовувати автоматизацію – Google Dialogflow, IBM Watson, Amazon Lex та ін.

Посилення операторів

Коли оператор колл-центра спілкується із клієнтом, можна фонові аналізувати мову й давати операторові додаткову інформацію, щоб він не витрачав свій час. Наприклад, клієнт може запитати, де найближчий банкомат – система розпізнає питання й виведе відповідь на екран оператора; останній просто зачитає відповідь, замість того щоб просити клієнта почекати.

Аналіз емоцій

У цьому зацікавлені приблизно всі, але цей самий важкий напрямок в CPaaS на даний момент, тому що люди схильні подавати ту саму інформацію по-різному, а також досить часто використовувати культурні відсилання в мові. Зараз багато компаній аналізують емоції, використовуючи текст. Зараз існують рішення в цьому напрямку, але не можна сказати, щоб вони були вдалими, тому

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

що на аналізі тільки лише тексту далеко не виїдеш; очевидно, що емоції – це не тільки ЩО саме сказано, але і ЯК. Тому переконливий аналіз емоцій у реальному часі – це питання (найближчого?) майбутнього.

Поліпшення аудіо/відео

Усі знають про noise reduction – коли ви говорите по телефоні, навчена модель «забирає» фонові шуми, щоб співрозмовник чув тільки вас. Іноді при цьому страждає голос самого мовця, тому що моделі не завжди можуть успішно розрізняти, які частоти ставляться до тла, а які – до голосу. Але в цілому це працює вже досить добре. Говорячи про картинку, ми знаємо, як сучасні смартфони роблять боці (розмивають тло) за допомогою AI. Такий підхід, але вже в рамках відеодзвінків теж буде затребуваний – представте, що вам не треба шукати ідеальне тло, тому що AI розміє будь-яке оточення за вашою спиною. Хоча чому «представте» – Skype уже має таку функціональність.

Аналіз відео

Аналіз відеопотоку або відеозаписів допомагає зрозуміти, що перебуває в кадрі. Поки що це досить ресурсномістке завдання, тому сьогодні з нею краще всіх справляються ті, у кого дуже багато обчислювальних потужностей – Google, Microsoft і інші великі гравці.

Аналітика дзвінків

Сюди ставиться не тільки класифікація й сегментація даних. Представте, що у вас є десятки тисяч записів дзвінків, і їх можна перевести в текст, а потім робити по ньому пошук. Але набагато ефективніше, якщо AI пройдеться по цих записах і розподілить їх на групи (це дзвінки про продажі, а це – гарантійні), виявить, де оператор колл-центра поведився коректно, а де – не дуже (плюс можна виявити, як саме поводитися людина, які були емоції), тут клієнт запитував тільки про покупку автомобіля, а тут – і про автомобіль, і про страховку, і про тест драйв. Можна вивудити скільки завгодно інформації з такого масиву даних за допомогою machine learning.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

Визначення автовідповідача

Особливий випадок, але теж являє гарний приклад: у своїй платформі ми реалізували визначення автовідповідача. Зараз платформа вміє розпізнавати автовідповідачі російською мовою – ми навчили модель на безлічі дзвінків, тепер вона вміє відрізняти живої людини від записаного повідомлення. Звичайні способи визначення не дуже ефективні (наприклад, по звуковому сигналі), але AI допоміг нам домогтися точності до 99%, при цьому на розпізнавання йде всього 2 секунди.

Труднощі

Машинне навчання вимагає багато ресурсів. І мова не тільки про обчислювальні потужності, але й про людей зі спеціальними навичками – data scientists, які створюють і налаштовують моделі навчання, а також знають, які дані потрібні. Таких людей непросто знайти і їхню працю дорого коштує. Також на них великий попит серед великих гравців, а конкурувати з умовним Google у плані наймання – це важко, хоча й можливо. Тому замість суперництва краще вибрати співробітництво з гігантами – більшість CPaaS-гравців використовують наробітки великих компаній, і це нормально. З іншого боку, це приводить до того, що гігант-партнер управляє витратами інших гравців – установлює/мінє розцінки на розпізнавання й синтез мови (згадуємо WaveNet від Google). Тобто, якщо ви користуєтеся рішеннями гіганта, а він раптом вирішує змінити розцінки, то ви змушені зробити те ж саме, що може не дуже обрадувати ваших користувачів. Додамо сюди те, що ви будете відправляти дані цьому гігантові – для деяких бізнесів це проблема. Однак завжди можна не залежати тільки від одного партнера, користуватися рішеннями декількох гігантів зі схожою функціональністю. Нарешті, таке співробітництво зручно й вигідно для CPaaS-гравців.

Грядуть нові технології, які вплинуть на комунікації так само, як у свій час вплинув WebRTC – це 5G і AV1.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

5G покликаний втілити в життя принцип «завжди онлайн» – це кінцева мета, але ясно, що це трапиться не в один день. Із приходом цієї технології в SPaaS з'явиться більше можливостей, тому що навіть ті, хто раніше не користувався мобільною передачею даних, почнуть це робити. Інфраструктура комунікацій зміниться, а разом з нею зміняться й звичні телекомунікаційні бізнеси.

Відеокодек AV1 теж буде корисний для SPaaS, тому що він безкоштовний, а значить не треба буде піклуватися про ліцензії. Безкоштовний кодек, що ефективніше чим H.265 і буде доступний усім, теж змінить мир комунікацій.

AV1 у деталях

Chroma from Luma

Chroma from Luma prediction (далі – CfL) – один з нових способів прогнозування, що застосували в AV1. CfL прогнозує кольору в зображенні (chroma), ґрунтуючись на значенні яскравості (luma). Спочатку кодуються/декодуються значення яскравості, потім CfL робить спробу пророчити кольору. Якщо спроба вдала, то знижується кількість інформації про колір, яку треба закодувати; отже, зберігається місце.

Варто відзначити, що CfL уперше з'явився не в AV1. Основний документ по CfL датується 2009 роком; тоді ж LG і Samsung запропонували ранню імплементацію CfL під ім'ям LM Mode, але все це згорнулося під час розробки HEVC/H.265. Кодек Thor від Cisco використовує схожу техніку, а HEVC імплементував поліпшену версію за назвою Cross-Channel Prediction (CCP).

Поліпшене внутрикадрове прогнозування

До останнього часу стиск відео ґрунтувалося на межкадровому прогнозуванні, тобто на відмінності кадру від інших, коли в основі прогнозування лежать опорні кадри. Незважаючи на те, що ця техніка могутньо розвилася, вона усе ще вимагає опорні кадри, які не покладаються на інші кадри. У підсумку опорні кадри використовують тільки внутрикадрове прогнозування.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Перші 60 кадрів тестового відео. Гістограма починається з опорного кадру в ~20 разів більше інших.

Опорні кадри набагато більше, ніж проміжні – тому їх намагаються використовувати як можна рідше. Але якщо опорних кадрів стає багато, це підвищує бітрейт відео. Щоб упоратися із цим і зменшити розмір опорних кадрів, дослідники кодеків зосередилися на поліпшенні внутрикадрового прогнозування (яке так само можна застосовувати й до проміжних кадрів).

Резюмуючи, можна затверджувати, що CfL – це саме і є просунута техніка внутрикадрового прогнозування, тому що вона працює на основі яскравості в межах кадру.

Кольорові крейди

CfL у своїй основі – це розфарбовування монохромного зображення виходячи з обґрунтованого, точного пророкування. Пророкування полегшується тим, що зображення б'ється на дрібні блоки, у яких кодування відбувається незалежно.

Розбивка на блоки, щоб максимізувати точність кодування.

Тому що кодер працює не з усім зображенням, а з його шматочками, досить виявити кореляції в невеликих областях – цього вистачить, щоб прогнозувати кольору на основі яскравості. Візьмемо маленький блок зображення:

На основі цього фрагмента, кодер установить, що яскраво = зеленому кольору, а чим темніше, тим менше насиченість. І так само з іншими блоками.

CfL в AV1

В CfL не стали використовувати алгоритм PVQ, тому витрати при піксельній і частотній областях приблизно однакові. Плюс AV1 використовує дискретне синусне перетворення й перетворення піксельної області (pixel domain identity transform), тому виконати AV1 CfL у частотній області не дуже легко. Але – сюрприз – AV1 і не має потреби в CfL у частотній області, тому що основні рівняння CfL однаково працюють в обох областях.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

CfL в AV1 покликаний максимально спростити реконструкцію. Для цього необхідно явно кодувати α і на його основі обчислити β , хоча... Можна й не обчислювати β , а замість нього використовувати вже спрогнозоване кодером DC-зсув кольоровості (воно буде менш точним, але все-таки підходящим):

Порівняння DC-прогнозу за замовчуванням (розрахунок на основі сусідніх пікселів) з розрахованим β -значенням (розрахунок на основі пікселів у поточному блоці).

Таким чином, складність апроксимації на стороні кодера макисмально оптимізується за рахунок використання прогнозування. Якщо прогнозування недостатньо, то виконуються інші перетворення; якщо ж прогнозування не дає вигоди в бітах, то воно не використовується зовсім.

3.2 Розробка структурної схеми

У четвертій промисловій революції всенаціональне спілкування опиниться в основі успішних бізнес-операцій. Це робить важливим для підприємств різних галузей інвестування в комунікаційну платформу як послугу (CPaaS).

Сьогодні майже кожен має смарт-мобільний пристрій, який забезпечує безперебійний зв'язок через декілька каналів. Однак бізнесу може бути важко йти в ногу без високо пристосованого комунікаційного рішення.

Наприклад, клієнти можуть витягнути свій смартфон і надіслати запит на отримання додаткової інформації або подати скаргу протягом декількох секунд, але компаніям буде доручено знайти інноваційні способи адаптувати свої відповіді, зберігаючи їхні повідомлення.

Перш ніж ми підемо далі, давайте спершу визначимо CPaaS. В основному, CPaaS можна визначити як злиття програмованих хмарних і телекомунікаційних технологій, які забезпечують підключення під ключ інструментів зв'язку, таких як SMS, голос та відео, які не вимагають власної фізичної інфраструктури.

Насправді CPaaS – це все та багато іншого, тому що це єдине вікно для

всіх ваших потреб у спілкуванні. Тому ми також повинні додати штучний інтелект, спілкування в чаті, обмін повідомленнями, розпізнавання мови та віртуальної реальності.

CPaaS є прискорювачем для бізнесу. Це полегшує час виходу на ринок для бізнесу, дозволяючи підприємствам просто «підключати і грати» рішення, пропускаючи всі види операційних головних болів, бюрократії, зв'язків з операторами та ін. Це суттєво зменшує перешкоди для входу в місцевість зв'язку. Це дуже цінно для бізнесу з кількома регіональними офісами.



Рисунок 3.1 – Структурна схема системи

Отже, які основні характеристики визначають CPaaS? Погляньмо.

1. Традиційний стандарт мобільного телефону та Інтернет-комунікація

Хмарні комунікаційні платформи об'єднують функціональність як традиційних телефонних мереж, так і сучасних систем зв'язку IP.

Підприємства можуть отримати вигоду від того, щоб усі свої канали комунікації мали на одній платформі, не залучаючи телекомунікаційну групу чи

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

купуючи необхідне обладнання для доповнення для її досягнення.

До появи CPaaS комунікації в режимі реального часу проходили лише в системах, побудованих з певною метою. Тепер ви можете залучати своїх клієнтів не лише за допомогою традиційного телефонного дзвінка, але й безперешкодно переходити до відео чи спілкуватися в чаті (якщо цього вимагає ситуація).

2. Безшовна інтеграція

Провідні рішення CPaaS оснащені інтерфейсами прикладного програмування (API) та наборами програмного забезпечення (SDK). Тож розробники можуть легко інтегрувати свої системи з хмарною платформою та швидко розгортати готові продукти для корпоративного класу.

Коли підприємства інтегрують засоби комунікації у свої застосунки, це заперечує необхідність переходу клієнтів на традиційну (PSTN) систему телефонії, щоб викликати службу підтримки для підтримки. Цей підхід надає абсолютно новий сенс уніфікованим комунікаціям.

3. Забезпечує регіональний \ глобальний доступ

Коли бренди співпрацюють з відомим постачальником CPaaS, вони можуть скористатися негайним доступом до сотень операторів по всій планеті. Для підприємств CPaaS може допомогти їм збільшити регіональний та глобальний обсяг, не потребуючи єдиних переговорів з телекомунікаційним оператором. Оскільки комунікаційні платформи на основі хмар можуть бути налаштовані під ваші конкретні потреби, підприємства можуть застосовувати орієнтований на клієнтів підхід, який підходить для кожної конкретної аудиторії.

Ви також можете зробити це на крок далі та взаємодіяти з кожним окремим споживачем, виходячи з його конкретних переваг у спілкуванні (враховуючи, коли, де та як кожна людина любить займатися).

Наприклад, якщо клієнт вважає за краще зв'язатись з ним через застосунок для чату ввечері через телефонний дзвінок або електронну пошту, ви зможете залучати цього конкретного клієнта відповідно до його переваг через ту саму платформу CPaaS.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

4. Вмикає звітування в реальному часі та Analytics

Платформи CPaaS можуть похвалитися аналітикою в реальному часі та інструментами динамічної звітності, які можна використовувати для визначення стану доставки SMS та даних використання дзвінків. До всієї цієї інформації можна легко дістатися в одному місці через дуже зручну інформаційну панель.

Хмарні комунікаційні платформи допомагають підняти бізнес-аналітику на новий рівень. Завдяки потужності аналітики даних тепер ви можете переглядати дані в реальному часі, одночасно аналізувати історичні звіти та прогнозування користувачів, щоб приймати «інтелектуальні» рішення, керовані даними.

5. Підтримує кілька мов програмування

Провідні платформи CPaaS сьогодні підтримують різноманітні мови програмування. Як результат, ваші внутрішні розробники можуть розпочати налаштування систем комунікацій підприємств за кодом, який вони вже знають.

