

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 віртуалізованої
комунікаційної платформи OpenTouch Enterprise Cloud”

Виконав здобувач вищої освіти
IV курсу, групи КІ-22-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Давиденко Ю.М.
«___» _____ 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

Давиденку Юрію Миколайовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud*

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 віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud*

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

Давиденко Ю.М.
(прізвище та ініціали)

АНОТАЦІЯ

Давиденко Ю.М. Програмне забезпечення системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud. 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 віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

Метою розробки є програмне забезпечення системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

Результат роботи – програмна реалізація системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

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, OpenTouch Enterprise Cloud

ABSTRACT

Davydenko Yu.M. Software for the OpenTouch Enterprise Cloud virtualized communication platform system. 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 that is intended for the OpenTouch Enterprise Cloud virtualized communication platform system.

The purpose of the development is the software for the OpenTouch Enterprise Cloud virtualized communication platform system.

The result of the work is the software implementation of the OpenTouch Enterprise Cloud virtualized communication platform system.

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 PCs with Windows 10/11.

The program is developed in the Python environment.

Keywords: computer engineering, OpenTouch Enterprise Cloud

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	5
1.1 Призначення системи.....	5
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.....	24
2.3 Розгорнута постановка завдання	27
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	29
3.1 Опис функціонування системи	29
3.2 Розробка структурної схеми.....	33
3.3 Розробка функціональної схеми	35
3.4 Розробка діаграми процесів.....	40
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	42
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	42
4.2 Захист розробленого програмного забезпечення.....	65
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	69
6 ОСНОВНІ ВИСНОВКИ.....	74
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	76

					ВКРБ-123.26.0040.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	Програмне забезпечення системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud	Лім.	Аркуш	Аркушів
Розроб.	Давиденко Ю.М.					Б	1	83
Перев.	Буравченко К.О.							
Н.контр.	Коваленко А.С.					ЦНТУ КІ-22-2		
Затв.	Смірнов О.А.							

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

АТМ	—	автоматична телефонна мережа
БД	—	база даних
БЗ	—	база знань
КОМ	—	банківські обчислювальні мережі
ЛОМ	—	локальні обчислювальні мережі
МА	—	мережне адміністрування
ПЗ	—	програмне забезпечення
СМ	—	система моніторингу
СУБД	—	система управління базою даних
ТфОП	—	телефонний оператор
СА	—	Call Agent, контролер сигналізацій
ІР	—	Internet Protokol
МІВ	—	база даних інформації керування
MG	—	транспортний шлюз
MGC	—	контролером шлюзу
MGCP	—	протокол керування шлюзами
RSVP	—	Resource Reservation Protocol
RTP	—	транспортний протокол реального часу
RTSP	—	протокол передачі потокової інформації в реальному часі
SDP	—	протокол опису параметрів зв'язку
SIP	—	протокол ініціювання сеансів
SNMP	—	простий протокол керування мережею
SS	—	програмний контролер
WINS	—	служба зіставлення netbios-імен комп'ютерів з ір-адресами вузлів
VoIP	—	Voice over IP

ВСТУП

Актуальність теми. Віртуалізована комунікаційна платформа OpenTouch Enterprise Cloud (OTEC), що надає телефонію, уніфіковані комунікації й застосунки для контакт-центрів. OTEC підтримує різні сценарії розгортання – у приватній або гібридній хмарі – і може інтегруватися з наявними комунікаційними системами.

Більшість виробників телекомунікаційного устаткування надають кошти Unified Communication (UC), включаючи такі сервіси як чат, телефонний дзвінок, голосову пошту, відео/аудіо конференції. З розвитком різного виду гаджетів таких, як планшети й смартфони, акцент на їхнє використання усе більше ставати затребуваним у сполученні з UC. Підтримка цих пристроїв і набір можливостей з їхнім використанням, найімовірніше стане головним «бойовищем» виробників. Однак, стаціонарна відеоконференція й можливістю спільного використання робочого стола комп'ютера, тією самою мірою затребуваний застосунок UC. Ще один істотний вектор у розвитку UC – різні хмарні застосунки включаються по ліцензіях або на базі іншого типу полісів, а так само гнучка міграція між хмарним і внутрікорпоративним використанням. Імовірно, у швидкому майбутньому ми побачимо сполучення таких продуктів як UC і Software as a Service (SAAS), пропонованих з однієї сторони виробниками устаткування й з іншої сторони постачальниками послуг.

Системи UC не повинні бути замкнутими самі на себе, навіть якщо виробник надає весь спектр рішень сам. Вони повинні підтримувати інтеграцію із зовнішнім миром за допомогою найпоширеніших протоколів, таких як SIP, XMPP і ін. Ще однією ключовою особливістю успішної системи UC є її інтуїтивність у використанні. На зорі мобільних телефонів, компанія Nokia завоювала своїх клієнтів саме простотою й очевидністю використання різних функцій телефону. Те ж можна сьогодні сказати й із приводу UC.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

Мета й завдання дослідження. Метою роботи є програмне забезпечення системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

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

– Огляд існуючих систем віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

– Дослідження системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

– Програмна реалізація системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

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

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Як було сказано вище, віртуалізована комунікаційна платформа OpenTouch Enterprise Cloud (OTEC), що надає телефонію, уніфіковані комунікації й застосунки для контакт-центрів. OTEC підтримує різні сценарії розгортання – у приватній або гібридній хмарі, – і може інтегруватися з наявними комунікаційними системами. Відповідно призначення системи – це реалізація IP-телефонії.

IP телефонія працює в такий спосіб. Голос у комп'ютері (або в IP-телефоні) перетворюється в цифровий сигнал і передається на комп'ютер (IP-телефон) вашого співрозмовника. Там він перетворюється в аналоговий сигнал і відтворюється динаміком.

У випадку, якщо дзвінок надходить на звичайний телефон, сигнал проходить через IP-шлюз, що перетворює цифровий сигнал (VoIP) в аналоговий. Те ж саме відбувається при дзвінках через IP-телефонію на мобільні телефони.

Для передачі голосу через IP-мережу, він звичайно стискується спеціальною програмою – кодеком. Це робиться для збільшення швидкості передачі, і, отже, для підвищення якості зв'язку. Крім того, з метою безпеки сигнал може шифруватися (знову ж кодеком). Тому зв'язок між двома VoIP терміналами можлива тільки якщо вони мають сумісні кодеки.

Крім того, для того, щоб два VoIP-термінали могли знайти один одного в мережі (інтернеті) і почати сеанс зв'язку, вони обоє повинні підтримувати однаковий протокол комутації. Найпоширенішим протоколом для цієї мети є SIP.

SIP (Session Initiation Protocol) – це найпоширеніший протокол IP-телефонії. Пристрої й програми для IP-телефонії, що підтримують протокол SIP, утворюють SIP-мережу. У цій мережі кожному абонентові привласнюється

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

унікальна SIP адреса, подібний імейлу. Формат адреси "sip:user@host.com". Зайшовши в SIP-мережу з будь-якого комп'ютера або пристрою, незалежно від географічного положення, ви можете дзвонити й приймати дзвінки на свій віртуальний номер.

Найбільш популярні альтернативні протоколи – Skype (закритий протокол, використовуваний тільки в мережі Skype) і H.323 (наприклад, по цьому протоколі працює програма NetMeeting)

1.2 Область застосування

Областю застосування є Call Center. У загальному контексті, концепція побудови Call Center / Contact Center перебуває на стику технології керування взаєминами із клієнтами (CRM) і технології комп'ютерно-телефонної інтеграції (CTI) і націлена на забезпечення персоналізованого обслуговування клієнтів на найвищому рівні незалежно від місця й способу контакту. У розрізі інформаційних технологій:

- Call Center – це центр обробки телефонних викликів, принцип побудови якого ґрунтується на маршрутизації викликів агентів за певними правилами,.

- Contact Center не обмежує клієнтів у способах одержання інформації й фіксує повідомлення, що прийшли по будь-яких каналах зв'язку.

Основні напрямки розвитку галузі клієнтського сервісу

- Переміщення в хмари: компанії, що вперше застосували хмарні технології, всерйоз почали перехід в 2012 р. – на два роки раніше, ніж це передвіщалося. Тепер же, бізнес всіх розмірів у всіх галузях приймає хмари як стандарт.

- Самообслуговування в контакт-центрах: людям набридли погані IVR-системи – хіба можна їх за це винити? Call-центри будуть використовувати більші дані й інтелектуальну аналітику для кращої організації обробки викликів.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

Це допоможе встановити більше зроблену модель самообслуговування, орієнтовану на клієнта.

– Ріст мобільності й мобільних застосунків: обслуговування клієнтів через персональні пристрої розширюють можливості call-центру. Кількість бізнес-застосунків буде збільшуватися, виводячи мобільні комунікації на крос-канальний рівень.

– Обслуговування в соцмережах: компанії впритул займаються цим напрямком, контактуючи все з більшою кількістю клієнтів по каналі, що воліється ними.

– Збільшення обсягів більших даних: додавання інформації, такий як номери мобільних телефонів і каналів обміну, що воліються, повідомленнями, до наявним в call-центрах даним будуть стимулювати й оптимізувати клієнтський сервіс.

– Обслуговування по всіх каналах: збереження контексту взаємодії із клієнтами в різних каналах у доступному й персоналізованому виді, і також його проактивне використання, переведе клієнтський сервіс із розряду мультиканального в розряд «всеканального».

– Злиття й поглинання: цього року продовжиться активність по об'єднанню компаній, що працюють на ринку клієнтського сервісу, і, відповідно, піде вдосконалення технологій обслуговування споживачів.

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

					ВКРБ-123.26.0040.00.00.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

AaStra Technologies

BluStar – продукт AaStra об'єднуючі різні пристрої від персонального комп'ютера до планшетів і смартфонів у систему UC.MX-ONE – флагманський продукт лінійки підтримуючий аудіо й відео зв'язок. Так само компанія пропонує продукти AaStra 5000 і AaStra Clearspan. BluStar підтримує SIP/SIMPLE, але ще не підтримує XMPP для інтеграції із зовнішніми пристроями. Інвестиції в дослідження й науково-технічні розробки в AaStra нижче, ніж у його конкурентів, що найімовірніше приведе до подальшого відставання. У цих умовах, Європейська криза не допомагає компанії, де її найбільший ринок.

Таблиця 2.1 – Зведений опис системи BluStar

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	MX-ONE
Конференц-зв'язок	До 8мі учасників конференції з функціоналом MX-ONE, доповнене модулем InConference (частина продукту CMG), BluStar конференц-зв'язок
Обмін повідомленнями	OneBox
ІМ і Присутність	OneBox
Клієнти	BluStar клієнт (мобільні й стаціонарний) і AaStra Mobile Client, AaStra BluStar клієнти для iPhone і iPad

Продовження таблиці 2.1

Сумісність і інтеграція	MX-ONE підтримує стандартні протоколи зв'язку, такі як SIP, CSTA1/TSAPI/TAPI, CSTA3/XML/ASN1, GICI, TR87, і має SDK. Web сервіси й сервіс-орієнтована архітектура (SOA)
Адміністрування	MX-ONE Система керування
Хмара	MX-ONE
Підтримка мобільності	AaStra BluStar клієнти for iPhone and iPad and AaStra Mobile Client (AMC). The AMC+ 2.0 client supports Symbian (3, 5, Anna, Belle), Android, 2.2, 2.3, 4.xBlackBerry/RIM, 5.0 and 6.0, 7.0, 7.1, так само офіційно підтримується Samsung Galaxy tablets, так само як і iOS 4.x and 5.x, 6.x for iPhone. BluStar client for iPad and iPhone підтримують голос і крапка-крапка відеоконференцію через Wi-Fi і LTE з'єднання (3G підтримка у випадку доступності смуги). BluStar підтримує iPad2/3/4, так само як і iPad mini з аудіо/відео дзвінками.

