

Центральноукраїнський національний технічний університет
Центр заочної та дистанційної освіти
Кафедра кібербезпеки та програмного забезпечення

”Допущено до захисту”
Завідувач кафедри кібербезпеки
та програмного забезпечення
д.т.н., професор
_____ Олексій СМІРНОВ
“ ____ ” _____ 2023 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи комунікації
Cloud-сервісів з використанням SIP/VoIP”

КБПЗ – 2023

Виконав здобувач вищої освіти
II курсу, групи КІ-22МЗ
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Іванченко Д.О.
« ____ » _____ 2023 р.

Керівник проекту
доктор технічних наук, професор
_____ Смірнов О.А.
« ____ » _____ 2023 р.
Рецензент _____

Центральноукраїнський національний технічний університет

Центр *Заочної та дистанційної освіти*

Кафедра *Кібербезпеки та програмного забезпечення*

Рівень вищої освіти *магістр*

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

« 6 » вересня 2023 року

**ЗАВДАННЯ НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ ЗА
ДРУГИМ (МАГІСТЕРСЬКИМ) РІВНЕМ ВИЩОЇ ОСВІТИ
ЗДОБУВАЧА ВИЩОЇ ОСВІТИ**

Іванченку Дмитру Олександровичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP*

2. Керівник роботи *Смірнов Олексій Анатолійович, докт. техн. наук, професор*

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу № 36-13 від 04.08.2023 року

3. Строк подання студентом роботи до захисту *10.12.2023 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP*

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

1. Призначення та область використання.

6. Наукова новизна.

2. Перегляд аналогічних існуючих систем.

7. Економічна ефективність розробленої програми.

3. Опис і обґрунтування проектних рішень.

8. Заходи з охорони праці та техніки безпеки.

4. Етапи програмування системи.

9. Висновки.

5. Впровадження системи в промислову експлуатацію

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

Наукова новизна

1 аркуш

Структурна схема системи

1 аркуш

Функціональна схема системи

1 аркуш

Діаграма процесів

1 аркуш

Блок-схема алгоритму роботи додатку

2 аркуша

Показники економічної ефективності

1 аркуш

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Економічний	Савеленко Г.В.	05.10.2023	14.11.2023
Охорона праці	Оришака О.В.	06.10.2023	16.11.2023

7. Дата видачі завдання « 6 » вересня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.10.2023 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2023 р.	
3.	Розробка моделі компонента	20.10.2023 р.	
4.	Розробка структур даних	25.10.2023 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2023 р.	
6.	Програмування алгоритмів	10.11.2023 р.	
7.	Розрахунок економічної ефективності	13.11.2023 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2023 р.	
9.	Оформлення ПЗ	17.11.2023 р.	
10.	Попередній захист роботи	10.12.2023 р.	

Дата видачі завдання
« 6 » вересня 2023 р.

Підпис керівника

_____ (прізвище та ініціали)

Завдання прийнято до виконання
« 6 » вересня 2023 р.

Підпис здобувача

_____ (прізвище та ініціали)

АНОТАЦІЯ

Іванченко Д.О. Дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2023.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи комунікації Cloud-сервісів з використанням SIP/VoIP.

Метою розробки є дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP.

Об'єктом дослідження є процес комунікації Cloud-сервісів з використанням SIP/VoIP.

Предметом дослідження є методи комунікації Cloud-сервісів з використанням SIP/VoIP.

Методи дослідження базуються на методах теорії телекому, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP.

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 архітектури IBM PC з ОС Windows 10/11.

Програму розроблено в середовищі Delphi 10.4 Sydney.

Ключові слова: комп'ютерна інженерія, SIP/VoIP

ABSTRACT

Ivanchenko D.O. Research and software implementation of the Cloud services communication system using SIP/VoIP. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2023.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the communication system of Cloud services using SIP/VoIP.

The purpose of the development is the research and software implementation of the Cloud services communication system using SIP/VoIP.

The object of the study is the communication process of Cloud services using SIP/VoIP.

The subject of the study is methods of communication of Cloud services using SIP/VoIP.

Research methods are based on telecom theory methods, mathematical statistics methods, and software development methods.

The result of the work is the software implementation of the Cloud services communication system using SIP/VoIP.

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 tools are provided.

The program can be used on PCs of IBM PC architecture with Windows 10/11 OS.

The program was developed in the Delphi 10.4 Sydney environment.

Keywords: computer engineering, SIP/VoIP

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	70
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	70
7.2 Розрахунок трудомісткості розробки програмної продукції.....	72
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	74
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	78
7.5 Визначення собівартості розробки та ціни програмної продукції.....	82
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	86
7.7 Визначення експлуатаційних витрат.....	86
7.8 Визначення економічної ефективності програмної продукції.....	88
7.9 Висновок.....	90
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	91
8.1 Вступ.....	91
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	93
8.3 Пожежна безпека.....	94
8.4 Розрахункова частина	95
8.5 Висновки до розділу.....	96
9 ОСНОВНІ ВИСНОВКИ.....	98
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	

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

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

ВСТУП

Актуальність теми. Популярність хмарних рішень стрімко росте. Однак, при дворазовому процентному росту в абсолютних величинах обсяг ринку хмарної телефонії поки невеликий, що в основному пов'язане з тим, що його головними споживачами є компанії малого бізнесу. Для таких організацій дуже важливе швидке розгортання системи й низька вартість володіння. І якщо по такому факторі, як швидкість розгортання, хмарні АТМ виграють в «звичайних» IP-АТМ, то їхня вартість на перевірку не завжди виявляється настільки привабливою. Віртуальні системи бувають дійсно вигідні – але для тих компаній, які створюються для реалізації коротких проектів.

Локально встановлені рішення, як правило, мають більше широку функціональність, а їхня модульна структура дозволяє, у міру росту компанії й зміни завдань, доповнювати таке рішення необхідними складовими, що відповідають новим потребам. Вони вказують на ряд ризиків, властивим віртуальним рішенням – зокрема, на те, що у випадку використання хмарних технологій безпека інформації повністю віддається на відкуп провайдеру. Їхній вивід такий: незважаючи на гадану привабливість віртуальних АТМ, у багатьох випадках вигідніше й безпечніше використовувати власні IP-системи.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP.

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

- Огляд існуючих систем комунікації Cloud-сервісів з використанням SIP/VoIP.
- Дослідження системи комунікації Cloud-сервісів з використанням SIP/VoIP.
- Програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

Об'єктом дослідження є процес комунікації Cloud-сервісів з використанням SIP/VoIP.

Предметом дослідження є методи комунікації Cloud-сервісів з використанням SIP/VoIP.

Методи дослідження базуються на методах теорії телекому, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод комунікації Cloud-сервісів з використанням SIP/VoIP.

– Розроблено вітчизняний продукт комунікації Cloud-сервісів з використанням SIP/VoIP, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі комунікації Cloud-сервісів з використанням SIP/VoIP.

Достовірність наукових результатів підтверджена теоретичними викладеннями, даними комп'ютерного моделювання, коректними дослідженнями параметрів на функціонуючій обчислювальній мережі, а також відповідністю отриманих результатів окремим результатам, наведеним у науковій літературі.

Робота апробована на LVII Науково-технічній конференції здобувачів вищої освіти «Наука – виробництву», 2023, основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №14.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Одна з основних переваг хмарного сервісу – відсутність капітальних витрат на інфраструктурне встаткування (здобувати прийдуть тільки термінали, та й щодо них можна домовитися про оренду). Основні витрати становлять щомісячні платежі за хмарні сервіси, але й у частині OPEX можна говорити про економію. Обслуговування систем уніфікованих комунікацій (УфК) обходиться в середньому в 1000 доларів на місяць (що порівняно з місячною вартістю хмарного сервісу) плюс ще витрати на зарплату кваліфікованого персоналу, здатного експлуатувати такі системи. Не можна забувати й про те, що при використанні своїх власних систем прийде ще витратитися на відновлення ПЗ – провайдери ж хмарних сервісів самостійно забезпечують подібне відновлення.

Інша незаперечна перевага – гнучкість і оплата в міру використання. Вартість хмарного сервісу телефонії й УфК звичайно розраховується виходячи з кількості користувачів: найняв замовник нових співробітників – оперативно підключив додаткові ліцензії, довелося когось звільнити – не буде потрібно оплачувати ліцензії, що простоюють. Важливо й те, що працездатність хмарного сервісу звичайно не залежить від фізичного розташування офісу. Тому у випадку переїзду замовника в новий офіс, не виникне складностей з переносом комунікаційної системи.

Ще одна перевага – швидкий старт. Як правило, впровадження власних складних IT-рішень займає від декількох місяців до декількох років. Необхідний час на формалізацію вимог, технічне проектування, закупівлю й поставку встаткування, а також на налаштування. Якщо ж ви користуєтеся хмарним сервісом, вся необхідна інфраструктура рішення вже розгорнута на стороні провайдеру. У підсумку строк впровадження сервісів скорочується в кілька разів.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

1.2 Область застосування

Хмарні провайдери намагаються використовувати надійні комерційні ЦОД рівня Tier III, тому здатні представити замовникам фінансову гарантію доступності сервісу не нижче 99,9% часу на місяць. Домогтися такого рівня надійності силами невеликої компанії досить складно й дуже дорого. І це ще один аргумент на користь хмарних сервісів.

Обґрунтоване порівняння моделей впровадження («своя система» або «із хмари») можливо провести тільки розглянувши конкретний випадок і взявши до уваги особливості бізнесу клієнта. Відповідно до розрахунків СТІ, проведеним на період п'ять років, у загальному випадку хмарна пропозиція в порівнянні зі створенням власного рішення аналогічного рівня надійності, масштабованості й захищеності буде вигідніше.

Вибір між локальною, хмарною й гібридною моделями розгортання/споживання комунікаційних сервісів в остаточному підсумку залежить від пріоритетів, специфіки бізнесу й ІТ-стратегії конкретного замовника.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Пропозиція «ЛАНКЕЙ»

Представники «ЛанКей» відразу підкреслили, що, перед тим як пропонувати замовникові конкретне рішення, їм важливо вивчити наявну в нього ІТ-інфраструктуру, з'ясувати, яке встаткування й ПЗ вже є, щоб максимально задіяти їх у проекті й здійснити впровадження нового рішення найбільше ефективно. Вони припустили, що замовник уже використовує (або планує задіяти – якщо компанія створюється з нуля) такі інфраструктурні сервіси, як служба каталогів Microsoft Active Directory і корпоративна система електронної пошти Microsoft Exchange, а з документами користувачі працюють в Microsoft Office. В 70-80% випадків так воно і є. З урахуванням цих припущень, як найбільше підходяще рішення обране система об'єднаних комунікацій Microsoft Lync.

Під об'єднаними комунікаціями розуміється рішення, що поєднає всі види зв'язку в єдиному продукті. Крім властиво бізнес-телефонії, Microsoft Lync забезпечить єдину корпоративну адресну книгу зі статусами присутності всіх абонентів, безкоштовні дзвінки абонентам Skype, ВКЗ, обмін миттєвими повідомленнями, спільний доступ до робочого стола, Web-конференції, мобільний доступ. Очевидною перевагою впровадження Microsoft Lync у цьому випадку стане тісна інтеграція з усіма сервісами замовника, а саме Active Directory, Exchange, продуктами Microsoft Office і Microsoft CRM (про майбутнє впровадження якого замовник замислюється). Як відзначають представники

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

«Ланкей», замовник одержує досить значний список можливостей, яких йому не надасть жодна класична офісна АТМ.

Фахівці «ЛанКей» проаналізували два підходи до питання використання Microsoft Lync: локальне впровадження в інфраструктурі замовника й оренда в хмарі (по моделі SaaS). З урахуванням невеликого розміру компанії замовника можна припустити, що в нього навряд чи є свій ЦОД, велика кількість потужних недозавантажених серверів, а саме головне – значне число кваліфікованих ІТ-фахівців. Таким чином, уклали в «ЛанКей», з великою часткою ймовірності замовникові буде значно вигідніше орендувати сервіс Cloud Lync у хмарі.

Ціна цього сервісу «Ланкей» дорівнює 500 грн./мес. за абонента за самого повного функціонала. Оскільки в замовника 100 співробітників, витрати складуть 50 000 грн./мес. Для порівняння: якщо впроваджувати Lync на 100 користувачів усередині компанії, то тільки вартість ПЗ перевищить 1 млн грн. і це без обліку витрат на сервери й впровадження. Крім того, у випадку реалізації свого власного сервісу замовникові прийде раз у три роки здобувати ПЗ заново й оплачувати роботи з міграції. Компанія «Ланкей» сама обновляє свій сервіс на нові версії Lync у міру їхнього виходу, а також безкоштовно надає його технічну підтримку й адміністрування.

Фахівці «Ланкей» самим ретельним образом підійшли до питання вибору прикінцевого встаткування, представивши замовникові велика кількість варіантів і рекомендацій, але залишивши волю вибору: «беріть, що подобається й що дозволяє бюджет». При цьому вони відзначили, що з ростом популярності концепції BYOD устаткування співробітникам часто вибирає вже не компанія, а самі співробітники.

Типові термінали системи Microsoft Lync – комп'ютери й ноутбуки із установленими на них програмними клієнтами. Для використання всіх можливостей системи досить підключити до комп'ютера провідну або бездротову гарнітуру. Гарнітура може бути будь-який, але фахівці «ЛанКей» рекомендували провідну гарнітуру Microsoft LX6000 (за ціною 2000 грн.), на якій їсти кнопки

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

взяти/покласти трубку, виключити мікрофон і гучність. Крім того, вони відзначили широкий вибір провідних і бездротових гарнітур від Jabra, Plantronics і Logitech. Ті співробітники, кому не подобається гарнітура, можуть використовувати USB-телефони – наприклад, від Polycom, Jabra і Plantronics за ціною від 3000 до 6000 грн.

Вибір IP-телефонів не так широкий, як гарнітур, але варіантів теж досить: від відносно недорогих моделей, таких як Snom 710 і AudioCodes HD420 (5000 грн.), до топових апаратів з кольоровими екранами для керівництва, наприклад Polycom cx600 і HP 4120 (13 000 грн.).

Мобільні співробітники зможуть використовувати будь-які смартфони й планшети під керуванням Android, iOS, Windows Phone і RT. Додаток Lync Mobile можна скачати безкоштовно з магазину додатків. Смартфон, що перебуває в мережі Wi-Fi, є бездротовим телефоном для Lync. Крім того, навіть при відсутності Wi-Fi додаток Lync Mobile однаково працює – у цьому випадку дзвінки на мобільний переадресуються через мережу GSM.

ВКЗ в Microsoft Lync організується за допомогою звичайного комп'ютера й USB-камери. Для ПК фахівці «ЛанКей» рекомендують Full HD-камеру Logitech C930e за 5000 грн., для невеликих груп – керовану камеру Logitech CC3000e (35 000 грн.). Для круглих столів є камери з кутом огляду 360° (Round Table), наприклад Polycom cx5000 за 200 000 грн. Оскільки Microsoft Lync інтегрується практично з будь-якими ВКЗ-системам від Polycom, то, якщо дозволяють засобу, замовник може придбати персональні термінали з монітором для топ-менеджерів (наприклад, Polycom HDX 4500), і кодеки для переговорних – наприклад, HDX 7000 або HDX 8000, які коштують від 400 000 грн. Для переговорних кімнат є рішення з більшими здвоєними сенсорними екранами і якісними камерами за 1 млн грн. і більше.

Пропозиція CISCO

З огляду на поточні потреби замовника й потенціал подальшого розвитку його інфраструктури, фахівці Cisco запропонували використовувати

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

комунікаційну платформу Cisco Business Edition 6000 (далі ВЕ6К), в основу якої покладена система УфК Cisco Unified Communications Manager. Ця платформа спеціально була розроблена для невеликих підприємств і оптимізована з метою зниження сукупної вартості володіння завдяки консолідації встаткування, скороченню витрат на живлення й охолодження, застосуванню зручних засобів налаштування й адміністрування. Як відзначають в Cisco, базове ПЗ й набір інтегрованих додатків надають повнофункціональні, комплексні сервіси спільної роботи й природно інтегруються з мережею передачі даних. Cisco ВЕ6К підтримує розширення ємності до 1000 абонентів і 2500 зареєстрованих пристроїв і забезпечує можливість об'єднання в єдину мережу до 50 віддалених офісів.

Для реалізації проекту пропонується використовувати відказостійкий апаратно-програмний комплекс Cisco ВЕ6К складається із двох серверів Cisco UCS і гіпервізора VMware vSphere 5. На базі даної платформи розгортається ряд додатків (у вигляді віртуальних машин):

– Cisco Unified Communications Manager 10.5 (CUCM) забезпечує реєстрацію абонентських пристроїв і додатків, керування номерним планом, установленням аудіо- і відеоз'єднань.

– Cisco IM & Presence базується на протоколі XMPP і забезпечує сервіс контролю статусу користувачів (Presence) і обміну миттєвими повідомленнями (IM) для програмних клієнтів Cisco Jabber (у тому числі й для мобільних пристроїв). Даний сервіс доступний не тільки для абонентів CUCM, але й (безкоштовно) для всіх співробітників організації в рамках програми Jabber for Everyone.

– Cisco Expressway забезпечує захищений віддалений доступ абонентів системи до всіх корпоративних комунікаційних сервісів без необхідності організації VPN-підключень, а також можливість повнофункціональних зовнішніх комунікацій з партнерами по бізнесі й клієнтами.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

– Cisco Unity Connection являє собою сервіс голосової пошти, інтегрованої з поштовими сервісами й програмними клієнтами Cisco Jabber.

– Cisco Unified Contact Center Express, рішення для організації контакт-центра до 100 операторів, забезпечує багаторівневий IVR, розгорнуту статистику по викликах, автоматизоване робоче місце операторів і супервізорів КЦ, інтеграцію з бізнес-системами й CRM.

Для реалізації єдиного багатоканального номера замовникові пропонується використовувати цифрове або IP-підключення до ТфОП або оператора зв'язку на базі маршрутизатора Cisco з функцією шлюзу або граничного елемента захисту (CUBE) для SIP-транка. Залежно від потреб замовника цей серійний номер може бути термінований на системі автоматичного секретаря, контакт-центрі або робочих місцях секретарів, обладнаних консоллю Unified Attendant.

Адміністрування платформи здійснюється за допомогою убудованих додатків Cisco Prime Collaboration. Як відзначають в Cisco, зручний користувальницький інтерфейс і інструменти автоматизації цього засобу значно прискорюють розгортання системи й істотно скорочують час, необхідне для внесення поточних змін. Крім того, радикально знизити навантаження на адміністратора системи по налаштуванню й кастомізації функцій дозволяє користувальницький портал самообслуговування, убудований в CUCM. Для користувачів реалізується сервіс «одного вікна», де вони самостійно (через Web-інтерфейс) можуть просто й наочно персоніфікувати основні функції телефонії, підключити мобільні пристрої, настроїти мобільний номер або запланувати конференцію.

Як засоби зв'язку для мобільних співробітників Cisco запропонували свої програмні клієнти Jabber (для Windows, MAC, iPad, iPhone, Android), а для керівників – ще й настільні IP-телефони Cisco 8851 з функціями Intelligence Proximity для інтеграції з мобільними пристроями. Надомним співробітникам також пропонуються клієнти Jabber. «Стаціонарним» офісним співробітникам –

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

настільні IP-телефони Cisco 7821 і теж клієнти Jabber, які забезпечать обмін миттєвими повідомленнями, контроль статусу присутності й керування настільним телефоном. Тим 20 співробітникам, які активно переміщуються в межах офісів, рекомендовані телефони Cisco 7821 у режимі Hot-Desking і клієнти Jabber для iPhone і Android.