Деякі мови програмування, що підтримуються хмарними платформами зв'язку, включають наступне:

- Java.
- .NET.
- PHP.
- Python.

Оскільки CPaaS також підтримуватиме модель перетягування "зроби сам" ("Зроби сам"), ви можете налаштувати взаємодію зі своєю цільовою аудиторією, не записуючи жодного рядка коду. Цей підхід виявляється ідеальним рішенням щодо дефіциту талантів вищих технологій у регіоні MEA.

6. Модель виставлення рахунків з оплатою

Коли підприємства включають модель CPaaS, їм не потрібно вкладати значних інвестицій у всю інфраструктуру, що вимагається традиційними комунікаційними рішеннями. Насправді, частіше за все, потрібен нульовий авансовий капітал, щоб перейти на смугу CPaaS.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Коли компанії підписуються на модель оплати за виплату, вони зможуть налаштувати свої комунікаційні рішення та визначити інноваційні способи взаємодії з новими та існуючими клієнтами.

7. Масштабованість та надійність

Хмарні комунікаційні рішення відрізняються високою безпекою, масштабуванням та спритністю. Ви також можете масштабувати вгору або вниз залежно від сучасних вимог і платити лише за те, що використовуєте.

Підприємства, які приймають рішення CPaaS, також добре оснащені для безперебійного масштабування від прототипу до світових виробничих потужностей. Оскільки все це буде підкріплено хмарою, підприємства також будуть сприятливі для покращення безперервності бізнесу.

У епоху цифрових технологій CPaaS буде основною силою, яка допоможе просунути бізнес вперед. Її функція "зроби сам" підключає та грає також полегшує компаніям користь від покращеного спілкування без усіх головних болів в роботі, які притаманні традиційним системам зв'язку.

Це дуже спритне та масштабоване рішення не вимагає інвестицій, тому бізнес може скористатися інноваційними системами бізнес-аналітики, які допоможуть просунути бізнес вперед.

3.3 Розробка функціональної схеми

В процесі практичної реалізації теоретичних принципів розробки системи, додатково розглянутих вище, була розроблена функціональна схема системи, яка зображена на рисунку 3.2.

Завдяки функціональній схемі можна чітко побачити основні функціональні блоки системи та взаємозв'язки між ними. При розробці функціональної схеми основний упор робився на існуючі розробки ПЗ і їх модулі допомоги. Аналіз рисунка 3.2 дозволяє чітко прослідити як працює програма. Розглянемо схему зверху вниз, в напрямку від пристрою до кінцевої програми –

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

за допомогою внутрішнього пула доступу що забезпечує закриті канали зв'язку абоненти генеральний директор, відділи бухгалтерії, логістики, складу, збуту, економічної безпеки – можуть взаємодіяти з дочірніми виробниками з використанням внутрішньої АТМ реалізованої у вигляді системи комунікаційної платформи як сервісу CPaaS. і чітко налагодженої технічної системи взаємодії. Технічна частина взаємодії забезпечується VoIP та гейткіпером з використанням маршрутизатора.

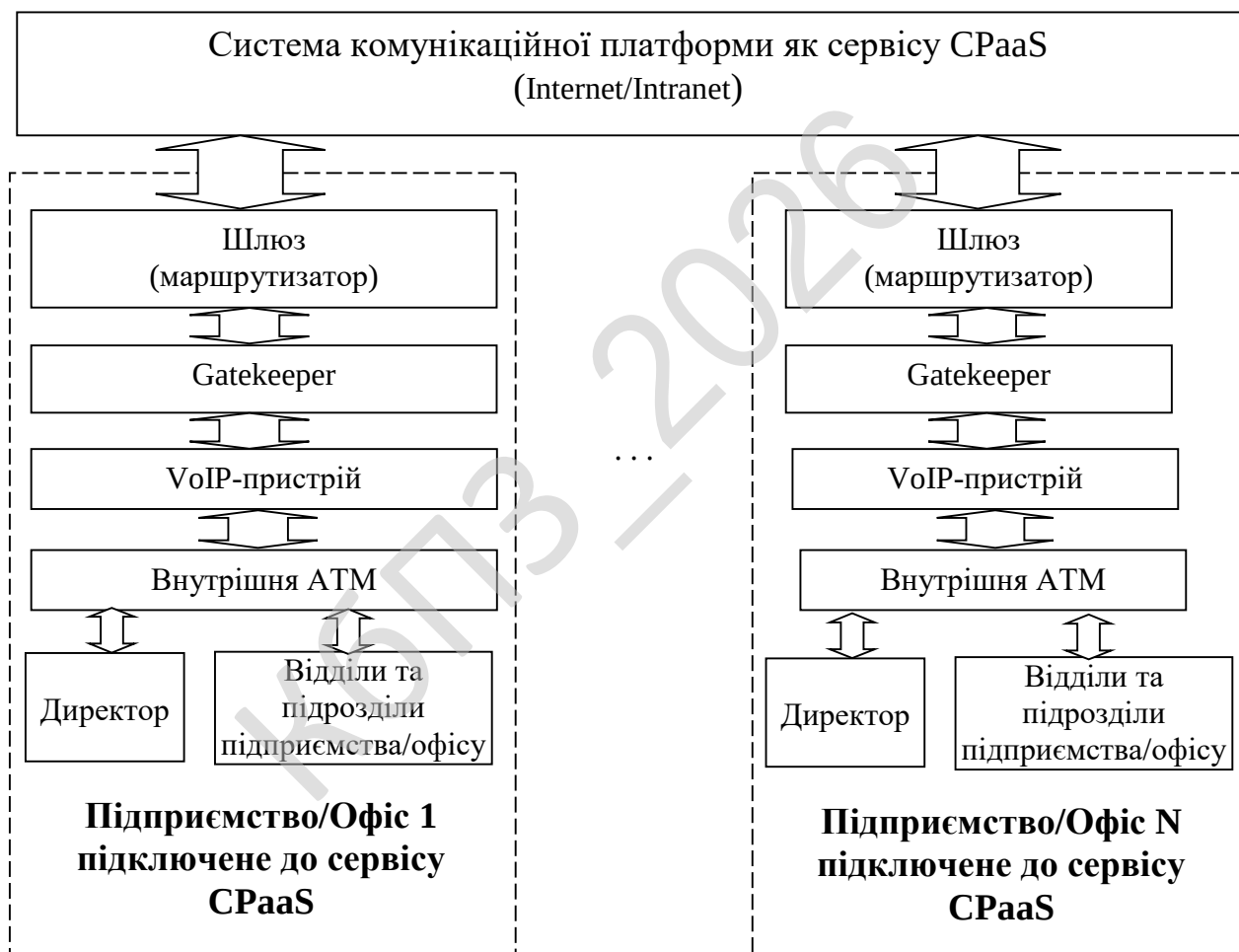


Рисунок 3.2 – Функціональна схема системи

Gatekeeper

Контролер зони Н.323 або гейткіпер Н.323 або воротар (англ. Н.323 Gatekeeper) – пристрій або програма в мережі ІР-телефонії, за стандартом Н.323, що виступає як центр обробки викликів усередині своєї зони й виконує найважливіші функції керування викликами. Зона в мережі Н.323 визначається як сукупність всіх терміналів, шлюзів і пристроїв керування конференціями (MCU) під керуванням даного контролера зони.

Контролер зони – необов'язковий компонент мережі Н.323, однак, якщо він присутній у мережі, то термінали й шлюзи повинні використовувати його послуги.

Різні ділянки зони мережі Н.323 можуть бути територіально рознесені й з'єднуватися один з одним через маршрутизатори.

Основним протоколом, що використовує контролер зони є Н.225.

Основні функції:

- Трансляція адрес – Перетворення внутрішніх адрес ЛОМ і телефонних номерів формату Е.164 в адреси протоколів ІР/ІРХ.
- Керування доступом – Авторизація доступу в Н.323 мережі для користувачів за допомогою RAS.
- Керування смугою пропускання – Дозвіл або заборона запитуваної терміналом смуги пропускання.
- Маршрутизація сигнальних повідомлень між терміналами, розташованими в одній зоні; воротар може організовувати сигнальний канал безпосередньо між терміналами або ретранслювати сигнальні повідомлення від одного терміналу до іншого.

Додаткові функції:

- Керування процесом устанавлення з'єднання – При двосторонній конференції воротар здатний обробляти службові повідомлення протоколу сигналізації Q.931, а також може служити ретранслятором таких повідомлень від кінцевих точок

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

– Авторизація з'єднання – Допускається відхилення воратарем запиту на встановлення з'єднання. Підстави – обмеження прав або часу доступу, і інші, лежачі поза рамками Н.323

– Керування викликами – Воратар може відслідковувати стан всіх активних з'єднань, що дозволяє управляти викликами, забезпечуючи виділення необхідної смуги пропускання й баланс завантаження мережних ресурсів за рахунок переадресації викликів на інші термінали й шлюзи

VoIP

VoIP (Voice over Internet Protocol) або IP-телефонія – це технологія, що забезпечує передачу голосу в мережах з пакетною комутацією за протоколом IP, часткою случаємо яким є мережі Інтернет, а також інші IP-мережі (наприклад, виділені цифрові канали). Для зв'язку мережі Інтернет (IP-мережі) з телефонною мережею загального користування PSTN (span>Public Switched Telephone Network), що ставиться до глобальних мереж з комутацією каналів, використовуються спеціальні аналогові VoIP-шлюзи.

Необхідно відзначити, що мережі Інтернет через цифрові шлюзи VoIP пов'язані із цифровими телефонними мережами ISDN (Integrated Services Digital Network). Крім того, інтеграція VoIP у мережі стільникового зв'язку є практично неминучим процесом, інтеграція забезпечить більше низьку в порівнянні із традиційною стільниковою телефонією вартість розмов.

На сьогоднішній день доступ в Інтернет можливий безпосередньо з мобільних телефонів, які підтримують технології: CSD (Circuit Switched Data або GSM Data), GPRS (General Packet Radio Service), EDGE (Enhanced Data rates for Global Evolution), CDMA (Code Division Multiple Access), EV-DO (Evolution-Data Optimized), які забезпечують широкий спектр послуг "Мобільний Інтернет" і WAP. Необхідно відзначити, що в мобільному зв'язку вже впроваджуються нові технології бездротового широкополосного доступу в Інтернет на базі технології зв'язку 4G (GSM/WiMAX/ Wi-Fi mobile phone).

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

У цей час до мереж PSTN і ISDN підключені центри комутації стільникового зв'язка (стільникові мережі різних операторів з'єднані між собою), що забезпечує дзвінки зі стільникових телефонів на стаціонарні телефони (PSTN або ISDN) і навпаки. Характерним для мереж 3G мобільного зв'язку є швидкісна бездротова передача даних і потужні магістральні мережі пакетної комутації. У зв'язку з тим, що мережі PSTN пов'язані з мережами Інтернет і мережами стільникового зв'язка, може бути забезпечена передача голосових сигналів між цими мережами.

Голосовий сигнал з каналу VoIP може безпосередньо надходити на IP-телефон, підключений до IP-мережі або маршрутизуватися на мобільний телефон мобільного оператора, або на аналоговий телефон, підключений до звичайної телефонної мережі PSTN, або на цифровий телефонний апарат, підключений до цифрової мережі з інтеграцією послуг ISDN.

Таким чином, IP-телефонія забезпечує передачу голосових сигналів з комп'ютера на комп'ютер, з комп'ютера на телефон (аналоговий телефон, цифровий телефон, IP-телефон, мобільний телефон) і з телефону на телефон. Дзвінки здійснюються через провайдеру послуг VOIP. Якість передачі голосу залежить від VoIP-провайдеру й способу підключення до Інтернету.

Одне з переваг IP-телефонії – це економія фінансових засобів на ведення міжнародних і міжміських телефонних переговорів за рахунок того, що значну частину відстані між абонентами голосовий сигнал у цифровому виді (у стислому стані) проходить по мережах пакетної комутації (по мережі Інтернет), а не по телефонних мережах з комутацією каналів. У цей час IP-телефонія забезпечує найдешевші або безкоштовні міжміські й міжнародні дзвінки необхідно тільки оплатити використаний трафік Інтернет-провайдеру.

Висока вартість передачі голосу в мережах PSTN пояснюється тим, що ці мережі мають низький коефіцієнт використання каналів, що комутируються. Комутація каналів має на увазі утворення безперервного складеного фізичного каналу з послідовно з'єднаних окремих каналних ділянок для прямої передачі

голосу між АТМ на час ведення переговорів двох абонентів. Під час пауз у процесі розмови складений фізичний канал не несе ніякого корисного навантаження, але ці паузи оплачуються абонентами.

Мережі з пакетною комутацією ефективно використовують мережу, тому що пакети передаються по поділюваному середовищу (загальному для всіх розмов каналу передачі даних). Комутація пакетів – це комутація повідомлень, що представляються у вигляді адресуємих пакетів, коли канал передачі даних зайнятий тільки під час передачі пакета й по її завершенню звільняється для передачі інших пакетів. Таким чином, паузи в ІР-мережах не оплачуються, тому передача голосу по ІР-мережах дешевше, ніж по мережах PSTN.

Ведення міжнародних і міжміських телефонних переговорів значно заощаджують фінансові засоби, як приватних осіб, так і компаній. Але основна перевага технології VoIP для компаній – це створення систем корпоративної або офісної ІР-телефонії з малими фінансовими витратами, але з більшою кількістю сервісних функцій VoIP. Системи офісної ІР-телефонії називаються ІР PBX або ІР АТМ, або Soft PBX, або програмні АТМ, або VoIP міні-АТМ, які є АТМ для установ – УАТМ, тобто телефонною системою для приватного користування.