Alcatel-Lucent

Система OpenTouch від Alcatel-Lucent містить у собі набір продуктів для забезпечення комплексного рішення UC. Система може поєднувати різні канали зв'язку й ряд пристроїв, включаючи смартфони й планшети, а також свої власні асортименти настільних пристроїв. У рішенні є перемикання однієї й тої ж сесії між різними пристроями, не перериваючи розмови, що робить продуктий досить привабливим. У системі є інтеграція з Twitter і розмови через Ipad. Компанія фокусує свої рішення на корпоративному бізнесі. Однак, у силу зміни керівництва в компанії, повинні бути підтверджені раніше обрані напрямки розвитку.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

Таблиця 2.2 – Зведений опис системи OpenTouch

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	OmniPCX Enterprise (OXE), OpenTouch Communications Suite
Конференц-зв'язок	OpenTouch Conversation for iPad (OTC for iPad), OpenTouch Video Store (OTVS), OmniTouch 8600 My Instant Communicator (HD Video), LifeSize full range of endpoints, LifeSize Video Center, OpenTouch Conversation for iPad, OmniTouch 8400 Instant Communications Suite (ICS) Communications Suite
Обмін повідомленнями	OpenTouch OmniTouch 8400 Instant Communications Suite OmniTouch 8440 Обмін повідомленнями Software OmniTouch 8450 Fax Software
ІМ і Присутність	OpenTouch OpenTouch Conversation for iPad OmniTouch 8660 My Instant Communicator OmniTouch 8660 My Teamwork
Клієнти	OpenTouch Conversation for iPadOmniTouch 8082 My IC Phone OpenTouch 8000 series ^[1] 40x8 Series, enterprise IP terminals range
Сумісність і інтеграція	Alcatel-Lucent provides standard APIs like CSTA, TAPI and TSAPI, as well as proprietary APIs: Based on XML (OmniTouch 8460 Advanced Communications Server-ACS) Based on SOAP (OmniTouch 8410 Web Services - ICS for OmniPCX Enterprise and Open Touch) REST (OpenTouch -OTC)

Продовження таблиці 2.2

Адміністрування	Alcatel-Lucent OmniVista 8770 Alcatel-Lucent VitalSuite
Хмара	Alcatel-Lucent OpenTouch Enterprise Cloud Solution Alcatel-Lucent OpenTouch Communications Suite Alcatel-Lucent OmniPCX Enterprise (OXE)
Підтримка мобільності	OpenTouch Conversation for iPad OmniTouch 8600 My Instant Communicator (MIC)

Avaya

Система Aura компанії Avaya разом з модулем конференц-зв'язку й інтеграцією з Radvision для аудіо/відео конференції доступна через Avaya Flare Experience. Відео частина системи оснащена Scopia Elite 6000 і інтегрована з Cisco TelePresence, так само як і з LifeSize і Polycom. Доступність застосунків чата й електронної пошти стало простіше із включенням їх у базову ліцензію Aura Foundation. Рішення компанії спрямовані на подальшу інтеграцію з іншими виробниками, однак є ще не мало проблем на рівні інтеграції із клієнтським устаткуванням. Мобільні застосунки у великому ступені спрямовані на стільникових операторів і потрібно прикласти сили для інтеграції з мобільними рішеннями по мережі VoWLAN.

Таблиця 2.3 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	Avaya Aura, Communication Manager and Communications Server 1000
Конференц-зв'язок	Avaya Aura Конференц-зв'язок, Avaya Radvision Scopia, AvayaLive Engage, Avaya One Touch Video, Avaya Desktop Video Device, Avaya Flare and Avaya one- X клієнти

Продовження таблиці 2.3

Обмін повідомленнями	Avaya Aura Обмін повідомленнями
ІМ і Присутність	Avaya Aura Presence Services, Avaya Flare, Avaya one-X, Avaya Client Applications (for integrations with Microsoft Lync and IBM Sametime)
Клієнти	Avaya Flare, Avaya one-X, Avaya Client Applications
Сумісність і інтеграція	Avaya Agile Communication Environment (Avaya ACE), Avaya Aura Collaboration Environment
Адміністрування	Avaya Aura System Manager, Avaya eVident
Хмара	AvayaLive Engage, AvayaLive Connect
Підтримка мобільності	Avaya Flare Experience (for iPad), Avaya one-X Mobile (for iOS, Android, BlackBerry), Avaya Scopia Mobile (for iOS and Android)

Huawei

eSpace від компанії Huawei система UC, що увібрала в себе кращі компоненти Internet Protocol Multimedia Subsystem (IMS). Система надає розширювану UC операторського класу. Вона підтримує iPad і Android планшети, так само як і власні MediaPad пристрою. Серія U шлюзів дозволяє eSpace підключатися як до малих, так і більшим середовищам. У добавок компанія представила пристрій TI30 – невеликий відеоадаптер для кімнатних рішень.

Таблиця 2.4 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	eSpace U1900 series IP PBX
Конференц-зв'язок	eSpace Meeting and Telepresence
Обмін повідомленнями	eSpace UMS, eSpace UC інтегрована з Microsoft Exchange, Microsoft Outlook and IBM Lotus Notes, iPhone Mail/Calendar

Продовження таблиці 2.4

ІМ і Присутність	eSpace PGM (Presence/Group/Обмін повідомленнями) supports SIP/SIMPLE
Клієнти	eSpace Desktop, eSpace Mobile Client підтримує iPhone і Android phone; eSpace HD Mobile Client підтримує iPad і Android tablet (e.g., Huawei MediaPad series). eSpace Web client практично всі браузери (e.g., Microsoft IE, Firefox, Safari, Chrome and Opera)
Сумісність і інтеграція	eSpace OBG – шлюзи UC, для підтримки сумісності eSpace UC і сторонніх застосунків через Web інтерфейс (SOAP, Parlay)
Адміністрування	eSpace EMS
Хмара	eSpace hosted UC solutions
Підтримка мобільності	eSpace Mobile Client for iPhone and Android phone; eSpace HD Mobile Client for iPad and Android tablet (MediaPad) eSpace Mobile software development kit (SDK)

IBM

Продукти Sametime і Sametime Unified Telephony (SUT) від IBM є флагманами UC компанії. Sametime надає власні рішення для аудіо й відео комутації, але використовує клієнтське устаткування конкурентів для телефонії й присутності. Не дивлячись на свою славу історію в сфері телекомунікації, сьогоднішні рішення компанії навряд чи задовольняють вимогам ринку. Скоріше рішення від IBM можуть претендувати на об'єднуючу ланку устаткування конкурентів, ніж на рішення UC.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

Таблиця 2.6 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	IBM Sametime Unified Telephony (SUT), IBM Sametime Unified Telephony Lite (SUT Lite), IBM Sametime
Конференц-зв'язок	IBM Sametime
Обмін повідомленнями	IBM Unified Обмін повідомленнями for WebSphere Voice Response, IBM Lotus Notes
ІМ і Присутність	IBM Sametime
Клієнти	Sametime Connect rich client (Windows, Mac and Linux), IBM Sametime for smartphones (Android, iOS, BlackBerry, Symbian and Windows Mobile), IBM Sametime for tablet (Android 2.2+, iPad)
Сумісність і інтеграція	Sametime SDK, Sametime Connect plug-ins, IBM Sametime Meetings Remote Client SDK
Адміністрування	IBM Sametime/SUT, IBM Tivoli Monitoring Agent for IBM Sametime, Meetings Statistics Utility and Configuration Validator tool
Хмара	IBM Sametime and SUT
Підтримка мобільності	IBM Sametime (IM and meetings) for smartphones (Android, iOS, BlackBerry, Symbian and Windows Mobile), IBM Sametime for tablet (Android 2.2+, iPad), SUT Dialer for iOS and Android

Interactive Intelligence

Customer Interaction Center 4.0 від Interactive Intelligence це платформа для надання голосових сервісів і UC рішень. Система добре інтегрована із продуктом Lync від Microsoft і з Sametime від IBM для підтримки текстових повідомлень, присутності, відео й WEB конференцій. Так само добре представлена підтримка

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

мобільних пристроїв, як планшетів, так і смартфонів. Основний ринок системи в контакт-центрах, як локальних, так і розподілених.

Таблиця 2.7 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	Customer Interaction Center
Конференц-зв'язок	Customer Interaction Center. Web конференція й стаціонарне відео представлено через інтеграцію з Microsoft Lync або IBM Sametime. Обидві інтеграції щільно убудовані в настільний клієнт CIC's.
Обмін повідомленнями	Customer Interaction Center.
ІМ і Присутність	Customer Interaction Center підтримує інтеграцію з IBM SameTime і Microsoft Lync
Клієнти	Interaction Client .NET Edition – обслуговує телефонні дзвінки, ACD, присутність, конференцію, ІМ, каталоги, історію дзвінків, запис розмов, VM, fax, screen pop, WFM та ін. Може бути убудована в сторонні застосунки. Interaction Client Browser Edition – WEB версія.NET клієнта. Interaction Client Mobile Browser Edition – браузерний клієнт для будь-якого мобільного пристрою (iPhone, iPad, Android, etc.) підтримує присутність, пошук по групах, доступ до корпоративної голосової пошти й історії дзвінків.

Продовження таблиці 2.7

Сумісність і інтеграція	CIC підтримує інтеграцію через Web Services (SOAP, WSDL, XML, etc.) для загального використання даних між застосунками. ICESLib (API) – надає клієнтам можливість вбудовувати функції поточного Interaction Client .NET Edition (presence management, call control, contact center applications, directory lookups, extension dialing, outbound dialing, agent scripting, і ін.) безпосередньо в застосунки сторонніх розроблювачів.
Адміністрування	Customer Interaction Center
Хмара	
Підтримка мобільності	Всі наступні застосунки UC можуть бути запуснені на клієнтах IOS, Android and Windows 8 mobile devices: Interaction Client .NET Edition, Interaction Client Browser Edition, Interaction Client Mobile Browser Edition.

Mitel

Mitel перейменував своє рішення UC у два продукти: MiVoice Business – система керування дзвінками й прикінцевим устаткуванням (раніше продукти називалися Mitel Communications Director, UC360 і Mitel Desktop). MiCollab – вся інша функціональність UC для спільної роботи, включаючи миттєві повідомлення, аудіо/відео конференції, присутність і чат (раніше ці компоненти називалися Mitel Applications Suite, Unified Communicator Advanced, Mitel Collaboration Advanced, NuPoint UM). Обидва продукти можуть бути розгорнуті як на локальній, так і на хмарній основі. Mitel так само має своя приватна хмара в Північній Америці. Це забезпечує той самий функціонал у всіх установках і гнучкій комерційній моделі (CAPEX/OPEX). Використання єдиного програмного

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

забезпечення дозволяє організаціям розгортати деякі послуги локально й доповнювати функціонал через хмарні сервіси. Система UC від Mitel так само може інтегрувати рішення інших виробників таких, як Google, Microsoft, VMware і має інтеграцію із хмарним рішенням salesforce.com.

Таблиця 2.9 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	MiVoice Business
Конференц-зв'язок	MiCollab MiVoice Conference Unit and MiVoice Video Unit (formerly UC360), MiCollab Client and Server
Обмін повідомленнями	MiCollab
ІМ і Присутність	MiCollab Client supporting XMPP and SIP/SIMPLE extensions
Клієнти	MiCollab Client for desktop, mobile devices (iOS, Android and BlackBerry) and as a Web client; MiCollab Client
Сумісність і інтеграція	Mitel Open Integration Gateway (OIG) надає широкий спектр інтерфейсів API для зв'язку Mitel's UC з застосунками третіх сторін.
Адміністрування	Embedded System Management (MiVoice/MiCollab), and Enterprise Manager
Хмара	Mitel AnyWare, MiVoice, Mitel Multi-Instance Communications Director
Підтримка мобільності	MiCollab Client supports BlackBerry v4.5, v4.6, v4.7, v5.0, v6.0, v7.0, Android 1.7-4.0, iOS 4.x-6.x. MiCollab Client supports Tablets with Android 1.7, 2.x, iOS 4.0 and 5.0. MiCollab Mobile Portal supports the following Tablet browsers: BlackBerry PlayBook OS 1.0, 2.0, Microsoft Windows 7.0, Apple iPad 4.x, 5.0, Android 1.x, 2.x, 3.x, 4.0

NEC

Univerge 3C система UC від NEC являє собою набір модулів, інтегрованих між собою й що поставляються як одне рішення. Система легко встановлюється, проста в керуванні й підтримці. Вона позиціонується як рішення бізнес класу й може бути розгорнута на стандартних апаратних пристроях. При цьому система добре масштабована й може працювати в локальному, хмарному й змішаному виконанні. Сама система зв'язку орієнтована на користувача, а не на пристрій або застосунок і такий підхід безумовно дає перевага у використанні. Система UC інтегрується з іншими продуктами Univerge, що дає можливість міграції до UC і є комерційно ефективним.

Таблиця 2.10 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	Univerge 3C Unified Communications and Collaboration
Конференц-зв'язок	Univerge 3C Unified Communications and Collaboration
Обмін повідомленнями	Univerge 3C Unified Communications and Collaboration
ІМ і Присутність	Univerge 3C Unified Communications and Collaboration, support for XMPP
Клієнти	Univerge UC Client (Windows), Univerge 3C Mobile Client (Windows, iOS, Android), Univerge 3C Collaboration Client for iPad and Android 4.0+
Сумісність і інтеграція	Univerge 3C Unified Communications and Collaboration. Набір сервісів UCC доступний через APIs на базі Web Service Definition Language (WSDL) разом з Simple Object Access Protocol (SOAP) і Extensible Markup Language (XML)

Продовження таблиці 2.10

Адміністрування	Univerge 3C Unified Communications and Collaboration
Хмара	Univerge 3C Unified Communications and Collaboration
Підтримка мобільності	Univerge 3C Mobile Client for iPhone, Android, iPad and Android 4.0+ tablets

ShoreTel

ShoreTel 14 система UC від фірми ShoreTel призначена для централізованих і розподілених мереж. Система добре відома й має позитивні відкликання клієнтів, за рахунок своєї простоти в установці, керуванні й використанні. Миттєві повідомлення, конференц-зв'язок, спільне використання застосунків здійснюються модулем Service Appliance 400, що працює окремо від комутуючого устаткування, але перебуває під єдиною системою WEB керування. ShoreTel Communicator надає користувальницький інтерфейс із різних пристроїв, а застосунок ShoreTel Mobility 7 додає інтеграцію зі смартфонами й планшетами. ShoreTel Sky – хмарне рішення компанії засноване на придбанні в 2012 році M5 Networks. Поки ще повна інтеграція не закінчена, але цим компанія займається. Рішення націлене на середній бізнес, хоча ростуть застосування у великих компаніях. Зведені дані по системі наведені в таблиці нижче.