Хоча фахівці Cisco запропонували замовникові розгорнути платформу Cisco ВЕ6К у себе на площадці, вони не виключили й інші моделі споживання сервісів телефонії/УфК. Функціональність запропонованого рішення може бути реалізована на базі хмарного сервісу на платформі Cisco Hosted Collaboration Solution (HCS) від операторів – партнерів Cisco. Більше того, завдяки підтримці загального набору додатків, прикінцевого встаткування й програмних клієнтів може бути реалізована гібридна модель рішення. Так, наприклад, нерідко в головному офісі замовника історично застосовується власна платформа, а інфраструктура для швидкозростаючих філій або часто міняють дислокацію офісів продажів, засобу підтримки мобільних/надомних співробітників розгортаються із залученням хмарного сервісу. При цьому для співробітників компанії спосіб доставки сервісів “прозорий” – ніякої різниці вони не відчують.

Cisco запропонувала широкі можливості по розвитку свого рішення в частині ВКЗ. Як персональні термінали ВКЗ керівники можуть використовувати термінали серії DX, які реєструються й працюють під керуванням CUCM, що входить до складу платформи ВЕ6К. Платформа Cisco ВЕ6К забезпечує ряд додаткових функцій – зокрема, для сумісності з прикінцевим устаткуванням H.323/SIP і організації міжкорпоративного конференц-зв'язку (TelePresence Video Communication Server) і проведення Web-конференцій (WebEx Meeting Center). Крім того, як варіант хмарної або гібридної моделі споживання комунікаційних сервісів можна розглянути розширення функціонала телефонії сервісом ВКЗ. Такі пропозиції є в портфелі українських провайдерів – партнерів Cisco.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

Всім гарна пропозиція Cisco, от тільки інформацію про його вартість замовник не одержав. Без цін важко зіставити ця пропозиція з іншими й прийняти остаточне рішення.

Пропозиція СТІ

Компанія СТІ пропонує замовникові можливість використовувати сервіси телефонії/УфК як на базі серверного рішення, придбаного у власне користування, так і із хмари. Але при цьому фахівці компанії акцентують його увагу на перевагах хмарної моделі. За допомогою брендovanого порталу замовник може одержати повне керування всіма послугами, підключати й видаляти користувачів, міняти їхні профілі, одержувати звіти по використанню сервісів і статистику дзвінків.

Рішення від СТІ забезпечує всі запитані замовником функції бізнесу-телефонії. Всі дзвінки усередині компанії, у тому числі між філіями в регіонах, стають безкоштовними, міжміські дзвінки на місцеві номери оплачуються по локальних тарифах, а завдяки використанню мобільних додатків для смартфона або ноутбука досягається економія на мобільних переговорах. Для підключення міських телефонних номерів можливо кілька варіантів. Якщо в замовника вже є номери, якими давно користуються клієнти, то їх можна зберегти, купивши й додаткові, якщо необхідно. Можна й з нуля забезпечити багатоканальний номер.

Представники СТІ підкреслюють, що при підключенні послуги замовник одержує закінчену екосистему з повним функціоналом сервісів УфК, тісно інтегрованих між собою, включаючи мультимедійний чат і текстові повідомлення, показ статусу присутності користувача, уніфіковану пошту, відеодзвінки. Запропонований сервіс забезпечує доступність по єдиному номері з автоматичним перемиканням виклику між планшетом, смартфоном і настільним телефоном, а також єдиний корпоративний довідник у єдиному корпоративному середовищі.

Єдиний телефонний довідник, доступний на будь-якому пристрої спільно зі статусом присутності, дозволяє швидко зв'язатися з потрібною людиною.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

Одним кличем можна відправити миттєве повідомлення, передзвонити колезі, перемкнутися в режим відеодзвінка, зібрати аудіоконференцію або спільно попрацювати з документом по Webex. Сервіси IP-АТМ ідеально розширюють можливості інших корпоративних інформаційних систем. Наприклад, дзвінки колегам можна робити прямо з довідника співробітників на порталі або до нової системи підключити застарілі аналогові АТМ.

Для тих співробітників, які більшу частину часу планують «тримати зв'язок» зі своїх мобільних пристроїв, фахівці СТІ запропонували повнофункціональний програмний клієнт Cisco Jabber, що підтримує всього функціонала УфК, включаючи відеозв'язок. Для тих співробітників, які переміщуються по офісі, також рекомендований Cisco Jabber плюс USB-гарнітура – для використання на робочих місцях. Таким чином, наше рішення не вимагає будівництва й змісту додаткової інфраструктури DECT, також немає необхідності закуповувати Wi-Fi-трубки – досить використовувати існуючі смартфони співробітників із установленим Cisco Jabber.

Надомним співробітникам, які будуть використовувати для голосового зв'язку свої ПК, рекомендований клієнт Cisco Jabber Desktop з набором наступних сервісів: голос, відео, обмін миттєвими повідомленнями, статус присутності, а також доступність по єдиному номері. На додаток кожному такому користувачеві пропонується використовувати USB-гарнітуру. «Стаціонарним» офісним користувачам фахівці СТІ радять Cisco Jabber Desktop + IP-телефон Cisco 7821 з настільки ж повним набором сервісів.

Нарешті, топ-менеджери замовника одержать термінали Cisco DX80, які представники СТІ охарактеризували як «найсучасніше на даний момент настільне рішення для комунікацій». Ці пристрої із сенсорними екранами розміром 23 дюйма є повноцінними терміналами ВКЗ із камерами Full HD (дозвіл 1080p). У цілому рішення від СТІ готово до проведення багатокористувальницьких відеоконференцій. При цьому користувачі клієнтів

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Jabber одержують можливість підключатися до багатоточечних конференцій разом з відеотерміналами, які будуть установлені в переговорні.

Пропозиція СТІ виглядає одним із самих функціональних і привабливих. На жаль, компанія не надала замовникові розцінки. Один тільки термінал DX80 коштує близько 4 тис. доларів, що більше всіх витрат на термінальне встаткування, передбачених пропозицією Mango Office.

Пропозиція MANGO

Компанія запропонувала рішення завдань замовника на основі віртуальної АТМ (ВАТМ) Mango Office. Як і просив замовник, йому був запропонований єдиний 100-канальний номер для всіх офісів. Але фахівці Mango рекомендували використовувати єдиний місцевий багатоканальний номер у кожному місті присутності й, при необхідності, номер 800. Дзвінок на кожній з номерів зможе прийняти будь-який співробітник компанії, незалежно від його місцезнаходження. На всіх номерах будуть доступні послуги місцевого й міжміського/міжнародного зв'язку. Внутрішній зв'язок між співробітниками, у тому числі тими, які знаходяться в різних містах, буде безкоштовною.

Базова функціональність ВАТМ досить багата: дворівневий IVR, голосова пошта, конференц-зв'язок, запис розмов, повідомлення абонентів про розрахунковий час до відповіді оператора, підключення номерів інших операторів, АВН (на вибір відображається номер, на який подзвонив абонент, або номер абонента), чорні/білі списки, дзвінок із сайту, приймання факсу на електронну пошту. Співробітників можна розбити на групи й використовувати в кожній з них один з п'яти алгоритмів розподілу викликів, схема якого може залежати від часу доби, номера що подзвонив, історії дзвінків абонентів (переадресація на співробітника, з яким недавно розмовляв). Для співробітників і груп надається докладна статистика по дзвінках: прийняті, зроблені, пропущені виклики, навантаження по годинниках, середня швидкість відповіді, середній час розмови.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

Налаштування ВАТМ, прослуховування записів розмови й доступ до інформаційно-аналітичних сервісів здійснюються через простий Web-інтерфейс особистого кабінету – спеціальні технічні знання для цього не потрібні.

У своїй пропозиції Mango докладно розписала переваги додаткового функціонала, реалізованого за допомогою її сервісу «Центр обробки викликів» (ЦОВ). Його зручно використовувати відділу продажів або обслуговування клієнтів, секретарям у приймальні, а також співробітникам, що обробляють великий потік дзвінків. Керівник відділу може контролювати роботу співробітників у режимі реального часу: відслідковувати їхній статус і кількість працюючих на лінії, виводити на лінію додаткових операторів у випадку зростання навантаження, бачити, хто з ким і як довго розмовляє, підключатися до розмов у режимі прослуховування, конференції або суфлювання. Крім того, керівник може одержати підсумкову статистику роботи співробітників.

У розглянутому випадку можливості контролю роботи співробітників будуть, швидше за все, дуже корисні для відстеження дій віддалених співробітників, що працюють вдома. У свою чергу, всі співробітники одержують можливість бачити черга дзвінків у реальному часі, хто з колег на лінії й хто вільний для перекладу дзвінка, перехоплювати виклики із черги, міняти свій статус («На лінії», «Перерва», «Не турбувати»). Дзвінки переадресуються, відбуваються й приймаються кличем миші, при цьому може використовуватися будь-який телефонний апарат або софтфон. При надходженні дзвінка на екрані спливає картка того, хто подзвонив з інформацією про нього й історією контактів.

Для співробітників, що працюють в офісі на фіксованих робочих місцях, найзручніший термінал, на думку фахівців Mango, – стаціонарний SIP-телефон. Правда, це й найдорожчий варіант. Такі телефони для 30 офісних користувачів можна вибрати в інтернет-магазині Mango Office. Вартість найдешевшої моделі (Panasonic KX-UT113) – 2900 грн. Популярні моделі телефонів – Yealink SIP-T21 (4100 грн.) і Grandstream GXP1400 (3100 грн.), останній варіант і включений у пропозицію для замовника. Інші можливі варіанти для офісних працівників –

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

звичайний аналоговий телефон, що підключається через шлюз (вартість починається від приблизно 1300 грн. за порт) або софтфон для ПК. Останній варіант, очевидно, буде основним для надомних працівників. Власники ПК із ОС Windows можуть установити безкоштовний софтфон Mango Talker, для MacOS є безкоштовні софтфони Telephone і Linphone, а для Linux – Zoiper, також доступний ряд інших додатків.

Співробітники, що переміщаються по офісі, можуть використовувати софтфони (Android або iOS), що працюють через офісну мережу Wi-Fi (безкоштовний варіант), або, що набагато зручніше, DECT-SIP-телефони. Популярна модель – Panasonic TGP500 (3800 грн.), що підтримує до шести трубок і до трьох одночасних розмов.

Співробітники, які планують працювати в основному з мобільних пристроїв, можуть використовувати SIP-софтфони або GSM-зв'язок. Фахівці Mango рекомендують для Android такі безкоштовні софтфони, як CSIPSimple, Linphone, Zoiper і Bria, для iOS – Zoiper і Linphone, також доступні й інші варіанти. Всі функції ВАТМ будуть підтримуватися. Якщо в місці знаходження співробітника IP-Зв'язок поганої якості, то можна приймати дзвінки, що надходять на єдиний номер ВАТМ, по GSM. Повний функціонал ВАТМ буде підтримуватися для вхідних дзвінків.

Mango Office надає три інтегровані хмарні сервіси – ВАТМ, ЦОВ і CRM-систему. Якщо компанія вирішить підключити CRM, то об'єднати її з ВАТМ не важко буде. Всі дзвінки будуть ураховуватися в CRM, через її інтерфейс можна буде зручно шукати й прослуховувати запису розмов, що ставляться до обраного клієнта або співробітника. Спливаюча картка надасть співробітникові повну інформацію про клієнта при дзвінку, дозволить оперативно внести дані за підсумками розмови й запланувати подальшу роботу.

Відеозв'язок між співробітниками може бути здійснена в рамках поточного проекту, якщо придбати відеотермінали SIP (наприклад, стаціонарні SIP-відеотелефони).

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

У числі численних переваг свого рішення фахівці Mango називають принцип «одного вікна»: ВАТМ, телефонні номери, послуги зв'язку, телефонні апарати (для допомоги в налаштуванні телефонних апаратів можна замовити виїзд інженера) плюс просте підключення додаткових систем ЦОВ і CRM з інтеграцією сервісів без додаткової плати й програмних доробок. Крім того, вони вказують на підвищену відказостійкість рішення: у випадку аварій (наприклад, відключення електроживлення в одному з офісів) дзвінки будуть автоматично переадресовуватися через альтернативні канали зв'язку (наприклад, через стільникових операторів) або направлятися співробітникам інших офісів. За твердженням Mango, налаштування й адміністрування ВАТМ не вимагають залучення сторонніх кваліфікованих фахівців.

Пропозиція PANASONIC

Компанія запропонувала замовникові реалізувати систему зв'язку на базі IP-платформ (IP-АТМ) Panasonic серії KX-NS, які відрізняються багатою функціональністю, а також широкими можливостями по оптимізації витрат на зв'язок усередині організації. Розроблене рішення дозволить мати єдиний корпоративний довідник і номерний план, централізовано управляти всіма АТМ через Web-інтерфейс, забезпечити мобільність співробітників, одержувати докладну статистику по дзвінках і виконати інші зазначені в завданні завдання.

У центральному офісі запропоновано встановити IP-АТМ KX-NS1000, а у філіях – IP-АТМ KX-NS500. Як відзначають представники Panasonic, одним з основних переваг їх IP-АТМ є можливість швидкого розширення за рахунок модульної структури й наступності поколінь устаткування. Нові IP-платформи сумісні з усім телекомунікаційним устаткуванням Panasonic, у тому числі з телефонними апаратами. Раніше встановлену АТМ можна використовувати й у якості стекового шлюзу для розширення ємності й здійснення дзвінків по каналах традиційної телефонії.

IP-АТМ KX-NS1000 дозволяє підключити до 256 зовнішніх ліній з різними типами сигналізації: зокрема, до 256 IP-Ліній по протоколі SIP і до 96 зовнішніх IP-Ліній по протоколі H.323. Система підтримує до 640 внутрішніх

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

ліній і стільки ж SIP/ IP-телефонів, а при необхідності до неї можна підключити до 512 мікростільникових телефонів DECT і 64 базові IP- або SIP-станції. Очевидно, що така функціональність забезпечує широкий простір для розвитку системи замовника.

Малі IP-платформи IP-АТМ КХ-NS500, які планується встановити в офісах, сполучать переваги традиційної й IP-телефонії, а також UC-серверів (КХ-NS1000). На думку фахівців Panasonic, це ідеальне рішення для малого й середнього бізнесу завдяки можливості розширення ємності із шести базових аналогових ліній і 18 абонентів до 288 внутрішніх ліній. В КХ-NS500 реалізований широкий набір функцій, які можуть бути доповнені відповідно до конкретних завдань клієнта. Зокрема, убудована система уніфікованих комунікацій дозволяє використовувати найбільш корисні функції на базі IP-телефонії. У системі також передвстановлені розширена голосова пошта й додатки для створення невеликого колл-центра.

Для того щоб співробітники компанії могли вільно переміщатися по офісі, у кожному місті повинні бути встановлені базові IP-станції для розгортання мікростільникової системи бездротового зв'язку. У центральному офісі як базова станція запропонована використовувати КХ-NCP058, а в якості системних DECT-телефонів – моделі серії КХ-ТСА.

Для всіх офісів запропоновано використовувати СТІ-додаток Communication Assistant – багатофункціональну програму зі зручним інтерфейсом, що дозволяє одержувати й обробляти дзвінки, а також управляти багатьма іншими сервісами IP-АТМ Panasonic. Цей додаток класу «клієнт – сервер», при цьому його серверна частина вже убудована в IP-АТМ серії КХ-NS. Замовникові потрібно тільки встановити на комп'ютери співробітників клієнтську частину. Communication Assistant підтримує інтеграцію з різними CRM-додатками.

Для організації телефонних конференцій у філіях рекомендовані IP-конференц-телефони КХ-NT700. Відповідно до проведених Panasonic тестам, цей конференц-телефон забезпечує краща якість звуку в порівнянні з аналогічним

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

устаткуванням даного сегмента, представленим на українському ринку. Важлива відмітна риса KX-NT700 – можливість перетворення мови в реальному часі. Залежно від ситуації ця опція дозволяє, наприклад, сповільнити прослуховування розмови за рахунок скорочення пауз у мові й у такий спосіб поліпшити сприйняття співрозмовника. Завдяки убудованому ретранслятору в моделях серії KX-NS, IP-телефони можна встановлювати не тільки безпосередньо в офісі, але й будинку у віддалених співробітників, а також включати в мережу мобільний персонал, що використовує софтли на своїх смартфонах, – останні також реєструються як внутрішні абоненти корпоративної мережі. Нові IP-платформи Panasonic дозволяють привласнювати кожному співробітникові компанії короткий єдиний внутрішній номер, на який можна записати до трьох телефонів – як мобільний або SIP, так і традиційний. Для зв'язку з абонентом досить набрати його короткий внутрішній номер, і він відповість по кожному зі зручних йому апаратів незалежно від того, де саме перебуває. Як відзначено в пропозиції Panasonic, реалізація подібних проєктів – технічно складне завдання, один із ключових факторів її успішного рішення – установка й налаштування системи кваліфікованими фахівцями. Фахівці компанії завжди рекомендують клієнтам звертатися до партнерів, що мають статус технічного центра АТМ Panasonic. У цьому випадку установка й технічне обслуговування будуть проведені на найвищому рівні, а на все встаткування надана п'ятирічна гарантія. У компанії думають, що багаторічний досвід Panasonic в області корпоративної телефонії, сучасні термінали з високочутливими мікрофонами, широкополосними динаміками й різноманітними можливостями забезпечують організації оптимальне за вартістю й можливостями рішення. У пропозиції Panasonic немає інформації з вартості запропонованих рішень, але зазначено, що в середньому вартість власної АТМ у перерахуванні на 20 абонентів приблизно збігається з витратами на оплату послуг хмарної АТМ протягом року за тих же абонентів, а після закінчення першого року вартість володіння власної IP-АТМ стає нижче, ніж хмарної.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

Embarcadero Delphi, раніше Borland Delphi і Codegear Delphi, – інтегроване середовище розробки ПЗ для Microsoft Windows, Mac OS, iOS і Android мовою Delphi (що раніше носила назву Object Pascal), створена спочатку фірмою Borland і на даний момент приналежна й розроблювальна Embarcadero Technologies. Embarcadero Delphi є частиною пакета Embarcadero RAD Studio і поставляється в чотирьох редакціях: Community (поширюється безкоштовно й має обмежену ліцензію на використання в комерційних цілях), Professional, Enterprise і Architect.

Delphi 10.4 Sydney

Випущено 26 травня 2020 року. RAD Studio Delphi 10.4 забезпечує значно поліпшену високопродуктивну нативну підтримку Windows, кращу продуктивність розробки, миттєві підказки code completion, прискорення виконання коду із синтаксисом керованих записів, поліпшення виконання паралельних завдань на сучасних багатоядерних CPU, а також містить більш 1000 виправлень багів, поліпшення продуктивності середовища й бібліотек і багато чого крім того.

Основні можливості Delphi 10.4.1:

– Істотні розширення для Windows: поліпшення для застосунків на моніторах 4K High DPI, інтеграція з новим WebView2 на базі Chromium, використання розширених title bars, таких же, як в Office, Explorer, Google Chrome.

– Керування пам'яттю в Delphi тепер стандартизоване на всіх підтримуваних платформах – мобільних, настільних і серверних – використовувачи класичну реалізацію керування пам'яттю об'єктів.

– Істотне поліпшення Delphi Code Insight (без можливого блокування IDE – в окремому процесі), що допоможе при роботі з великими проектами.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

– Тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання.

– Розширена підтримка бібліотек C++: ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode.

– Відладник Win 64 (на LLDB) і збирач для C++.

– Поліпшення для C++: Включена велика кількість поліпшень STL з Dinkumware.

– Підтримка Metal Driver GPU для macOS і iOS.

– Вбудований Fmxlinux.

– Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.

Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Реалізований заново стилізуємий FMX компонент TМето на платформі Windows значно поліпшений і тепер має відмінну підтримку ІМЕ.

– Численні поліпшення швидкості й стабільності роботи нашої бібліотеки The Parallel Programming Library (PPL).

– Додані оновлені драйвери для FireBird, PostgreSQL і SQLite.