Розглянемо системний взаємозв'язок програмних модулів і інформаційних файлів. Технологія внутримашинної організації задається послідовністю реалізованих процедур – схем взаємозв'язку програмних модулів і інформаційних масивів. Така схема являє собою декомпозицію загального процесу рішення задачі на окремі процедури перетворення масивів, іменованими модулями (це – виклик ІР-адресату, завершення сеансу зв'язку, редагування легенди ІР-номерів, редагування бази даних ІР-адрес, моніторинг бази даних ІР-адрес та легенд або організація селективного зв'язку і т.і.).

Основне призначення створюваної корпоративної мережі зв'язку на основі ІР-технології – це автоматизація процесу керування. Отже, структуру програм можна описати наступними основними блоками.

Робота з програмою починається з введення інформаційного вікна й

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

активізації системи меню. Робота програми здійснюється по діалоговому і подійному режиму, при цьому по діалогом розуміється надання користувачу декількох альтернатив і обробка його вибору. У діалогову систему входять головне меню з відповідними спливаючими підменю а також діалогові вікна. Під подіями розуміються процеси, що активізуються користувачем (наприклад – натискання функціональних клавіш), а також програмні події – одержання з’єднання з абонентом або закінчення з’єднання з абонентом. На підставі даних подій активізуються процедури контролю допустимості даних.

Модуль “Головне меню” призначений для запуску основних процедур програми і завершення роботи з програмою.

Модуль роботи з довідниками містить у собі два довідники:

- Довідник – Довідкова система.
- Довідник – Інструкція користувача.

Модуль роботи з базою даних абонентів корпоративної мережі включає в себе наступні підмодулі:

- Редагування легенди IP-номерів.
- Редагування бази даних IP-адрес.
- Моніторинг бази даних IP-адрес та легенд.

Призначення даного модуля є пошук і перегляд інформації з телефонних даних адрес абонентів корпорації, а також їх легенд.

Інформаційною базою даного модуля є таблиці: Значення IP-номерів та Легенда IP-номерів. Дані в інформаційну базу заносяться за допомогою спеціальних форм, що викликаються з головного меню програми.

Модуль «Формування вхідної інформації» призначений для введення первинних даних і перегляду раніше занесених. Даний модуль реалізує задачі обліку телефонних номерів, забезпечуючи введення номерів та їх легенд, ранжування за важливістю та їх знищенню.

У комп’ютерних системах користувачі для введення, перегляду та редагування інформації бази даних (IP-адрес та відповідних легенд) можуть

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

застосовувати форми. Основні переваги використання форм наступні:

- При введенні даних у поля-форми, додаток може зчитувати словник даних сервера й автоматично перевірити допустимість даних відповідно до правил цілісності.
- Поле введення у формі може представляти список допустимих значень, з яких користувачі можуть легко вибрати потрібне.
- Область форми може виводити шаблон, що відповідає поточної виведеної у формі запису.
- Командні кнопки у формі можуть виконувати дії, зв'язані з виведеної у формі поточною записом.

3.4 Розробка діаграми процесів

Розглянемо розроблену діаграму процесів яка зображена на рисунку 3.3. Основна будова діаграми процесів полягає у графічному представленні складу сукупностей даних, що характеризуються як співвідношення різних частин кожної з сукупностей. Склад статистичної сукупності графічно може бути представлений як за допомогою абсолютних, так і відносних показників. Графічне зображення складу сукупності по абсолютними і відносними показниками сприяє проведенню більш глибокого аналізу і дозволяє проводити аналіз системи.

Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи.

Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі. Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі.

Діаграми потоків даних містять чотири типи елементів:

- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.
- Процеси які являють собою трансформацію даних в рамках описуваної системи.
- Сховища даних (репозиторії).



Рисунок 3.3 – Діаграма взаємодії процесів

Таким чином, розглянувши опис системи, структурну, функціональну схеми системи, та діаграму взаємодії процесів перейдемо до опису блок-схем основної програми, та підпрограм, які використовуються, для реалізації системи.

4 РЕАЛІЗАЦІЯ ПРОЕКТУ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ПРАВИЛЬНІСТЬ ПРОЕКТНИХ РІШЕНЬ

4.1 Блок-схеми та опис алгоритмів функціонування системи

Розглянемо реалізацію бакалаврської дипломної роботи. Були проведені розрахунки і підібрані набори тестових даних для перевірки правильності реалізації проектних рішень.

Було створено блок-схеми роботи системи. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується. Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо.

Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

У інформаційних технологіях функціональна схема складається з функціональних блоків, які являють собою конструктивно відособлені частини (елементи або пристрої) автоматичних систем, які виконують певні функції. Функціональні блоки на схемі позначають прямокутниками, всередині яких надписують їх найменування відповідно до функцій, що виконуються. Зв'язки між функціональними блоками (внутрішні впливи) позначаються лініями зі стрілками, які вказують напрям впливів.

Функціональні схеми можуть виконуватися в укрупненому і розгорненому вигляді. У першому випадку на схемі зображають найважливіші блоки системи і зв'язки між ними.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

У другому варіанті схема відображається більш детально, що полегшує її читання та ілюструє принцип роботи.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач. Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції. Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента.

Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується.

Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи. Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані. Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи). З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми.

Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

Блок-схеми показують весь процес роботи системи з підсистемами та частково доказують правильність вибраних проектних рішень. Тому від точності і детальної блок-схеми залежить результат всієї програми.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає високого рівня декомпозиції задач на класи.

На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

З яких видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підсистеми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ. При роботі підпрограми виконується основний функціонал системи з циклічними послідовностями, перевіркою поточного стану та поверненням в основну програму прапорів стану виконання.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення.

UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем.

UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		45

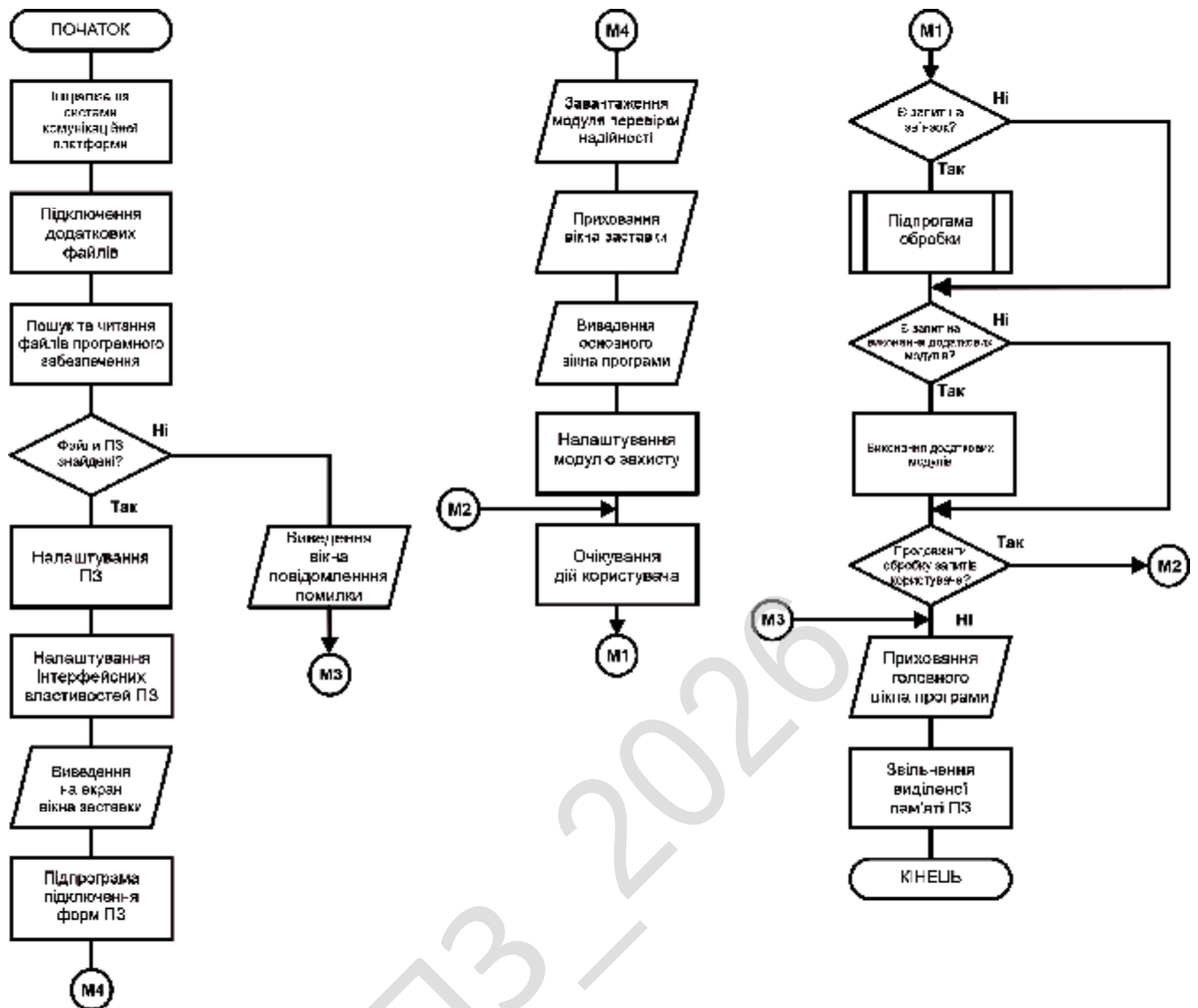


Рисунок 4.1 – Блок-схема основної програми

Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу). Діаграма діяльності. Це візуальне представлення графу діяльності. Граф діяльності є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання

діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

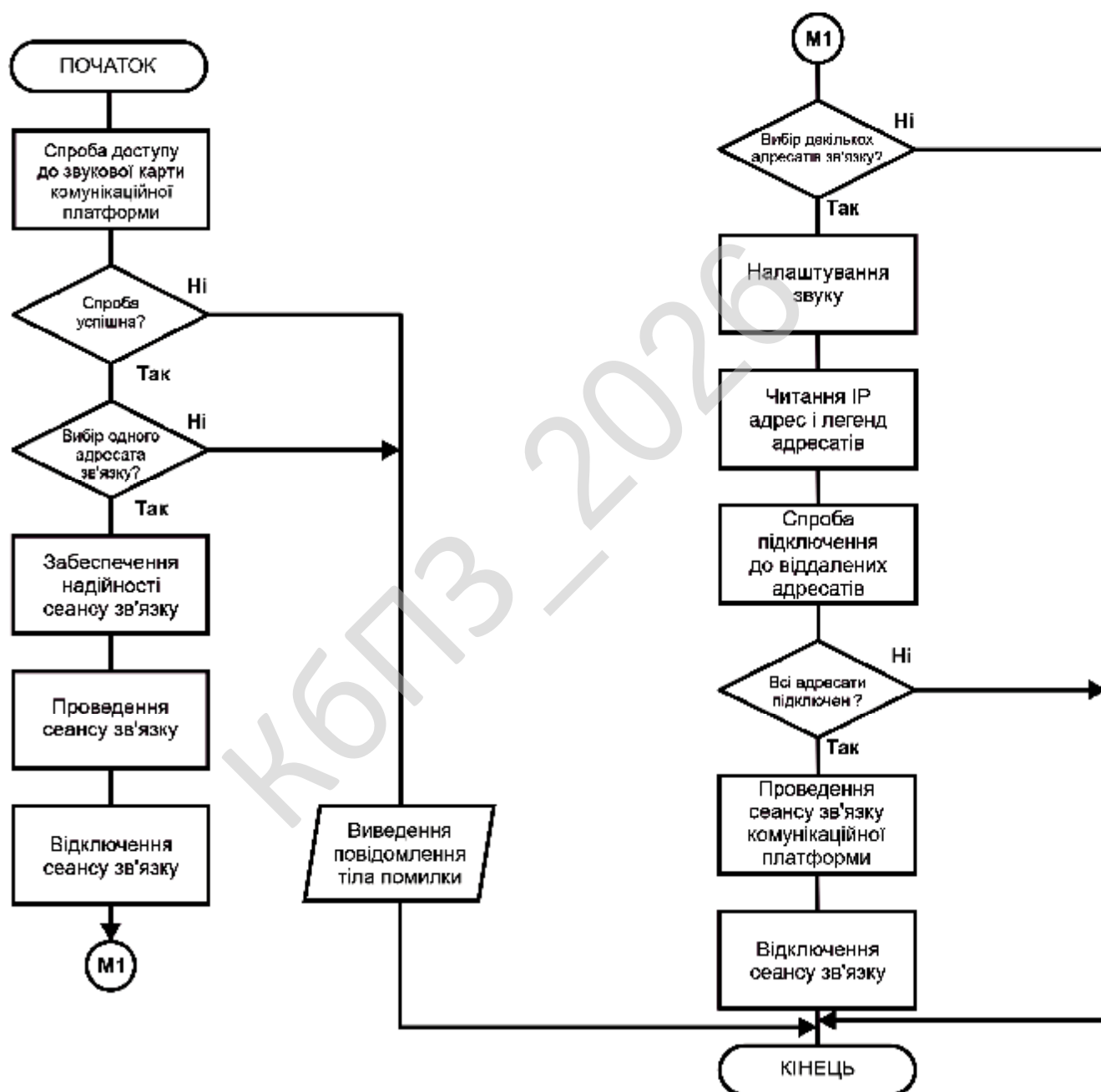


Рисунок 4.2 – Блок-схема роботи підпрограми

Специфікація діяльності (на вищих рівнях сумісності) може дозволяти виконання декількох (логічних) потоків, та існування механізмів синхронізації для гарантування виконання дій у правильному порядку.

Система керування базами даних (СКБД, Database Management System, DBMS) – набір взаємопов'язаних даних (база даних) і програм для доступу до цих даних. Надає можливості створення, збереження, оновлення та пошуку інформації в базах даних з контролем доступу до даних.

Першим поколінням СКБД прийнято вважати ієрархічні й мережеві системи. Ці системи отримали широке поширення в 1970-х роках, а першою комерційною системою цього типу була система IMS компанії IBM.