Таблиця 2.11 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	ShoreTel 14
Конференц-зв'язок	ShoreTel 14 Конференц-зв'язок, ShoreTel 14 Communicator – Professional Access, ShoreTel Конференц-зв'язок для Apple iOS
Обмін повідомленнями	ShoreTel 14
ІМ і Присутність	ShoreTel Service Appliance 100/400

Продовження таблиці 2.11

Клієнти	ShoreTel Communicator разом з Personal Access надає керування викликами, довідник контактів, візуальна голосова пошта, спільна робота, ІМ для ПК. ShoreTel Communicator для Mac надає керування викликами, довідник контактів і візуальний UM. ShoreTel Communicator разом з Professional Access додає відео, софтверні й інтелектуальну комутацію дзвінків. ShoreTel Mobility – VoIP через Wi-Fi і GSM клієнти – PBX сервіси, присутність, мобільні через GSM і Wi. ShoreTel Конференц-зв'язок для Apple iOS – клієнт для проведення або участь у конференціях ShoreTel.
Сумісність і інтеграція	ShoreTel 14 SDK providing an HTTP and JavaScript Object Notation (JSON)-based interface. Microsoft Outlook telephony, voice mail and calendar/конференц-зв'язок integration
Адміністрування	ShoreTel 14 Director
Хмара	Поки ShoreTel Sky рішення не інтегроване з локальними застосунками, але такі функції як ShoreTel Mobility і ShoreTel Dock доступні в обох середовищах.

Продовження таблиці 2.11

Підтримка мобільності	ShoreTel Mobility 7. Platforms підтримує Apple (iPhone 3,4,5), Android (2.3,4,4.1), BlackBerry (5.0, 7,7,1). Платформи підтримують мобільні дзвінки VoIP через Wi-Fi, VoIP через GSM і мобільний дзвінок з call-back. BlackBerry 6.0 так само підтримується, але для цієї платформи VoIP дзвінки не підтримуються в силу обмежень BlackBerry. Apple iPad/iPad mini, Samsung Galaxy Note II підтримують VoIP через WI-FI і VoIP через GSM. Apple iPhone і iPad підтримуються в ShoreTel Dock через 30-pin коннектор для прямого підключення мікрофона й динаміка або Bluetooth з'єднання.
-----------------------	--

Siemens Enterprise Communications

OpenScape рішення UC від Siemens можуть бути розгорнуті як локально, так і в хмарі. Рішення доповнене OpenScape Voice продуктом, для керування розподіленими голосовими й відео пристроями. HiPath 4000 і OpenScape Branch розширюють можливості системи в розподілених мережах. Модуль системи OpenScape Session Border Controller може підтримувати до 8000 сесій. Модуль OpenScape Mobile поєднує VoIP і UC в одному мобільному клієнті. OpenScape Web Collaboration – мобільний клієнт надає WEB і відеоконференцію для iOS і Android пристроїв. У такий спосіб рішення для аудіо/відео зв'язку є всеосяжним і має власний SDK пакет, однак слабо представлені миттєві повідомлення.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк. 21
Вим.	Арк.	№ докум.	Підпис	Дата		

Таблиця 2.12 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	OpenScape Voice HiPath 4000, OpenScape Session Border Controller, OpenScape Branch, OpenScape Office
Конференц-зв'язок	OpenScape UC Application, OpenScape Web Collaboration, OpenScape Video, OpenScape Media Server and OpenScape UC Application, OpenScape Voice, OpenScape Soft Клієнти, Managed Video Service
Обмін повідомленнями	OpenScape UC Application, OpenScape Xpressions
ІМ і Присутність	OpenScape UC Application, third-party integration though XMPP
Клієнти	UC Desktop SIP client, UC Web client, OpenScape UC Application Personal edition, OpenScape Mobile Client (iOS, Android), OpenScape Web Collaboration Mobile client (Android, iOS)
Сумісність і інтеграція	OpenScape Fusion Software Developer Toolkit suite
Адміністрування	OpenScape Common Management Platform OpenScape Deployment Service OpenScape User Management, OpenScape Accounting, OpenScape Fault Management, OpenScape Performance Management
Хмара	OpenScape Cloud Services, OpenScape Enterprise and OpenScape Business

Продовження таблиці 2.12

Підтримка мобільності	OpenScape Mobile Client. Android: v2.1 to v2.x, v3.0.1, v4.x (when available) Apple: iOS 3.x, iOS 4.x, iOS 5.x, iOS 6.x, BlackBerry: RIM OS 4.2.1, 4.3, 4.6, 5.0, 6.0, Windows Mobile: V5 Phone Edition, V6.1, V6.5, Symbian Platform (Nokia v3.1, v3.2, and v5)
-----------------------	--

Toshiba

Продукт IPedge Unified Communications фірми Toshiba є ядром системи UC, що вбудовується в загальний продукт IPedge, установлюваний на локальному сервері Strata CIX або хмарній платформі VIPedge. IPedge Meeting застосунок надає аудіо/відео й WEB конференцію. Чат і присутність надається за допомогою Call Manager, що є клієнтським додатком інтегрованим з телефонією й XMPP сервером для інформації про статус і календар, однак поки він не сполучається з Microsoft і Google. Call Manager Mobile нова розробка, що підтримує HTML5 у планшетах і смартфонах. Вектор розвитку компанії спрямований на інтеграцію рішення UC у вже встановлених серверах Strata CIX, що підтримують цифрову й аналогову телефонію, а так само зв'язок меду ними.

Таблиця 2.13 – Зведений опис системи

Критична здатність	Найменування й опис продукту / послуги
Голос і Телеком	IPedge IP Business Communication System Strata CIX IP Converged Business Communication System with IPedge Application Server
Конференц-зв'язок	IPedge Meeting
Обмін повідомленнями	IPedge Unified Обмін повідомленнями
ІМ і Присутність	IPedge Call Manager PC (Win) IPedge Call Manager Mobile (Android and iOS)

Продовження таблиці 2.13

Клієнти	Call Manager PC (Win), Call Manager Mobile (Android and iOS)
Сумісність і інтеграція	Call Manager PC
Адміністрування	Enterprise Manager for IPedge and VIPedge and Network eManager for Strata CIX
Хмара	CIX/IPedge and VIPedge Hybrid Solution
Підтримка мобільності	Call Manager PC (Win), Call Manager Mobile (Android and iOS), and IPMobility (Inbound/Outbound Twinning/Calling; Android and iOS)

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Python – високорівнева мова програмування, яку називають другою за популярністю в світі. Її використовують для розробки вебзастосунків, програмного забезпечення, машинного навчання. Python застосовують для вирішення робочих завдань у компаніях Google, Instagram, Facebook, IBM, NASA, Dropbox, Netflix та інших. Розробники цінують цю мову програмування за простоту у вивченні, ефективність та мультиплатформність.

Python – скриптова мова програмування з досить простим синтаксисом. Для розуміння достатньо порівняти принципи написання найпростішої програми, яка виводить на екран текстове повідомлення. Саме тому мова програмування Python більш доступна для новачків, а професіонали встигли адаптувати її для вирішення великої кількості завдань. Це мультиплатформне рішення, тому знання Python дає можливість працювати у різних сферах: від розробки мобільних застосунків до ігрової індустрії та штучного інтелекту.

У мови програмування динамічна типізація: є можливість передавати до функцій будь-який тип даних без попереднього вказання. Інтерпретованість дозволяє знаходити помилки у коді ще до повної збірки у робочий застосунок. При цьому Python дуже чітко дає зрозуміти, де та через що виникла помилка.

Це мова об'єктноорієнтованого програмування (ООП). Програмне забезпечення на Python оформлене у вигляді моделей, які можуть бути зібраними у пакети. Тип та структуру кожного об'єкта можна запитати під час виконання програми. Для кожного з об'єктів можна отримати всю інформацію щодо його внутрішньої структури. Окрім того:

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;
- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
- у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

Проте є декілька особливостей, які можна віднести до недоліків. Це повільність (ця мова програмування хоч і універсальна, проте повільніша за інші), велика кількість ресурсів, необхідних для роботи та «прив'язаність» до системних бібліотек.

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.
2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).

3. Ігри. Багато сучасних ігор для комп'ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.

4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.

5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.

6. Тестування (автоматизація цього процесу).

7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

Бібліотеки Python

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

Найвідоміші фреймворки для мови програмування Python

Фреймворки Python допомагають створити зручне та функціональне середовище для розробки. У них міститься набір інструментів, модулів та бібліотек, корисних для виконання конкретних завдань. Це значно полегшує роботу: наприклад, дає змогу не витратити час на розписування дій, які повторюються, а використати релевантний інструмент. Тож є можливість позбутися рутинних процесів та сконцентруватися на логіці проекту.

Серед найпопулярніших фреймворків для Python:

– Django – найстаріший та найвідоміший. Створений для реалізації великих інтерактивних проєктів;

– Pyramid – зручний у налаштуваннях, і дає можливість реалізувати складні нестандартні ідеї;

– Web2py – підходить в першу чергу для вебзастосунків і може використовуватись на будь-яких архітектурах.

Популярні Python IDE

IDE або інтегровані середовища розробки – це програмне забезпечення, яке надає розробникам необхідні інструменти для написання, редагування, тестування та налаштування коду. Для розробки на Python найчастіше використовують IDE PyCharm, IDLE, Spyder та Atom.

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 віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

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;

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.

К6ПЗ-2026

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Важливо розуміти, що УС – це середовище об'єднуючі різні сервіси (чат, голос, відео, факс і ін.) з однієї сторони й пристрої доступу (стаціонарний комп'ютер, мобільний телефон, факс машину, планшет, смартфон та ін.) з іншого боку, і передавати інформацію між ними по всіх існуючих каналах зв'язку. У який то ступеня УС це перекладач інформації між різними пристроями що використовує різні середовища. Більшість виробників збирають у рішенні свої пристрої зв'язку, часто придбані на основі OEM угоди в третіх компаній, однак необхідність співіснувати з конкурентним устаткуванням має важливе значення для багатьох організацій. Звичайно, використання мультивендорного рішення покладає на плечі інтегратора більші складності по сумісності устаткування, але це часто дає економічні вигоди, а іноді й просто необхідно в силу прихильностей користувача і його звичок у використанні прикінцевого устаткування. Важливим параметром при виборі виробника є його стійкість на ринку. При цьому, якщо компанія займається тільки цією сферою, і має хоча б 5 років позитивної репутації (це можна побачити по росту продажів і/або прибутку) їй варто віддавати перевагу при всіх рівних. Якщо ж УС є побічним продуктом, і виступає як каталізатор продажів основного товару, то варто дуже уважно поставитися до такого рішення. Істотною особливістю УС є те, що цей продукт завжди повинен бути кастомізован під конкретного користувача. Це починається із завдання внутрішніх номерів користувачів, інтеграцією з їхніми базами даних телефонів і електронних пошт, самих сервісів, використовуваних у компанії, наприклад факси або WEB навчання, і далі до розподілу/обмеження сервісів, використовуваних різними групами співробітників. Таким чином, роль грамотного інтегратора зростає в багато разів. Тепер інтегратор не тільки той, хто

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

«все включає», але й той хто планує групи користувачів, задає їм ACL (Access Control List) і підтримує функціональність усього комплексу. Навіть якщо компанія має у своєму штаті навченого фахівця, що може створити нового користувача, буде дуже дорогим задоволенням тримати штат співробітників, що могли б реагувати на мінливі компоненти системи. А мінятися вони будуть обов'язково, адже UC і покликане до того, щоб об'єднати все для всіх. Корпоративний бізнес використовує UC для зв'язку своїх підрозділів. Але в цьому випадку, у вартість UC, як правило, не включають ціну модернізації мережі, що, як правило, необхідна для відео-конференції. У цьому випадку, буде важливо мати різні групи користувачів, що мають право використовувати внутрішню телефонію й відео-конференц-зв'язок роздільно. В останньому випадку може бути вигідно використовувати хмарні рішення із сервісом по запиті UCaaS.

Цей розділ написаний для того, щоб користувачам було легше зорієнтуватися в необхідному їм функціоналі й виборі конкретного постачальника UC по чотирьох наступних критеріях:

- Повне рішення UCS з більшим упором на телефонію.
- Повне рішення UCS з більшим упором на кооперацію.
- Базове рішення UCS інтегроване з устаткуванням конкурентів.
- Інтегроване рішення локальних і хмарних сервісів.

Який би критерій не був обраний, головним аргументом використання UC буде підвищення продуктивності праці співробітників, полегшення завдань ними виконуваних і, як наслідок, одержання конкурентної переваги в роботі. UC постійно розвиваються й всі нові сервіси включаються під це поняття. У даній статті розглядаються дев'ять найбільш важливих функцій UC:

– Телефонія, що включає в себе фіксований телефон, інтеграцію із класичної PBX, мобільні телефони, програмні клієнти, застосунки на смартфонах і відео дзвінки.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– Конференц-зв'язок, що включає в себе аудіо, відео й WEB конференцію. Конференція повинна мати віртуальні кімнати, WEB камерами й спільним використанням робочого стола комп'ютера.