– Клієнтські бібліотеки HTTP і REST Client розширені застосунковими можливостями роботи з HTTPS. Також були розширені можливості підтримки Amazon AWS services

– У технологію Visual LiveBindings внесена безліч поліпшень, у тому числі швидкодії, що стосуються, застосунків на VCL і FireMonkey

RAD Studio 10.4 Короткий огляд:

– Істотні розширення для Windows. Створення застосунків, що чудово виглядають, із чіткими елементами інтерфейсу на 4k моніторах High DPI за допомогою нової гнучкої підтримки стилів елементів керування на екрані. Інтеграція із сучасними, безпечними web-технологіями від Microsoft – новим WebView2 на базі Chromium. Використання сучасних розширених title bars, таких же, як в Office, Explorer, Google Chrome, у своїх проектах. Істотні поліпшення надійності налагодження в новому відладнику для C++ Windows 64-bit.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

– Зросла продуктивність розробки. Ріст продуктивності за рахунок миттєвої реакції підказок code completion у середовищі IDE. Краща сумісність із уже наявною кодовою базою, і спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю. Швидке зв'язування даних і візуальних елементів за допомогою розширеної технології Visual LiveBindings з підвищеною швидкодією. Просте використання розповсюджених бібліотек C++, наприклад, ZeroMQ, SDL2, SOCI, libSIMDpp і Nematode. Оновлена підтримка Amazon AWS cloud.

– Поліпшення швидкодії і якості. Більш 1000 поліпшень швидкодії і якості. Краща ефективність коду за допомогою нового синтаксису custom managed records. Більш швидке виконання паралельних завдань на сучасних багатоядерних CPU. Переконаєтеся в прискоренні відображення на екрані з підтримкою Metal API на macOS і iOS. Краща сумісність із уже наявною кодовою базою й спрощення програмування за рахунок уніфікованої архітектури керування пам'яттю.

Істотне поліпшення Delphi Code Insight

Як найбільше й головне поліпшення інструментів програмування Delphi за багато років, в 10.4 Delphi Code Insight реалізований через Language Server Protocol (LSP). LSP – це технологія генерації результатів для code completion, навігації й інших сервісів в окремому процесі. Це значить, що code completion і Code Insight одержать більш точні результати без блокування IDE. 10.4 забезпечує набагато більш високу продуктивність розроблювачів, які працюють із більшими проектами, що містять мільйони рядків коду.

Delphi Custom Managed Records

Ключове розширення мови Delphi: тип даних Delphi «record» тепер підтримуть довільні ініціалізацію, фіналізацію й операції копіювання. Управляйте тем, як ці структури створюються, копіюються й звільняються з допомогу вашого коду, який буде виконуватися у відповідний момент. Це

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Поліпшена кроссплатформеність

- Додана підтримка Metal Driver GPU для macOS і iOS.
- Крім підтримки останнього iOS SDK, в RAD Studio 10.4 розроблювачі можуть задовольнити нові вимоги Apple до набору стартових екранів.
- Реалізований заново стилізуємий FMX компонент TМемо на платформі Windows значно поліпшений і тепер має відмінну підтримку IME.
- Користувачам редакцій Enterprise або Architect доступна повна інтеграція Fmxlinux з IDE для створення клієнтських застосунків Linux з GUI.
- Компонент Twebbrowser для iOS тепер реалізований на Wkwebview API.
- Реалізація компонента Media Player для macOS тепер використовує Avfoundation.

Оновлений менеджер пакетів Getit

Менеджер пакетів Getit в IDE був значно вдосконалений.

Дати випуску релізів пакетів тепер видні, і можливе сортування списку по цих датах; відбір тільки встановлених пакетів, контенту, доступного тільки при наявності підписки, багато чого іншого.

Універсальний інсталятор для установки Online і Offline

В 10.4 включений новий універсальний інсталятор, який використовує технологію на базі Getit. Цей інсталятор підтримує як online, так і offline (з ISO) варіанти установки.

Тепер обоє варіанта установки дозволяють вам указати початковий набір можливостей RAD Studio для установки, наприклад, свою комбінацію мов програмування й цільових платформ, мов інтерфейсу, і додавати до нього або видаляти непотрібне в будь-який момент.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускні кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи комунікації Cloud-сервісів з використанням SIP/VoIP.

В процесі розробки випускної кваліфікаційної роботи за другим (магістерським) 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.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Хоча ще довго TDM-телефонія буде як і раніше кращою для деяких ситуацій, в останні роки спостерігається масовий перехід на IP-телефонію. По оцінці J'son & Partners Consulting, якщо в 2022 році в Україні налічувалося 11 млн користувачів IP-телефонії, включаючи користувачів P2P-телефонії на зразок Skype (це відповідає рівню проникнення близько 14%), то в 2023 році ці цифри виросли приблизно до 14 млн (17,3%), а до кінця 2016 року очікується уже 17 млн (23,5%). Основними факторами росту ринку VoIP в Україні залишаються подальший розвиток ринку широкополосного доступу (включаючи мобільний, у тому числі за технологією LTE) і збільшення активності українських і міжнародних провайдерів IP-телефонії.

Стимулами до впровадження VoIP у системах корпоративної телефонії служать скорочення витрат на зв'язок між віддаленими офісами, можливість розширення функціонала систем за рахунок розробки нових додатків і інтеграція з іншими бізнес-додатками.

На українському ринку в першу чергу затребувані технологічні рішення по організації й об'єднанню в єдину мережу віддалених офісів, а також системи для організації колективної роботи. Збільшення попиту на IP-телефонію в корпоративному секторі відбувається не тільки через потребу скорочення витрат на регіональний зв'язок, але й завдяки непрямим економічним вигодам від її впровадження, у тому числі росту продуктивності праці після підключення додаткових сервісів. Високий ступінь гнучкості й відкритості IP-систем, можливості їхнього використання як єдина платформа для різних комунікаційних рішень (голос, відео й дані), провідних і бездротових технологій стимулюють підвищення інтересу до IP-телефонії.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Платформи SIP/VoIP дозволяють будувати високонадійні уніфіковані телефонні мережі із широким спектром можливостей. Серед них – короткі номери для дзвінків між офісами, гнучка переадресація, багатоточечні конференції. Абоненти можуть робити й приймати виклики з комп'ютера й смартфону, одержувати по електронній пошті оповіщення про пропущені дзвінки й т.д. Все це зробить користування корпоративною телефонією більше зручним.

Перехід на VoIP зачіпає й приватних осіб. У квартирах установлюється встаткування GPON, що дозволяє крім телефонії одержати широкий спектр послуг зв'язку: домашній Інтернет на швидкості до 200 Мбіт/с, інтерактивне телебачення, а в перспективі – сервіси відеоспостереження й керування комунальними ресурсами. Проект початий в 2022 році й, відповідно до планів, буде завершений в 2024-м.

Крім того 2023 рік, безсумнівно, був роком BYOD («принеси свій власний пристрій») і навіть BYOA («принеси свій додаток») – роком проникнення в офіси сторонніх додатків і хмарних сервісів.

Ця тенденція зміцніє в 2016 році, а забезпечення ефективного й безпечного керування BYOA стане головним завданням ІТ-директорів і системних адміністраторів. Інструменти для спільної роботи в реальному часі будуть другим об'єктом їхньої пильної уваги.

По прогнозом IDC, у році, що наступив, витрати на ІТ по усьому світі перевищать 2,1 трлн доларів, що на 5,7% більше показників 2022 року, а самим значимим драйвером галузі стане мобільність: продажу смартфонів і планшетів виростуть на 20%, згенерував 20% продажів в ІТ. В Україні вже близько 12 млн чоловік виходять в Інтернет з мобільних пристроїв. У сукупності з переходом на VoIP і SIP цей фактор буде впливати й на ринок корпоративної телефонії.

В Citrix виділяють наступні основні тенденції 2015 року в корпоративному сегменті: відмова від настільних ПК, перехід на мобільний стиль роботи, усе більше широке використання хмарних сервісів і додатків. Мобільність – загальне поняття, що описує, як компанії змінюють принципи ведення бізнесу, освоюючи

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

нові технології, будь то BYOD або віддалене співробітництво. В 2016 році багато компаній із всіх сегментів ринку стануть інвестувати в інструменти, необхідні для організації повноцінної мобільності співробітників.

Залежно від специфічних потреб компаній мобільність буде розумітися ними по-різному, але очевидно, що дана тенденція охопить усіх. З іншого боку, ті, хто вже вклався в ці технології, почнуть шукати нові можливості, щоб надати ще більше волі своїм співробітникам. За прогнозами Forrester, в 2024 році можна чекати інтеграції мобільних додатків і хмарних сервісів, а формула «хмари + мобільність» дасть користувачам щось набагато більше, ніж сума цих складових.

Крім того, на ринок корпоративної телефонії впливають потреба замовників у розширенні функціонала мобільного зв'язку, мобільного й бездротового доступу усередині офісу й за його межами, бажання звертатися до звичних функцій УАТС при віддаленій роботі й більш широко використовувати протокол SIP, а також зростаючий інтерес до відеотелефонії й додаткових функцій. Найбільшим потенціалом на ринку корпоративного зв'язку володіють додатки, що збагачують функції телефонних систем. Прикладами є додатки відео-конференц-зв'язку, рішення для обліку й аналізу внутрікорпоративних витрат на відповідні послуги.

За прогнозами Avaya, у найближчі рік-півтора чимало цікавого з'явиться в області відеокommунікацій, використання відео в бізнес-процесах (відеоспівробітництво). Якщо в 2022 році відео ще не входило в обов'язковий список засобів для ділового спілкування, то в 2023 році відбулася помітна зміна в їхньому сприйнятті. Сьогодні відео – це просто доступна технологія. Люди з комфортом використовують його й активно впроваджують у роботу. Провідні вендори можуть забезпечити «відео в одне торкання» на екрані планшета або смартфона: 2023 рік змінив правила гри.

В Cisco підкреслюють, що в усьому світі спільна робота за допомогою різних пристроїв у будь-якому місці й у будь-який час поступово стає самим затребуваним способом співробітництва. А коли для цього залучаються

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

своїх інфокомунікаційних систем, одержати все необхідне від провайдеру хмарної АТМ, у тому числі додаткові функції.

Запущений хмарний сервіс для малого й середнього бізнесу Business 365 з убудованими функціями VoIP телефонії й інтерфейсом у стилі соціальних мереж – ще один характерний приклад сучасних комунікаційних рішень. Ця розробка поєднує в собі функції трьох систем: керування компанією, спільної роботи й продажів. А українські клієнти можуть скористатися «уніфікованими комунікаціями як послугою» (UaaS), розгорнутими на платформі Cisco.

Керовані комунікаційні сервіси хмарної телефонії, ВКЗ, уніфікованих комунікацій і спільної роботи над проектами по моделі SaaS, що набирають популярність за кордоном, стають доступними й для українських замовників. Як прогноують аналітики ABI Research, на світовому ринку в найближчі роки в результаті масової міграції корпоративних комунікацій у хмару електронна пошта, телефонія, аудіо-, відео- і Web- конференції стануть реалізовуватися як хмарні рішення.

Згідно із прогнозами, уже до 2030 року більше 40% корпоративних комунікаційних додатків будуть перенесені в хмари. Про широке поширення аутсорсингу телефонних послуг і хмарної телефонії в Україні говорити ще рано, але це одна з найбільш яскравих тенденцій.

Замовниками хмарних телекомунікаційних сервісів стають організації, бізнес яких характеризується високими темпами росту або пікових сплесків активності. Вони затребувані проектними групами будь-якого розміру, компаніями з територіально розподіленою структурою, а також окремими користувачами у випадку застосування таких схем, як надомна робота або часткова зайнятість. У міру збільшення попиту оператори будуть доповнювати комунікаційні сервіси послугами аутсорсингу корпоративних додатків.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

3.2 Розробка структурної схеми

Для перевірки коректності реалізації даної задачі було виконано багато розрахунків та експериментальних матеріалів. Цьому питанню приділялась особлива увага тому, що помилка при розрахунку привела б до ряду негативних наслідків. Відлагодження та перевірка, що підтверджує вірність програмних рішень відбувалась за декількома етапами:

- математична перевірка окремих модулів;
- математична перевірка всієї системи (з допомогою математичної логіки будується логічна схема всієї системи);
- практична перевірка підпрограм (перевіряється процедурна частина кожної підпрограми окремо);
- практична перевірка всієї системи у дії (перевіряється ситема в цілому за допомогою вводу різних даних у програму, потім на виході з програми перевіряємо отриману інформацію з очікуваною).

Для підтвердження правильності розрахунку програми були використані експерементальні дані різних аудіо форматів, були проведені консультації з даного питання зі спеціалістами.

Простота мови проектування та маніпулювання даними, зручність спілкування користувача з системою до мінімуму вивчення цієї програми. Користувач програми – це людина, яка повинна володіти азами програмування. При написанні програми я намагалася, щоб програма відповідала наступним параметрам:

- Швидкодія. Програма працює постійно з великою кількістю кінцевих абонентів (селективний зв'язок).
 - Захист. Забезпечити надійний захищений канал зв'язку.
 - Відсутність проблеми дороговизни сучасних персональних комп'ютерів.
- Система, що написана може встановлюватись на будь-якому персональному комп'ютері – використовувати відносно швидкі алгоритми захисту зв'язку.
- Можливість зручно і швидко формувати приклади і теорію для користувача.

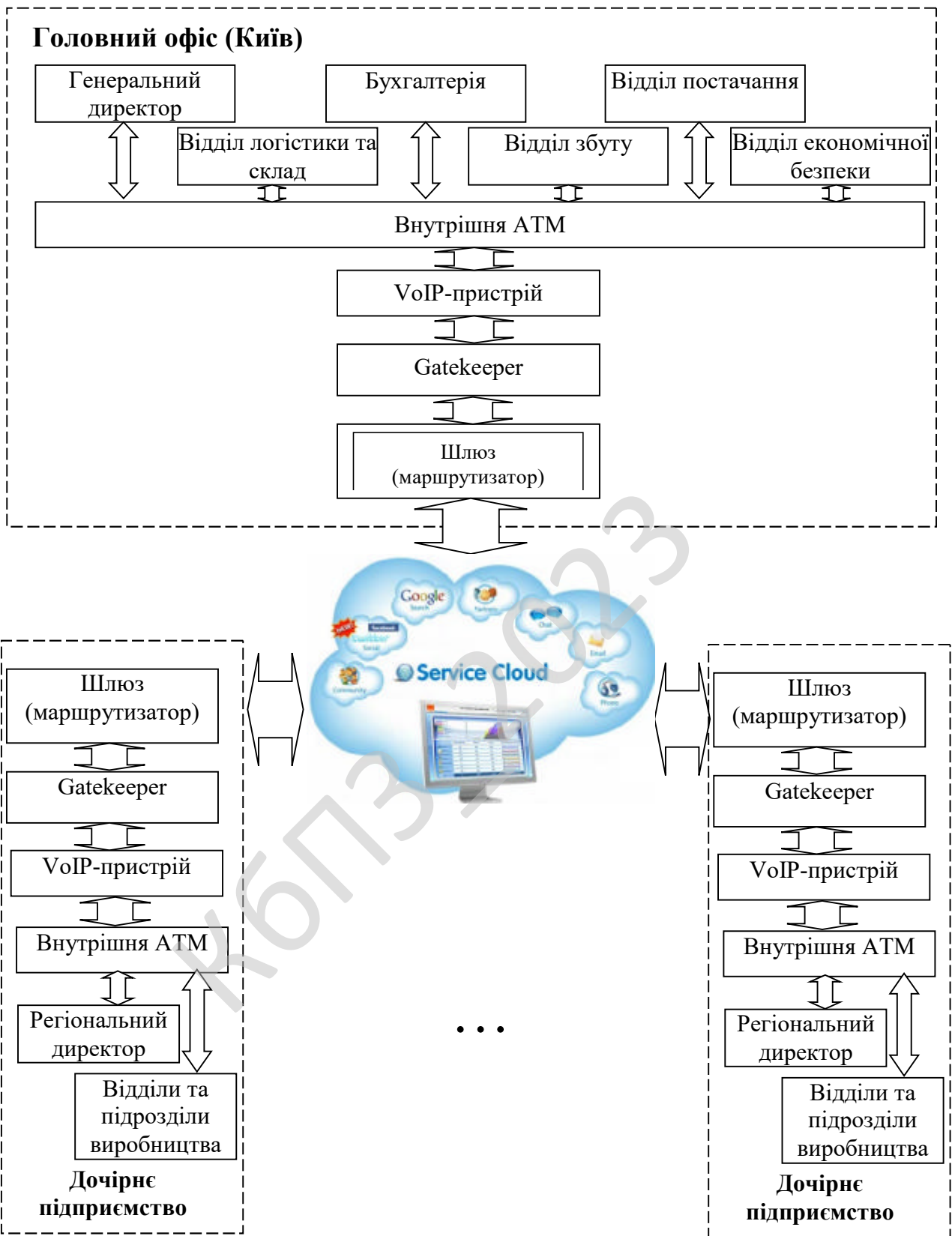


Рисунок 3.1 – Структурна схема системи

Завдяки структурній схемі можна чітко побачити основні структурні блоки системи та взаємозв'язки між ними. При розробці структурної схеми основний упор робився на існуючі розробки ПЗ і їх модулі допомоги. Аналіз рисунка 3.1 дозволяє чітко прослідити як працює програма. Розглянемо схему зверху вниз, в напрямку від пристрою до кінцевої програми – за допомогою внутрішнього пула доступу що забезпечує закриті канали зв'язку абоненти генеральний директор, відділи бухгалтерії, логістики, складу, збуту, економічної безпеки – можуть взаємодіяти з дочірніми виробниками з використанням внутрішньої АТМ. і чітко налагодженої технічної системи взаємодії. Технічна частина взаємодії забезпечується VoIP та воратарем з використанням маршрутизатора.

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

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

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

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

з'єднання з абонентом або закінчення з'єднання з абонентом. На підставі даних подій активізуються процедури контролю допустимості даних.

Модуль “Головне меню” призначений для запуску основних процедур програми і завершення роботи з програмою. Схема основного меню розглянута нижче у структурній схемі системи керування (рисунок 3.4).

Модуль роботи з довідниками містить у собі два довідники:

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

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

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

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

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

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

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

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

- Поле введення у формі може представляти список допустимих значень, з яких користувачі можуть легко вибрати потрібне.
- Область форми може виводити шаблон, що відповідає поточної виведеної у формі запису.
- Командні кнопки у формі можуть виконувати дії, зв'язані з виведеної у формі поточною записом.

3.3 Розробка функціональної схеми

Н.323

Як було обґрунтовано раніше, сучасна ІР-телефонія будується на основі протоколу Н.323. Розглянемо більш докладно цей протокол. У рекомендаціях, що входять у сімейство Н.323, визначені протоколи, методи й мережні елементи, необхідні для організації мультимедійного зв'язку між двома або більше користувачами [15-22].

Найбільш затребуваною з послуг, специфікованих у рекомендації Н.323, є послуга передачі мовної інформації з мереж з маршрутизацією пакетів ІР. Найпоширенішим підходом до побудови мереж ІР-телефонії сьогодні є саме підхід, запропонований ІТУ-Т у рекомендації Н.323.

Сімейство протоколів Н.323 містить у собі три основних протоколи: протокол взаємодії кінцевого устаткування з gatekeeper – RAS, протокол керування з'єднаннями – Н.225 і протокол керування логічними каналами – Н.245. Ці три протоколи, разом з Інтернет-протоколами TCP/IP, UDP, RTP і RTCP, а також з описаним в [6] протоколом Q.931 складають основу технології ІР-телефонії. Суть ієрархії цих протоколів полягає в наступному. Для переносу сигнальних повідомлень Н.225 і керуючих повідомлень Н.245 використовується протокол з встановленням з'єднання й з гарантованою доставкою інформації – TCP. Сигнальні повідомлення RAS переносяться протоколом з негарантованою доставкою інформації – UDP. Протокол RAS забезпечує контроль використання

мережних ресурсів, підтримує автентифікацію користувачів і може забезпечувати нарахування плати за послуги. Для переносу мовної і відеоінформації використовується протокол передачі інформації в реальному часі – RTP. Контроль переносу користувальницької інформації виробляється протоколом RTCP. Процедура встановлення з'єднання в таких мережах IP-телефонії базується на рекомендації Q.931 і аналогічна процедурі, використовуваної в ISDN.