У 1980-х роках ці системи були витіснені системами другого покоління – повсюдно використовуваними і донині реляційними СКБД. У цих системах використовувалися непроцедурні мови управління даними (SQL) і передбачався значний ступінь незалежності даних. Реляційні системи внесли значні удосконалення в управління даними: графічний користувацький інтерфейс (GUI), клієнт-серверні застосунки, розподілені бази даних, паралельний пошук даних та інтелектуальний аналіз даних.

Але вже до кінця 1980-х років існуюча тоді реляційна модель перестала задовольняти розробників в силу низки обмежень. Відповіддю на зростаючу складність програм баз даних стали два нових напрямки розвитку СКБД: об'єктно-орієнтовані СКБД і об'єктно-реляційні СКБД.

У 1991 був утворений консорціум ODMG, основною метою якого стало вироблення промислового стандарту об'єктно-орієнтованих баз даних. Між 1993 та 2001 роками ODMG опублікувала п'ять ревізій своїх специфікацій. Остання версія стандарту має індекс 3.0, після чого група розпустилася. До кінця 1990-х існувало близько десяти компаній, що виробляли комерційні продукти, що позиціонуються на ринку як ООСКБД. Найбільш відомими системами даного класу стали Objectivity, Versant виробництва однойменних компаній, а також СКБД Jasmine, випущена компанією CA. Незважаючи на переваги, що дозволяють

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

ефективніше вирішувати певний ряд завдань, об'єктно-орієнтовані системи так і не змогли завоювати значущу частку ринку СКБД, залишившись «нішевим» продуктом.

Постачальниками традиційних реляційних СКБД також була проведена значна робота з об'єднання об'єктно-орієнтованих і реляційних систем. Розробники постаралися розширити мову SQL, щоб включити в неї концепції об'єктно-орієнтованого підходу, зберігаючи переваги реляційної моделі (об'єктні розширення мови SQL були зафіксовані в стандарті SQL:1999).

Основний принцип – це еволюційний розвиток можливостей СКБД без поломки попередніх підходів та зі збереженням наступності з системами попереднього покоління.

Поняття СКБД третього покоління, якими, власне кажучи, і є об'єктно-реляційні СКБД, з'явилося після опублікування групою відомих фахівців в області баз даних «Маніфесту систем баз даних третього покоління». Основні принципи СКБД третього покоління, позначені в маніфесті:

Крім традиційних послуг з управління даними, СКБД третього покоління повинні забезпечити підтримку розвиненіших структур об'єктів і правил. Розвинутіша структура об'єктів характеризує засоби, необхідні для зберігання і маніпулювання нетрадиційними елементами даних (тексти, просторові дані, мультимедіа).

СКБД третього покоління повинні включити в себе СКБД другого покоління. Системи другого покоління внесли вирішальний вклад у двох областях – непроцедурний доступ за допомогою мови запитів SQL і незалежність даних. Ці досягнення обов'язково повинні враховуватися в системах третього покоління.

СКБД третього покоління повинні бути відкриті для інших підсистем. Це включає оснащення різноманітними інструментами підтримки прийняття рішень, доступом з багатьох мов програмування, інтерфейсами до існуючих популярних систем і бізнес-застосунків, можливістю запуску програм з бази даних на іншій

машині і розподілені СКБД. Весь набір інструментів і СКБД має ефективно функціонувати на різноманітних апаратних платформах з різними операційними системами.

Крім того, СКБД, що розраховує на широку сферу застосування, повинна бути оснащена мовою четвертого покоління (4GL).

У середині 1990 років було лише кілька дослідних прототипів СКБД, які поєднали найкращі риси реляційних і об'єктно-орієнтованих СКБД. Першим комерційним продуктом, якому були властиві об'єктно-реляційні риси, став Universal Server компанії Informix(згодом була поглинена IBM). В даний час більшість цих ідей вже втілено в реальних комерційних рішеннях, в тому числі і в продуктах основних постачальників СКБД (Oracle Database і IBM DB2).

Розвиток індустрії систем керування базами даних базується на значних фундаментальних наукових дослідженнях. Найчастіше, між самими дослідженнями та їхньою конкретною реалізацією в прикладних рішеннях минають роки, а іноді й десятиліття. Роботу в області управління даними проводять як університетські дослідницькі групи (MIT, Berkeley), так і центри розробок основних постачальників СКБД (Oracle, IBM, Microsoft). Інвестування в управління даними – це довгострокове, і разом з тим, вигідне вкладення коштів. В даний час дослідники мають у своєму розпорядженні засоби, що дозволяють ефективно реалізувати найскладніші запити, що маніпулюють терабайтами й петабайтами різних даних.

Основними тенденціями, які дали привід для проведення різних масштабних досліджень в області баз даних стали:

Експонентний ріст даних. Обсяг даних, у тому числі синтетичних, що генеруються автоматизованими системами, значно зріс. Збільшилося і число прикладних областей, в яких вимагається обробка великих обсягів даних. До таких областей тепер відносяться не тільки традиційні корпоративні програми, пошук у веб, але також і наукові дослідження, обробка природних мов, аналіз соціальних мереж тощо.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Значне ускладнення структур використовуваних даних. Прості види даних у вигляді чисел і символьних рядків стали доповнятися численною мультимедійною інформацією, просторовими, процедурними даними та великою кількістю інших складних форматів.

Широке поширення дешевих високопродуктивних апаратних засобів. Щорічно ми спостерігаємо зростання обчислювальних можливостей мікропроцесорів, збільшення ємності і зниження вартості доступних і зручних в експлуатації пристроїв дискової оперативної пам'яті.

Активний розвиток засобів комунікації та «всесвітньої павутини» World Wide Web. WWW стає єдиним інформаційним середовищем, що пронизує весь світ і об'єднує величезне число користувачів та електронних пристроїв.

Поява нових важливих областей застосування СКБД. У першу чергу, це пов'язано з інтелектуальним аналізом даних, сховищами даних, а останнім часом – з паралельними обчисленнями і хмарними технологіями.

Основні характеристики СКБД

1. Контроль за надлишковістю даних.
2. Несуперечливість даних.
3. Підтримка цілісності бази даних (коректність та несуперечливість).
4. Цілісність описується за допомогою обмежень.
5. Незалежність прикладних програм від даних.
6. Спільне використання даних.
7. Підвищений рівень безпеки.

Можливості СКБД

1. Дозволяється створювати БД (здійснюється за допомогою мови визначення даних DDL (Data Definition Language)).

2. Дозволяється додавання, оновлення, видалення та читання інформації з БД (за допомогою мови маніпулювання даними DML, яку часто називають мовою запитів).

3. Можна надавати контрольований доступ до БД за допомогою: Системи забезпечення захисту, яка запобігає несанкціонованому доступу до БД; Системи керування паралельною роботою прикладних програм, яка контролює процеси спільного доступу до БД; Система відновлення – дозволяє відновлювати БД до попереднього несуперечливого стану, що був порушений в результаті збою апаратного або програмного забезпечення.

Основні компоненти середовища СКБД

1. Апаратне забезпечення.
2. Програмне забезпечення.
3. Дані.
4. Процедури – інструкції та правила, які повинні враховуватись при проектуванні та використанні БД.
5. Користувачі: адміністратори даних (керування даними, проектування БД, розробка алгоритмів, процедур) та БД (фізичне проектування, відповідальність за безпеку та цілісність даних); розробники БД; прикладні програмісти; кінцеві користувачі.

Архітектура СКБД

Існує трирівнева система організації СКБД ANSI-SPARC, при якій існує незалежний рівень для ізоляції програми від особливостей представлення даних на нижчому рівні.

Рівні:

1. Зовнішній – представлення БД з точки зору користувача.
2. Концептуальний – узагальнене представлення БД, описує які дані зберігаються в БД і зв'язки між ними. Підтримує зовнішні представлення, підтримується внутрішнім рівнем.
3. Внутрішній – фізичне представлення БД в комп'ютері.

Логічна незалежність – повна захищеність зовнішніх моделей від змін, що вносяться в концептуальну модель.

Фізична незалежність – захищеність концептуальної моделі від змін, які вносяться у внутрішню модель.

Було використано MongoDB – документо-орієнтована система керування базами даних (СКБД) з відкритим вихідним кодом, яка не потребує опису схеми таблиць.

MongoDB займає нішу між швидкими і масштабованими системами, що оперують даними у форматі ключ/значення, і реляційними СКБД, функціональними і зручними у формуванні запитів.

Код MongoDB написаний на мові C++ і поширюється в рамках ліцензії AGPLv3.

MongoDB підтримує зберігання документів в JSON-подібному форматі, має досить гнучку мову для формування запитів, може створювати індекси для різних збережених атрибутів, ефективно забезпечує зберігання великих бінарних об'єктів, підтримує журналювання операцій зі зміни і додавання даних в БД, може працювати відповідно до парадигми Map/Reduce, підтримує реплікацію і побудову відмовостійких конфігурацій.

У MongoDB є вбудовані засоби із забезпечення шардінгу (розподіл набору даних по серверах на основі певного ключа), комбінуючи який з реплікацією даних можна побудувати горизонтально масштабований кластер зберігання, в якому відсутня єдина точка відмови (збій будь-якого вузла не позначається на роботі БД), підтримується автоматичне відновлення після збою і перенесення навантаження з вузла, який вийшов з ладу.

Розширення кластера або перетворення одного сервера на кластер проводиться без зупинки роботи БД простим додаванням нових машин.

При розробці автори виходили з необхідності спеціалізації баз даних, завдяки чому їм вдалося відійти від принципу «один розмір під усе». За рахунок мінімізації семантики для роботи з транзакціями з'являється можливість вирішення цілого ряду проблем, пов'язаних з нестачею продуктивності, причому горизонтальне масштабування стає простішим. Використовувана модель

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

документів зберігання даних (JSON/BSON) простіше кодується, простіше управляється (у тому числі за рахунок застосування так званого безсхемного стилю (schemaless style), а внутрішнє угруповання релевантних даних забезпечує додатковий вииграш в швидкодії.

Нереляційний підхід досить зручний для створення баз даних, у яких горизонтальне масштабування означає розгортання на множині машин. Можливість забезпечувати найкращу продуктивність повинна існувати паралельно з підтримкою більшої функціональності, ніж це дозволяє використання пар «ключ-значення» (у чистому вигляді).

Технологія баз даних має працювати скрізь, починаючи з серверів користувача та віртуальних машин і закінчуючи хмарними технологіями.

MongoDB, на думку розробників, має заповнити розрив між простими сховищами даних типу «ключ-значення» (швидкими і легко масштабованими) і великими СКБД (зі структурними схемами і потужними запитами).

Основні можливості MongoDB:

1. Документо-орієнтоване сховище (проста та потужна JSON-подібна схема даних).
2. Досить гнучка мова для формування запитів.
3. Динамічні запити.
4. Повна підтримка індексів.
5. Профілювання запитів.
7. Швидкі оновлення «на місці».
8. Ефективне зберігання бінарних даних великих обсягів, наприклад, фото та відео.
9. Журналювання операцій, що модифікують дані в БД.
10. Підтримка відмовостійкості і масштабованості: асинхронна реплікація, набір реплік і шардінг.
11. Може працювати відповідно до парадигми MapReduce.

СКБД управляє наборами JSON-подібних документів, що зберігаються в бінарному форматі BSON. Зберігання і пошук файлів в MongoDB відбувається завдяки викликам протоколу GridFS.

Подібно до інших документо-орієнтованих СКБД (CouchDB тощо), MongoDB не є реляційною СКБД.

Є докладна і якісна документація, велике число прикладів і драйверів для популярних мов Java, C++, C#, PHP, Python, Perl, Ruby.

При випуску одразу було заявлено, що реліз MongoDB 1.0 готовий до використання у виробництві як одиничний хост, так і у зв'язках master/slave.

Код цього релізу досить стабільний і перевірений в промисловій експлуатації протягом 1,5 років. MongoDB рекомендується розгортати мінімум на двох серверах використовуючи реплікацію Master/Slave.

Це забезпечує наявність актуальних даних при виході з ладу однієї з СКБД. MongoDB – продукт досить молодий, і відтак у ньому зустрічаються помилки, з'являються нові можливості тощо. Характерний високий темп розробки (проект пишуть не тільки волонтери, а й компанія людей на повній зайнятості). Компанія-розробник надає платні підтримку, хостинг, консультації.

Подійно-орієнтована архітектура (Event-driven architecture, надалі EDA) – шаблон архітектури програмного забезпечення, який призначений для створення подій, їх виявлення, споживання і реагування на них.

Подія може бути визначена як значна зміна стану. Наприклад, коли споживач купує автомобіль, стан автомобіля змінюється з "на продаж" до "продано". Архітектура системи дилера автомобілів може трактувати цю зміну стану як подію, поява якої може стати відомою іншим програмам даної архітектури.

З формальної точки зору, те, що виробляється, публікується, поширюється, виявляється і споживається (як правило, асинхронно) є повідомленням, яке називають сповіщенням про подію (або нотифікацією), а не самою подією, яка є зміною стану, що викликає появу повідомлення.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

Події не подорожують, вони просто відбуваються. Проте термін подія часто використовується метонімічно для позначення самого нотифікаційного повідомлення, що може призвести до певної плутанини.

Цей архітектурний шаблон може застосовуватися при проектуванні і реалізації ПЗ і систем, які передають події між слабкозв'язаними компонентами програмного забезпечення і сервісами (службами).

Подійно-орієнтована система як правило складається з емітерів подій (або агентів) і споживачів подій (або стоків).