– Повідомлення й чат. Інтеграція з електронною поштою й текстовими повідомленнями через уніфікований інтерфейс UC.

– Присутність. Це можливість знати статичний стан абонента, наприклад не турбувати, на зустрічі й ін., а також динамічні стан, наприклад знятий трубка, дзвонить, розмовляє. До присутності також ставиться й геолокація. Вся ця інформація повинна виходити від всіх компонентів системи UC, таких як стаціонарний і мобільний телефон, софт клієнт та ін.

– Різні клієнти зв'язків. Це здатність забезпечити обмін інформацією з різними пристроями, включаючи WEB і мобільні пристрої, а так само стаціонарні телефони, SIP телефони софтверні під різні операційні системи й нативні застосунки смартфонів. Так само для багатьох компаній підтримка WebRTC стане обов'язковою.

– Сумісність і інтеграція. Можливість інтегрувати застосунки UC з іншими продуктами, такими як CRM, поштові клієнти та ін. Це досягається підтримкою TAPI, W-TAPI і іншими засобами й інструментами. У деяких випадках засобу інтеграції можуть бути убудовані в налаштування, наприклад у випадку вхідного дзвінка з'являється попередньо настроєна WEB сторінка CRM системи, що відразу підвищить ефективність роботи кола-центра.

– Керування. Централізоване керування системою з можливістю аналізу й налаштування всіх елементів зв'язку. Система керування повинна мати автоматичне розпізнавання нових елементів на мережі, завантаженням конфігурації й відновлення на них.

– Хмарні рішення – це здатність використовувати частину функціонала локально в компанії й частину віддалено в хмарному просторі. Це також повинне дозволяти компаніям міняти систему ліцензування й використання.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

– Підтримка мобільності – це здатність інтегрувати мобільні пристрої в єдину систему, включаючи присутність, пейджер, голосову й відео зв'язок, конференції й спільне використання робочого стола стаціонарного комп'ютера.

Варіанти використання

Підприємства розвивають свою інфраструктуру поступово залежно від зростаючих потреб і відповідно до бюджету. Як наслідок, УС представлений у великому ціновому діапазоні, але, як правило, це можна звести до одному з описаних чотирьох випадків.

Повне рішення USS з більшим упором на телефонію. Використовується великими й середніми організаціями для підвищення ефективності голосової інфраструктури. Добре сполучається із прогнозованим розвитком суміжних областей ІТ.

Повне рішення USS з більшим упором на кооперацію. Так само використовують середні й великі компанії для інтеграції взаємодії персоналу й програм. Керівництво особлива увага приділяє конференц-зв'язку, факс-серверу, електронній пошті, чату і їхньої інтеграції з іншими застосунками. Інтеграція з телефонією так само корисна, але не розглядається як головне.

Базове рішення USS інтегроване з устаткуванням конкурентів. У такому рішенні присутнє бажання зберегти наявний «зоопарк» устаткування й програм, об'єднавши його в одному середовищі.

Інтегроване рішення локальних і хмарних сервісів. Рішення дозволяє компаніям використовувати частину передових сервісів не чекаючи розвитку власної інфраструктури зв'язку. Компаніям буде цікаво надійне зберігання інформації в хмарі, а в деяких випадках і захищеність її від санкцій силових структур. При цьому гнучкість переміщення між локальним і хмарним використанням досить істотна.

Виробники устаткування включені в аналіз, представлений нижче, на підставі наступних критеріїв:

– Виробник повинен мати функціональність на ринку в 4х і більше функціях УС, зі списку представленого в таблиці вище. Сама присутність на ринку може проявлятися одним із двох способів – фактична частка ринку, зайнята виробником або різні нововведення в цій сфері.

– Продукт повинен мати корпоративні впровадження з посиланнями й можливістю контакту з користувачем.

– Пропоноване рішення повинне бути закінченим і покривати всі сфери УС, навіть якщо частина пропонованих рішень узята як рішення третіх сторін.

Критична оцінка можливостей

Життєздатність – це оцінка виробника по супроводу рішення в плинні циклу його життя. Чотири основні показники включають цю оцінку: стратегія, підтримка, виконання й інвестиції. Стратегія визначає, як дане рішення вписується в загальну стратегію компанії, її маркетингові устремління й загальний напрямок бізнесу. Підтримка містить у собі технічна якість і відкритість/простоту роботи із клієнтами. Виконання відбиває структуру виробника, а так само процес продажів, маркетингу, цін і переговорів. Інвестиції визначають фінансове здоров'я виробника і його готовність розширювати даний напрямок. Кожний пункт оцінюється по 5-ти бальній шкалі.

3.2 Розробка структурної схеми

OpenTouch Enterprise Cloud прискорює цифрову трансформацію завдяки безпечному, гнучкому та простому в управлінні комунікації як послуги.

OpenTouch Enterprise Cloud – це рішення цифрової ери «комунікація як послуга», розроблене для прискорення цифрової трансформації організацій. Воно дозволяє співробітникам працювати з будь-якого місця, покращуючи безперервність бізнесу та взаємодію з клієнтами без шкоди для безпеки.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

Платформа пропонує гнучкі моделі підписки, автоматизовані операції та надійні функції управління викликами. ІТ-відділи отримують вигоду від спрощеного управління та оптимізації витрат, а співробітники насолоджуються безперебійним спілкуванням на різних пристроях. OpenTouch Enterprise Cloud інтегрується з існуючими екосистемами та пропонує варіанти хмарного або локального хостингу.

Особливості

Ключові функції включають комплексні можливості керування викликами на різних пристроях (стаціонарні телефони, DECT, WLAN-телефони); безшовну інтеграцію з існуючими екосистемами, такими як системи медичної сигналізації, захист самотніх працівників та CRM-додатки; надійне рішення для контакт-центру віддалених агентів; та гнучкі моделі підписки з опціями хмарного або локального хостингу. Платформа забезпечує адміністрування на основі ролей через веб-портал та пропонує візуальні автоматизовані оператори зв'язку для кількох клієнток, спливаючі вікна, функцію «дзвінок за кліком» та візуальну співпрацю. Платформа забезпечує високу доступність та безперервність бізнесу завдяки керуванню викликами з можливістю гарячого перемикання та резервуванню центру обробки даних.

Переваги

OpenTouch Enterprise Cloud пропонує кілька ключових переваг, зокрема покращену безпеку завдяки виділеному програмному забезпеченню та ресурсам центру обробки даних; покращений досвід клієнтів завдяки забезпеченню безперервності дзвінків та автоматизованій взаємодії; оптимізовані ІТ-операції з узгодженням витрат та зменшенням кількості завдань налаштування; а також мінімізацію ризиків простою завдяки функціям резервування та забезпечення відмовостійкості. Це дозволяє використовувати гнучкі моделі підписки, що відповідають потребам вашого бізнесу. Платформа пропонує безперешкодну інтеграцію з іншими бізнес-додатками та системами. Платформа сертифікована за стандартом ISO27001 для безпечного зв'язку.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34



Рисунок 3.1 – Структурна схема системи

3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема розробленого програмного забезпечення. Розглянемо стек протоколів H.323, які використовуються у віртуалізованій комунікаційній платформі OpenTouch Enterprise Cloud (OTEC).

Стек протоколів H.323

Стандарт H.323 визначає широкі вимоги для багатьох різних протоколів, які становлять повний стек протоколів H.323.

Стек H.323 складають 7 груп протоколів:

- керування й сигналізація;
- обробка звукових сигналів;
- обробка відеосигналів;
- конференц-зв'язок;
- передача мультимедійної інформації;

– забезпечення інформаційної безпеки;

– додаткові послуги;

1. Керування з'єднанням і сигналізація:

– H.225.0: протоколи сигналізації й пакетування мультимедійного потоку (використовує підмножину протоколу сигналізації Q.931).

– H.225.0/RAS: процедури реєстрації, допуску й стану.

– H.245: протокол керування для мультимедіа.

2. Обробка звукових сигналів:

– G.711: імпульсно-кодова модуляція тональних частот.

– G.722: кодування звукового сигналу 7 кГц в 64 кбіт/с.

– G.723.1: мовні кодери на дві швидкості передачі для організації мультимедійного зв'язку зі швидкістю передачі 5.3 і 6.3 кбіт/с.

– G.728: кодування мовних сигналів 16 кбіт/с за допомогою лінійного пророкування з кодуванням сигналу порушення з малою затримкою.

– G.729: кодування мовних сигналів 8 кбіт/с за допомогою лінійного пророкування з алгебраїчним кодуванням сигналу порушення сполученої структури.

3. Обробка відеосигналів:

– H.261: відеокодеки для аудіовізуальних послуг зі швидкістю 64 кбіт/с.

– H.263: кодування відеосигналу для передачі з малою швидкістю.

4. Конференц-зв'язок для передачі даних:

– T.120: це стек протоколів (який включає T.123, T.124, T.125) для передачі даних між прикінцевими пунктами. Він може використовуватися для різних застосунків в області спільної роботи (Collaboration Work), такий як колективне редагування растрових зображень, спільне використання застосунків і спільна організація документів. В T.120 застосовується багаторівнева архітектура, подібна моделі OSI.

5. Мультимедійна передача:

– RTP: транспортний протокол реального часу.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		36

– RTSP: протокол керування передачею в реальному часі.

6. Забезпечення безпеки:

– Н.235: забезпечення безпеки й шифрування для мультимедійних терміналів мережі Н.323.

7. Додаткові послуги:

– Н.450.1: узагальнені функції для керування додатковими послугами в Н.323.

– Н.450.2: переклад з'єднання на телефонний номер третього абонента.

– Н.450.3: переадресація виклику.

– Н.450.4: утримання виклику.

– Н.450.5: паркування виклику (park) і відповідь на виклик (pick up).

– Н.450.6: повідомлення про виклик, що надійшов, у стані розмови.

– Н.450.7: індикація повідомлення, що очікує.

– Н.450.8: служба ідентифікації імен.

– Н.450.9: служба завершення з'єднання для мереж Н.323.

Тепер розглянемо функціональні можливості системи.

З функціональної схеми ми бачимо, що основний функціональний модуль включає в себе наступні функціональні підблоки:

– IP-телефонія.

– Селективний зв'язок.

– Відеоконференція.

– Формування вхідної інформації.

– База даних абонентів банківської мережі.

– Білінг.

– Виведення статусу ПЗ.

– Довідник.

Дані, які поступають з основного функціонального модуля перетворюються згідно протоколу Н.323, після чого передаються до IP-мережі.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Робота з програмою починається з введення інформаційного вікна й 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.

Головне вікно програми 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 банківської 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 знищення.

Основний функціональний модуль віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud

ІР-телефонія

Селективний зв'язок

Відеоконференція

Формування вхідної інформації

База даних абонентів банківської мережі:

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

Біллінг

Виведення статусу ПЗ

Довідник:

- Довідкова система.
- Інструкція користувача.



Н.323

Керування з'єднанням і сигналізація: Н.225.0, Н.225.0/RAS, Н.245

Обробка звукових сигналів: G.711, G.722, G.723.1, G.728, G.729

Обробка відеосигналів: Н.261, Н.263

Конференц-зв'язок для передачі даних: Т.120, Т.123, Т.124, Т.125

Мультимедійна передача: RTP, RTCP

Забезпечення безпеки: Н.235

Додаткові послуги: Н.450

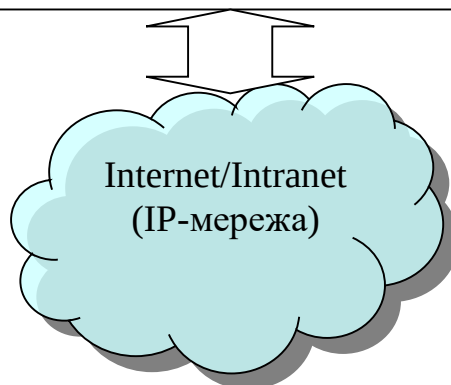


Рисунок 3.2 – Функціональна схема системи

Вим.	Арк.	№ докум.	Підпис	Дата

ВКРБ-123.26.0040.00.00.ПЗ

Арк.

39

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

Біллінгова система – автоматизована система розрахунків з абонентами за надані послуги. Керування й статистика може бути доступна з будь-якої точки мережі через Веб-інтерфейс.

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.



Рисунок 3.3 – Діаграма взаємодії процесів

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).