Основними пристроями мережі є: термінал, шлюз, gatekeeper і пристрій керування конференціями.

Термінал H.323 – це кінцевий пристрій мережі IP-телефонії, що забезпечує двосторонній мовний або мультимедійний зв'язок з іншим терміналом, шлюзом або пристроєм керування конференціями. Користувальницький інтерфейс керування системою дає користувачеві можливість створювати й приймати виклики, а також конфігурувати систему й контролювати її роботу.

Таблиця. 3.1 – Сімейство протоколів H.323

Гарантована доставка інформації із протоколу TCP		Негарантована доставка інформації із протоколу UDP		
H.245	H.225	Потоки мови й відеоінформації		
	Керування з'єднанням (Q.931)	RAS	RTCP	RTP
TCP		UDP		
IP				
Канальний рівень				
Фізичний рівень				

Модуль керування підтримує три види сигналізації: H.225, H.245 і RAS. Цей модуль забезпечує реєстрацію терміналу у gatekeeper, установлення й завершення з'єднання, обмін інформацією, необхідної для відкриття мовних каналів, надання додаткових послуг і техобслуговування.

Телематичні додатки забезпечують передачу користувальницьких даних, нерушливих зображень і файлів, доступ до баз даних і т.п. Стандартним протоколом для підтримки таких додатків є протокол Т. 120.

Модуль Н.225.0 відповідає за перетворення відеоінформації, мови, даних і сигнальної інформації у вид, придатний для передачі по мережах з маршрутизацією пакетів ІР, і за зворотне перетворення. Крім того, функціями модуля є розбивка інформації на логічні кадри, нумерація послідовно переданих кадрів, виявлення й корекція помилок.

Мережний інтерфейс забезпечує гарантовану передачу керуючих повідомлень Н.245, сигнальних повідомлень Н.225.0 (Q.931) і користувальницьких даних за допомогою протоколу TCP і негарантовану передачу мовної й відеоінформації, а також повідомлень RAS, за допомогою протоколу UDP.

Блок синхронізації вносить затримку на прийомній стороні з метою забезпечити синхронізацію джерела інформації з її приймачем, узгодження мовних і відеоканалів або згладжування варіації затримки інформації.

Відеокодеки кодують відеоінформацію, що надходить від зовнішнього джерела відеосигналів (відеокамери або відеомагнітофона), для її передачі по мережі з маршрутизацією пакетів ІР і декодують сигнали, що надходять із мережі, для наступного відображення відеоінформації на моніторі або телевізорі.

Аудіокодеки кодують аудіоінформацію, що надходить від мікрофона (або інших джерел аудіоінформації), для її передачі по мережі з маршрутизацією пакетів ІР і декодують сигнали, що надходять із мережі, для відтворення.

Слід зазначити, що при організації децентралізованої конференції термінал Н.323 може приймати більш ніж один потік мовної інформації. У цьому випадку термінал повинен уміти змішувати або перемикати пакетовану мову, що надходить від інших учасників конференції.

Основною функцією шлюзу Н.323 є перетворення мовної (мультимедійної) інформації, що надходить із боку ТфОП з постійною швидкістю, у вид,

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

придатний для передачі по IP-мережах.

При відсутності в мережі gatekeeper повинна бути реалізована ще одна функція шлюзу – перетворення номера ТфОП у транспортну адресу IP-Мережі.

З боку мереж з маршрутизацією пакетів IP, так само, як і з боку ТфОП, шлюз може брати участь у з'єднаннях як термінал або пристрій керування конференціями. У випадку, коли термінал Н.323 зв'язується з іншим терміналом Н.323, розташованим у ті ж самій IP-мережі, шлюз у цьому з'єднанні не бере участь.

Шлюз, у сукупності з gatekeeper мережі IP-телефонії, утворить універсальну платформу для надання всього спектра послуг зв'язку.

У gatekeeper зосереджений весь інтелект мереж IP-телефонії, що базуються на рекомендації ITU Н.323. Мережа Н.323 має зонну архітектуру. Gatekeeper виконує функції керування зоною мережі IP-телефонії, у яку входять термінали, шлюзи й пристрої керування конференціями, зареєстровані в цього gatekeeper. Різні ділянки зони мережі Н.323 можуть бути територіально рознесені й з'єднуватися один з одним через маршрутизатори. Варто звернути увагу на те, що комутатори кадрів Ethernet і маршрутизатори пакетів IP не є мережними елементами Н.323, тому що вони працюють на ланковому або мережному рівнях відповідно, у той час як устаткування Н.323 працює на прикладному рівні стека протоколів TCP/IP.

У число найбільш важливих функцій, виконуваних gatekeeper, входять:

- перетворення так званої alias-адреси (ім'я абонента, телефонного номера, адреси електронної пошти й ін.) у транспортну адресу мережі з маршрутизацією пакетів IP (IP адреса й номер порту TCP);
- контроль доступу користувачів системи до послуг IP-телефонії за допомогою сигналізації RAS (використовуються повідомлення ARQ/ACF/ARJ);
- контроль, керування й резервування пропускної здатності мережі;
- маршрутизація сигнальних повідомлень між терміналами, розташованими в одній зоні; gatekeeper може організовувати сигнальний канал

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

безпосередньо між терміналами або ретранслювати сигнальні повідомлення від одного терміналу до іншого.

У тому випадку, коли абонент, який викликає, знає IP-адресу терміналу абонента, який викликається, з'єднання між двома пристроями може бути встановлене без допомоги gatekeeper. Але, при наявності gatekeeper я забезпечується мобільність абонентів, тобто здатність користувача одержати доступ до послуг з будь-якого терміналу в будь-якому місці мережі й здатність мережі ідентифікувати користувачів при їхньому переміщенні з одного місця в інше. При відсутності в мережі gatekeeper перетворення адреси викликуваного абонента, що надходить із боку ТфОП у форматі E. 164, у транспортну адресу IP-мережі повинне виконуватися шлюзом. В одній мережі може перебувати декілька gatekeeper, які повинні взаємодіяти між собою. Слід особливо зазначити, що хоча gatekeeper є окремим логічним елементом мережі, він може бути реалізований у терміналі, у шлюзі, у пристрої керування конференціями або в пристроях, не специфікованих у рекомендації H.323.

Рекомендація H.323 передбачає три види конференцій.

Перший вид – централізована конференція, у якій кінцеві пристрої з'єднуються в режимі точка-точка із пристроєм керування конференціями (MCU), що контролює процес створення й завершення конференції, а також обробку потоків користувацької інформації.

Другий вид – децентралізована конференція, у якій кожний її учасник з'єднується з іншими учасниками в режимі точка – група точок, і кінцеві пристрої самі обробляють потоки інформації, що надходять від інших учасників.

Третій вид – змішана конференція, тобто комбінація двох попередніх видів.

Перевага централізованої конференції – порівняно прості вимоги до термінального встаткування, недолік – більша вартість пристрою керування конференціями.

Для децентралізованої конференції потрібно більше складне термінальне встаткування, крім того, бажано, щоб у мережі підтримувалася передача пакетів

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

IP у режимі багатоадресного розсилання (IP multicasting). Якщо мережа не підтримує цей режим, термінал може передавати інформацію до кожного з інших терміналів, що беруть участь у конференції, у режимі точка-точка, але це стає неефективним при числі учасників більше чотирьох. Пристрій керування конференціями MCU містить один обов'язковий елемент – контролер багатоточкових з'єднань – Multipoint controller (MC). Крім того, MCU може містити один або більше процесорів для обробки інформації користувачів при багатоточкових з'єднаннях – Multipoint processor (MP). Слід зазначити, що контролер MC і процесор MP є самостійними логічними пристроями H.323 і що контролер може існувати незалежно від процесора (зворотне невірно). Контролер може бути фізично сполучений з gatekeeper, зі шлюзом або з MCU, а MCU, у свою чергу, може бути сполучене зі шлюзом або з gatekeeper.

Контролер конференцій повинен використовуватися для організації конференції будь-якого виду. Він організує обмін між учасниками конференції даними про функціональні можливості їхніх терміналів, указує, у якому режимі (з використанням яких кодеків) учасники конференції можуть передавати інформацію, причому цей режим може змінюватися в ході конференції, наприклад, при підключенні до неї нового учасника. Таким чином, контролер MC визначає режим конференції, що може бути загальним для всіх учасників конференції або окремим для кожного з них. В зв'язку з тим, що в мережі може бути кілька контролерів MC, то для кожної знову створюваної конференції повинна проводитися процедура визначення встаткування, для того, щоб виявити той з контролерів MC, що буде управляти даною конференцією. При організації централізованої конференції, крім контролера MC, повинен використовуватися процесор MP, що обробляє користувальницьку інформацію й відповідає за перемикання або змішування мовних потоків, відеоінформації й даних. При організації децентралізованої конференції процесор MP не використовується.

На функціональній схемі, зображеній на рисунку 3.2, є можливість прослідкувати за можливими шляхами проходження сигналів від одного функціонального блоку до іншого. Тонка стрілочка вказує шлях проходження сигналів між функціональними блоками, товста – мережна взаємодія.

Розглянемо основні функціональні взаємодії між головним офісом та дочірніми підприємствами. З головного офісу з застосуванням ІР технології проводиться зв'язок з дочірніми підприємствами, такими абонентами як – регіональний директор, підрозділи. Є можливість проведення різних режимів зв'язку при різній кількості абонентів з обліком абонентів (за допомогою БД).

Взаємодія користувача із системою здійснюється в діалоговому режимі. Основним сполучним елементом розроблювального програмного забезпечення корпоративної мережі зв'язку є система меню, що складається з головного меню та підменю. Розглянемо структурну схему системи керування (рисунок 3.4). Розроблена система є меню-орієнтованою. У вікні завантаження, яке спливає при завантаженні програми, відображено інформацію про програмне забезпечення та інформацію про автора.

На головній формі існують наступні закладки:

- про програму;
- система доступу;
- системне меню програми;
- настройки програмного забезпечення;
- вихід з програми.

На закладці «Про програму» міститься коротка інформація про дану програму: ПІБ розробника, місце та рік розробку, версію програми.

На закладці «Система доступу» сформована форма автентифікації користувача, на якій потрібно ввести паролльні дані, для доступу до мережі корпоративного зв'язку, що також допомагає вести моніторинг дзвінків всередині корпорації. Для підвищення захищеності корпоративної мережі зв'язку передбачено можливість «заміни пароля».

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

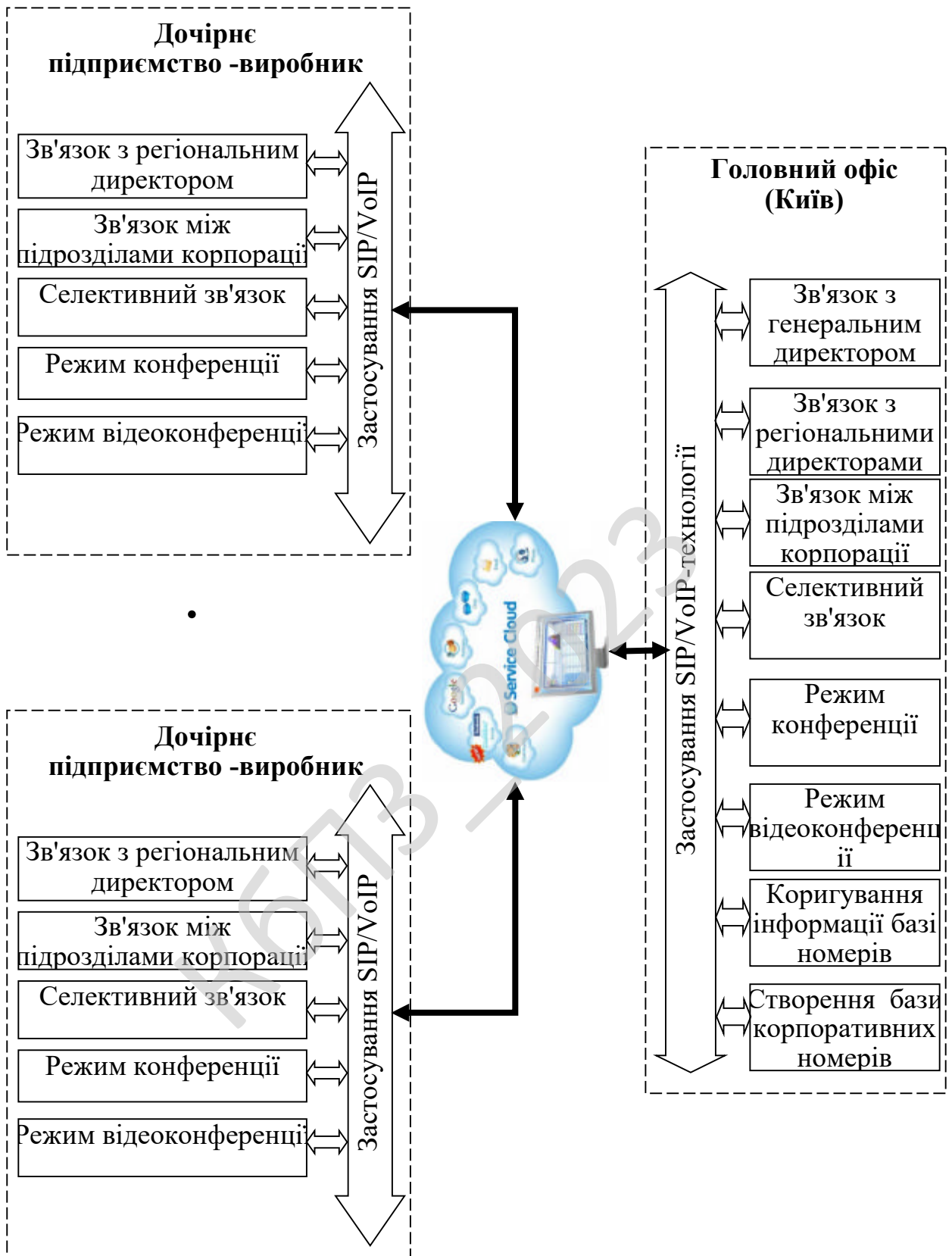


Рисунок 3.2 – Функціональна схема системи

На закладці «Системне меню програми» реалізована можливість настройки зовнішнього вигляду програмного забезпечення.

На закладці «Настройки програмного забезпечення» впроваджена система створення піктограм до вигляду додатків MS-Office, які є стандартом де-факто інтерфейсного забезпечення програмних додатків.

Закладка «Вихід із програми» дозволяє користувачу закінчити роботу з програмою і вийти з додатка.

На «Головній формі» розташовані наступні програмні блоки для реалізації функцій мережі корпоративного зв'язку усередині корпорації:

- Рядок статусу програмного забезпечення.
- Виклик IP-адресату.
- Завершення сеансу зв'язку.
- Редагування легенди IP-номерів.
- Редагування бази даних IP-адрес.
- Моніторинг бази даних IP-адрес та легенд.
- Організація селективного зв'язку.

У пункті меню «Рядок статусу програмного забезпечення» відображено поточний статус програмного забезпечення, тобто, що у даний момент виконує програма: Виклик IP-адресату, Завершення сеансу зв'язку, Редагування легенди IP-номерів, Редагування бази даних IP-адрес, Моніторинг бази даних IP-адрес та легенд або Організація селективного зв'язку. При цьому, якщо іде розмова, або селективний зв'язок то указується з якими абонентами відбувається ця дія.

Пункт меню «Виклик IP-адресату» дозволяє по вибраному з бази даних або набраному вручну IP-адресу установити зв'язок з відповідним адресатом.

Кнопка «Завершення сеансу зв'язку» нажимається при завершенні розмови з відповідним абонентом, або при завершенні селективної наради.

В пункті меню «Редагування легенди IP-номерів» вноситься легенда для кожної IP-адреси або телефонного номера абоненту, наприклад: 127.0.0.1 – Генеральний директор.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

- Сховища даних (репозиторії).
- Зовнішні по відношенню до системи сутності.
- Потоки даних між елементами трьох попередніх типів.

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.

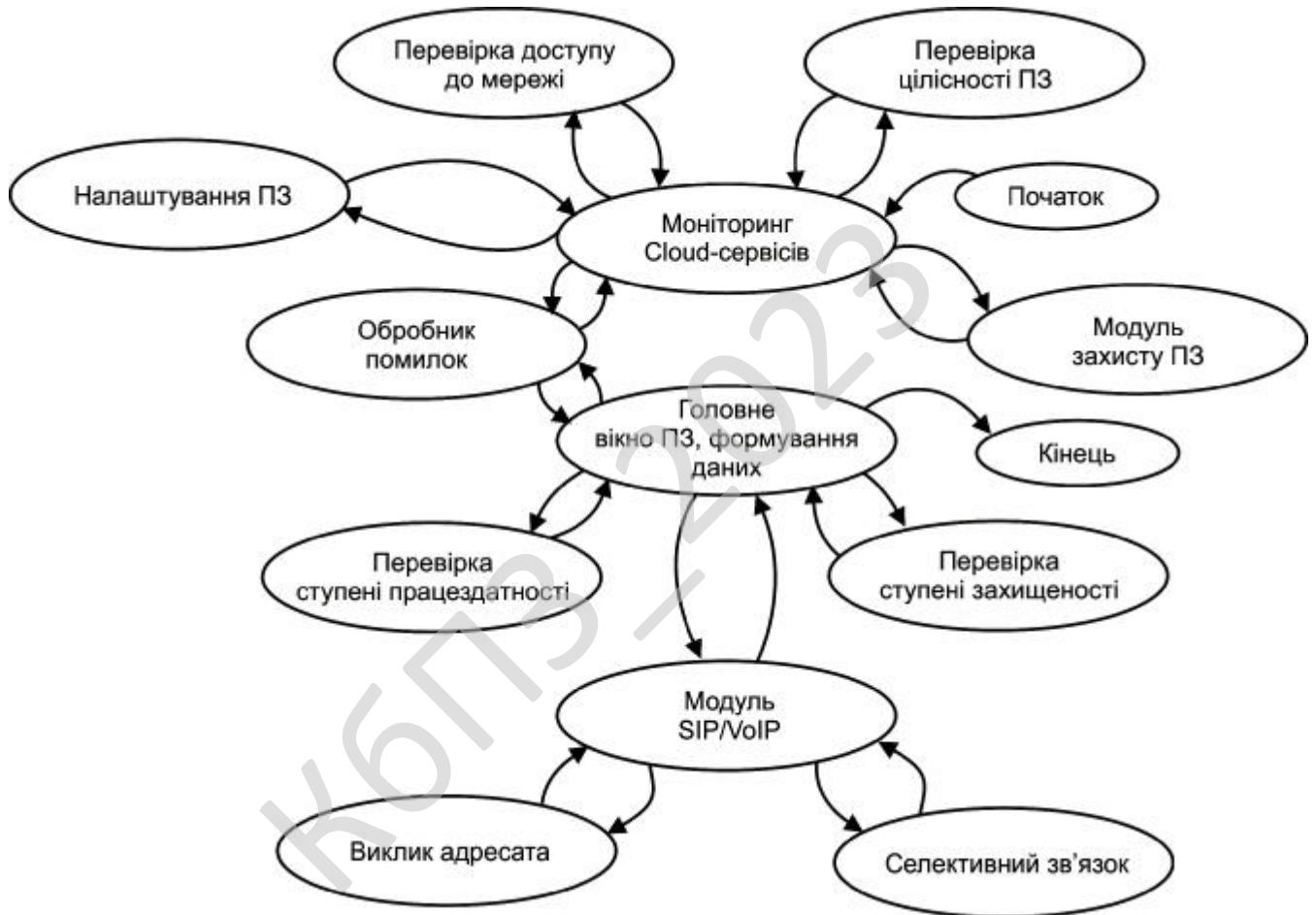


Рисунок 3.3 – Діаграма взаємодії процесів

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Під час роботи над магістерською дипломною роботою було створено блок-схеми. 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.