Стоки несуть відповідальність за здійснення реагування на появу події. Реакція не завжди може бути повністю забезпечена самим стоком. Наприклад, стік, може бути відповідальним лише за фільтрацію, трансформацію і відправку події до іншого компонента або він може забезпечити повністю самостійну реакцію на таку подію. Перша категорія стоків може бути заснована на традиційних компонентах, таких як проміжне програмне забезпечення, орієнтоване на обробку повідомлень (message oriented middleware, MOM), в той час, як друга категорія стоків (самостійна реакція в режимі он-лайн) може вимагати більш придатної платформи (фреймворку) для виконання транзакцій.

Розробка ПЗ і систем в подійно-орієнтованій архітектурі дозволяє їм бути сконструйованими способом, який більш відповідає вимогам до їх створення, оскільки такі системи в більшій мірі пристосовуються до непередбачуваних і асинхронних середовищ.

Подійно-орієнтована архітектура (EDA) може доповнювати сервісно-орієнтовану архітектуру (SOA), оскільки сервіси (служби) можуть бути активовані тригерами, які ініціюються при настанні подій.

Ця парадигма особливо корисна, коли стік не забезпечує власного виконання будь-яких дій.

Подійно-орієнтована SOA (SOA-2) розвиває архітектури SOA і EDA для забезпечення більш глибокого і надійного рівня сервісів за рахунок використання раніше невідомих причинно-наслідкових зв'язків, щоб сформувати новий шаблон

подій. Цей новий шаблон бізнес-аналітики дає поштовх до небаченого раніше зростання рівня автоматизації підприємства за рахунок привнесення додаткової цінної інформації в описану раніше модель діяльності.

Обчислювальна техніка та сенсорні пристрої (сенсори, датчики, контролери) можуть виявляти зміни стану об'єктів або умов і створювати події, які потім можуть бути оброблені сервісом (службою) або системою.

Подія може складатися з двох частин: заголовка події та тіла події. Заголовок події може включати в себе інформацію таку як, наприклад, назва події, часова мітка події і тип події. Тіло події – це частина, яка описує факт, що стався в дійсності. Тіло події не слід плутати з шаблоном або логікою, яка може бути застосована як реакція на саму подію.

Архітектура, керована подіями, складається з чотирьох логічних рівнів (шарів). Вона починається з виявлення факту, його технічного подання у формі події і закінчується непустою множиною реакцій на цю подію.

Генератор подій.

Першим логічним шаром є генератор подій, який виявляє факт і представляє цей факт подією. Оскільки фактом може бути практично все, що може бути сприйнято, то ним може бути і генератор подій. Наприклад, генератором може бути клієнт електронної пошти, система електронної комерції або певний тип датчика.

Перетворення різних даних, отриманих від датчиків, в єдину стандартизовану форму даних, які можуть бути оцінені, є основною проблемою при розробці та реалізації цього шару. Однак, враховуючи, що подія є строго декларативною, можна легко застосовувати будь-які операції трансформації, тим самим усуваючи необхідність забезпечення високого рівня стандартизації.

Канал подій.

Канал подій – це механізм, через який інформація від генератора подій передається до обробника подій (event engine) або стоку.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Це може бути з'єднання TCP/IP або вхідний файл будь-якого типу (простий текст, формат XML, e-mail тощо). В один і той же час може бути відкрито кілька каналів подій. Як правило, оскільки обробник подій повинен працювати в режимі, наближеному до реального часу, канали подій зчитуються асинхронно. Події зберігаються в черзі, очікуючи наступної обробки механізмом обробки подій.

Механізм обробки подій.

Механізм обробки подій (event processing engine) є місцем, де подія ідентифікується і вибирається відповідна реакція на нього, яка потім виконується. Це також може призвести до породження ряду тверджень. Якщо подія, яка надійшла до механізму обробки подій, є наприклад такою «Запаси продукту ID досягли нижнього допустимого рівня», це може ініціювати, наприклад, такі реакції як «Замовити продукт ID» і «Сповістити персонал».

Наступна подійно-орієнтована дія (післядія).

Щодо того, як можуть проявлятися наслідки події, слід відмітити, що вони можуть проявитись багатьма різними способами і у різноманітних формах (наприклад, повідомлення електронної пошти, надіслане комусь, або ПЗ, що виводить деяке попередження на екран). Залежно від рівня автоматизації, який забезпечується стоком (механізмом обробки подій), ці дії можуть виявитись зайвими.

Є три основні стилі обробки подій: простий, потоковий і складний. Часто ці три стилі використовуються спільно у розвинутій подійно-орієнтованій архітектурі.

Проста обробка подій.

Проста обробка подій стосується подій, які безпосередньо належать до специфічних вимірних змін умов. У випадку простої обробки подій, мають справу з появою відомих подій, що ініціюють післядію (післядії). Проста обробка подій зазвичай використовується для управління потоком робіт в реальному часі, скорочуючи тим самим час затримки і вартість робіт.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

Наприклад, прості події можуть створюватись (породжуватись) датчиком, що виявляє зміну тиску в шині або температуру навколишнього середовища.

Обробка потоку подій.

При обробці потоку подій (event stream processing, далі ESP) відбуваються як звичайні, так і відомі події. Звичайні події (заявки, передачі RFID) перевіряються на те, чи є вони відомими, і передаються інформаційним передплатникам. Обробка потоку подій зазвичай використовується для управління потоком інформації в реальному часі і на рівні підприємства, що дозволяє своєчасно приймати рішення.

Обробка складних подій.

Обробка складних подій (Complex event processing (CEP)) дозволяє за шаблонами простих і звичайних подій проводити аналіз того, чи настанула складна подія. Обробка складних подій полягає в оцінюванні взаємного впливу подій і в наступному виконанні дій. При цьому, типи подій (відомих або звичайних) можуть перетинатись, а події можуть виникати протягом тривалого періоду часу.

Кореляція подій може бути причинною, тимчасовою або просторовою. CEP вимагає використання складних інтерпретаторів подій, визначення і підбору шаблонів подій, а також відповідних кореляційних методів. Обробка складних подій зазвичай використовується для виявлення і реагування на аномальну поведінку, загрози і можливості у бізнесі.

При розробці ПЗ було використано підходи ризик-менеджменту – це система управління ризиками, яка включає в себе стратегію та тактику управління, направлені на досягнення основних цілей. Ефективний ризик-менеджмент включає:

- систему управління;
- систему ідентифікації і вимірювання;
- систему супроводження (моніторингу та контролю).

Сучасна наука представляє ризик як вірогідну подію, в результаті настання якої можуть відбутися позитивні, нейтральні або негативні наслідки. Якщо ризик припускає наявність як позитивних, так і негативних результатів, він відноситься до спекулятивних ризиків. Якщо ж наслідки негативні, або відсутні взагалі, такий ризик іменується чистим.

Мета ризик-менеджменту – підвищення конкурентоспроможності господарюючих суб'єктів за допомогою захисту від реалізації чистих ризиків.

Теорія ризик-менеджменту ґрунтується на трьох базових поняттях: корисності, регресії і диверсифікації.

У 1738 швейцарський математик Даніель Бернуллі доповнив теорію вірогідності методом корисності або привабливості того або іншого результату подій. Ідея Бернуллі полягала в тому, що в процесі ухвалення рішення люди приділяють більше уваги розміру наслідків різних результатів, ніж їх вірогідність.

В кінці XIX століття англійський дослідник Ф. Гальтон запропонував вважати регресію або повернення до середнього значення універсальною статистичною закономірністю. Суть регресії трактувалася ним як повернення явищ до норми з часом. Згодом було доведено, що правило регресії діє в найрізноманітніших ситуаціях, починаючи з азартних ігор та розрахунку вірогідності виникнення нещасних випадків, і закінчуючи прогнозуванням коливань економічних циклів.

У 1952 аспірант Університету Чикаго Гарі Марковіц в статті «Диверсифікація вкладень» («Portfolio Selection») математично обґрунтував стратегію диверсифікації інвестиційного портфеля, зокрема, він показав, як шляхом продуманого розподілу вкладень мінімізувати відхилення прибутковості від очікуваного показника. У 1990 Г. Марковіцу присуджена Нобелівська премія за розробку теорії і практики оптимізації портфеля фондових активів.

Етапи ризик-менеджменту

У ризик-менеджменті прийнято виділяти декілька ключових етапів:

- на першому етапі відбувається виявлення ризику з супутньою оцінкою вірогідності його реалізації і масштабу наслідків;
- на другому етапі здійснюється розробка ризик-стратегії з метою зниження вірогідності реалізації ризику і мінімізації можливих негативних наслідків;
- на третьому етапі вибираються методи і інструменти управління виявленим ризиком;
- на четвертому етапі проводиться безпосереднє управління ризиком;
- на завершальному етапі оцінюються досягнуті результати і коректується ризик-стратегія.

За ключовий етап ризик-менеджменту вважається етап вибору методів і інструментів управління ризиком.

Методи і інструментарій ризик-менеджменту

Базовими методами ризик-менеджменту є відмова від ризиків, зниження, передача і ухвалення.

Ризик-інструментарій значно ширший. Він включає політичні, організаційні, правові, економічні, соціальні інструменти, причому ризик-менеджмент як система допускає можливість одночасного застосування декількох методів і інструментів ризик-управління.

Найбільш часто вживаним інструментом ризик-менеджменту є страхування. Страхування припускає передачу відповідальності за відшкодування передбачуваного збитку сторонній організації (страхової компанії).

Прикладами інших інструментів можуть бути відмова від надмірно ризикової діяльності (метод відмови), профілактика або диверсифікація (метод зниження), аутсорсинг витратних ризикових функцій (метод передачі), формування резервів або запасів (метод ухвалення).

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

Redmine – вільне серверне ПЗ для управління проектами та відстежування помилок. До системи входить календар-планувальник та діаграми Ганта для візуального представлення ходу робіт за проектом та строків виконання. Redmine написано на мові Ruby і є ПЗ розробленим з використанням відомого веб-фреймворку Ruby on Rails, що означає легкість в розгортанні системи та її адаптації під конкретні вимоги. Для кожного проекту можна вести свої вікі та форуми.

Функціональні можливості:

- Ведення декількох проектів.
- Гнучка система доступу з використанням ролей.
- Система відстеження помилок.
- Діаграми Ганта та календар.
- Ведення новин проекту, документів та управління файлами.
- Сповіщення про зміни за допомогою RSS-потоків та електронної пошти.
- Власна Wiki для кожного проекту.
- Форуми для кожного проекту.
- Облік часових витрат.
- Налаштування власних (custom) полів для задач, затрат часу, проектів та користувачів.
- Легка інтеграція із системами керування версіями (SVN, CVS, Git, Mercurial, Bazaar и Darcs).
- Створення записів про помилки на основі отриманих листів
- Підтримка LDAP автентифікації.
- Можливість самореєстрації нових користувачів.
- Багатомовний інтерфейс (у тому числі українська мова).
- Підтримка СКБД: MySQL, PostgreSQL, SQLite.

Діаграма Ганта (*Gantt chart*, також стрічкова діаграма, графік Ганта) – це популярний тип діаграм, який використовується для ілюстрації плану, графіка

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

робіт за будь-яким проектом. Є одним з методів планування та управління проектами.

Діаграма Ганта являє собою відрізки (графічні плашки), розміщені на горизонтальній шкалі часу. Кожен відрізок відповідає окремому завданню або підзадачі. Завдання і підзадачі, складові плану, розміщуються по вертикалі. Початок, кінець і довжина відрізка на шкалі часу відповідають початку, кінцю і тривалості завдання. На деяких діаграмах Ганта також показується залежність між завданнями.

Діаграма може використовуватися для представлення поточного стану виконання робіт: частина прямокутника, що відповідає завданню, заштриховується, відзначаючи відсоток виконання завдання. показується вертикальна лінія, що відповідає моменту «сьогодні».

Часто діаграма Ганта використовується спільно з таблицею зі списком робіт, рядки якої відповідають окремо взятій задачі, зображеній на діаграмі, а стовпці містять додаткову інформацію про задачу.

Система відстеження помилок Багтрекер – прикладна програма для допомоги розробникам програмного забезпечення (програмістам, тестувальникам тощо) враховувати і контролювати помилки, знайдені у програмах, питання щодо функціональності, рішення та оновлення, побажання користувачів, а також стежити за процесом їх виконання.

Кожному, хто розробляв програмні продукти, добре знайоме співвідношення «20/80» – останні 20 % роботи тривають 80 % часу.

Як це не парадоксально, але нічого дивного в цій пропорції немає, адже саме на завершальній стадії починається тестування проекту, коли виявляються помилки, і що більший проект, то більше буде знайдено помилок.

Водночас досить часто виявляється, що більшість цих помилок були відомі та могли бути виправлені з меншими витратами на попередніх стадіях роботи, але не були вчасно описані, а потім загубилися серед інших важливих завдань.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Отже, система відстеження помилок у найпростішому варіанті – це процес, що включає в себе виявлення помилки, її опис, виправлення і перевірку цього виправлення, тобто процес «стеження» за багом протягом всього як його життєвого циклу, так і життєвого циклу розробки в цілому.

Сукупність інформації про дефект. Головний компонент такої системи – база даних, що містить відомості про виявлені дефекти. Ці відомості можуть включати в себе:

- номер (ідентифікатор) дефекту;
- хто повідомив про дефект;
- дата і час виявлення дефекту;
- версія продукту, в якій виявлено дефект;
- серйозність (критичність) дефекту та пріоритет рішення;
- опис кроків для відтворення дефекту (неправильної поведінки програми);
- відповідальний за усунення дефекту;
- обговорення можливих рішень та їх наслідків;
- поточний стан виправлення дефекту;
- версії продукту, в якій дефект виправлений.

Крім того, розвинені системи надають можливість прикріплювати файли, які допомагають описати проблему, наприклад, дамп пам'яті або скріншот.

Використання. Основна перевага систем відстеження помилок полягає в забезпеченні чітких централізованих оглядів, запитів на розробку (включаючи помилки і виправлення) та їх стан. У корпоративному середовищі, системи відстеження помилок можуть бути використані для генерації звітів по продуктивності програмістів виправлення помилок. Однак, це може іноді приводити до неточних результатів, тому що різні помилки можуть мати різні ступені пріоритету та серйозності, що пов'язано з складністю їх фіксації.