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

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

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-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

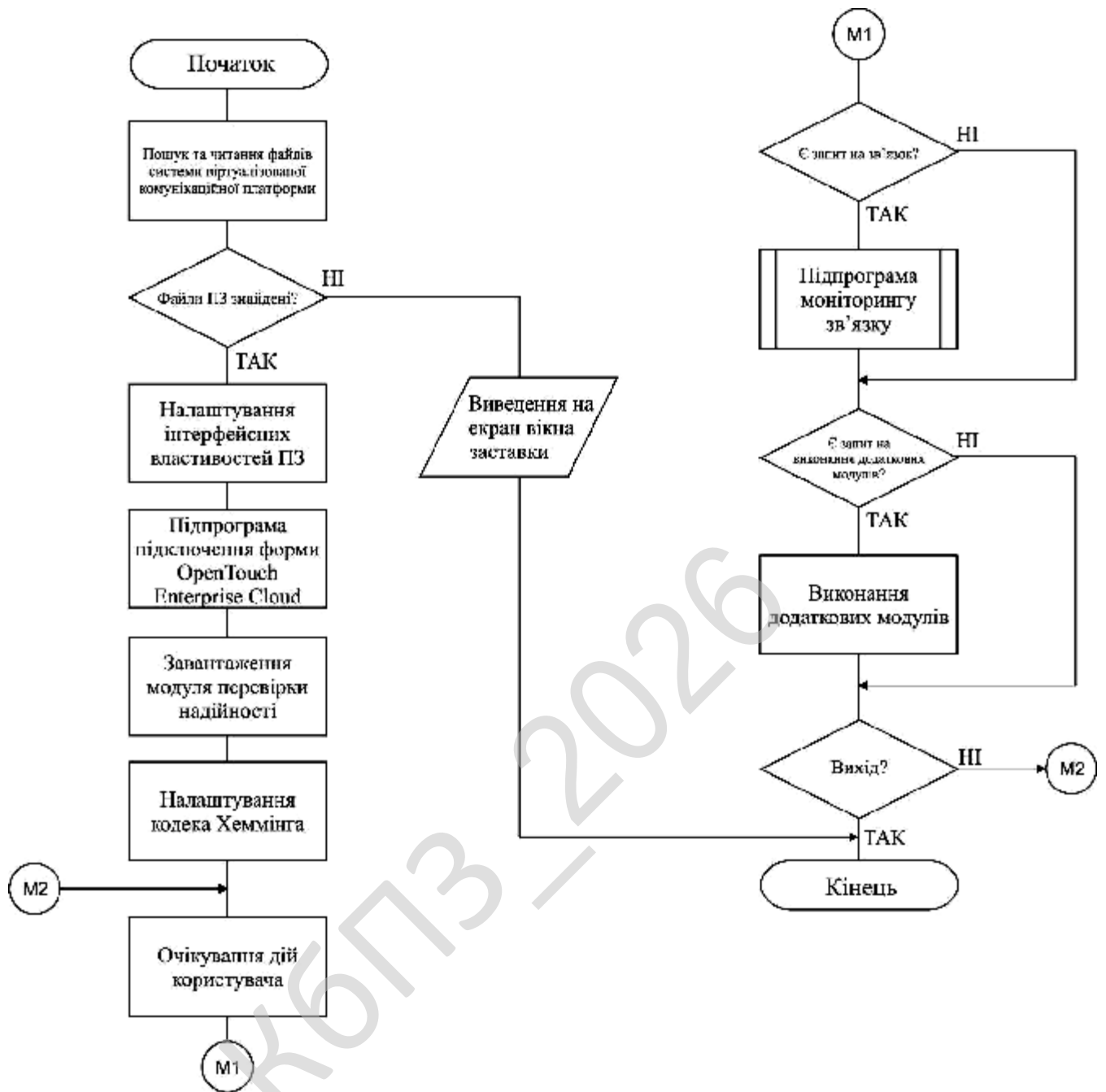


Рисунок 4.1 – Блок-схема основної програми

Було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент; Діаграма об'єктів; Діаграма розгортання.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей 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.

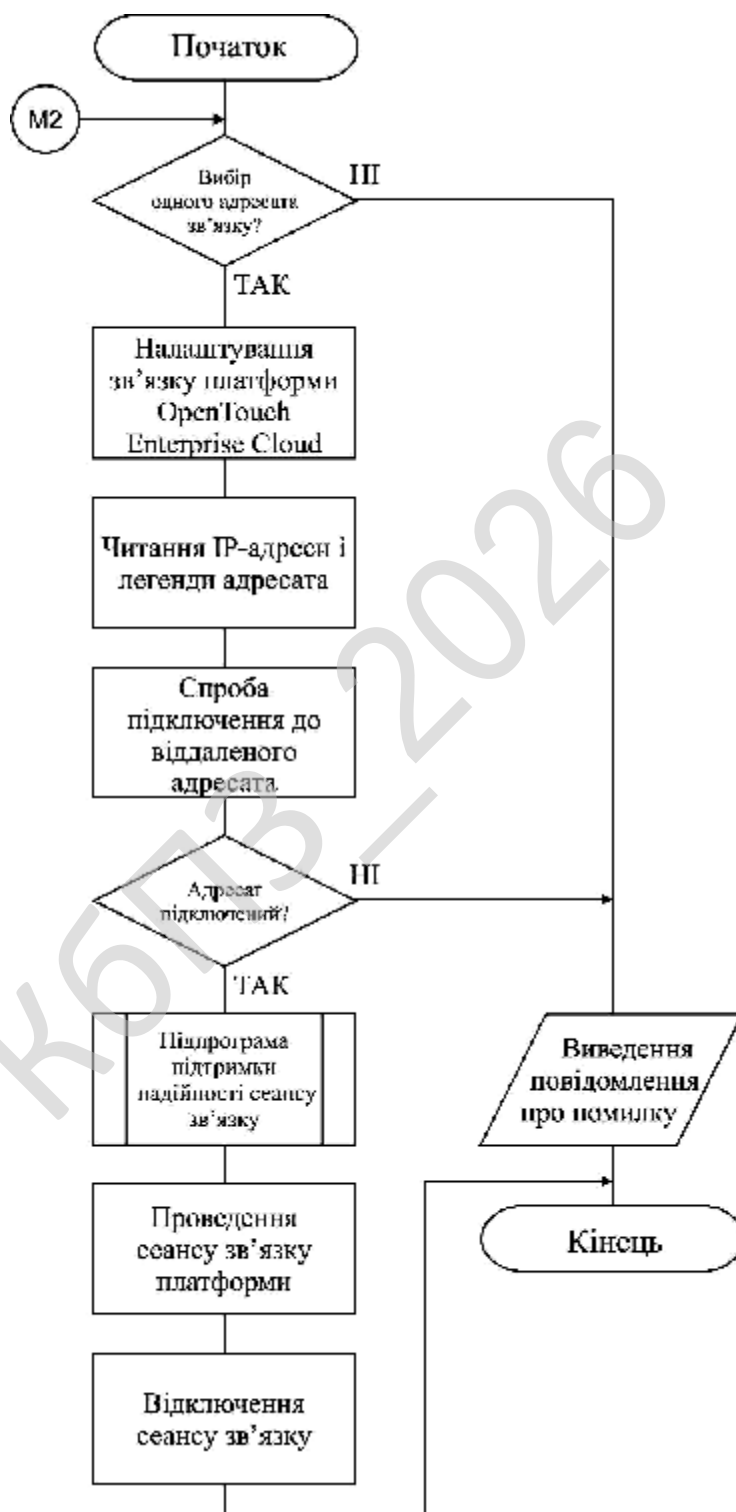


Рисунок 4.2 – Блок-схема роботи підпрограми

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.

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

Діаграма прецедентів це діаграма, на якій зображено відношення між акторами та прецедентами в системі. Також, перекладається як діаграма варіантів використання.

Діаграма прецедентів є графом, що складається з множини акторів, прецедентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами. Діаграми прецедентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (use case) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором.

При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

У мові UML є кілька стандартних видів відношень між акторами і варіантами використання:

– асоціації (association relationship);

- включення (include relationship);
- розширення (extend relationship);
- узагальнення (generalization relationship).

При цьому загальні властивості варіантів використання можуть бути представлені трьома різними способами, а саме – за допомогою відношень включення, розширення і узагальнення.

Відношення асоціації – одне з фундаментальних понять у мові UML і в тій чи іншій мірі використовується при побудові всіх графічних моделей систем у формі канонічних діаграм.

Включення (include) у мові UML – це різновид відношення залежності між базовим варіантом використання і його спеціальним випадком. При цьому відношенням залежності (dependency) є таке відношення між двома елементами моделі, при якому зміна одного елемента (незалежного) приводить до зміни іншого елемента (залежного).

Відношення розширення (extend) визначає взаємозв'язок базового варіанта використання з іншим варіантом використання, функціональна поведінка якого задіюється базовим не завжди, а тільки при виконанні додаткових умов.

Діаграма класів це статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення.

Діаграма класів, також, може містити позначення для пакетів та може містити позначення для вкладених пакетів. Також, діаграма класів може містити позначення деяких елементів поведінки, однак їх динаміка розкривається в інших типах діаграм.

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування.

На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

В UML існують наступні типи зв'язків які використовуються у діаграмі класів: Асоціації; Агрегація; Композиція.

Асоціації це якщо між двома класами визначена асоціація, то можна переміщатися від об'єктів одного класу до об'єктів іншого. Цілком припустимі випадки, коли обидва кінці асоціації відносяться до одного і того ж класу. Це означає, що з об'єктом деякого класу дозволено зв'язати інші об'єкти з того ж класу. Асоціація, що зв'язує два класи, називається бінарної. Можна, хоча це рідко буває необхідним, створювати асоціації, що зв'язують відразу кілька класів. Графічно асоціація зображується у вигляді лінії, що з'єднує клас сам з собою або з іншими класами.

Асоціації може бути присвоєно ім'я, яке описує природу відносини. Зазвичай ім'я асоціації не вказується, якщо тільки ви не хочете явно задати для неї рольові імена або у вашій моделі настільки багато асоціацій, що виникає необхідність посилатися на них і відрізняти один від одного. Ім'я буде особливо корисним, якщо між одними і тими ж класами існує кілька різних асоціацій.

Клас, що бере участь в асоціації, грає в ній деяку роль. По суті, це "обличчя", яким клас, що знаходиться на одній стороні асоціації, звернений до класу з іншого її боку. Можна явно позначити роль, яку клас грає в асоціації.

Часто при моделюванні буває важливо вказати, скільки об'єктів може бути пов'язано допомогою одного примірника асоціації. Це число називається кратністю (Multiplicity) ролі асоціації та записується або як вираз, значенням якого є діапазон значень, або в явному вигляді.

Вказуючи кратність на одному кінці асоціації, ви тим самим говорите, що на цьому кінці саме стільки об'єктів повинно відповідати кожному об'єкту на протилежному кінці. Кратність можна задати рівною одиниці (1), можна вказати діапазон: "нуль або одиниця" (0..1), "багато" (0 .. *), "одиниця або більше" (1 .. *). Дозволяється також вказувати певне число (наприклад, 3). За допомогою списку можна задати і більш складні кратності, наприклад 0. . 1, 3..4, 6 .. *, що означає "будь-яке число об'єктів, крім 2 і 5".

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

Агрегація це проста асоціація між двома класами відображає структурний відношення між рівноправними сутностями, коли обидва класу знаходяться на одному концептуальному рівні і ні один не є більш важливим, ніж інший. Але іноді доводиться моделювати відношення типу «частина/ціле», в якому один з класів має більш високий ранг (ціле) і складається з декількох менших за рангом (частин).

Ставлення такого типу називають агрегацією; воно зараховане до відносин типу «має» (з урахуванням того, що об'єкт-ціле має кілька об'єктів-частин). Агрегація є окремим випадком асоціації і зображується у вигляді простої асоціації з незафарбованим ромбом з боку «цілого».

Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

Композиція це більш суворий варіант агрегації. Відома також як агрегація за значенням.

Композиція має жорстку залежність часу існування екземплярів класу контейнера та примірників містяться класів. Якщо контейнер буде знищений, то весь його вміст буде також знищено. Графічно представляється як і агрегація, але з зафарбовані ромбиком.

Діаграма компонент в UML це діаграма, на якій відображаються компоненти, залежності та зв'язки між ними.

Діаграма компонент відображає залежності між компонентами програмного забезпечення, включаючи компоненти вихідних кодів, бінарні компоненти, та компоненти, що можуть виконуватись.

Модуль програмного забезпечення може бути представлено в якості компоненти. Деякі компоненти існують під час компіляції, деякі – під час компонування, а деякі під час роботи програми.

Діаграма компонент відображає лише структурні характеристики, для відображення окремих екземплярів компонент слід використовувати діаграму розгортання.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Компоненти об'єднуються разом використовуючи структурні зв'язки (assembly connector) щоб об'єднати інтерфейси двох компонент. Це ілюструє зв'язок типу «клієнт-сервер».

Структурна взаємодія – «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

При використанні діаграми компонент щоб показати внутрішню структуру компонента, клієнтські та серверні інтерфейси можуть утворювати пряме з'єднання з внутрішніми. Таке з'єднання називається з'єднанням делегації.

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

Діаграми об'єктів не мають власної нотації. Оскільки діаграми класів можуть відображати об'єкти, то діаграма класів, на якій відображено лише об'єкти, та не відображено класи, може вважатись діаграмою об'єктів.

Діаграма об'єктів відображає об'єкти та зв'язки в певний момент роботи програми. Об'єкти можуть містити інформацію про власні значення а не про описання. Для відображення загальних шаблонів об'єктів та зв'язків, що можуть багаторазово створюватись під час роботи програми, слід використовувати діаграму взаємодії, яка може відображати характеристики об'єктів та зв'язків. Екземпляр діаграми взаємодії створює діаграму об'єктів.

Діаграма об'єктів не відображає еволюцію системи під час роботи. Натомість, слід використовувати діаграми взаємодії з повідомленнями, або діаграми послідовності.