Основні елементи схем алгоритму це термінатор, процес, рішення, зумовлений процес (підпрограма), дані та з'єднувач.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

- діаграма прецедентів (діаграми поведінки типу);
- 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.

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: проєктована 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 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.

На рисунку 4.1 наведено блок-схему основної програми, на рисунку 4.2 зображено роботу підпрограм.

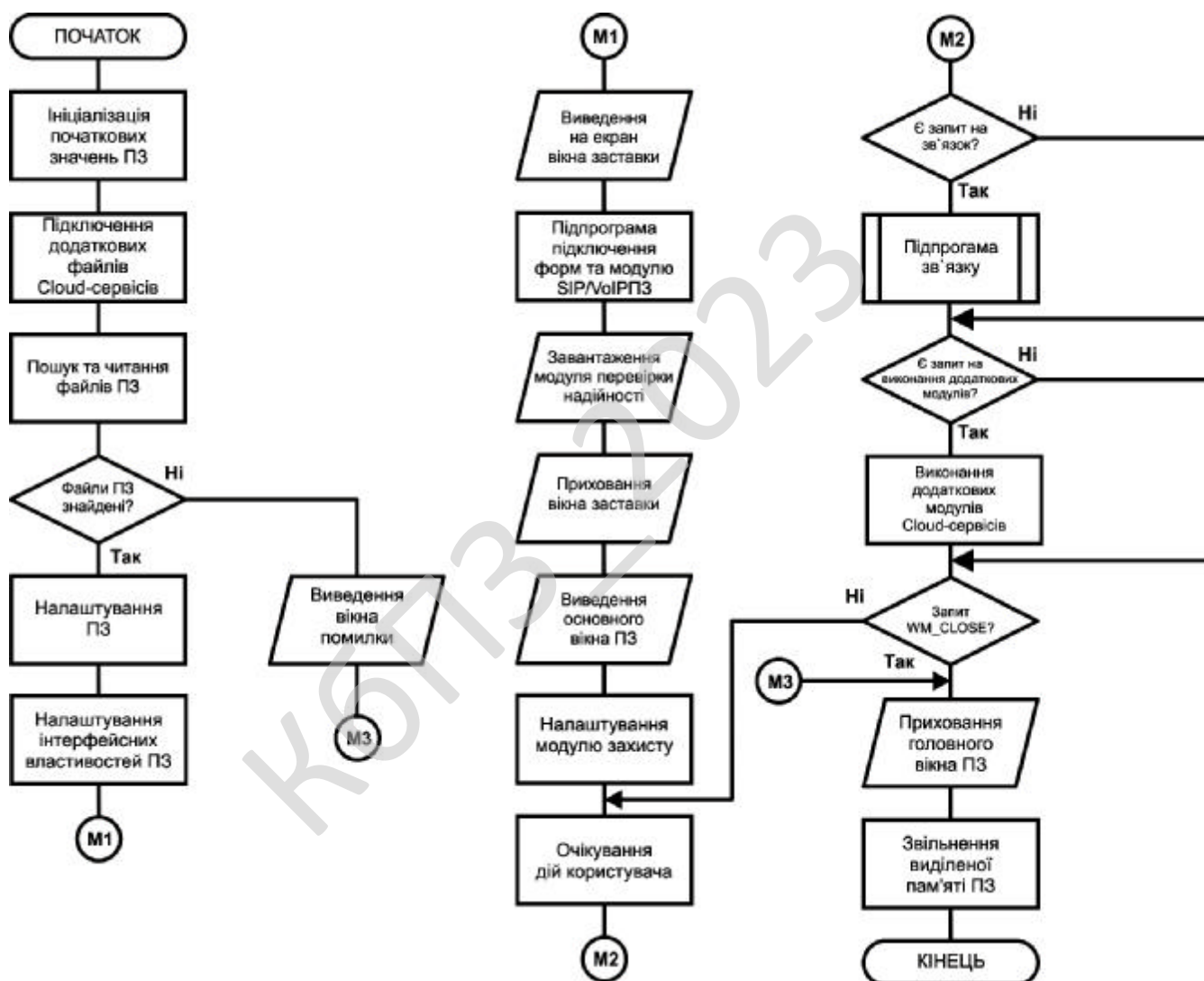


Рисунок 4.1 – Блок-схема основної програми

Розглянемо обґрунтування інформаційного забезпечення. Перелік первісних даних.

Під вхідною інформацією розуміється вся інформація, необхідна для вирішення задачі і розташована на різних носіях: первинних документах, машинних носіях, у пам'яті персонального комп'ютера. Вхідною інформацією для розроблюваної в дипломному проєкті корпоративної системи зв'язку з використанням IP-технологій є мова або відеозображення.

Щоб передати мову через телефонну мережу, мовну інформацію потрібно перетворити в аналоговий електричний сигнал.

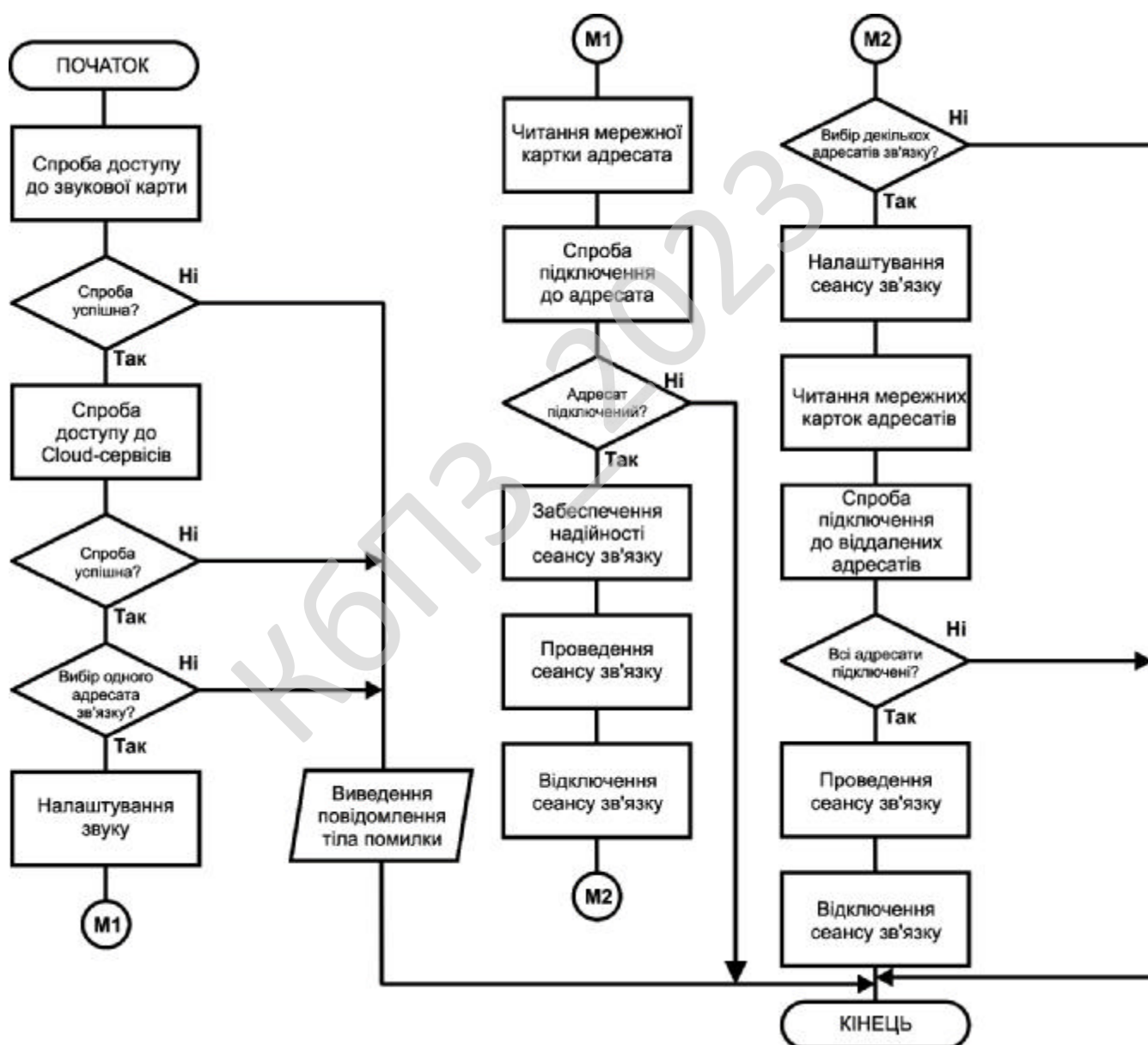


Рисунок 4.2 – Блок-схема роботи підпрограми

Америці використовується кодування по m-закону, а в Європі – по A-Закону. Тому при міжнародному зв'язку в багатьох випадках потрібне перетворення m-закону в A-закон, відповідальність за яке несе країна, у якій використовується m-закон кодування. В обох випадках кожний відлік кодується 8 бітами, або одним байтом, який можна вважати звуковим фрагментом. Для передачі послідовності таких фрагментів необхідна пропускна здатність каналу, рівна 64 Кбіт/с.

Оскільки ІКМ була першою стандартною технологією, що одержала широке застосування в цифрових системах передачі, пропускна здатність каналу, рівна 64 Кбіт/с, стала всесвітнім стандартом для цифрових мереж всіх видів, причому – стандартом, що забезпечує передачу мови з дуже гарною якістю. Однак така висока якість передачі мовного сигналу (що є еталоном при оцінці якості інших схем кодування) досягнута в системах ІКМ за рахунок явно надлишкова, при сучасному рівні технології, швидкості передачі інформації.

Щоб зменшити властиву ІКМ надмірність і знизити вимоги до смуги пропускання, послідовність чисел, отримана в результаті перетворення мовного аналогового сигналу в цифрову форму, піддається математичним перетворенням, що дозволяють зменшити необхідну швидкість передачі.

Ці перетворення «сирого» цифрового потоку в потік меншої швидкості називають «стиском» (а часто – кодуванням, розглядаючи ІКМ як якусь відправну точку для подальшої обробки інформації). Існує безліч підходів до «стиску» мовної інформації; всі їх можна розділити на три категорії: кодування форми сигналу (waveform coding), кодування вихідної інформації (source coding) і гібридне кодування, що представляє собою сполучення двох підходів.

Кодування форми сигналу. Імпульсно-кодова модуляція, по суті, і являє собою схему кодування форми сигналу. Однак цікавлять більш складні алгоритми, що дозволяють знизити вимоги до смуги пропускання. Розглянуті методи кодування форми сигналу використовують ту обставину, що між випадковими значеннями декількох послідовних обчислень існує деяка залежність.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

декодера, що приводить до катастрофічного погіршення якості відтворення мови навіть при малій імовірності втрат.

Перелік вихідних даних. У ході розробки корпоративної системи зв'язку з використанням ІР-технологій визначено, що вихідною інформацією є мова або відеозображення на пункті отримання інформації. Як може здатися на перший погляд, вузькополосне кодування мови, що вимагає обчислювальної потужності, є самим складним завданням, виконуваної устаткуванням ІР-телефонії.

Однак це не так: алгоритми кодування мови стандартизовані й відмінно документовані, більше того, на ринку доступні досить ефективні їхні реалізації для всіх популярних DSP-платформ. З іншого боку, в устаткуванні ІР-телефонії повинні бути реалізовані багато інших функцій, спосіб реалізації яких не є об'єктом стандартизації.

На передавальній стороні устаткування ІР-телефонії працює за принципом «закодував, передав і забув». На прийомній стороні все набагато складніше. Пакети приходять із мережі із затримкою, що міняється за випадковим законом. Більше того, пакети можуть прийти не в тій послідовності, у якій були передані, а деякі пакети можуть взагалі бути загублені. Приймач повинен справлятися з усіма цими труднощами, забезпечуючи на виході нормальний звуковий потік з тактовою синхронізацією, або генерованим на основі прийнятого потоку даних, або одержуваним із ТфОП по каналах Е1.

Прив'язка мовних потоків до місцевого тактового синхросигналу здійснюється шляхом непомітної на слух деформації періодів мовчання у відтвореному сигналі. До цього залишається додати необхідність передачі факсимільної інформації в реальному часі з автоматичним розпізнаванням сигналів факсимільних апаратів і передачу DTMF-сигналів з коректним їхнім відновленням у приймачі.

Сигнали багаточастотного набору номера (DTMF) – просто звукові сигнали, передані по телефонному каналі.

При передачі їх по цифровій телефонній мережі не виникає ніяких

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

проблем, тому що кодування за допомогою алгоритму G.711 не накладає ніяких обмежень на вид звукових сигналів – це може бути мова, сигнали модему, або тональні сигнали – всі будуть успішно відтворені на приймачі.

Вузькополосні кодеки, щоб досягти низьких швидкостей передачі, використовують той факт, що сигнал, який вони кодують, представляє саме мову. Сигнали DTMF при проходженні через такі кодеки спотворюються й не можуть бути успішно розпізнані приймачем на прийомній стороні.

Коли користувачеві ТфОП потрібно ввести якусь додаткову інформацію у віддалену систему при вже встановленому з'єднанні, необхідно забезпечити можливість надійної передачі DTMF-Сигналів через мережу IP-телефонії. У випадках, коли система, взаємодіючи з користувачем, просто ставить запитання й чекає введення, тривалість і момент передачі сигналу не важливі.

В інших випадках система видає користувачеві список і просить його натиснути, наприклад, кнопку «#», як тільки він почує потрібну інформацію; тут ситуація більше складна, і необхідна більше точна прив'язка вчасно.

Існуючі методи передачі сигналів DTMF по мережах IP-телефонії.

– Обов'язковий метод. Спеціальне повідомлення протоколу H.245 може містити символи цифр і «*», «#». У цьому випадку використовується надійне TCP-з'єднання, так що інформація не може бути загублена. Однак через особливості TCP можуть мати місце значні затримки;

– Нестандартний метод. Він може бути застосований у терміналах H.323v2 при використанні процедури fastStart і відсутності каналу H.245. Для передачі сигналів DTMF відкривається спеціальна RTP-сесія, у якій передаються кодовані значення прийнятих цифр, а також дані про амплітуду й тривалість сигналів. Може бути використана та ж сесія, що й для мови, але зі спеціальним типом корисного навантаження. Використання RTP дозволяє прив'язати DTMF- сигнали до реального часу, що є важливою перевагою даного методу.

У принципі, перший метод може бути більше кращим, однак у випадку міжнародних викликів і при використанні віддалених систем, що вимагають

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61

твердої прив'язки введення користувача до часу, може виявитися необхідним застосувати другий метод. Шлюзи IP-телефонії повинні обов'язково придушувати перекручені сигнали DTMF, що пройшли через основний мовний канал. У протилежному випадку, при відновленні сигналів, про які була прийнята інформація, можуть виникнути неприємні ефекти накладення й розмноження сигналів.

На основі даного огляду функцій устаткування IP-телефонії можна зробити висновок, про те що, незважаючи на існування стандартних алгоритмів кодування мови, у розроблювачів є величезний простір для діяльності, спрямованої на подальше вдосконалювання технології IP-телефонії.

4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм SEED – у криптографії симетричний блоковий криптоалгоритм на основі Мережі Фейстеля, розроблений Корейським агентством інформаційної безпеки (Korean Information Security Agency, KISA) в 1998 році. В алгоритмі використовується 128-бітний блок і ключ довжиною 128 біт. Алгоритм одержав широке поширення й використовується фінансовими й банківськими структурами, виробничими підприємствами й бюджетними установами Південної Кореї, оскільки 40-бітний SSL не забезпечує на даний момент мінімально необхідного рівня безпеки. Агентством по захисту інформації специфіковане використання шифру SEED у протоколах TLS і S/MIME. У той же час, алгоритм SEED не реалізований у більшості сучасних браузерів і інтернет-додатків, що утрудняє його використання в даній сфері поза межами Південної Кореї.

SEED являє собою Мережа Фейстеля з 16 раундами, 128-бітовими блоками й 128-бітовим ключем. Алгоритм використовує дві 8×8 таблиці підстановки, які, як такі з Safer, виведені з дискретного зведення в ступінь (у цьому випадку, x^{247} і x^{251} – плюс деякі «несумісні операції»). Це є деякою

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

подібністю с MISTY1 у рекурсивності його структури: 128-бітовий повний шифр – мережа Фейстеля з F-функцією, що впливає на 64-бітові половини, у той час як сама F-функція – Мережа Фейстеля, складена з G-функції, що впливає на 32-розрядні половини. Однак рекурсія не простягнеться далі, тому що G-функція – не Мережа Фейстеля. В G-функції 32-розрядне слово розглядають як чотири 8-бітових байта, кожний з яких проходить через одну або іншу таблицю підстановки, потім поєднується в помірковано комплексному наборі булевих функцій таким чином, що кожний біт виводу залежить від 3 з 4 вхідних байтів.

SEED має складний ключовий розклад, генеруючи тридцять два 32-розрядних додаткових символу, використовуючи G-функції на серіях обертань вихідного неопрацьованого ключа, комбінованого зі спеціальними раундовими константами (як в TEA) від «Золотого співвідношення» (англ. Golden ratio).

Згідно з дослідженнями KISA, алгоритм SEED «надійно протистоїть відомим атакам».

КБПЗ – 2023

					VKPM-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

Для перегляду короткої довідки про програму слід натиснути на основному вікні кнопку авторського права, після чого на екрані з'явиться вікно показане на рисунку 5.2.

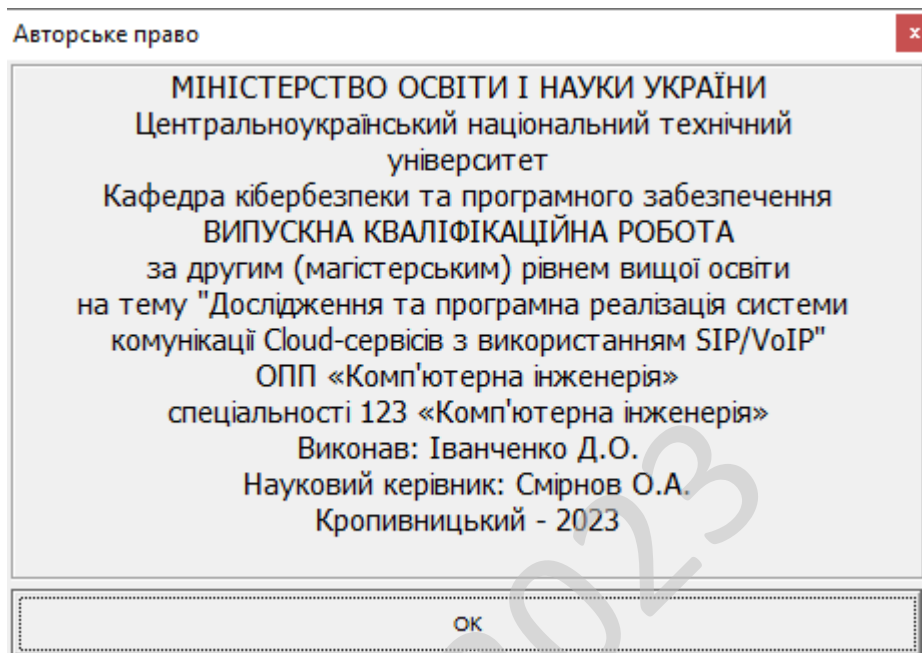


Рисунок 5.2 – Вікно розробника ПЗ

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

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

Проводилась оцінка:

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

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		65

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

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

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.
- Як приймаються вхідні 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», чию 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).

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

закінчення цього терміну користувач вирішить продовжити використання ПЗ, він зобов'язаний купити його (zareєstrуватися), заплативши авторові певну суму. В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

КБПЗ_2023

					VKPM-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи комунікації Cloud-сервісів з використанням SIP/VoIP.

Метою розробки є дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP.

Об'єктом дослідження є процес комунікації Cloud-сервісів з використанням SIP/VoIP.

Предметом дослідження є методи комунікації Cloud-сервісів з використанням SIP/VoIP.