Життєвий цикл дефекту. Як правило, система відстеження помилок використовує той чи інший варіант «життєвого циклу» помилки, стадія якого визначається поточним станом помилки.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Типовий життєвий цикл дефекту:

1. Новий – дефект зареєстрований тестувальником.
2. Призначений – призначений відповідальний за виправлення дефекту.
3. Дозволений – дефект переходить назад у сферу відповідальності тестувальника. Як правило, супроводжується резолюцією, наприклад:

– Виправлено (виправлення включені у версію таку-то).

- Дубль (повторює дефект, що вже знаходиться в роботі).
- Не виправлено (працює відповідно до специфікації, має занадто низький пріоритет, виправлення відкладено до наступної версії тощо).

– «В мене все працює» (запит додаткової інформації про умови, в яких дефект проявляється).

4. Далі тестувальник проводить перевірку виправлення, залежно від чого дефект або знову переходить у стан «Призначений» (якщо він описаний як виправлений, але не виправлений), або у стан «Закрито».

5. Відкрито повторно – дефект знайдено знову в іншій версії.

Система може надавати адміністраторові можливість налаштування користувачі, які можуть переглядати і редагувати помилки залежно від їх стану, переводити їх в інший стан або видаляти.

У корпоративному середовищі, система відстеження помилок може використовуватися для отримання звітів, що показують продуктивність програмістів при виправленні помилок. Однак, часто такий підхід не дає достатньо точних результатів через те, що різні помилки мають різну ступінь серйозності та складності. При цьому серйозність проблеми прямо не стосується складності її усунення. Незважаючи на те що я працював над ПЗ один в реалізації програми я використовував підходи пришвидшення розробки на основі методологій Agile.

Гнучка розробка програмного забезпечення (Agile software development, agile-методи) – клас методологій розробки програмного забезпечення, що

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

базується на ітеративній розробці, в якій вимоги та розв'язки еволюціонують через співпрацю між самоорганізовуваними багатофункціональними командами.

Гнучка розробка – найкращий засіб для підвищення продуктивності розробників програмного забезпечення.

Більшість гнучких методологій націлені на мінімізацію ризиків, шляхом зведення розробки до серії коротких циклів, що мають назву ітерацій, які зазвичай тривають один-два тижні. Кожна ітерація сама по собі виглядає як програмний проект в мініатюрі, і включає всі завдання, необхідні для видачі мінімального приросту за функціональністю: планування, аналіз вимог, проектування, кодування, тестування і документування. Хоча окрема ітерація, як правило, недостатня для випуску нової версії продукту, мається на увазі те, що гнучкий програмний проект готовий до випуску наприкінці кожної ітерації. Після закінчення кожної ітерації, команда виконує переоцінку пріоритетів розробки.

Agile акцентує увагу на безпосередньому спілкуванні «віч-на-віч». Більшість agile команд розташовані в одному офісі, його іноді називають bullpen. Як мінімум вона включає і «замовників» (замовники, які визначають продукт, також це можуть бути менеджери продукту, бізнес аналітики або клієнти). Офіс може також включати тестувальників, дизайнерів інтерфейсу, технічних авторів і менеджерів.

Основною метрикою agile методів є робочий продукт. Віддаючи перевагу безпосередньому спілкуванню, agile-методи зменшують обсяг письмової документації в порівнянні з іншими методами. Це привело до критики цих методів як недисциплінованих.

Agile – родина процесів розробки, а не єдиний підхід в розробці програмного забезпечення, і визначається Agile Manifesto. Agile не включає практик, а визначає цінності та принципи, якими керуються успішні команди.

Agile Manifesto розроблений і прийнятий 17 розробниками 11-13 лютого 2001 року на лижному курорті The Lodge at Snowbird в горах Юти. Маніфест підписали представники наступних методологій Extreme programming, Scrum,

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

DSDM, Adaptive software development, Crystal Clear, Feature driven development, Pragmatic Programming. Agile Manifesto містить 4 основні ідеї та 12 принципів. Примітно, що Agile Manifesto не містить практичних порад.

Основні ідеї:

- Особистості та їхні взаємодії важливіші, ніж процеси та інструменти.
- Робоче програмне забезпечення важливіше, ніж повна документація.
- Співпраця із замовником важливіша, ніж контрактні зобов'язання.
- Реакція на зміни важливіша, ніж дотримання плану.

Принципи, які роз'яснює Agile Manifesto:

- задоволення клієнта за рахунок ранньої та безперебійної поставки коштовного програмного забезпечення;
- вітання змін вимог навіть наприкінці розробки (це може підвищити конкурентоспроможність отриманого продукту);
- часта поставка робочого програмного забезпечення (кожен місяць або тиждень або ще частіше);
- тісне, щоденне спілкування замовника з розробниками впродовж всього проекту;
- проектом займаються мотивовані особистості, які забезпечені потрібними умовами роботи, підтримкою і довірою;
- рекомендований метод передачі інформації – особиста розмова (віч-на-віч);
- робоче програмне забезпечення – найкращий вимірювач прогресу;
- спонсори, розробники та користувачі повинні мати можливість підтримувати постійний темп на невизначений термін;
- постійну увагу поліпшенню технічної майстерності та зручному дизайну;
- простота – мистецтво не робити зайвої роботи;
- найкращі технічні вимоги, дизайн та архітектура виходять у самоорганізованої команди;
- постійна адаптація до мінливих обставин.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		67

Маніфест та Принципи гнучкої розробки містять високорівневі ідеї щодо того, як потрібно вибудовувати процес розробки програмного забезпечення, щоб успішно завершувати проекти й створювати команди, в яких приємно та цікаво працювати.

Документи визначають, що потрібно для цього зробити, але не говорять, як це зробити. По-іншому й не могло бути, оскільки Маніфест та Принципи народилися внаслідок консенсусу представників різних (хоча й споріднених) напрямів, які могли знайти спільну основу лише на рівні базових цінностей та принципів.

Критика. Багато керівників проектів, що працюють у традиційних методологіях на кшталт «водоспаду», критикують agile-методи.

Один з повторюваних пунктів критики: при agile-підході часто нехтують створенням «дорожньої карти» розвитку продукту, так само як і управлінням вимогами, в процесі якого і формується така «карта». Гнучкий підхід до управління вимогами не має на увазі далекосяжних планів (по суті, управління вимогами просто не існує в даній методології), а має на увазі можливість замовника раптом і несподівано наприкінці кожної ітерації виставляти нові вимоги, що часто суперечать архітектурі вже створеного і поставленого продукту. Таке іноді призводить до катастрофічних «авралів» з масовим рефакторингом і переробками практично на кожній черговій ітерації.

Крім того вважається, що робота в agile мотивує розробників вирішувати всі прибулі завдання найпростішим і найшвидшим можливим способом, при цьому часто не звертаючи уваги на коректність коду з точки зору вимог базової платформи (підхід «працює, та й добре», при цьому не враховується, що може перестати працювати при найменшій зміні або ж породити важкі до відтворення дефекти після реального розгортання у клієнта). Це призводить до зниження якості продукту і накопиченню дефектів.

Методології. Існують методології, які дотримуються цінностей і принципів заявлених в Agile Manifesto, деякі з них:

1. Agile Modeling – набір понять, принципів і прийомів (практик), що дозволяють швидко і просто виконувати моделювання і документування в проектах розробки програмного забезпечення. Не включає в себе детальну інструкцію з проектування, не містить описів, як будувати діаграми на UML.

Основна мета – ефективне моделювання і документування; але не охоплює програмування та тестування, не включає питання управління проектом, розгортання і супроводу системи. Однак включає в себе перевірку моделі кодом.

2. Agile Unified Process (AUP) спрощена версія IBM Rational Unified Process (RUP), розроблена Скоттом Амблером, яка описує просте і зрозуміле наближення (модель) для створення програмного забезпечення для бізнес-додатків.

3 Agile Data Method – група ітеративних методів розробки програмного забезпечення, в яких вимоги та рішення досягаються в рамках співпраці різних крос-функціональних команд.

4. DSDM заснований на концепції швидкої розробки додатків (Rapid Application Development, RAD). Являє собою ітеративний і інкрементний підхід, який надає особливого значення тривалій участі в процесі користувача/споживача.

5. Essential Unified Process (EssUP).

6. Екстремальне програмування (Extreme programming, XP).

7. Feature driven development (FDD) – функціонально-орієнтована розробка. Використовуване в FDD поняття функції або властивості (feature) Системи досить близько до поняття прецеденту використання, використовуваному в RUP, істотна відмінність – це додаткове обмеження: «кожна функція повинна допускати реалізацію не більше, ніж за два тижні». Тобто якщо сценарій використання досить малий, його можна вважати функцією. Якщо ж великий, то його треба розбити на декілька відносно незалежних функцій.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

8. Getting Real – ітераційний підхід без функціональних специфікацій, що використовується для веб-додатків. У даному методі спершу розробляється інтерфейс програми, а потім її функціональна частина.

9. OpenUP – це ітераційно-інкрементний метод розробки програмного забезпечення. Позиціюється, як легкий і гнучкий варіант RUP. OpenUP ділить життєвий цикл проекту на чотири фази: початкова фаза, фази уточнення, конструювання та передачі. Життєвий цикл проекту забезпечує надання зацікавленим особам та членам колективу точок ознайомлення і прийняття рішень впродовж усього проекту. Це дозволяє ефективно контролювати ситуацію і вчасно приймати рішення про задовільність результатів. План проекту визначає життєвий цикл, а кінцевим результатом є остаточний додаток.

10. Scrum встановлює правила керування процесом розробки та дозволяє використовувати вже існуючі практики кодування, коректуючи вимоги або вносячи тактичні зміни. Використання цієї методології дає можливість виявляти і усувати відхилення від бажаного результату на більш ранніх етапах розробки програмного продукту.

11. Бережлива розробка програмного забезпечення (lean software development). Використовує підходи з концепції бережливого виробництва.

При розробці ПЗ було використано V-Model (або VEE модель) є моделлю розробки інформаційних систем (ИС), спрямованої на спрощення розуміння складнощів, пов'язаних з розробкою систем. Вона використовується для визначення єдиної процедури розробки програмного забезпечення, апаратного забезпечення та людино-машинного інтерфейсу.

Концепція V-подібної моделі була розроблена Німеччиною та США в кінці 1980-х років незалежно один від одного:

– Німецька V-модель була розроблена аерокосмічної компанією IABG в Оттобрунні поряд з Мюнхеном у сприянні з Федеральним департаментом з закупівлі озброєнь в Кобленці, для Міністерства оборони Німеччини. Модель

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

була прийнята німецькою федеральною адміністрацією для цивільних потреб влітку 1992.

– Американська V-Model (VEE) була розроблена національною радою з системної інженерії (міжнародна – з 1995 року) для супутникових систем, включаючи обладнання, програмне забезпечення та взаємодію з користувачами.

Сучасною версією V-Model є V-Model XT, яка була затверджена в лютому 2005 року. V-модель використовується для управління процесом розробки програмного забезпечення для німецької федеральної адміністрації.

Зараз вона є стандартом для німецьких урядових і оборонних проектів, а також для виробників ПЗ в Німеччині. V-Model являє собою скоріше набір стандартів у галузі проектів, що стосуються розробки нових продуктів. Ця модель багато в чому схожа з Prince2 і описує методи як для проектного управління, так і для системного розвитку.

Основні принципи

Основний принцип V-подібної моделі полягає в тому, що деталізація проекту зростає при русі зліва направо, одночасно з плином часу, і ні те, ні інше не може повернути назад. Ітерації в проекті виробляються по горизонталі, між лівою і правою сторонами літери.

Стосовно до розробки інформаційних систем V-Model – варіація каскадної моделі, в якій завдання розробки йдуть зверху вниз по лівій стороні букви V, а завдання тестування – вгору по правій стороні букви V. Усередині V проводяться горизонтальні лінії, що показують, як результати кожної з фаз розробки впливають на розвиток системи тестування на кожній із фаз тестування.

Модель базується на тому, що прийнятно-здавальні випробування ґрунтуються, насамперед, на вимогах, системне тестування – на вимогах та архітектури, комплексне тестування – на вимогах, архітектурі та інтерфейсах, а компонентне тестування – на вимогах, архітектурі, інтерфейсах та алгоритмах

Цілі

V-модель забезпечує підтримку у плануванні та реалізації проекту. В ході проекту ставляться такі завдання:

1. Мінімізація ризиків: V-подібна модель робить проект більш прозорим і підвищує якість контролю проекту шляхом стандартизації проміжних цілей і опису відповідних їм результатів та відповідальних осіб. Це дозволяє виявляти відхилення в проекті і ризики на ранніх стадіях і покращує якість управління проектом.

2. Підвищення та гарантії якості: V-Model – стандартизована модель розробки, що дозволяє домогтися від проекту результатів бажаної якості. Проміжні результати можуть бути перевірені на ранніх стадіях. Універсальне документування полегшує читаність, зрозумілість та контрольованість.

3. Зменшення загальної вартості проекту: Ресурси на розробку, виробництво, управління і підтримку можуть бути заздалегідь прораховані та проконтрольовані. Отримувані результати також універсальні і легко прогножуються. Це зменшує витрати на подальші стадії та проекти.

4. Підвищення якості комунікації між учасниками проекту: Універсальний опис усіх елементів та умов полегшує взаєморозуміння всіх учасників проекту. Таким чином, зменшуються неточності у розумінні між користувачем, покупцем, постачальником і розробником.

Переваги:

– Користувачі V-Model беруть участь у розробці та підтримці V-моделі. Комітет з контролю за змінами підтримує проект і збирається раз на рік для обробки всіх отриманих запитів на внесення змін до V-Model.