Діаграма розгортання (deployment diagram) це діаграма в UML, на якій відображаються обчислювальні вузли під час роботи програми, компоненти, та об'єкти, що виконуються на цих вузлах. Компоненти відповідають представленню робочих екземплярів одиниць коду. Компоненти, що не мають представлення під час роботи програми на таких діаграмах не відображаються; натомість, їх можна

відобразити на діаграмах компонент. Діаграма розгортання відображає робочі екземпляри компонент, а діаграма компонент, натомість, відображає зв'язки між типами компонент.

Архітектура клієнт-сервер є одним із архітектурних шаблонів програмного забезпечення та є домінуючою концепцією у створенні розподілених мережних програм і передбачає взаємодію та обмін даними між ними. Вона передбачає такі основні компоненти:

- набір серверів, які надають інформацію або інші послуги програмам, які звертаються до них;
- набір клієнтів, які використовують сервіси, що надаються серверами;
- мережа, яка забезпечує взаємодію між клієнтами та серверами.

Сервери є незалежними один від одного. Клієнти також функціонують паралельно і незалежно один від одного.

Немає жорсткої прив'язки клієнтів до серверів. Більш ніж типовою є ситуація, коли один сервер одночасно обробляє запити від різних клієнтів; з іншого боку, клієнт може звертатися то до одного сервера, то до іншого. Клієнти мають знати про доступні сервери, але можуть не мати жодного уявлення про існування інших клієнтів.

Дуже важливо ясно уявляти, хто або що розглядається як «клієнт». Можна говорити про клієнтський комп'ютер, з якого відбувається звернення до інших комп'ютерів. Можна говорити про клієнтське та серверне програмне забезпечення. Нарешті, можна говорити про людей, які бажають за допомогою відповідного програмного та апаратного забезпечення отримати доступ до тієї чи іншої інформації.

Загальноприйнятим є положення, що клієнти та сервери – це перш за все програмні модулі.

Найчастіше вони знаходяться на різних комп'ютерах, але бувають ситуації, коли обидві програми – і клієнтська, і серверна, фізично розміщуються на одній машині; в такій ситуації сервер часто називається локальним.

Модель клієнт-серверної взаємодії визначається перш за все розподілом обов'язків між клієнтом та сервером. Логічно можна відокремити три рівні операцій:

- рівень представлення даних, який по суті являє собою інтерфейс користувача і відповідає за представлення даних користувачеві і введення від нього керуючих команд;
- прикладний рівень, який реалізує основну логіку ПЗ і на якому здійснюється необхідна обробка інформації;
- рівень управління даними, який забезпечує зберігання даних та доступ до них.

Дворівнева клієнт-серверна архітектура передбачає взаємодію двох програмних модулів – клієнтського та серверного. В залежності від того, як між ними розподіляються наведені вище функції, розрізняють:

- модель тонкого клієнта, в рамках якої вся логіка ПЗ та управління даними зосереджена на сервері. Клієнтська програма забезпечує тільки функції рівня представлення;
- модель товстого клієнта, в якій сервер тільки керує даними, а обробка інформації та інтерфейс користувача зосереджені на стороні клієнта. Товстими клієнтами часто також називають пристрої з обмеженою потужністю: кишенькові комп'ютери, мобільні телефони та ін.

Типовим прикладом клієнт-серверної взаємодії є WWW. Існує величезна кількість веб-серверів, на яких розміщується та чи інша інформація. У найпростішому випадку ця інформація являє собою набір веб-сторінок, які можуть зберігатися на сервері у вигляді файлів, розмічених за допомогою мови розмітки HTML.

Але ситуація, як правило, є складнішою; значна частина веб-ресурсів на сучасному етапі є динамічними, тобто вони не існують в заздалегідь підготовленому вигляді, а створюються безпосередньо в процесі обробки запиту від користувача.

Для того, щоб людина, яка працює в Інтернеті, могла переглянути ту чи іншу сторінку, на її комп'ютері повинно бути встановлено відповідне програмне забезпечення. Програми для перегляду веб-сторінок називаються браузером.

Але, крім браузерів, до серверів можуть звертатися і інші клієнти, а саме – автономні програми. Вони можуть передбачати взаємодію з людиною, а можуть працювати в цілком автоматичному режимі. Типовим класом таких програм є роботи, призначені для автоматичного перегляду веб-ресурсів. Зокрема, роботи є важливим елементом пошукових систем і використовуються ними для перегляду сторінок і збору інформації про них.

Для запиту до веб-сервера клієнтська програма повинна задати місцезнаходження комп'ютера, на якому розміщується серверна програма, назву потрібного документа і, можливо, інші дані, які специфікують запит. Мережа забезпечує знаходження сервера і передачу йому клієнтського запиту. Серверні програми обробляють цей запит, відповідь пересилається по мережі клієнтові.

Трирівнева клієнт-серверна архітектура, яка почала розвиватися з середини 90-х років, передбачає відділення прикладного рівня від управління даними. Відокремлюється окремий програмний рівень, на якому зосереджується прикладна логіка ПЗ. Програми проміжного рівня можуть функціонувати під управлінням спеціальних серверів ПЗ, але запуск таких програм може здійснюватися і під управлінням звичайного веб-сервера. Нарешті, управління даними здійснюється сервером даних.

Для роботи з системою користувач використовує стандартне програмне забезпечення –звичайний браузер. Це позбавляє його необхідності завантажувати та інсталиувати спеціальні програми (хоча інколи така необхідність все-таки виникає).

Але користувачеві слід надати в розпорядженні інтерфейс, який дозволяв би йому взаємодіяти з системою і формувати запити до неї. Форми, що визначають цей інтерфейс, розміщуються на веб-сторінках та завантажуються разом з ними.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

Веб-оглядач формує запит та пересилає його до сервера, який здійснює обробку. При необхідності сервер викликає серверні програмні модулі, які забезпечують обробку запиту і в разі потреби звертаються до сервера даних. Сервер даних здійснює операції з даними, що зберігаються в системі та складають її інформаційну основу. Зокрема, він може здійснити вибірку з інформаційної бази відповідно до запиту та передати її модулю проміжного рівня для подальшої обробки. Дані, з якими працює сервер даних, найчастіше організовані як реляційна база даних.

Найчастіше веб-сервер і серверні модулі проміжного рівня розміщуються на одному комп'ютері, хоч і являють собою окремі і логічно незалежні програмні модулі.

На сучасному етапі для програмування модулів проміжного рівня використовується мова серверних сценаріїв PHP, а для управління даними – СУБД MySQL. Таким чином, зв'язку PHP-MySQL слід розглядати як стандартний інструмент для створення порівняно простих інтерактивних веб-сайтів та систем електронної комерції; близько 90% комерційних систем сьогодні створюється саме на цій основі. Водночас як засоби управління даними, так і middleware-засоби можуть бути найрізноманітнішими. Так, для створення серверних програм, крім PHP, широко застосовуються Java, Perl, Python, Delphi.

Взагалі, технології створення розподілених, зокрема веб-програм, стрімко розвиваються. Слід згадати про технології EJB (Enterprise Java Beans), CORBA, а також про .NET – порівняно нову ініціативу компанії Microsoft. Для зберігання даних та їх передачі часто використовується так звана розширювана мова розмітки XML (Extensible Markup Language).

Розглянемо визначення API. Це прикладний програмний інтерфейс (Application Programming Interface, API) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

Спрощено – це набір чітко визначених методів для взаємодії різних компонентів. API надає розробнику засоби для швидкої розробки програмного

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

забезпечення. API може бути для веб-базованих систем, операційних систем, баз даних, апаратного забезпечення, програмних бібліотек.

Одним з найпоширеніших призначень API є надання набору широко використовуваних функцій, наприклад для малювання вікна чи іконок на екрані. API є абстрактним поняттям – програмне забезпечення, що пропонує деякий API, часто називають реалізацією (implementation) даного API. У багатьох випадках API є частиною набору розробки програмного забезпечення, водночас, набір розробки може включати як API, так і інші інструменти/апаратне забезпечення, отже ці два терміни не є взаємозамінювані.

Високорівневі API часто програють у гнучкості. Виконання деяких функцій нижчого рівня стає набагато складнішим, або навіть неможливим.

В об'єктно-орієнтованих мовах, прикладний програмний інтерфейс зазвичай включає в себе опис набору визначень класу, з набором форм поведінки, пов'язаних з цими класами. Це абстрактне поняття пов'язане з реальними функціями, які надані або надаватимуться, класами, які реалізуються в методах класу.

Прикладний програмний інтерфейс в даному випадку можна розглядати як сукупність всіх методів, які публічно доступні в класах (зазвичай званий інтерфейс класу). Це означає, що прикладний програмний інтерфейс вказує методи, за допомогою яких взаємодіє з об'єктами, отриманими з визначень класів і обробляє їх.

У більш загальному плані можна визначити Прикладний Програмний Інтерфейс як сукупність усіх видів об'єктів, які можна вивести з визначення класу, і пов'язаних з ними можливих варіантів поведінки.

Наприклад: клас, що представляє Stack, може просто виставити публічно два методи Push() (для додавання нового елемента в стек) і Pop() (для вилучення останнього пункту, ідеально розташований на вершині стека).

У цьому випадку Прикладний Програмний Інтерфейс може бути інтерпретованим як два методи pop() і push(), або, більш широко,

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

використовується варіант, коли можна використовувати елемент типу Stack, який реалізує поведінку стека, надаючи йому можливість для додавання / видалення елементів з вершини. Друга інтерпретація видається більш доречною в дусі об'єктно-орієнтованого підходу.

Якість документації, пов'язаної з Прикладним Програмним Інтерфейсом, є часто ключовим фактором, що визначає його успішність з точки зору простоти використання.

ППІ, як правило, пов'язаний із бібліотеками програмного забезпечення: ППІ описує і вказує очікувану поведінку в той час, як бібліотека є фактичною реалізацією даного набору правил. Один ППІ може мати декілька реалізацій (або жодної, будучи абстрактним) у вигляді різних бібліотек, які мають такий же інтерфейс.

Прикладний програмний інтерфейс також може бути пов'язаним з платформами програмування: платформа може бути заснована на кількох бібліотеках реалізує декілька інтерфейсів ППІ, але на відміну від звичайного використання ППІ, доступ до поведінки вбудований в платформу опосередкований шляхом розширення його змісту новими класами і вставлений в саму платформу. Крім того, загальний потік управління програми може бути під контролем абонента.

Прикладний програмний інтерфейс може бути також реалізацією протоколу.

Коли ППІ реалізує протокол, він може бути заснованим на проксі-методах віддалених викликів, що засновані на протоколі зв'язку. Роль ППІ може заключатися саме в тому, щоб приховати деталі транспортного протоколу. Наприклад: RMI є ППІ, який реалізує протокол або JRMP IIOP як RMI-IIOP.

Протоколи, як правило, розподіляються між різними технологіями і зазвичай дозволяють різним технологіям обмінюватися інформацією, діючи як абстракція між двома світами. ППІ, як правило, є специфічним для конкретної технології: звідси, інтерфейси даної мови не можуть бути використані на інших

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

мовах, якщо виклики функції не будуть перетворені з конкретної адаптації бібліотеки.

Деякі мови, серед яких такі, що працюють на віртуальних машинах (наприклад: мови, сумісні з NET CLI середовища CLR і JVM сумісних мов у віртуальній машині Java) можуть ділитися програмними інтерфейсами.

У цьому випадку віртуальна машина дозволяє мові взаємодії завдяки спільному знаменнику віртуальної машини, що абстрагується від конкретної мови, використовувати проміжний байт-код і його мову.

При використанні прикладного програмного інтерфейсу в контексті веб-розробки, як правило, ППП визначається набором повідомлень запиту HTTP, також визначається структура повідомлень-відповідей, зазвичай у розширенні мови розмітки XML або в форматі об'єктного запису JavaScript (JSON). У той час як прикладний програмний інтерфейс у Web історично був практично синонімом для веб-служби, останнім часом тенденція змінилась (так званий Web 2.0) на відхід від Simple Object Access Protocol (SOAP) на основі веб-сервісів і сервіс-орієнтованої архітектури (SOA) на більш прямі передачі репрезентативного стану (REST) стилів веб-ресурсів та ресурсів-орієнтованої архітектури (ROA).

Частина цієї тенденції пов'язана з рухом Семантичного веб-ресурсу до Опису Платформ (RDF), Концепції розвитку веб-технологій інженерних онтологій. Прикладні програмні інтерфейси у Web, що дозволяють комбінувати декількома прикладними програмними інтерфейсами в нові додатки називають гібридними.

Хоча я реалізовував програму сам, було використано підходи Scrum для саморозвитку та пришвидшенню розробки, розглянемо цей метод. Scrum – підхід управління проектами для гнучкої розробки програмного забезпечення. Скрам чітко робить акцент на якісному контролі процесу розробки.

Підхід вперше описали Гіротакі Такеучі та Ікуджіро Нонака в статті The New New Product Development Game (Гарвардський Діловий Огляд, січ–лют 1986). Вони відзначили, що проекти, над якими працюють невеликі, крос–

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

функціональні команди, зазвичай систематично продукують кращі результати, і пояснили це, як «підхід регбі». У 1991 році ДеҐрейс та Шталь у книжці Злі проблеми, справедливі рішення послалися на цей підхід, як на Scrum (штовханина; сутичка навколо м'яча (у регбі)), спортивний термін, згаданий в статті Такеучі і Нонака. Кен Швабер на початку 1990-х використовував підхід який привів Scrum в його компанію.