Методи дослідження базуються на методах теорії телекому, методах математичної статистики, методах розробки програмного забезпечення.

Наукова новизна отриманих результатів. У процесі рішення завдань, обумовлених цілями дослідження, отримані наступні результати:

– Удосконалено метод комунікації Cloud-сервісів з використанням SIP/VoIP.

– Розроблено вітчизняний продукт комунікації Cloud-сервісів з використанням SIP/VoIP, який має більш широкі можливості, на відміну від існуючих аналогів.

					VKPM-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

7 ДАНІ ПРО ЕКОНОМІЧНУ ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ

7.1 Техніко-економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти

Після ознайомлення з підприємством та засобами розробки програмної продукції був розроблений план розробки програми. Був підрахований необхідний час для розробки та впровадження програми. Цей час склав 24 днів (один місяць).

В магістерській роботі було проведено дослідження та виконана програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP.

Розроблене програмне забезпечення має достатню надійність і задовольняє усім поставленим умовам, а саме:

- а) невеликий розмір;
- б) невеликі системні потреби;
- в) незалежність від встановлених на комп'ютері баз даних;
- г) зручність у користуванні та надійність.

Таблиця 7.1 – Початкові дані

Показники	Позначення	Характеристика або величина
1	2	3
1. Кількість розроблених програм період, шт.	N	1
2. Кількість екземплярів програм, шт.	Ne	30
3. Запланований термін розробки, днів	Frq	24 (1 місяць)
4. Група задачі підсистеми управління (1-6)	–	1
5. Ступінь новизни задачі (А, Б, В, Г)	–	Г
6. Складність алгоритму (1, 2, 3)	–	2

Продовження таблиці 7.1

1	2	3
7. Кількість макетів вхідної інформації	–	8
8. Кількість форм вихідної інформації.	–	6
9. Мова програмування (1-6)	–	2
10. Попередній досвід (1-6)	–	3
11. Гнучкість проекту ПП (1-6)	–	3
12. Детальність проекту ПП (1-6)	–	1
13. Рівень спрацьованості колективу (1-6)	–	2
14. Ступінь вимірності процесів (1-6)	–	3
15. Необхідна надійність програмного забезпечення (1-6)	–	3
16. Розмір бази даних (порівняно з розміром програми) (1-6)	–	4
17. Складність кінцевого програмного продукту (1-6)	–	5
18. Необхідний рівень забезпечення повторного використання (1-6)	–	2
19. Документованість відповідно до планованого життєвого циклу (1-6)	–	3
20. Вимоги до швидкодії ПП (1-6)	–	3
21. Обмеження на розміри основного сховища даних (1-6)	–	2
22. Різноманітність використовуваних обчислювальних платформ (1-6)	–	4
23. Професійний рівень аналітиків (1-6)	–	3
24. Професійний рівень програмістів (1-6)	–	4
25. Постійність складу команди розробників (1-6)	–	2
26. Досвід розробки додатків (1-6)	–	1
27. Досвід роботи з обчислювальною платформою (1-6)	–	2

Продовження таблиці 7.1

1	2	3
28. Досвід роботи з мовою і інструментами середовища розробки (1-6)	–	2
29. Досвід роботи з програмними інструментами розробки (1-6)	–	3
30. Розробка ПЗ для декількох серверів одночасно (1-6)	–	3
31. Вимоги до дотримання встановленого графіка робіт (1-6)	–	2
32. Вартість ПЗ у розробника (НМА), грн.	–	30000
33. Норматив додаткової зарплати, % :	Н _д	10
34. Норматив відрахувань у соціальні фонди, %	Н _с	22
35. Норматив загальногосподарських витрат, %	Н _г	15
36. Норматив витрат на освоєння нових мов програмування, %	Н _п	15
37. Рівень рентабельності програмної продукції, %	Р _е	40
38. Ставка податку на додану вартість, %	Н _{дв}	20

7.2 Розрахунок трудомісткості розробки програмної продукції

Значення трудомісткості розробки програмного забезпечення для стадій ТЗ, ЕК, ТП та ВП визначаємо по типовим нормам часу приведеним в додатках МВ. Стадія РП є найбільш тривалою і трудомісткою, що робить значний вплив на інші стадії проекту.

Визначимо трудомісткість розробки ПЗ для стадії РП.

Обчислюємо номінальні трудовитрати, люд-міс.:

$$T_{ном} = A \text{ Size}^B, \quad (7.1)$$

де: A – коефіцієнт Боема, $A = 2,45$;

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

Size – загальний об'єм відлагодженого програмного коду, тис. рядків;

B – показник ступеня, що визначається співвідношенням:

$$B = 1,01 + 0,001 \sum W_i, \quad (7.2)$$

де: W_i – сумарне значення п'яти показників (МВ, додаток 2), що відображають особливості розробки проекту програмного продукту (ПП) і колективу розробників.

$$B = 1,01 + 0,001(2,43 + 3,64 + 4,22 + 3,95 + 2,73) = 1,027.$$

$$T_{ном} = 2,45 \cdot 2,2^{1,027} = 5,5 \text{ люд-міс.}$$

Визначаємо уточнені (з урахуванням приведених в МВ додатку 3 сімнадцяти додаткових коефіцієнтів) трудовитрати, люд-міс.:

$$T_{уточн} = T_{ном} \prod V_j, \quad (7.3)$$

де: $\prod V_j$ – добуток сімнадцяти додаткових коефіцієнтів, приведених в МВ додатку 3.

$$T_{уточн} = 5,5 \cdot (1 \cdot 1,09 \cdot 1,30 \cdot 0,91 \cdot 1 \cdot 1 \cdot 1 \cdot 1,15 \cdot 1 \cdot 0,87 \cdot 1,10 \cdot 1,22 \cdot 1,12 \cdot 1,10 \cdot 1 \cdot 1 \cdot 1,10) = 12,9 \text{ люд-міс.}$$

Ці коефіцієнти дозволяють диференційовано оцінювати результати роботи програмістів, беручи до уваги швидкодію програми, використання різноманітних обчислювальних платформ і інструментів розробки, взаємодію декількох серверів, вимоги до об'ємів баз даних і ін.

Визначаємо підсумкові трудовитрати по стадії робочий проект, люд-дні:

$$T_{РП} = 0,3 C T_{уточн}^{0,33 + 0,2(B-1,01)} S, \quad (7.4)$$

де: C – визначений емпірично коефіцієнт, запропонований авторами методики, (МВ, додаток 4); S – коефіцієнт стиснення (або подовження) графіка робіт %, що дозволяє коректувати терміни розробки ПЗ згідно встановленим вимогам. Вибираємо в межах (25...350)%.

$$T_{РП} = 0,3 \cdot 2,66 \cdot 12,9^{0,33 + 0,2(1,027 - 1,01)} \cdot 100 = 131 \text{ люд/день.}$$

Для зручності визначення загальної трудомісткості на розробку програмного забезпечення результати розрахунків по стадіям зводимо до таблиці 7.2.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

Таблиця 7.2 – Визначення трудомісткості розробки програмного забезпечення

Стадії розробки	Трудомісткість за типовими нормами та розрахунками	
	Величина, люд/дні	Підстава
Технічне завдання	9	Д5
Ескізний проект	10	Д6
Технічний проект	15	Д7
Робочий проект	66	Ф 7.1-7.4
Впровадження	15	Д13
Всього	115	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

Чисельність ставок інженерів-програмістів для розробки програмного забезпечення визначається за формулою:

$$Ч = \frac{T_{нз} N}{F_{pq} - H_{ев}}, \quad (7.5)$$

де: F_{pq} – плановий фонд робочого часу одного спеціаліста, днів;

$T_{нз}$ – трудомісткість розробки програмного забезпечення люд-дні.

$$Ч = \frac{115 \cdot 1}{24 \cdot 3} = 5,5 \text{ ставки.}$$

Чисельність інженерів-електронщиків для проведення технічного обслуговування та ремонту комп'ютерних мереж визначається в залежності від наявності технічних засобів і норм витрат часу на виконання профілактичних робіт на протязі року.

Визначаємо затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за період розробки. Результати розрахунку зводимо до таблиці 7.3.

Таблиця 7.3 – Затрати часу на виконання профілактичних робіт по обслуговуванню обладнання за розрахунковий період

Найменування обладнання	Профілактичне обслуговування			
	Кількість хв. на один. обл.	Кількість обладнання	Затрати часу в хв.	Затрати часу в год.
Системний блок ПК	90	12	1080	18
Монітор	60	12	720	12
Клавіатура	30	12	360	6
Маніпулятор «мишка»	30	12	360	6
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	2	240	4
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	3	90	1,5
Кабельні господарства ЛОМ на 1 м.п.	2,5	400	1000	16,67
Копіювальний апарат	140	1	140	2,33
Усього за рік:			3 _ч	67,83

Час на профілактику обладнання в загальному балансі робочого часу інженерів-електронщиків не повинен складати більше 10%.

Виходячи з цього фонд робочого часу інженерів-електронщиків складає:

$$\Phi_{op}^c = \frac{3_{ч} \cdot n_{mic}}{1,2}, \quad (7.6)$$

$$\Phi_{op}^c = \frac{67,83 \cdot 1}{1,2} = 56,5 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

$$Ч_{ел} = \frac{\Phi_{op}^c}{F_{op} \cdot T_{зм}}, \quad (7.7)$$

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	1	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	1	
	Контроль взаєморозрахунків з постачальниками	1	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	1	0,5
	Створення графічних і стилістичних елементів сайту	1	
	Оформлення банерів і промо-сторінок	1	
	Розміщення графіки і контенту на Інтернет сторінках	1	
Всього		4	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,5
	Верстка друкованих видань	1	
	Додрукова підготовка макетів	1	
	Розміщення графіки і контенту на Інтернет сторінках	1	
Всього		4	

Складемо штатний розклад виконавців.

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	18412	18412
Продакт-менеджер	0,5	14000	7000
Інженер-програміст	5,5	13000	71500
Інженер-електронщик	0,3	10000	3000
Інженер-системотехнік	0,5	10000	5000
Адміністратор мережі	0,5	10000	5000
Дизайнер WEB	0,5	10000	5000
Всього за період розробки	$R_{cn} = 8,8$	-	$\Phi_{роб} = 114912$

Розрахуємо середньоденну зарплату одного виконавця:

$$z_{сд} = \frac{\Phi_{роб}}{R_{cn} F_{pq}}, \quad (7.8)$$

де: $\Phi_{роб}$ – загальна сума зарплати за плановий період, грн.

$$z_{сд} = \frac{114912}{9,5 \cdot 24} = 504 \text{ грн.}$$

7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника

Балансова вартість будівель визначається з урахуванням кількості робочих місць виконавців, питомої площі на одне робоче місце, та вартості одного квадратного метра виробничої площі:

$$B_{yд} = R_{cn}^1 S_y \Pi_{пл}, \quad (7.9)$$

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 13 робочих місць;

S_y – питома площа на одне робоче місце, m^2 ;

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

$C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» (м. Кіровоград) ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 400...1600 $у.о./м^2$. Приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ $м^2$. На кожне робоче місце у середньому потрібно 8 $м^2$. З урахуванням цього:

$$B_{уд} = 13 \cdot 8 \cdot 20000 = 2080000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 208000 грн. Балансова вартість інвентарю розраховується за нормою 3500 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{сн}^1 \cdot C_{м}, \quad (7.10)$$

де: $C_{м}$ – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 13 \cdot 3500 = 45500 \text{ грн.}$$

Балансова вартість обчислювальної техніки визначається по оптовим цінам постачальника з врахуванням витрат на транспортування.

Специфікація на обчислювальну техніку наведена в таблиці 7.7.

Дані по оптовій ціні на обладнання та комплектуючі вибирались по прайсу фірми Brain за 19.10.23 – джерело <http://brain.com.ua>.

Таблиця 7.6 – Специфікація

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Персональний комп'ютер		11457
Системний блок		7509
Процесор	INTEL Pentium G6405 (BX80701G6405) 1200, 2 ядра, 4 потоки, 4.1 GHz, TDP – 58 Вт, 14nm	-

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
Системна плата	MSI PRO H510M-B сокет – 1200, LAN – 1 Гбіт/с, D-Sub (VGA), HDMI, 1 x M.2 2280, 4 x SATA 6.0 Gb/s, Micro-ATX, BOX	-
Відеокарта	Intel UHD Graphics 610	-
Жорсткий диск	SSD M.2 2280 256GB ADATA (ALEG-700-256GCS) Серія – LEGEND 700, 256 GB, 3D NAND, M.2, PCI Express 3.0 x4	-
Оперативна пам'ять	DDR4 8GB 2666 MHz Kingston (KVR26N19S8/8)	-
DVD-привод	-	-
Корпус	Gamemax MT520-450W, Miditower, ATX, Mini – ITX, PSU – 450 Вт	-
Кардрідер внутрішній	USB 2.0 Card reader STORM CR-35U1A4-B, int. 3.5", 1*USB2.0+AUDIO+1394, multi: All Type Cards, black	-
інше	Клавіатура, мишка	-
Монітор	22" TFT, ASUS VW223D (5ms, 300/3000: 1, 170/160, D-SUB, Wide)	2600
Принтер лазерний	Canon i-SENSYS LBP6030W	2700
Принтер струминний	Epson Stylus Photo P50 (C11CA45341) + USB cable	5500
Сканер	Epson Perfection V37 Photo	2970
Копіювальний апарат	Canon i-SENSYS MF217W with Wi-Fi	5965
Пристрій безперебійного живлення	UPS APC BACK-UPS ES 525VA 230V RUSSIA (BE525-RS)	1348

Витрати на транспорт, монтаж та випробування можуть бути прийняті в межах до 10% від оптової ціни.

Для визначення необхідної кількості капітальних вкладень складемо таблицю 7.8.

Таблиця 7.7 – Балансова вартість обчислювальної техніки

Найменування обчислювальної техніки	Кількість, шт.	Ціна за одиницю, грн.	Витрати на транспортування, монтаж та випробування.	Загальна вартість, грн.
Персональні комп'ютери	13	11457	14894,1	163835,1
Принтер лаз.	2	2700	540	5940
Принтер струм.	1	5500	550	6050
Сканери	1	2970	297	3267
Копіюв. апарат	1	5965	596,5	6561,5
Всього	–	–	–	185653,6

Таблиця 7.8 – Вартість основних фондів та амортизаційні відрахування розробника

Групи та види основних фондів	Балансова вартість, грн.	Амортизація	
		Норма, %	Відрахування, грн.
1	2	3	4
Група 3			
1. Будівлі	2080000	-	-
2. Передавальні пристрої	208000	-	-
Всього по групі	2288000	5	114400

Продовження таблиці 7.8

1	2	3	4
Група 4			
3. Обчислювальна техніка	185654	-	-
Всього по групі	185654	50	92827
Група 5, 6			
4. Вимірювальні пристрої	3999	25	-
5. Транспортні засоби	0	20	-
6. Господарський інвентар	45500	25	-
Всього по групі	49499	-	12374,75
7. Нематеріальні активи	30000	10	3000
Разом	$K_p = 2553153$		$A_p = 222602$

7.5 Визначення собівартості розробки та ціни програмної продукції

Визначимо основну зарплату виконавців:

$$Z_o = \frac{Z_{cd} \cdot T_{nz}}{N_e}, \quad (7.11)$$

де: N_e – кількість екземплярів програм, шт.

$$Z_o = 504 \cdot 115 / 30 = 1932 \text{ грн.}$$

Визначимо додаткову зарплату (оплата відпусток, виконання державних та суспільних обов'язків) на рівні 10%:

$$Z_d = Z_o \cdot H_q \cdot 0,01, \quad (7.12)$$

де: H_q – норматив додаткової зарплати, %.

$$Z_d = 1932 \cdot 10 \cdot 0,01 = 193 \text{ грн.}$$

Відрахування на соціальні потреби за нормативом $H_c = 22\%$ від суми основної та додаткової зарплати:

$$C_{oc} = 0,01 \cdot H_c (Z_o + Z_d), \quad (7.13)$$

де: H_c – відрахування на соціальні потреби, %.

$$C_{oc} = 0,01 \cdot 22(1932+193) = 468 \text{ грн.}$$

Визначимо загальногосподарські витрати (електроенергію, ремонт і утримання приміщень і т.д) за нормативом $H_z = 15\%$ від основної зарплати:

$$G_{ocn} = Z_o \cdot H_z \cdot 0,01, \quad (7.14)$$

де: H_z – загальногосподарські витрати, %.

$$G_{ocn} = 1932 \cdot 15 \cdot 0,01 = 290 \text{ грн.}$$

Визначимо витрати на матеріали для розробки програмної продукції за нормами споживання та діючими цінами за одиницю виміру:

$$Z_M = (Z_{M1} + Z_{M2} + Z_{M3})/N_e, \quad (7.15)$$

де: Z_{M1} – вартість паперу, грн.;

Z_{M2} – вартість запам'ятовуючих пристроїв, грн.;

Z_{M3} – вартість фарби, картриджів, тонеру, грн.;

N_e – кількість екземплярів програм, шт.

Згідно виданих норм приймаємо $n = 1/5$ пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $Ц_n = 200$ грн., визначаємо вартість паперу за період розробки $N_m = 1$ міс:

$$Z_{M1} = Ц_n \cdot N_m \cdot n. \quad (7.16)$$

$$Z_{M1} = 200 \cdot 1 \cdot 1/5 = 40 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD/DVD дисків в кількості 3 примірника:

$$Z_{M2} = \sum Ц_d, \quad (7.17)$$

де: $Ц_d$ – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 31 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 31 грн./шт.

$$Z_{M2} = 31 \cdot 3 = 93 \text{ грн.}$$

					БКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

Згідно виданих норм одноразовій заправці підлягають усі друкуючі пристрої і становить:

$$Z_{M3} = \sum C_3, \quad (7.18)$$

де: C_3 – вартість розхідних матеріалів друкуючих пристроїв: відновлення та заправка картриджу для Canon i-SENSYS LBP6030W – 574 грн.; картридж для Epson Stylus Photo P50 – 558 грн.; відновлення картриджу для MF217W – 570 грн.

$$Z_{M3} = 574 + 558 + 570 = 1702 \text{ грн.}$$

$$Z_M = (40 + 93 + 1702) / 30 = 61 \text{ грн.}$$

Визначимо витрати на освоєння нових мов програмування або операційних систем за нормативом ($H_n = 15\%$) від основної зарплати виконавців:

$$O_n = Z_o \cdot H_n \cdot 0,01, \quad (7.19)$$

де: H_n – норматив витрат на освоєння нових мов програмування, %.

$$O_n = 1932 \cdot 15 \cdot 0,01 = 290 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 30$ прим.):

$$A_m = \frac{A_p \cdot N_{mic}}{N_e \cdot 12}, \quad (7.20)$$

де: A_p – загальна річна сума амортизаційних відрахувань, грн.

$$A_m = 222602 \cdot 1 / (30 \cdot 12) = 618 \text{ грн.}$$

Повна собівартість ПЗ визначається як сума витрат за попередніми статтями калькуляції:

$$C_n = Z_o + Z_d + C_{oc} + \Gamma_{ocn} + Z_M + O_n + A_m. \quad (7.21)$$

$$C_n = 1932 + 193 + 468 + 290 + 61 + 290 + 618 = 3852 \text{ грн.}$$

Величини ціна підприємства, податок на додану вартість, відпускна ціна програмної продукції визначаються за формулами, приведеними в таблиці 7.9

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Таблиця 7.9 – Нормативна калькуляція собівартості розробки програмного забезпечення задачі

Найменування статей витрат	Позначення	Величина, грн
1	2	3
1. Основна зарплата виконавців	Z_o	1932
2. Додаткова зарплата виконавців	Z_d	193
3. Відрахування на соціальні потреби	C_{oc}	468
4. Загальногосподарські витрати	G_{ocn}	290
5. Витрати на матеріали	Z_M	61
6. Освоєння нових операційних систем, мов програмування	O_n	290
7. Амортизація основних фондів	A_M	618
8. Повна собівартість програмного забезпечення	C_n	3852
9. Плановий прибуток	P_p	1541
10. Ціна підприємства $C_n = C_n + P_p$	C_n	5393
11. Податок на додану вартість $ПДВ = 0,01 \cdot H_{об} \cdot C_n$	$ПДВ$	1077
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	6470

Визначимо плановий прибуток за рівнем рентабельності (P_n) програмної продукції, яка залежить від складності програми та ступеня новизни задачі.