– На старті будь-якого проекту V-подібна модель може бути адаптована під цей проект, так як ця модель не залежить від типів організацій та проектів.

– V-model дозволяє розбити діяльність на окремі кроки, кожен з яких буде включати в себе необхідні для нього дії, інструкції до них, рекомендації та докладне пояснення діяльності.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Обмеження. Наступні моменти не враховуються в V-моделі, але можуть бути розглянуті окремо, або можливо адаптувати модель під них:

- Не регулюється розміщення контрактів на обслуговування.
- Організація і виконання управління, обслуговування, ремонту та утилізації системи не враховуються в V-моделі. Однак, планування і підготовка до цих операцій моделлю розглядаються.
- V-подібна модель більше стосується розробки програмного забезпечення в проекті, ніж всієї організації процесу.

Переваги:

- У моделі особливе значення надається плануванню, спрямованому на верифікацію та атестацію розроблювального продукту на ранніх стадіях його розробки. Фаза модульного тестування підтверджує правильність деталізованого проектування. Фази інтеграції та тестування реалізують архітектурне проектування або проектування на вищому рівні. Фаза тестування системи підтверджує правильність виконання етапу вимог до продукту і його специфікації.
- У моделі передбачені атестація та верифікація всіх зовнішніх і внутрішніх отриманих даних, а не тільки самого програмного продукту.
- У V-подібній моделі визначення вимог виконується перед розробкою проекту системи, а проектування ПЗ – перед розробкою компонентів.
- Модель визначає продукти, які повинні бути отримані в результаті процесу розробки, причому кожен отриманий дани повинні піддаватися тестуванню.
- Завдяки моделі менеджери проекту можуть відслідковувати хід процесу розробки, так як в даному випадку цілком можливо скористатися тимчасовою шкалою, а завершення кожної фази є контрольною точкою.

Недоліки:

- Модель не передбачає роботу з паралельними подіями.
- У моделі не передбачено внесення вимоги динамічних змін на різних етапах життєвого циклу.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

– Тестування вимог в життєвому циклі відбувається занадто пізно, внаслідок чого неможливо внести змін, не вплинувши при цьому на графік виконання проекту.

– У модель не входять дії, спрямовані на аналіз ризиків.

– Деякий результат можна отримати тільки при досягненні низу букви V.

4.2 Захист розробленого програмного забезпечення

Розроблене програмне забезпечення захистимо за допомогою наступного алгоритму захисту інформації DSA.

Відправник і одержувач електронного документа використовують при обчисленні великі цілі числа: G і P – прості числа, L біт кожне ($512 < L < 1024$); q – просте число довжиною 160 біт (дільник числа $(P-1)$). Числа G , P , q є відкритими й можуть бути загальними для всіх користувачів мережі.

Відправник вибирає випадкове ціле число X , $1 < X < q$. Число X є секретним ключем відправника для формування електронного цифрового підпису. Потім відправник обчислює значення $Y = G^X \bmod P$. Число Y є відкритим ключем для перевірки підпису відправника. Число Y передається всім одержувачам документів.

Цей алгоритм також передбачає використання однобічної функції гешування $h(-)$. У стандарті DSS визначений алгоритм безпечного гешування SHA. Для того щоб підписати документ M , відправник гешує його в ціле геш-значення m : $m = h(M)$, $1 < m < q$, потім генерує випадкове ціле число K , $1 < K < q$, і обчислює число r : $r = (G^K \bmod P) \bmod q$. Потім відправник обчислює за допомогою секретного ключа X ціле число s :

$$s = \frac{m + r * X}{K} \bmod q .$$

Пара чисел r і s утворить цифровий підпис $S = (r, s)$ під документом M . Таким чином, підписане повідомлення являє собою трійку чисел $[M, r, s]$. Одержувач підписаного повідомлення $[M, r, s]$ перевіряє виконання умов $0 < r < q$, $0 < s < q$ і відкидає підпис, якщо хоча б одна із цих умов не виконана. Потім одержувач обчислює значення $w = 1/s \bmod q$, геш-значення $m = h(M)$ і числа $u_1 = (m * w) \bmod q$, $u_2 = (r * w) \bmod q$. Далі одержувач за допомогою відкритого ключа Y обчислює значення $v = ((G^{u_1} * Y^{u_2}) \bmod P) \bmod q$ і перевіряє виконання умови $v = r$. Якщо умова $v = r$ виконується, тоді підпис $S = (r, s)$ під документом M визнається одержувачем справжнім.

КБПЗ-2026

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система комунікаційної платформи як сервісу CPaaS. Терміни комунікаційна платформа як послуга (CPaaS) і уніфікована комунікація як послуга (UCaaS) можуть вводити в оману, оскільки як CPaaS, так і UCaaS пропонують безліч однакових базових комунікаційних послуг. Однак вони відрізняються тим, як саме вони надають ці сервіси.

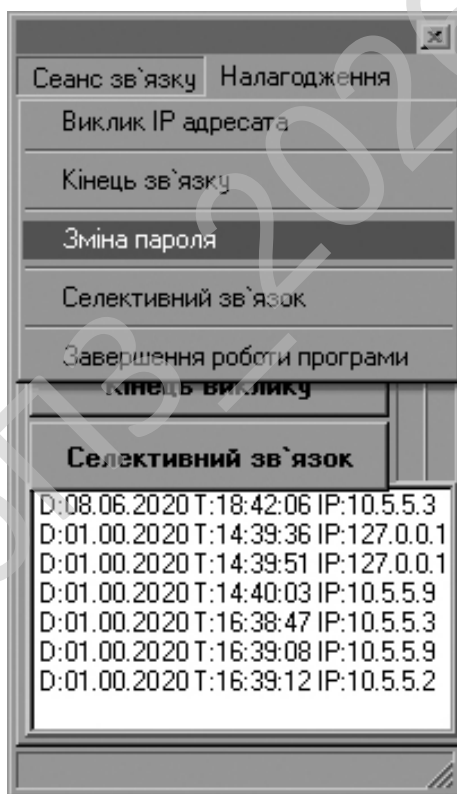


Рисунок 5.1 – Головне вікно ПЗ

UCaaS нагадує традиційну локальну платформу, всі функції й застосунки якої перебувають на платформі постачальника послуг. На відміну від цього, CPaaS просто надає розроблювачам необхідні інструменти для налаштування й

вбудовування комунікацій в існуючі бізнес-застосунки. Теоретично, відносини між UCaaS і CPaaS є взаємодоповнюючими, а не конкурентними. Багато постачальників платформ UCaaS всі частіше займають проміжну позицію в суперечках UCaaS проти CPaaS, пропонуючи як API, так і платформи CPaaS. Коли платформа CPaaS перебуває поверх платформи UCaaS, вона може розширити можливості організації взаємодіяти з більше широким колом клієнтів без повного розриву й заміни комунікаційної інфраструктури організації. Розроблена програма має дуже простий і зрозумілий інтерфейс з користувачем. Кожен, хто в достатньому обсязі володіє операційним середовищем Windows без особливих складностей освоїть і цю програму, оскільки її інтерфейс інтуїтивно зрозумілий. Якщо програма не видала ніяких помилок, і працює, то можна використовувати, інакше слід слідувати інструкціям, які пропонує програма.

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення.

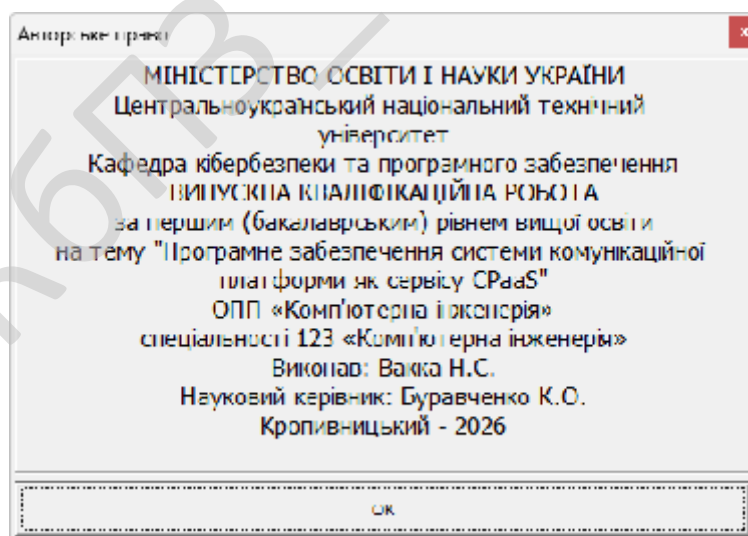


Рисунок 5.2 – Авторське право

Розглянемо процес впровадження програмного забезпечення, це процес налаштування програмного забезпечення під певні умови використання, а також

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

навчання користувачів роботі з програмним продуктом. Впровадження програмного забезпечення це усі дії, що роблять розроблену програмну систему готовою до використання. Даний процес є частинною життєвого циклу програмного забезпечення.

Загалом процес розгортання складається з кількох взаємопов'язаних дій із можливими переходами між ними. Ця активність може відбуватися як з боку виробника так і з боку споживача. Оскільки кожна програмна система є унікальною, то усі процеси та процедури під час розгортання важко передбачити. Тому, "розгортання" можна трактувати як загальний процес відповідно до певних вимог та характеристик. Розгортання може здійснюватись програмістом і в процесі розробки програмного забезпечення.

До діяльностей пов'язаних із розгортанням програмного забезпечення відносять:

- Випуск.
- Встановлення та активація.
- Деактивація.
- Адаптація.
- Оновлення.
- Вмонтування.
- Відстежування версій.
- Видалення.
- Вилучення з обігу.

При впровадженні програмного забезпечення потрібно урахувати наступні дії:

– Виділення критичних, з точки зору загального результату, процедур в діяльності організації. Коли набір таких процедур визначений, необхідно в першу чергу використовувати ІТ рішення для автоматизації операцій усередині саме цих процедур. Таким чином, розроблене ІТ рішення автоматично стає життєво

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

важливим і затребуваним для організації, а також буде забезпечена публічність процесу впровадження;

– Розширення нормативної бази організації шляхом включення до неї регламентів, що описують порядок виконання процедур автоматизованих процесів. В іншому випадку є небезпека виникнення неузгодженості між автоматизованими процедурами та іншими процесами організації.

– Виконання робіт з загальної стандартизації існуючої діяльності організації, коли виділяються кращі практики виконання процедур і включаються в ІТ рішення за принципом найбільшої корисності для більшості учасників. Відсоток таких процедур щодо загального обсягу автоматизації може бути невеликий, але це надає процесу побудови рішення вагу в організації за рахунок збільшення його необхідності.

Під час роботи над програмою було проведено тестування програмного забезпечення, тобто технічне дослідження, призначене для виявлення інформації про якість продукту відносно контексту, в якому воно має використовуватись.

Тестування включає як процес пошуку помилок або інших дефектів, так і випробування програмних складових з метою їх оцінки.

Проводилась оцінка:

- відповідності поставленим вимогам;
- правильна відповідь для усіх можливих вхідних даних;
- виконання функцій за прийнятний час;
- практичність;
- сумісність з ОС та стороннім ПЗ.

Оскільки число можливих тестів для програмних компонент практично нескінченне, тому стратегія тестування полягала в тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів.

Як результат ПЗ тестувалось стандартним виконанням програми з метою виявлення помилок або інших дефектів.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

Проводилось тестування форматом білої скриньки засноване на аналізі керуючої структури програми. Програма вважається повністю перевіреною, якщо проведено вичерпне тестування маршрутів (шляхів) її графа управління.

У цьому випадку формуються тестові варіанти, в яких:

- Гарантується перевірка всіх незалежних маршрутів програми.
- Знаходяться гілки True, False для всіх логічних рішень.
- Виконуються всі цикли (у межах їхніх кордонів та діапазонів).
- Аналізується правильність внутрішніх структур даних.

Недоліки тестування "білої скриньки":

- Кількість незалежних маршрутів може бути дуже велика.
- Повне тестування маршрутів не гарантує відповідності програми вихідним вимогам до неї.
- У програмі можуть бути пропущені деякі маршрути.
- Не можна виявити помилки, поява яких залежить від даних.

Переваги тестування "білої скриньки" пов'язані з тим, що принцип «білої скриньки» дозволяє врахувати особливості програмних помилок:

- Кількість помилок мінімально в «центрі» і максимально на «периферії» програми.
- Попередні припущення про ймовірність потоку керування або даних у програмі часто бувають некоректними. У результаті типовим може стати маршрут, модель обчислень за яким опрацьована слабо.
- При записі алгоритму програмного забезпечення у вигляді тексту на мові програмування можливе внесення типових помилок трансляції (синтаксичних та семантичних).
- Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Проводилось тестування чорної скриньки. Основне місце програми тестів «чорної скриньки» — інтерфейс ПЗ. Відомі: функції програми. Досліджується: робота кожної функції на всій області визначення.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

Ці тести демонструють:

- Як виконуються функції програми.
- Як приймаються вихідні дані.
- Як виробляються результати.
- Як зберігається цілісність зовнішньої інформації.

При тестуванні «чорної скриньки» розглядаються системні характеристики програм, ігнорується їхня внутрішня логічна структура. Вичерпне тестування, як правило, неможливе.

Наприклад, якщо в програмі 10 вхідних величин і кожна приймає по 10 значень, то кількість тестових варіантів становитиме 10^{10} . Тестування «чорної скриньки» не реагує на багато особливостей програмних помилок.

Тестування «чорної скриньки» (функціональне тестування) дозволяє отримати комбінації вхідних даних, які забезпечують повну перевірку всіх функціональних вимог до програми.

Програмний виріб тут розглядається як «чорна скринька», чію поведінку можна визначити тільки дослідженням його входів та відповідних виходів. При такому підході бажано мати:

- Набір, утворений такими вхідними даними, які призводять до аномалій у поведінці програми (назвемо його ІТс).
- Набір, утворений такими вхідними даними, які демонструють дефекти програми (назвемо його ОТ).