Вперше метод Scrum було представлено на загальний огляд задокументованим, чітко сформульованим та описаним спільно Сазерлендом та Швабером на OOPSLA'96 в Остіні. Швабер та Сазерленд протягом наступних років працювали разом щоб обробити та описати весь їхній досвід та найкращі практичні зразки для індустрії в одне ціле, в ту методологію, що відома сьогодні як Scrum. Швабер об'єднав зусилля з Майком Бідлом в 2001, щоб детально описати метод в книжці Agile Software Development with SCRUM. Не зважаючи на те, що для Scrum нарікли долю управління проектами з розробки ПЗ, він може також використовуватися в роботі команд обслуговувань програмного забезпечення (software maintenance teams), або як підхід управління розробкою і супроводом програм: Scrum of Scrums.

Scrum – це кістяк процесу, який включає набір методів і попередньо визначених ролей. Головні дійові особи – ScrumMaster, той хто опікується процесами, веде їх і працює як керівник проекту, Власник Продукту, людина, що представляє інтереси кінцевих користувачів та інших зацікавлених в продукті сторін, та Команду, яка включає розробників.

Протягом кожного спринту, 15-30 денного періоду (тривалість визначається командою), працівники створюють функціональний ріст програмного забезпечення.

Набір можливостей, які імплементуються кожного спринту, приходять з етапу, що має назву product backlog (документація запитів на виконання робіт), який має найвищу пріоритетність за рівнем вимог до роботи, що повинна бути виконана.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

Запити на виконання робіт (backlog items), що визначені протягом наради з планування спринту (sprint planning meeting), переміщуються в етап спринту. Протягом цієї наради Власник Продукту інформує про завдання, які він хоче, аби були виконані. Тоді Команда визначає, скільки з бажаного вони можуть зробити, щоб завершити необхідні частини протягом наступного спринту. Протягом спринту команда виконує визначений фіксований список завдань (т.з. backlog items). Впродовж цього періоду ніхто не має права змінювати перелік запитів на виконання робіт, що слід розуміти, як заморожування вимог (requirements) протягом спринту.

Product backlog – це документ, який має список вимог до функціональності, які упорядковані згідно зі ступенем важливості. Product backlog представляє список того, що повинно бути реалізовано. Елементи цього списку називається «історіями» (user story) або елементами backlog-у (backlog items). Product backlog відкритий для редагування усім учасникам Scrum-процесу.

Обов'язкові поля:

1. ID – унікальний ідентифікатор, порядковий номер, який використовується для ідентифікації історій у разі їх перейменування.

2. Назва (Name) – стислий опис історії. Він повинен бути однозначним, щоб і розробники і product owner могли зрозуміти, про що йдеться і відрізнити одну історію від іншої.

3. Важливість (Importance) – ступінь важливості даної історії на погляд product owner 'а. Зазвичай являє собою натуральне число, іноді для цієї цілі використовуються числа Фібоначчі. Чим більше значення, тим більше пріоритет.

4. Попередня оцінка (initial estimate) – початкова оцінка об'єму робіт, необхідного для реалізації історії порівняно з іншими історіями. Вимірюється у story point'ах. Приблизно відповідає числу «ідеальних людино-днів».

5. Як продемонструвати (how to demo) – стисле пояснення того, як завершена задача буде продемонстрована у кінці спринта. Дане поле може являти собою код автоматизованого приймального тесту.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Додаткові поля. Іноді, також, використовуються додаткові поля у product backlog, в основному для того, щоб допомогти product owner'у визначитися з його пріоритетами.

Категорія (track). Наприклад, «панель управління» чи «оптимізація». За допомогою цього поля product owner може легко вибрати усі пункти категорії «оптимізація» і задати їм низький пріоритет.

Компоненти (components) – указує, які компоненти (наприклад, база даних, сервер, клієнт) будуть зачеплені при реалізації історії. Дане поле складається з групи checkbox'ів, які відмічаються, якщо відповідні компоненти потребують змін.

Ініціатор запиту (requestor). Product owner може захотіти зберігати інформацію про усіх замовників, зацікавлених у даній задачі. Це потрібно для того, щоб тримати їх у курсі діла про хід виконання робіт.

ID у системі обліку помилок (bug tracking ID) – якщо ви використовуєте окрему систему обліку помилок, тоді у описі історії корисно зберігати посилання на всі дефекти, які до неї відносяться.

Sprint backlog – містить функціональність, обрану Product Owner із Product Backlog. Всі функції розбиті по задачах, кожна з яких оцінюється командою. Кожен день команда оцінює об'єм роботи, який необхідно провести для завершення задачі.

Burndown chart – показує, скільки вже виконано і скільки ще залишається зробити.

Планування спринта (Sprint Planning Meeting)

Проходить на початку нової ітерації Спринта:

– Із Product Backlog обираються задачі, зобов'язання по виконанню яких за спринт приймає на себе команда.

– На основі обраних задач створюється Sprint Backlog. Кожна задача оцінюється у ідеальних людино-годинах.

- Рішення задачі не повинно займати більше 12 годин або одного дня. При необхідності задача розбивається на підзадачі.
- Обговорюється та визначається, яким чином буде реалізовано цей об'єм робіт.
- Тривалість наради обмежена зверху 4–8 годинами в залежності від тривалості ітерації, досвіду команди тощо.
- (перша частина наради) Беруть участь Product Owner + Команда: обирають задачі із Product Backlog.
- (друга частина наради) Бере участь лише команда: обговорюють технічні деталі реалізації, наповнюють Sprint Backlog.

Щоденна нарада (Daily Scrum Meeting)

Відбувається кожен день протягом спринта. Є «пульсом» ходу спринта. Нараді властиві наступні обмеження:

- починається точно вчасно;
 - всі можуть спостерігати, але говорять тільки обрані;
 - триває не більш ніж 15 хвилин;
 - проводиться в одному і тому ж місці протягом одного спринта.
- Протягом наради кожен член команди відповідає на 3 запитання:
- Що зроблено з моменту попередньої щоденної наради?;
 - Які проблеми заважають досягненню цілей спринта? (Над рішенням цих проблем працює ScrumMaster. Зазвичай це рішення проходить за рамками щоденної наради і у складі осіб, що безпосередньо займаються даною перешкодою.)

Демонстрація (Sprint Review Meeting):

- Проходить у кінці ітерації (спринта).
- Команда демонструє внесок функціональності до продукту всім зацікавленим особам.
- Залучається максимальна кількість глядачів.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

– Усі члени команди беруть участь у демонстрації (одна людина на демонстрацію або кожен показує, що зробив за спринт).

– Обмежена 4–ма годинами в залежності від тривалості ітерації і змін у продукті.

Ретроспектива (Sprint Retrospective):

– Члени команди висловлюють свою думку про минулий спринт.

– Відповідають на два основних запитання: Що було зроблено добре у минулому спринті?; Що потрібно покращити в наступному?.

– Виконують покращення процесу розробки (вирішують питання та фіксують вдалі рішення).

– Обмежена 1–3ма годинами.

Jira – була використана комерційна система відслідковування помилок, призначена для організації взаємодії з користувачами, хоча в деяких випадках використовується і для управління проектами. Розроблено компанією Atlassian, є одним з двох її основних продуктів (поряд з вікі-системою Confluence). Має веб-інтерфейс.

Назва системи отримано шляхом усічення слова «Gojira» – Японського імені монстра Годзилла, що, в свою чергу, є відсиланням до назви конкуруючого продукту – Bugzilla; створювалася в якості заміни Bugzilla і багато в чому повторює її архітектуру. Система дозволяє працювати з декількома проектами. Для кожного з проектів створює і веде схеми безпеки і схеми оповіщення.

До версії 3.13.5 (включно) розрізнялися редакції Enterprise, Professional і Standard, після – Залишилася тільки редакція Enterprise (для великих організацій).

Система заснована на Java EE і працює на кількох популярних системах управління базами даних і операційних системах.

Основний елемент обліку в системі – завдання (ticket або issue). Завдання містить назву проекту, тему, тип, пріоритет, компоненти і зміст. Завдання може бути розширена додатковими полями (також і нові призначені для користувача поля можуть бути визначені), додатками (наприклад – Фотографіями,

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

скріншотами) або коментарями. Завдання може редагуватися або просто змінювати статус, наприклад, з «відкритий» в «закритий». Які переходи між станами можливі, визначається через настроюється потік операцій. Будь-які зміни в задачі записуються в журнал.

Jira має велику кількість можливостей конфігурації: для кожної програми може бути визначений окремий тип завдання з власним workflow, набором статусів, одним або декількома видами уявлення (screens). Крім того, за допомогою так званих «схем» можна визначити для кожного індивідуального Jira-проекту власні права доступу, поведінку і видимість полів і багато іншого.

Завдяки універсальному підходу можна пристосувати Jira для багатьох непрофільних завдань, наприклад, керування вимогами, керування ризиками, аж до реалізації невеликої системи бронювання, автоматизації процесу рекрутингу.

Для інтеграції з зовнішніми системами підтримує інтерфейси SOAP, XML-RPC і REST. Поставляється із засобами інтеграції з такими системами управління версіями, як Subversion, CVS, Git, Clearcase, Team Foundation Server, Mercurial і Perforce.

Існують доповнення, що дозволяють вбудувати Jira в інтегровані середовища розробки, в тому числі Eclipse і IntelliJ IDEA. Перекладена багатьма мовами, включаючи українську, англійську, японську, німецьку, французьку, іспанську.

Для сторонніх розробників надаються кошти розробки розширень системи – плагінів. Розробники розширень можуть викладати плагіни для продажу на спеціальний розділ сайту Atlassian.

Є комерційним продуктом, який може бути ліцензований для роботи на локальному сервері або доступний в якості віддаленого додатки. Ціноутворення залежить від максимального числа користувачів, при цьому близько \$50 за користувача для локального і \$7 на місяць за користувача для віддаленого доступу є типовими цінами.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

Для академічних і комерційних клієнтів доступний повний вихідний код під ліцензією розробника.

Для проектів з відкритим вихідним кодом Atlassian надає спеціальну безкоштовну ліцензію при дотриманні наступних правил:

- проект використовує ліцензії, схвалені Open Source Initiative;
- вихідний код проекту доступний для скачування;
- у проекту є публічно доступна веб-сайт;
- програмне забезпечення від Atlassian є на веб-сайті проекту.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою IDEA – симетричний блоковий алгоритм шифрування даних, запатентований швейцарською фірмою Ascom. Відомий тим, що застосовувався в пакеті програм шифрування PGP. У листопаді 2000 року IDEA був представлений як кандидат у проекті NESSIE в рамках програми Європейської комісії IST (англ. Information Societies Technology, інформаційні громадські технології).

Першу версію алгоритму розробили в 1990 році Лай Сюецзя (Xuejia Lai) і Джеймс Мессі (James Massey) зі Швейцарського інституту ETH Zürich (за контрактом з Hasler Foundation, яка пізніше влилася в Ascom-Tech AG) як заміна DES (англ. Data Encryption Standard, стандарт шифрування даних) і назвали її PES (англ. Proposed Encryption Standard, запропонований стандарт шифрування). Потім, після публікації робіт Біхамом і Шаміра по диференціальному криптоанализу PES, алгоритм був поліпшений з метою посилення криптостійкості і названий IPES (англ. Improved Proposed Encryption Standard, покращений запропонований стандарт шифрування). Через рік його перейменували в IDEA (англ. International Data Encryption Algorithm).

Так як IDEA використовує 128-бітний ключ і 64-бітний розмір блоку, відкритий текст розбивається на блоки по 64 біт. Якщо таке розбиття неможливо,

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

останній блок доповнюється різними способами певною послідовністю біт. Для уникнення витоку інформації про кожному окремому блоці використовуються різні режими шифрування. Кожен вихідний незашифрований 64 – біт ний блок ділиться на чотири підблока по 16 біт кожен, так як всі алгебраїчні операції, що використовуються в процесі шифрування, відбуваються над 16-бітними числами. Для шифрування і розшифрування IDEA використовує один і той же алгоритм.

Позначення операцій:

- $\boxplus$ Додавання за модулем 2^{16} .
- $\odot$ Множення за модулем $2^{16}+1$.
- $\oplus$ Побітова виключна диз'юнкція.

Фундаментальним нововведенням в алгоритмі є використання операцій з різних алгебраїчних груп, а саме:

Додавання за модулем 2^{16} .

Множення за модулем $2^{16}+1$.

Побітова виключна диз'юнкція (XOR).

Ці три операції несумісні в тому сенсі, що ніякі дві з них не задовольняють дистрибутивному закону, тобто:

$$a \odot (b \oplus c) \neq (a \odot b) \oplus (a \odot c).$$

Застосування цих трьох операцій ускладнює криптоаналіз IDEA в порівнянні з DES, який базується виключно на операції виключає АБО, а також дозволяє відмовитися від використання S-блоків і таблиць заміни. IDEA є модифікацією мережі Фейстеля.