Для даного програмного забезпечення рівень рентабельності складає 40%.

$$P_p = 0,01 \cdot P_n \cdot C_n, \quad (7.22)$$

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 40 \cdot 3852 = 1541 \text{ грн.}$$

7.6 Визначення об'єму капітальних вкладень у споживача програмної продукції

Об'єм капітальних вкладень у споживача програмної продукції визначаємо на основі балансової вартості основних фондів, яка враховує ціну, транспортно-заготівельні витрати, вартість будівель, монтажних та пусконаладжувальних робіт, а також витрати на випробування у виробничих умовах. Результати розрахунків зводимо у таблицю 7.10.

Таблиця 7.10 – Розрахунок об'єму капітальних вкладень у споживача програмної продукції

Найменування капітальних вкладень	Сума за варіантами, грн.	
	Базовий	Новий
Вартість програмної продукції	–	6470
Всього капітальних витрат	–	6470

7.7 Визначення експлуатаційних витрат

Експлуатаційні витрати у споживача програмної продукції визначаємо при умові роботи підсистеми на протязі року. Результати зводимо до таблиці 7.11.

Витрати на профілактичні роботи:

$$Z_p = T_p \cdot Z_z \cdot (1 + 0,01 \cdot H_q) \cdot (1 + 0,01 \cdot H_c), \quad (7.23)$$

де: T_p – кількість годин обслуговування кожного комп'ютера за рік, год.;

Z_z – заробітна плата обслуговуючого персоналу, грн/год.

Таблиця 7.11 – Розрахунок експлуатаційних витрат у споживача програмної продукції

Найменування статей витрат	Позначення	Сума витрат за варіантами, грн.	
		Базовий	Новий
1. Витрати на технічне обслуговування	Z_p	48312	28128
2. Витрати на електроенергію	$Z_{ел}$	0	0
3. Витрати на амортизацію	$Z_{ам}$	0	3235
Всього витрат за рік	I	48312	31363

Після купівлі нового програмного забезпечення кількість профілактичних годин робіт зменшилася з 360 годин на рік до 210 годин на рік, тому витрати на технічне обслуговування зменшилися з:

$$Z_{p\text{ баз}} = 360 \cdot 100 \cdot 1,1 \cdot 1,22 = 48312 \text{ грн},$$

до:

$$Z_{p\text{ нов}} = 210 \cdot 100 \cdot 1,1 \cdot 1,22 = 28128 \text{ грн}.$$

Витрати на електроенергію визначаються з урахуванням споживаємої потужності ($P_{ел}$) в кіловатах, часу експлуатації технічних засобів (T_p) в годинах та ціни однієї кіловат-години ($C_{ел}$):

$$Z_{ел} = P_{ел} \cdot T_p \cdot C_{ел}. \quad (7.24)$$

Витрати на електроенергію залишились без змін. $Z_{ел\text{ баз}} = Z_{ел\text{ нов}}$

Витрати по амортизації визначаються на основі норм амортизаційних відрахувань, вартості програмної продукції і основних фондів. Для розрахунку складаємо таблицю 7.12.

$$E_{cn} = (48312 - 31363) - 0,5 \cdot 6470 = 13714 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у споживача програмної продукції за рахунок зниження експлуатаційних витрат:

$$T_{cn} = \frac{K_n - K_{\bar{o}}}{I_{\bar{o}} - I_n}, \quad (7.28)$$

$$T_{cn} = \frac{6470}{48312 - 31363} = 0,38 \text{ року.}$$

Показники економічної ефективності програмної продукції зводимо до таблиці 7.13.

7.9 Висновки

Розроблена програма економічно вигідна. За рахунок впровадження програмного забезпечення досягається скорочення часу обробки інформації, підвищується культура праці, підвищення якості приймаючих управлінських рішень.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		90

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Техніка безпеки – це система правил і заходів, які допомагають запобігти травмам, хворобам і аваріям на робочому місці або в повсякденному житті. Знання техніки безпеки дуже важливе, бо воно рятує життя і здоров'я людей. Наприклад, якщо людина працює на шахті, то вона повинна знати правила безпеки у вугільних шахтах, щоб не потрапити під обвал або не підірватися на міні. Або якщо людина хоче навчитися програмувати, то вона повинна дотримуватися гігієнічних вимог і не сидіти за комп'ютером занадто довго, щоб не зашкодити своїм очам і спині.

Охорона праці та здоров'я у сфері ІТ – це комплекс заходів, які спрямовані на забезпечення безпечних і здорових умов праці для працівників, які використовують інформаційні технології, а також на запобігання травматизму, професійним захворюванням і стресу.

Характерною ознакою сучасного науково-технічного прогресу практично у всіх сферах діяльності людини є широке застосування комп'ютерних технологій, заснованих на використанні електронно-обчислювальних машин. Сьогодні, а тим більше, майбутнє, вже важко уявити без комп'ютерів та іншої електронної техніки. Адже саме завдяки їм стала можливою швидка переробка величезних обсягів інформації, проведення необхідних розрахунків, виконання різних видів робіт, пов'язаних обробкою текстових та ілюстраційних зображень, організація оперативного отримання та передачі інформації, збереження її значних обсягів електронним способом.

Законом України “Про охорону праці” [1] регламентуються загальні положення державної політики в галузі охорони праці, а конкретизуються ці положення нормативно-правовими актами про охорону праці, зокрема Наказом Міністерства соціальної політики України 14.02.2018 № 207 [4], який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		91

«Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» [2].

Програмісти у процесі роботи мають негативний вплив на органи зору, а також мають значну розумову напругою і нервово-емоційне навантаження. Руки (суглоби пальців та м'язи рук) при роботі з клавіатурою мають теж істотне навантаженням [2]. До шкідливих факторів, які впливають на робитників галузі інформаційних технологій (ІТ) спеціалісти відносять високочастотні електромагнітні коливання (випромінювання) роботи апаратної частини ЕОМ та виділення шкідливих газів.

Ці шкідливі фактори можуть привести до професійних захворювань.

При розгляді шкідливих чинників роботи програмістів та інших спеціалістів ІТ будемо керуватись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98 [2], та «Правила охорони праці під час експлуатації електронно-обчислювальних машин» НПАОП 0.00-1.28-10,

Умови праці програміста вулючають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;
- освітлення приміщення;
- параметри повітряного середовища в приміщенні, тощо.

Щоб запропонувати заходи щодо зменшення впливу комп'ютера на організм програміста визначемо фактори, які можуть викликати професійне захворювання і впливають на працездатність програміста.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		92

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машина (ЕОМ) та інше обладнання є джерелами небезпеки ураження електричним струмом. Так як робота програміста характеризується істотним зоровим навантаженням, то вимагає належного освітлення. У приміщенні, в якому працюють люди (у т.ч. програмісти) необхідно створити належний мікроклімат, параметри якого регламентуються, Державними санітарними правилами і нормами, зокрема ДСанПіН 3.3.2.007-98.

При роботі з використанням ЕОМ відзначають наступні небезпечні та шкідливі фактори:

- ризик виникнення надзвичайних ситуацій природного або штучного характеру на об'єкті або території.
- ризик виникнення пожежі;
- негативний вплив на органи зору людини;
- ризики ураження електричним струмом;
- недостатня, або надмірна освітленість робочого місця;
- електромагнітні (у т.ч. високочастотні) електромагнітні випромінювання (коливання);
- несприятливі мікрокліматичні умови;
- нервово-емоційна напруженість праці;
- інтелектуальні навантаження;
- монотонність праці;
- невідповідність ергономічних показників робочого місця діючим вимогам;
- шум;
- статичні навантаження на кістково-м'язовий апарат.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		93

8.3 Пожежна безпека

Пожежі в приміщеннях з оргтехнікою становлять особливу небезпеку, бо поєднані з великими матеріальними збитками. Пожежа може виникнути при взаємодії горючих речовин і джерел запалювання. Горючими речовинами є будівельні та опоряджувальні матеріали, пластмасові корпуси техніки, шнури тощо. Джерелами запалювання можуть бути електронні схеми комп'ютерів, принтерів, пристроїв електроживлення, де внаслідок різних порушень виникає перегрівання елементів, утворюються електричні іскри та дуги, здатні спричинити займання горючих матеріалів.

З метою виявлення початкової стадії займання необхідно використовувати пристрої систем автоматичного пожежогасіння там, де цього вимагають правила пожежної безпеки.

При обслуговуванні, ремонтних та профілактичних роботах використовуються різні легкозаймісті рідини, прокладаються тимчасові електропровідники, здійснюється паяння. Виникає додаткова пожежна небезпека, яка потребує відповідних заходів пожежного захисту. До засобів гасіння пожежі, призначених для локалізації невеликих займань, належать вогнегасники, сухий пісок, азбестові ковдри. Приміщення, в який встановлено комп'ютери і де немає необхідності влаштування систем автоматичного пожежогасіння, необхідно оснащувати переносними вуглекислотними з розрахунку 2 шт. на кожні 20 м² в приміщеннях. Звуковбирне облицювання стін, стель приміщень треба виконувати з негорючих та важко горючих матеріалів.

Електроустановки (можливість їх застосування, монтаж, накладка експлуатація) повинні відповідати вимогам чинних правил улаштування електроустановок, правил технічної експлуатації, електроустановок та інших нормативних документів.

Ймовірність виникнення пожежі від електротехнічного та іншого одиничного виробу не повинна перевищувати 10⁻⁶ на рік. При короткому замиканні

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		94

в місцях з'єднання проводів опір практично дорівнює нулю, звідси величина струму досягає дуже великих значень.

Персональні комп'ютери після закінчення роботи повинні відключатися від мережі не рідше 1 разу на квартал, необхідно очищати від пилу агрегати та вузли, кабельні канали та простір між підлогами. Не дозволяється розміщувати комп'ютерні зали ЕОМ у підвалах; проводити ремонт вузлів (блоків) ЕОМ безпосередньо у залах, де знаходяться ПК (персональні комп'ютери), залишати без нагляду ввімкнену в мережу електронну апаратуру, яка використовується для контролю ЕОМ.

Електричний струм силою 0,1 А є небезпечним для людини. Для попередження травм усе електричне обладнання повинне бути заземлене. Приступаючи до роботи необхідно перевірити справність обладнання, ізоляцію проводів і надійність заземлення. Доторкання до оголених струмоведучих і незахищених частин в електроустаткуванні забороняється. В разі виявлення порушень ізоляції електропроводів, відкритих струмоведучих частин електроустаткування або порушення заземлення треба негайно повідомити про це свого начальника для вжиття заходів щодо усунення несправності. Проводити самому ремонт електроустаткування забороняється.

8.4 Розрахункова частина

В приміщенні (де відсутні джерела виділення шкідливих речовин) працює одна людина. Робота пов'язана з використанням ПЕОМ. Розміри приміщення: $A = 4$ м, $B = 3,5$ м, $H = 2,8$ м, устаткування займає 15% об'єму. Визначити найменшу необхідну кількість повітря для вентиляції.

Для приміщень, в яких відсутні виділення шкідливих речовин у повітрі, розрахунок вентиляції здійснюється залежно від кількості працюючих.

Необхідна кількість повітря ($\text{м}^3 / \text{год.}$), яка забезпечує відповідність параметрів повітря робочої зони нормованим значенням, визначається за наступною формулою:

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

$$L = L' \cdot N, \quad (8.1)$$

де L' – нормативна кількість повітря на одного працюючого, яка залежить від питомого об'єму приміщення, $\text{м}^3 / (\text{год.} \cdot \text{люд.})$;

N – кількість працюючих.

Питомий об'єм приміщення V_n , ($\text{м}^3 / \text{люд.}$), визначається за формулою:

$$V_n = V / N, \quad (8.2)$$

де V – об'єм приміщення, м^3 .

Визначаємо вільний об'єм приміщення

$$V = A \cdot B \cdot H \cdot 0,85 = 4 \cdot 3,5 \cdot 2,8 \cdot 0,85 = 33,3 \text{ м}^3.$$

Питомий вільний об'єм складає

$$V' = V / N = 33,3 / 1 = 33,3 \text{ м}^3 / \text{люд.} < 20 \text{ м}^3 / \text{люд.}$$

Нормована кількість повітря на одну людину при $V' < 20 \text{ м}^3 / \text{люд.}$ становить $30 \text{ м}^3 / (\text{год.} \cdot \text{люд.})$.

8.5 Висновки до розділу

У даному розділі магістерської роботи проведено аналіз умов працівника робота якого пов'язана з комп'ютерною технікою. Проведено аналіз основних санітарно – гігієнічних показників в заданому приміщенні, де працівник зайнятий постійною роботою за комп'ютером.. Створені умови повинні забезпечувати комфортну роботу. На підставі вивченої літератури з даної проблеми, були зазначені оптимальні параметри мікроклімату, освітлення, допустимі рівні шуму та іонізуючого випромінювання при роботі з ПЕОМ, а також розраховано найменшу необхідну кількість повітря для вентиляції.

Дотримання умов, що визначають оптимальну організацію робочих місць працівників, дозволить зберегти гарну працездатність протягом усього робочого дня, підвищить як в кількісному, так і в якісному відношеннях продуктивність їх праці.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

Список використаних джерел інформації

1. Наказ Міністерства соціальної політики України 14.02.2018 № 207 «Про затвердження Вимог щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями». – *Режим доступу до ресурсу:* <https://zakon.rada.gov.ua/laws/show/z0508> (дата звернення 19.10.23).

2. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПІН 3.3.2-007-98. – *Режим доступу до ресурсу:* <https://zakon.rada.gov.ua/rada/show/v0007282-98>.

3. Наказ Міністерства освіти і науки України 15.08.2016 № 974 «Про затвердження Правил пожежної безпеки для навчальних закладів та установ системи освіти України» – *Режим доступу до ресурсу* <https://zakon.rada.gov.ua/laws/show/z1229-16#Text> (дата звернення 19.10.23).

4. Постанова № 42 від 01.12.1999 Головного державного санітарного лікар України «Санітарні норми мікроклімату виробничих приміщень ДСН 3.3.6.042-99 *Режим доступу до ресурсу:* <https://zakon.rada.gov.ua/rada/show/va042282-99> (дата звернення 19.10.23).

5. Державні будівельні норми України: ДБН В.2.5-28:2018. – *Режим доступу до ресурсу:* <https://goo.su/9AkQ> (дата звернення 19.10.23).

6. Методичні рекомендації до виконання розділу "Заходи з охорони праці та техніки безпеки" випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти для здобувачів вищої освіти спеціальностей 123 "Комп'ютерна інженерія" та 122 "Комп'ютерні науки" / М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. кібербезпеки та програм. забезпечення; [укл. О.В. Оришака, К.М. Марченко]. – Кропивницький: ЦНТУ, 2022. — 19 с. [Електронний ресурс]. – *Режим доступу :* <http://dspace.kntu.kr.ua/jspui/handle/123456789/12240> (дата звернення 19.10.23).

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи комунікації Cloud-сервісів з використанням SIP/VoIP.

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

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти наведені теоретичне узагальнення й рішення наукового завдання дослідження методів комунікації Cloud-сервісів з використанням SIP/VoIP.

Рішення даного завдання полягало у вирішенні наступних задач:

– Був проведений огляд існуючих систем комунікації Cloud-сервісів з використанням SIP/VoIP.

– Досліджена система комунікації Cloud-сервісів з використанням SIP/VoIP.

– На основі отриманих результатів досліджень створена програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання комунікації Cloud-сервісів з використанням SIP/VoIP.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

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.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

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 Delphi 10.4 Sydney. Дана мова програмування дозволяє найбільш ефективно обробляти дані. 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 SEED.

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.

Розроблена програма має реальний економічний ефект від її впровадження у виробництво у сумі 13714 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,38 роки.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		99

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Іванченко Д.О. Дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. ITU-T Recommendation G.723.1. Dual Rate speech coder for multimedia communication transmitting at 5.3 and 6.3 kbit / sec. – 1996.
3. ITU-T Recommendation G.729. Speech codec for multimedia telecommunications transmitting at 8 / 13 kbit / s. – 1996.
4. ITU-T Recommendation H.225.0. Call signaling protocols and media stream packetization for packet-based multimedia communication systems. -Geneva, 1998.
5. ITU-T Recommendation H.245. Control protocol for multimedia communication. -Geneva, 1998
6. ITU-T Recommendation H.248. Gateway control protocol. – Geneva, 2000.
7. ITU-T Recommendation H.320. Narrow-band Visual Telephone Systems and Terminal Equipment. – 1996.
8. ITU-T Recommendation H.321. Adaptation of H.320 Visual Telephone Terminals to B-ISDN Environments. – 1996.
9. ITU-T Recommendation H.322. Visual Telephone Systems and Terminal Equipment for Local Area Networks which Provide a Guaranteed Quality of Service. – 1996.
10. ITU-T Recommendation H.323. Packet based multimedia communication systems. – Geneva, 1998.
11. ITU-T Recommendation H.324. Terminal for Low Bit Rate Multimedia Communications. -1996.
12. ITU-T Recommendation Q.931. ISDN User-Network Interface Layer 3 Specification for Basic Call Control. – 1993.

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		100

13. RFC 2205. Resource Reservation Protocol (RSVP). Ver.1. Functional Specification. – September 1997.
14. RFC 2327. Session Description Protocol. M. Handley, V. Jacobson. April, 1998.
15. RFC 2543. SIP: Session Initiation Protocol. M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg. March 1999.
16. RFC 2705. Media Gateway Control Protocol (MGCP) Version 1.0. M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett. October 1999.
17. RFC 2865. Remote Authentication Dial In User Service (RADIUS). C.Rigney, S. Willens, A. Rubens, W. Simpson. June 2000.
18. RFC 2885. Megaco Protocol 0.8. A.Rayhan, B. Rosen, J. Segers. August 2000.
19. 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.
20. Smirnov, O., Neskrodieva, T., Fedorov, E., Rudakov, K., Neskrodieva, A. «Method Detection Audit Data Anomalies on Basis Restricted Cauchy Machine» CEUR Workshop Proceedings, Volume 3187, 2022,
21. 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.
22. 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>
23. Smirnov O., Kuznetsov A., Zhora V., Onikiyчук 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.

24. 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.

25. 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.

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

27. 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.

28. 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.

29. 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.

30. 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.

31. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhienko 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.

32. 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.

33. 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.

34. 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.

35. 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.

36. 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.

37. 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.

38. 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.

39. 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.

40. 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.

41. 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.

42. 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.

43. 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.

44. 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.

45. Т.В. Смірнова, О.М. Дреєв, О.А. Смірнов «Хмарна інформаційна система оцінювання шорсткості з використанням дискретного частотного аналізу макروفотografій». IV міжнародна науково-практична конференція

“Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 15-16 квітня 2021р. – Кропивницький: ЦНТУ. – 2021. – С. 30.

46. 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.

47. 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.

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

49. 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.

50. 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.

51. 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).

52. 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

					ВКРМ-123.23.0010.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

Додаток А
(обов'язковий)

Технічне завдання

Зміст

1 Найменування та область застосування.....	2
2 Підстава для розробки.....	2
3 Мета та призначення розробки.....	2
4 Джерела розробки.....	2
5 Технічні вимоги.....	2
5.1 Вміст проекту.....	2
5.2 Показники призначення.....	3
5.3 Вимоги до функціональних характеристик.....	3
5.4 Вимоги до архітектури.....	3
5.5 Вимоги до надійності.....	3
5.6 Умови експлуатації.....	4
5.7 Вимоги до складу та параметрів технічних засобів.....	4
5.8 Вимоги до інформаційної і програмної сумісності.....	4
5.8.1 Обладнання.....	4
5.8.2 Мова програмування.....	4
5.8.3 Вхідні дані.....	5
5.8.4 Вихідні дані.....	5
6 Вимоги до програмної документації.....	5
7 Економічні вимоги.....	5
8 Вимоги щодо охорони праці.....	5
9 Перелік документів, що розробляються.....	6
10 Етапи розробки.....	6
11 Порядок контролю та приймання.....	6

					ВКРМ-123.23.0010.00.00.ТЗ			
Вим.	Арк.	№ документа	Підпис	Дата				
Розробив	Іванченко Д.О.				<i>Дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP</i>	Літ.	Аркуш	Аркушів
Перевірів	Смірнов О.А.					М	1	6
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22МЗ			
Затв.	Смірнов О.А.							