Будь-який спосіб тестування «чорної скриньки» повинен:

- Виявити такі вхідні дані, які з високою ймовірністю належать набору ІТс.
- Сформулювати такі очікувані результати, які з високою ймовірністю є елементами набору ОТ.

Принцип «чорної скриньки» не альтернативний принципу «білої скриньки». Скоріше це доповнює підхід, який виявляє інший клас помилок.

Тестування «чорної скриньки» забезпечує пошук наступних категорій помилок:

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		81

- Некоректних чи відсутніх функцій.
- Помилки інтерфейсу.
- Помилки у зовнішніх структурах даних або в доступі до зовнішньої бази даних.

- Помилки характеристик (необхідна ємність пам'яті і т.д.).
- Помилки ініціалізації та завершення.

Обрано умови розповсюдження – Shareware.

Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми.

В угоді про використання (ліцензії для кінцевого користувача, EULA) також може бути обумовлена заборона на комерційне або професійне (не тестове) її використання.

Основний принцип умовно-безплатного ПЗ – «спробуй, перш ніж купити» (try before you buy). ПЗ що поширюється як умовно-безплатний, надається користувачам безоплатно. Звичайно користувач платить тільки за час завантаження файлів через Інтернет або за носій (CD диск, флешку, ключ). Протягом певного терміну, що становить зазвичай тридцять днів, він може користуватися програмою, тестувати її, освоювати її можливості.

Якщо після закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєstrуватися), заплативши авторові певну суму.

В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи комунікаційної платформи як сервісу CPaaS.

В межах України в недостатній мірі представлені вітчизняні розробки в цій області.

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем комунікаційної платформи як сервісу CPaaS.
- Досліджена система комунікаційної платформи як сервісу CPaaS.
- На основі отриманих результатів досліджень створена програмна реалізація системи комунікаційної платформи як сервісу CPaaS.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання комунікаційної платформи як сервісу CPaaS.

Розроблене програмне забезпечення має простий, дружній та зручний інтерфейс користувача, що забезпечує легкість у освоєнні роботи програмного продукту, зручність у використанні, і не потребує особливих спеціальних знань.

При створенні програмного забезпечення було використано об'єктно-орієнтований підхід, що відповідає сучасним тенденціям у галузі розробки комерційних програмних систем.

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані призначені для системи комунікаційної платформи як сервісу CPaaS. Це дозволило мінімізувати строк розробки програмного забезпечення, і, як слід, зменшити витрати на його розробку. Запропоноване програмне забезпечення ділиться на загальне програмне забезпечення, що поставляється із засобами обчислювальної техніки й спеціальне

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

програмне забезпечення, що спеціально розроблене для даної конкретної системи й включає програми, що реалізують її функції.

Програма призначена для виконання під управлінням багатозадачної операційної системи Windows 10/11.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм DSA.

В цілому створене програмне забезпечення підтверджує правильність використаних проектних рішень та повністю відповідає вимогам технічного завдання. Створене програмне забезпечення має потенційну можливість для подальшого вдосконалення і застосування у різних галузях.

К6ПЗ-2026

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
2. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
3. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
4. Інженерна комп'ютерна графіка: підручник / В.В. Проців [та ін.] / М-во освіти і науки України, Нац. гірн. унт-т. – Дніпро: НГУ, 2017. – 247 с.
5. Проців В.В. Прикладна комп'ютерна графіка [Текст]: Навч. посібник / В.В. Проців, К.А. Зіборов, К.М. Бас, Г.К. Ванжа; М-во освіти і наук, Нац. гірн. ун-т. - Д.: НГУ, 2016. - 187 с.
6. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
7. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
8. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 379–402.
9. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems*, 2024, pp. 403–447
10. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		85

технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.

11. Батрак О., Смірнова Т., Гнатюк В., Одарченко Р., Смірнов О. «Дослідження показників ефективності функціонування та перспектив розвитку систем IP-телефонії». *Підводні технології*, 2024, № 13, с. 28-35.

12. Al-Mudhafar Aqeel, A.M., Smirnova, T., Buravchenko, K., Smirnov, O. «The method of assessing and improving the user experience of subscribers in software-configured networks based on the use of machine learning». *Advanced Information Systems*, 2023, 7(2), pp. 49-56.

13. Smirnov, O., Sydorenko, V., Aleksander, M., Zhyharevych, O., Yenchев, S. «Simulation of the cloud IoT-based monitoring system for critical infrastructures». *CEUR Workshop Proceedings*, Volume 3530, 2023, pp. 256-265.

14. Smirnov, O., Odarchenko, R., Smirnova, T., Bondar, S., Volosheniuk, D. «Optimal Structure Construction of Private 5G Network for the Needs of Enterprises». *Lecture Notes on Data Engineering and Communications Technologies*, 2023, 178, pp. 208–223.

15. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувацького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.

16. Smirnov, O., Neskoriadiеva, T., Fedorov, E., Rudakov, K., Neskoriadiеva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,

17. Smirnov O., Smirnova T., Anas M. Al-Oraiqat, Drieiev O., Polishchuk L., Sheroz Khan, Yassin M. Y. Hasan, Aladdein M. Amro, Hazim S. AlRawashdeh «Method for Determining Treated Metal Surface Quality Using Computer Vision Technology». *Sensors (Basel, Switzerland)* Volume 22, Issue 16, 6223, 2022.

18. Smirnov O., Kuznetsov A., Kryvinska N., Kiian A., Kuznetsova K. «Full Non-Binary Constant-Weight Codes». SN Computer Science, Vol 2, 337, 2021. <https://doi.org/10.1007/s42979-021-00739-w>

19. Smirnov O., Kuznetsov A., Zhora V., Onikiychuk A., Pieshkova O. «Hiding Messages in Audio Files Using Direct Spread Spectrum». 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2021, Cracow, Poland, 22-25 September 2021. P. 414-418.

20. Smirnov O., Kuznetsov A., Lokotkova I., Kuznetsova T., Florov S., Lebid O. «Using Orthogonal Signals to Hide Information in Images». 4 IEEE International Conference on Advanced Information and Communication Technologies (AICT) - 2021, Lviv, Ukraine, September 21-25, 2021. P. 255-260.

21. Smirnov, O., Kuznetsov, A., Potii, O., Poluyanenko, N., Stelnyk, I., Mialkovsky, D. «Combining and filtering functions in the framework of nonlinear-feedback shift register». International Journal of Computing; 2020, Volume 19, Issue 2 – Research Institute for Intelligent Computer Systems – 2020. – P. 247-256.

22. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova T. «Non-binary constant weight coding technique». CEUR Workshop Proceedings. Volume 2740, 2020, Pages 102-114.

23. Smirnov O., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». Journal of theoretical and applied information technology Vol.98. No 21, 2020, P. 3334-3346.

24. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». CEUR Workshop Proceedings Volume 2654, 2020, Pages 1-14.

25. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». 2020 IEEE 11th International Conference on Dependable

Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

26. Smirnov O., Kuznetsov A., Kiian A., Cherep A., Kanabekova M., Chepurko I. «Testing of code-based pseudorandom number generators for post-quantum application». 2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT), Ukraine, Kyiv, May 14-18. 2020. P. 172-177.

27. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies, vol 48. Springer, Cham. 2021. pp 557-587.

28. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». CEUR Workshop Proceedings Volume 2616, 2020, Pages 366-379.

29. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», CEUR Workshop Proceedings Volume 2608, 2020, Pages 633-645.

30. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», CEUR Workshop Proceedings Volume 2608, 2020, Pages 646-660.

31. Zhurakovskiy, B., Tsopa, N., Batrak, Y., Odarchenko, R., Smirnova, T «Comparative analysis of modern formats of lossy audio compression». Workshop Proceedings, 2020, 2654, стр. 315-327.

32. Smirnov O. Kuznetsov A., Zaichenko Yu., Pastukhov M., Oleshko O., Kuznetsova K., «Formation of Discrete Signals with Special Correlation Properties». International Conference on Information and Telecommunication Technologies and Radio Electronics, UkrMiCo 2019; Odessa; Ukraine; 9-13 September 2019. P.22-28.

33. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». International Journal of Computing; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407.

34. Smirnov, O., Kuznetsov, A., Reshetniak, O., Ivko, N., Katkova, T., Kuznetsova, T., «Generators of Pseudorandom Sequence with Multilevel Function of Correlation». 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T), Kyiv, Ukraine, 8 – 11 October 2019 . P.517-522.

35. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». CEUR Workshop Proceedings, Vol 2588, P. 90-106, 2019.

36. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Averchev, A., Pastukhov, M., Kuznetsova, K., «Formation of Pseudorandom Sequences with Special Correlation Properties», 2019 3rd International Conference on Advanced Information and Communications Technologies, AICT -2019/ Lviv, Ukraine, 2-6 July, 2019, P. 395-399.

37. Smirnov, O., Kuznetsov, A., Kavun, S., Babenko, B., Nakisko, O., Kuznetsova, K., «Malware Correlation Monitoring in Computer Networks of Promising Smart Grids», 2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS), Kyiv, Ukraine April 17-19, 2019 P. 347-352.

38. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019, Pages 618-629.

39. Smirnov A.A., Kuznetsov A.A., Danilenko D.A., Berezovsky A., «The statistical analysis of a network traffic for the intrusion detection and prevention systems», Telecommunications and Radio Engineering. – Volume 74, Issue 1. – Begel House Inc. – 2015. – P. 61-78.

40. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New Technique for Hiding Data in Cover Images Using Adaptively Generated Pseudorandom Sequences». CEUR Workshop Proceedings Volume 2732, 2020, Pages 214-227.

41. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макрофотографій». IV міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

42. О.А. Смірнов, П.С. Усік, «Дослідження перспектив використання технологічних рішень в мережах 5G» у Кібербезпека та інформаційні технології: монографія. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.

43. О.А.Смірнов, Т.В.Смірнова, Л.І. Поліщук, К.О. Буравченко, А.О.Макевнін, «Дослідження хмарних технологій як сервісів», Кібербезпека: освіта, наука, техніка. № 3(7). С. 43-62. 2020.

44. Смірнов О.А., Дреєва Г.М., Дреєв О.М., Смірнова Т.В. «Фрактальний аналіз генератора самоподібного трафіку на основі ланцюга Маркова». Центральнотураїнський науковий вісник. Технічні науки. № 2(33). с. 161-172, 2019.

45. О. Смірнов, Є. Деменко, О. Онікійчук, А. Арищенко, Л. Горбачова, «Формування псевдовипадкових послідовностей для приховування даних в зображеннях» Комп'ютерні науки та кібербезпека. № 4. С. 30-37. 2019.

46. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В. Поліщук Л.І. Проектування комп'ютерних систем та мереж. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2019. – 264 с.

47. Smirnov, O., Kuznetsov, A., Kuznetsova., K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan

D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).

48. Смірнов О.А., Дреєва Г.М. Метод генерування фрактального трафіку за допомогою моделі генератора на графі. Монографія: Інформаційна безпека та інформаційні технології : монографія / за заг. ред. В. С. Пономаренка. – Х. : Вид. Рожко С.Г. 2019. С. 123-139

49. Дреєва Г.М., Смірнов О.А., Дреєв О.М. Метод генерування фрактальноподібної числової послідовності на основі скінченного автомату для моделювання трафіку у мережі. Центральнотуристський науковий вісник. Технічні науки. № 1(32). с. 173-183, 2019.

50. Смірнов О.А., Кавун С.В., Коваленко О.В., Дреєв О.М. Мережні інформаційні технології. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 159 с.

51. Смірнов О.А., Смірнов С.А. Дідик А.К., Дреєв О.М. Моделі системи нейромережових експертів безпечної маршрутизації у хмарних антивірусних системах. Збірник наукових праць "Системи обробки інформації". - Випуск 3 (140). - Х.: ХУПС - 2016. - С. 36-39.

52. Смірнов О.А., Кавун С.В., Коваленко О.В., Доренський О.П., Дреєв О.М., Вялкова В.І. Комп'ютерні мережі. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 233 с.

53. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник наукових праць "Системи обробки інформації". - Випуск 2 (118). т.2. - Х.: ХУПС - 2014. - С. 64-67

54. Смірнов О.А., Дреєв О.М. Порівняння бітових щільностей при використанні різних методів кодування інформації. Збірник тез VI міжнародної науково-практичної конференції “Проблеми та перспективи розвитку ІТ-індустрії”. м. Харків. 17-18 квітня 2014р. – Харків: ХНЄУ. - 2014. - С. 240.

55. Смірнов О.А., Коваленко О.В., Кожанова А.С., Лєвошко О.Л., Константинова Л.В. Основи системного програмування. Навчальний посібник. – Кіровоград: КНТУ 2013. – 257с.

56. Смірнов О.А., Дреєв О.М., Дорєнський О.П. «Дослідження впливу ступеня стиснення зображень на оперативність їх доставки у телекомунікаційній системі. Збірник наукових праць "Системи обробки інформації". – Випуск 8(115). – Х.: ХУПС – 2013. – С. 234-239.

57. Смірнов О.А., Дорєнський О.П., Дреєв О.М. Аналіз процесів стиснення та відновлення зображень на основі цифрових методів. Наука і техніка Повітряних сил Збройних Сил України. – Випуск 3(12). – Х.: ХУПС. – 2013. – С.122-127.

58. Смірнов О.А., Мелешко Є.В., Семенов С.Г. Методи та засоби обробки сигналів і даних в інформаційних системах. Навчальний посібник. – Кіровоград: КНТУ 2012. – 250 с.

59. Смірнов О.А., Євсєєв С.П., Жукарев В.Ю., Король О.Г., Сорокін В.Є., Мелешко Є.В. Технології і стандарти комп'ютерних мереж. – Кіровоград: КНТУ 2012. – 454 с

					ВКРБ-123.26.0003.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92