Генерація ключів

З 128-бітного ключа для кожного з восьми раундів шифрування генерується по шість 16-бітних підключів, а для вихідного перетворення генерується чотири 16-бітних підключа. Всього буде потрібно $52 = 8 \times 6 + 4$ різних підключів по 16 біт кожен. Процес генерації п'ятдесяти двох 16-бітних ключів полягає в наступному:

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

Насамперед, 128-бітний ключ розбивається на вісім 16-бітних блоків. Це будуть перші вісім підключів по 16 біт кожен – $(K_1^{(1)}K_2^{(1)}K_3^{(1)}K_4^{(1)}K_5^{(1)}K_6^{(1)}K_1^{(2)}K_2^{(2)})$

Потім цей 128-бітний ключ циклічно зсувається вліво на 25 позицій, після чого новий 128-бітний блок знову розбивається на вісім 16-бітних блоків. Це вже наступні вісім підключів по 16 біт кожен – $(K_3^{(2)}K_4^{(2)}K_5^{(2)}K_6^{(2)}K_1^{(3)}K_2^{(3)}K_3^{(3)}K_4^{(3)})$

Процедура циклічного зсуву і розбивки на блоки триває до тих пір, поки не будуть згенеровані всі 52 16-бітних підключа.

Шифрування

Структура алгоритму IDEA показана на рисунку 4.3.

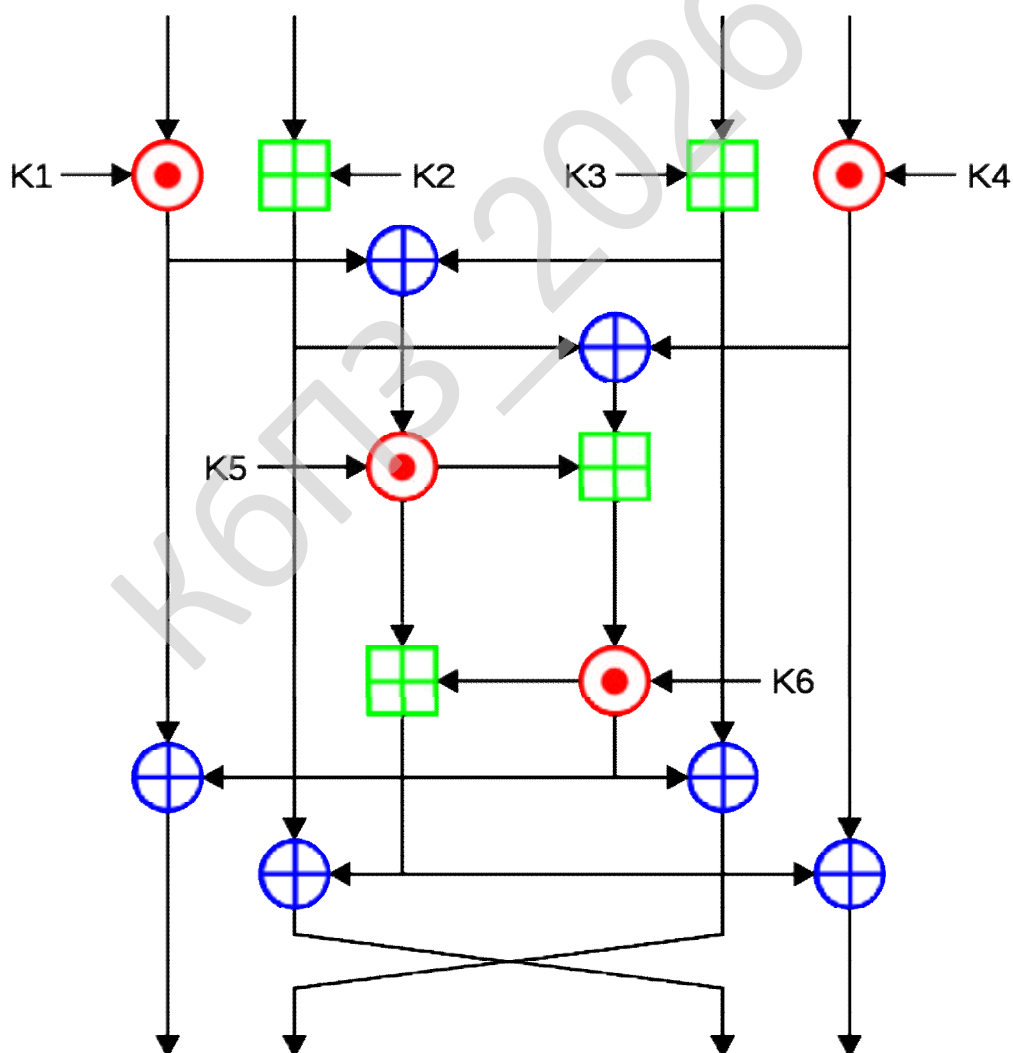


Рисунок 4.3 – Структура алгоритму IDEA

Процес шифрування складається з восьми однакових раундів шифрування і одного вихідного перетворення. Вихідний незашифрований текст ділиться на блоки по 64 біта. Кожен такий блок ділиться на чотири підблока по 16 біт кожен. На рисунку ці підблоки позначені D_1 , D_2 , D_3 , D_4 . У кожному раунді використовуються свої підключі згідно з таблицею підключів. Над 16-бітними підключами і підблока незашифрованого тексту проводяться наступні операції:

- Множення за модулем $2^{16}+1 = 65537$, причому замість нуля використовується 2^{16} .

- Додавання за модулем 2^{16} .

- Побітове виключне АБО.

В кінці кожного раунду шифрування є чотири 16-бітних підблоки, які потім використовуються як вхідні підблоки для наступного раунду шифрування. Вихідна перетворення являє собою скорочений раунд, а саме, чотири 16-бітних підблоки на виході восьмого раунду і чотири відповідних підключа піддаються операціям:

- Множення за модулем $2^{16}+1$.

- Додавання за модулем 2^{16} .

Після виконання вихідного перетворення конкатенація підблоків D_1' , D_2' , D_3' і D_4' являє собою зашифрований текст. Потім береться наступний 64-бітний блок незашифрованого тексту і алгоритм шифрування повторюється. Так продовжується до тих пір, поки не зашифрують всі 64-бітові блоки вихідного тексту.

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено розроблене у бакалаврської дипломної роботі система віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud. З рисунку можна побачити що інтерфейс головного вікна розподілено на наступні функціональні розділи: Функціональних кнопок ПЗ; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Верхнього меню; Розділу обрання групи; Розділу виведення результату роботи системи.

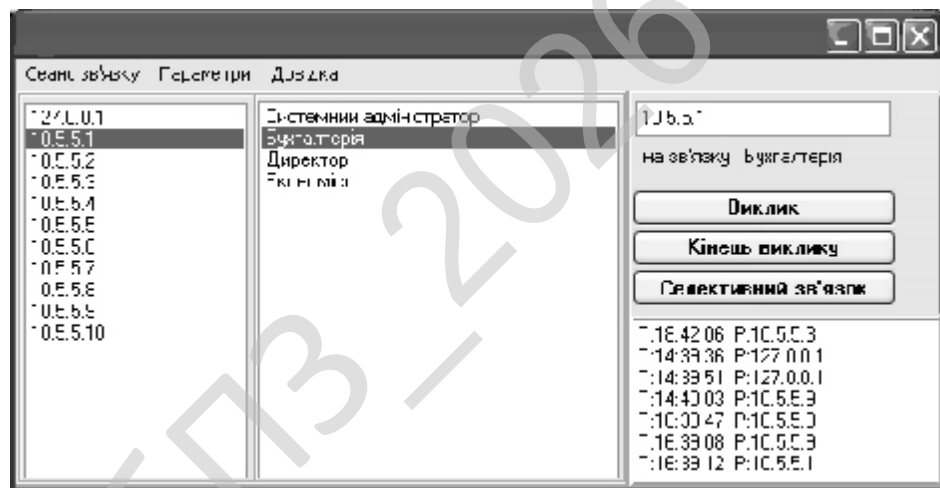


Рисунок 5.1 – Головне вікно ПЗ

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.

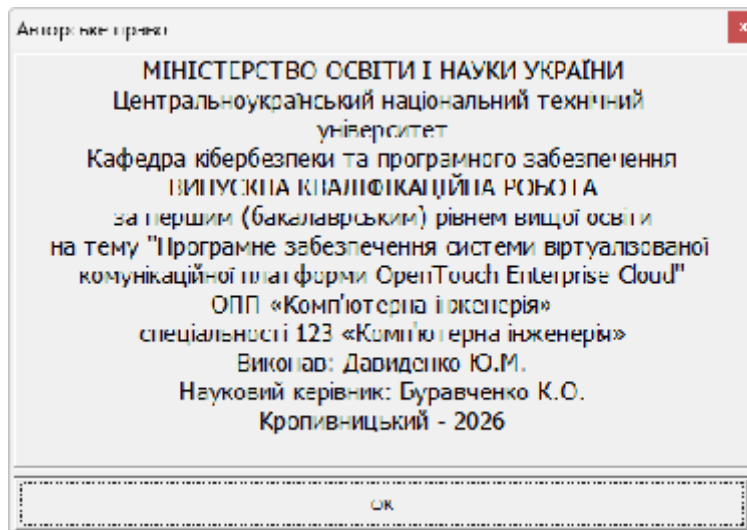


Рисунок 5.2 – Авторське право

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

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

Ці тести демонструють:

- Як виконуються функції програми.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

- 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 імовірністю 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:

- Некоректних чи відсутніх функцій.
- помилок інтерфейсу.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

– 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 – Freeware.

Це власницьке програмне забезпечення, котре можна Безоплатно використовувати протягом необмеженого терміну без обмежень у функціональності, і поширюване без сирцевих кодів.

Автори такого програмного забезпечення, як правило, хочуть «дати щось спільноті», але хочуть також контролювати його подальшу розробку. Іноді, коли програмісти вирішують припинити розробку, вони передають сирцевий код іншим програмістам, або ж спільноті як вільне програмне забезпечення.

Дуже часто плутають поняття «безплатне програмне забезпечення» та «вільне програмне забезпечення», хоча вони суттєво відрізняються.

Безплатне програмне забезпечення можна безоплатно встановлювати та використовувати (іноді з певними обмеженнями, як, наприклад, «безплатне для домашнього або некомерційного вжитку»), в той час як вільне програмне забезпечення можна продавати за будь-яку суму, але при тому, у користувача, котрий його отримує, повинні бути права на вивчення, модифікацію та поширення сирцевих кодів одержаної програми.

6 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти, призначено для системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

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

Рішення завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

- Досліджена система віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

- На основі отриманих результатів досліджень створена програмна реалізація системи віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

Розроблені під час виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

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 віртуалізованої комунікаційної платформи OpenTouch Enterprise Cloud.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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 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 IDEA.

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.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ranjan Parekh. Fundamentals of Image, Audio, and Video Processing Using MATLAB® With Applications to Pattern Recognition. CRC Press. 2021. 406 p.
2. Alasdair McAndrew. A Computational Introduction to Digital Image Processing. Chapman & Hall. 2021. 560 p.
3. Михайло Пічугін, Іван Канкін, Володимир Воротніков Комп'ютерна графіка. Навчальний посібник / Центр навчальної літератури 346 с. 2019р.
4. Маценко В.Г. Комп'ютерна графіка: Навчальний посібник. – Чернівці: Рута, 2009 – 343 с.
5. 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.
6. 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.
7. Kopf, Johannes and Lischinski, Dani. Depixelizing Pixel Art (англ.) // ACM Trans. Graph. – 2011. – Vol. 30, no. 4. – P. 99:1--99:8.
8. Giachetti, Andrea and Asuni, Nicola. Real-Time Artifact-Free Image Upscaling (англ.) // Trans. Img. Proc.. – 2011. – Vol. 20, no. 10. – P. 2760—2768.
9. 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.
10. 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

11. Смірнова Т.В., Коноплицька-Слободенюк О.К., Буравченко К.О., Смірнов С.А., Кравчук О.В., Козірова Н.Л., Смірнов О.А. «Дослідження технологій забезпечення кібербезпеки хмарних сервісів IaaS, PaaS та SaaS». *Кібербезпека: освіта, наука, техніка*. 2024. №4(24), С. 6-27.
12. 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.
13. 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.
14. 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.
15. 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.
16. Аль-Мудхафар Акіл Абдулхуссейн М., Смірнова Т.В., Буравченко К.О., Смірнов О.А. «Метод оцінки та підвищення користувальницького досвіду абонентів в програмно-конфігурованих мережах на основі використання машинного навчання». *Сучасні інформаційні системи*, 2023, том 7, № 2, С. 49-56.
17. Smirnov, O., Neskorodieva, T., Fedorov, E., Rudakov, K., Neskorodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» *CEUR Workshop Proceedings*, Volume 3187, 2022,
18. 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.

					ВКРБ-123.26.0040.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

19. 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>

20. 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.

21. 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.

22. 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.

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

24. 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.

25. 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.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. 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.

46. 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.

47. 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.

48. 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).

49. 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

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

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

52. 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.

53. 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.

54. 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

55. 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.

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

57. 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.

58. 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.

59. 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.

60. 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