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи комунікації Cloud-сервісів з використанням SIP/VoIP.

2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 36-13 від 04.08.2023 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи комунікації Cloud-сервісів з використанням SIP/VoIP.

4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є стосовна до теми література і існуючі аналоги.

5 Технічні вимоги

5.1 Склад продукції

Складниками розробки є:

- вибір і обґрунтування методів реалізації проекту;
- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;

					ВКРМ-123.23.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- техніко-економічне обґрунтування доцільності прийнятого до розробки програмного забезпечення;
- аналіз умов праці;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

5.2 Показники призначення

Система повинна забезпечувати:

- програмну реалізацію системи комунікації Cloud-сервісів з використанням SIP/VoIP;
- цілісність даних у процесі роботи та при зберіганні;
- простий, інтуїтивно зрозумілий інтерфейс.

5.3 Вимоги до функціональних характеристик

Розроблене програмне забезпечення не повинно мати обмежень на версію драйверів та операційної системи.

5.4 Вимоги до архітектури

Компонент, що розробляється повинен використовувати системні засоби та апаратні засоби, що на даному етапі розвитку обчислювальної техніки найбільше поширені.

5.5 Вимоги до надійності

Програмні модулі написані по всім правилам, які стосуються стандартних викликів процедур, функцій, методів і форм, визначених технічною документацією на середовище розробки.

					ВКРМ-123.23.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

- температура повітря: 19-20 град. по Цельсію;
- відносна вологість повітря до 80%;
- атмосферний тиск 107 кПа.

5.7 Вимоги до складу та параметрів технічних засобів

Програмне забезпечення повинно бути реалізоване на ПЕОМ архітектури IBM PC, працювати в ОС Windows 10/11 і з сумісними з цією платформою пристроями і прикладним програмним забезпеченням.

5.8 Вимоги до інформаційної і програмної сумісності

Переносність програмного забезпечення повинна бути забезпечена за рахунок його реалізації стандартного інтерфейсу взаємодії з ОС, що працюють під управлінням ОС Windows 10/11.

5.8.1 Обладнання

Комп'ютер Intel® Celeron/8 Mb/1.2 Gb/SVGA 14" 1Mb або сумісні з ним.

5.8.2 Мова програмування

Середовище Delphi 10.4 Sydney.

					ВКРМ-123.23.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

5.8.3 Вхідні дані

Опис алгоритму роботи запропонованої системи.

5.8.4 Вихідні дані

Робоча програма.

6 Вимоги до програмної документації

Програмна продукція повинна бути представлена у виді опису структури даних, схем та опису алгоритму, а також текстів вихідних модулів програмного забезпечення згідно ЄСПД .

7 Економічні вимоги

7.1 Для ПЗ необхідно виробити функціонально-вартісний аналіз варіантів розробки.

7.2 Виконати розрахунок витрат показників економічного ефекту з урахуванням цін на 3 вересня 2023 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута пожежна безпека.

					ВКРМ-123.23.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

9 Перелік документів, що розробляються

– Наукова новизна	– 1 аркуш.
– Структурна схема системи	– 1 аркуш.
– Функціональна схема системи	– 1 аркуш.
– Діаграма процесів	– 1 аркуш.
– Блок-схема алгоритму роботи програми	– 2 аркуша.
– Показники економічної ефективності	– 1 аркуш.
– Пояснювальна записка	– 105 аркушів.

10 Етапи розробки

10.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти (складання ТЗ).

10.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.

10.3 Розробка функціональних схем, блок схем алгоритмів роботи програмного забезпечення.

10.4 Побудова схем взаємодії даних.

10.5 Створення прототипу ПЗ.

10.6 Віднаходження ПЗ, аналіз отриманих результатів.

10.7 Робота над питанням охорони праці і техніки безпеки.

10.8 Розрахунок з техніко-економічного обґрунтування.

10.9 Оформлення пояснювальної записки і виконання робіт по графічній частині.

11 Порядок контролю та приймання

11.1 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на попередній захист 10.12.2023 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 22.12.2023 р.

					ВКРМ-123.23.0010.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		6

Додаток Б
(обов'язковий)

Міністерство освіти і науки України
Центральноукраїнський національний технічний університет

ЗАТВЕРДЖУЮ

Керівник випускної кваліфікаційної роботи за
другим (магістерським) рівнем вищої освіти
_____ Смірнов О.А.

*Дослідження та програмна реалізація
системи комунікації Cloud-сервісів з використанням SIP/VoIP*

Лістинг програми

Код документу 12

Носій: CD/DVD-диск / USB-флеш-накопичувач

Загальна кількість аркушів: 34

Літера: РП

Кропивницький – 2023 року

Програма організації зв'язку

```

program iptel;

uses
  Forms,
  Dialogs,
  Windows,
  Sysutils,
  Messages,
  usimple in 'usimple.pas' {Form1},
  Unit2 in 'Unit2.pas' {Form2},
  Unit3 in 'Unit3.pas' {Form3},
  Unit1 in 'Unit1.pas' {Form0},
  Unit4 in 'Unit4.pas' {Form4};
{$R simple.RES}
var
  i:integer;
{
  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;}
begin
Application.Initialize;
{
if FileExists('main.dat')=false then
begin
  MessageDlg('Файл main.dat не знайдено! завершення
програми',mtInformation,[mbOK],0);
  Application.Terminate;
end else
begin
  AssignFile(F, 'main.dat');
  Reset(F);
  Readln(F, S);
  CloseFile(F);
  if Crypt(InputBox('Увага!', 'Введіть пароль',Y),KEY,false)=S then
  begin
    MessageDlg('Дякую, пароль вірний!', mtInformation,[mbOK],0);}
    Try
      Form2:=TForm2.Create(Application);
      Form2.Show;
      Form2.Update;
      for i:=1 to 10 do
      begin
        sleep(200);
        Form2.ProgressBar1.Position:=i*10;
        Form2.Update;
      end;
      Application.Title := 'IpTel';
Application.CreateForm(TForm1, Form1);
Application.CreateForm(TForm3, Form3);
Application.CreateForm(TForm0, Form0);
Application.CreateForm(TForm4, Form4);
Finally
  Form2.Free;
End;
Application.Run;
{
  end
  else
  begin
    MessageDlg('Невірний пароль!',mtInformation,[mbOK],0);

```

```
Application.Terminate;  
end;  
end;}  
end.
```

КБПЗ_2023

Підпрограма запуску завантажувального вікна

```
unit U_SPLASH;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,
  Dialogs, jpeg, ExtCtrls, StdCtrls, Gauges;

type
  TForm_SPLASH = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Gauge1: TGauge;
    Timer1: TTimer;
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form_SPLASH: TForm_SPLASH;

implementation

{$R *.dfm}

end.
```

КБПЗ_2023

Головне вікно програми

```

unit usimple;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  telefoon, StdCtrls, ComCtrls, ExtCtrls, jpeg, Menus, Buttons;

type
  TForm1 = class(TForm)
    Telefoon1: TTelefoon;
    StatusBar1: TStatusBar;
    PageControl1: TPageControl;
    TabSheet1: TTabSheet;
    Panel1: TPanel;
    Panel3: TPanel;
    Panel4: TPanel;
    Panel2: TPanel;
    Label1: TLabel;
    Label4: TLabel;
    Edit1: TEdit;
    Button2: TButton;
    Button1: TButton;
    Panel5: TPanel;
    TabSheet2: TTabSheet;
    Panel6: TPanel;
    Label2: TLabel;
    Panel7: TPanel;
    MainMenu: TMainMenu;
    Panel8: TPanel;
    ListBox1: TListBox;
    BitBtn1: TBitBtn;
    BitBtn2: TBitBtn;
    ListBox2: TListBox;
    Panel9: TPanel;
    Label3: TLabel;
    Panel10: TPanel;
    BitBtn3: TBitBtn;
    BitBtn4: TBitBtn;
    N1: TMenuItem;
    IP1: TMenuItem;
    N5: TMenuItem;
    N6: TMenuItem;
    N7: TMenuItem;
    N8: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    BitBtn5: TBitBtn;
    Memol: TMemo;
    TabSheet3: TTabSheet;
    Panel11: TPanel;
    Image3: TImage;
    Label5: TLabel;
    Label6: TLabel;
    Label7: TLabel;
    Label8: TLabel;
    N11: TMenuItem;
    N12: TMenuItem;
    ListBox3: TListBox;
    Label9: TLabel;
    Label10: TLabel;
    N2: TMenuItem;
    PB1: TProgressBar;
    PB2: TProgressBar;
    T1: TTimer;
    Imagel: TImage;
    Timer1: TTimer;
    procedure Button1Click(Sender: TObject);
  end;

```

```

    procedure Button2Click(Sender: TObject);
    procedure ListBox1Click(Sender: TObject);
    procedure ListBox2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn4Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure BitBtn5Click(Sender: TObject);
    procedure N12Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
    procedure N3Click(Sender: TObject);
    procedure N6Click(Sender: TObject);
    procedure N2Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure T1Timer(Sender: TObject);
    procedure Timer1Timer(Sender: TObject);
private
    { Private declarations }
public
    { Public declarations }
end;

var
    Form1: TForm1;

implementation

uses Unit3, Unit2, Unit4, Unit1;
var
    KEY:string='2#$T%&(*qwrda@@@#45';
    {$R *.DFM}
    procedure TForm1.Button1Click(Sender: TObject);
    begin
        Telefoon1.placecall(Edit1.text);
        ListBox3.items.add('D:'+DateToStr(now)+' T:'+TimeToStr(now)+'
        IP:'+Edit1.text);
        T1.Enabled:=true;
    end;
    procedure TForm1.Button2Click(Sender: TObject);
    begin
        Telefoon1.calling:=false;
        T1.Enabled:=false;
        PB1.Position:=0;
        PB2.Position:=0;
    end;
    procedure TForm1.ListBox1Click(Sender: TObject);
    begin
        Edit1.text:=ListBox1.Items.Strings[ListBox1.ItemIndex];
    end;
    procedure TForm1.ListBox2Click(Sender: TObject);
    begin
        Label4.Caption:=ListBox2.Items.Strings[ListBox2.ItemIndex];
    end;
    procedure TForm1.BitBtn1Click(Sender: TObject);
    var
        InputString: string;
    begin
        InputString:= InputBox('Додавання ip адреси', 'Прошу', 'Введення IP не
        здійснено');
        if (InputString<>'Введення IP не здійснено') then
            begin
                Listbox1.Items.add(InputString);
            end;
    end;
    procedure TForm1.BitBtn3Click(Sender: TObject);
    var
        InputString: string;

```

```

begin
  InputString:= InputBox('Додавання легенди ip адреси', 'Прошу', 'Введення
легенди не здійснено');
  if (InputString<>'Введення легенди не здійснено') then
  begin
    Listbox2.Items.add(InputString);
  end;
end;
procedure TForm1.BitBtn2Click(Sender: TObject);
begin
  ListBox1.Items.Strings[ListBox1.ItemIndex]:= '';
  ListBox1.Items.Delete(ListBox1.ItemIndex);
end;

procedure TForm1.BitBtn4Click(Sender: TObject);
begin
  ListBox2.Items.Strings[ListBox1.ItemIndex]:= '';
  ListBox2.Items.Delete(ListBox1.ItemIndex);
end;
procedure TForm1.N10Click(Sender: TObject);
begin
  Application.Terminate;
end;
procedure TForm1.BitBtn5Click(Sender: TObject);
var
  i:integer;
  G:boolean;
begin
  Form3.ListBox1.Clear;
  g:=false;
  for i:=0 to (ListBox1.Items.Count - 1) do
  begin
    if ListBox1.Selected[i] then
    begin
      Form3.ListBox1.Items.add(ListBox1.Items.Strings[i]);
      g:=true;
    end;
    if g then Form3.show;
  end;
end;
procedure TForm1.N12Click(Sender: TObject);
var
  F: TextFile;
  H,S:string;
  OLD:string;
  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;
begin
  if FileExists('main.dat') then
  begin
    AssignFile(F, 'main.dat');
    Reset(F);
    Readln(F, S);
    CloseFile(F);
    if Crypt(InputBox('Увага!', 'Введіть старий пароль', ''),KEY,false)=S then
    begin
      H:=Crypt(InputBox('Увага!', 'Введіть новий пароль', ''),KEY,false);
      if H<>' ' then
      begin
        MessageDlg('Дякую пароль змінено',mtInformation,[mbOK],0);
      end;
    end;
  end;
end;

```

```

        AssignFile(F, 'main.dat');
        Rewrite(F);
        Writeln(F,H);
        CloseFile(F);
    end else MessageDlg('Введіть значення!',mtInformation,[mbOK],0);
end
else
begin
    MessageDlg('Файл main.dat не знайдено чи пароль невірний! завершення
програми',mtInformation,[mbOK],0);
    Application.Terminate;
end;
end;
end;
procedure TForm1.FormCreate(Sender: TObject);
begin
    randomize;
    PB1.Position:=0;
    PB2.Position:=0;
    T1.Enabled:=false;
    Timer1.Enabled:=true;
    if FileExists('MainLegend.dat')=false then
    begin
        MessageDlg('Файл MainLegend.dat не знайдено!',mtInformation,[mbOK],0);
    end else
    begin
        ListBox2.Items.LoadFromFile('MainLegend.dat');
    end;
    if FileExists('MainIP.dat')=false then
    begin
        MessageDlg('Файл MainIP.dat не знайдено!',mtInformation,[mbOK],0);
    end else
    begin
        ListBox1.Items.LoadFromFile('MainIP.dat');
    end;
    if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не знайдено!',mtInformation,[mbOK],0)
    else ListBox3.Items.LoadFromFile('MainHISTORY.dat');
    if FileExists('mainHelp.dat')=false then MessageDlg('Файл mainHelp.dat не
знайдено!',mtInformation,[mbOK],0)
    else Memol.Lines.LoadFromFile('mainHelp.dat');
    end;
    procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
    begin
        if FileExists('MainLegend.dat')=false then MessageDlg('Файл MainLegend.dat не
знайдено!',mtInformation,[mbOK],0)
        else ListBox2.Items.SaveToFile('MainLegend.dat');
        if FileExists('MainIP.dat')=false then MessageDlg('Файл MainIP.dat не
знайдено!',mtInformation,[mbOK],0)
        else ListBox1.Items.SaveToFile('MainIP.dat');
        if FileExists('MainHISTORY.dat')=false then MessageDlg('Файл MainHISTORY.dat
не знайдено!',mtInformation,[mbOK],0)
        else ListBox3.Items.SaveToFile('MainHISTORY.dat');
    end;
    procedure TForm1.N3Click(Sender: TObject);
    begin
        TabSheet2.Visible:=true;
    end;
    procedure TForm1.N6Click(Sender: TObject);
    begin
        Telefoon1.calling:=false
    end;
    procedure TForm1.N2Click(Sender: TObject);
    begin
        Form0.show;
    end;
    procedure TForm1.FormShow(Sender: TObject);
    begin
        PB1.Position:=0;

```

```
PB2.Position:=0;
end;
procedure TForm1.T1Timer(Sender: TObject);
var
  i1,i2:integer;
begin
  PB1.Position:=random(100);
  PB2.Position:=PB1.Position;
end;
procedure TForm1.Timer1Timer(Sender: TObject);
begin
  Form4.show;
  Form1.hide;
  Timer1.Enabled:=false;
end;
end.
```

К6П3_2023

Підпрограма налагодження звукової карти AudioSettings

```

unit Unit1;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ComCtrls, StdCtrls, AMixer, MMSystem;
type
  TForm0 = class(TForm)
    ComboBox1: TComboBox;
    ComboBox2: TComboBox;
    TrackBar: TTrackBar;
    CheckBox: TCheckBox;
    Label1: TLabel;
    Label2: TLabel;
    Mixer: TAudioMixer;
    Label3: TLabel;
    Label4: TLabel;
    ComboBox3: TComboBox;
    LabelStereo: TLabel;
    Button1: TButton;
    procedure FormCreate(Sender: TObject);
    procedure ComboBox1Change(Sender: TObject);
    procedure ComboBox2Change(Sender: TObject);
    procedure MixerControlChange(Sender: TObject; MixerH, ID: Integer);
    procedure TrackBarChange(Sender: TObject);
    procedure CheckBoxClick(Sender: TObject);
    procedure ComboBox3Change(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure FormClose(Sender: TObject; var Action: TCloseAction);
  private
    { Private declarations }
    Setting: Boolean;
  public
    { Public declarations }
  end;
var
  Form0: TForm0;
implementation
uses usimple;
{$R *.DFM}
procedure TForm0.FormCreate(Sender: TObject);
var A: Integer;
begin
  For A := 0 to Mixer.MixerCount - 1 do
    ComboBox3.Items.Add ('Mixer '+IntToStr(A));
  If (ComboBox3.Items.Count > 0) then
    ComboBox3.ItemIndex := 0;
  ComboBox3Change (Sender);
end;
procedure TForm0.ComboBox1Change(Sender: TObject);
var A: Integer;
begin
  ComboBox2.Items.Clear;
  ComboBox2.Items.Add
(Mixer.Destinations[ComboBox1.ItemIndex].Data.szName);
  For A:=0 to Mixer.Destinations[ComboBox1.ItemIndex].Connections.Count-1 do
    ComboBox2.Items.Add(Mixer.Destinations[ComboBox1.ItemIndex].Connections[A].Data.szName);
  If ComboBox2.Items.Count>0 then
    begin
      ComboBox2.ItemIndex:=0;
      ComboBox2Change (Self);
    end;
end;
procedure TForm0.ComboBox2Change(Sender: TObject);
var L,R,M: Integer;
    VD,MD: Boolean;
    Stereo: Boolean;
    IsSelect: Boolean;
begin

```

```

Mixer.GetVolume                               (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,L,R,M,Stereo,VD,MD,IsSelect);
Setting:=True;
TrackBar.Visible:=not VD;
Label1.Visible:=not VD;
Label3.Visible:=VD;
If TrackBar.Visible then
  TrackBar.Position:=L;
CheckBox.Visible:=not MD;
Label2.Visible:=not MD;
Label4.Visible:=MD;
If CheckBox.Visible then
  CheckBox.Checked:=M<>0;
If (Stereo) then
  LabelStereo.Caption := '- stereo -'
else
  LabelStereo.Caption := '- mono -';
Setting:=False;
end;
procedure TForm0.MixerControlChange(Sender: TObject; MixerH, ID: Integer);
begin
  ComboBox2Change (Self);
end;
procedure TForm0.TrackBarChange(Sender: TObject);
begin
  If (not Setting) then
  begin
    Setting:=True;
    Mixer.SetVolume                               (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer(CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.CheckBoxClick(Sender: TObject);
begin
  If not Setting then
  begin
    Setting:=True;
    Mixer.SetVolume                               (ComboBox1.ItemIndex,ComboBox2.ItemIndex-
1,TrackBar.Position,TrackBar.Position,Integer(CheckBox.Checked));
    Setting:=False;
  end;
end;
procedure TForm0.ComboBox3Change(Sender: TObject);
var A:Integer;
begin
  If (ComboBox3.ItemIndex >= 0) AND (ComboBox3.ItemIndex < Mixer.MixerCount)
  then
    Mixer.MixerId := ComboBox3.ItemIndex;
    ComboBox1.Items.Clear;
    If Mixer.MixerCount>0 then
    begin
      For A:=0 to Mixer.Destinations.Count-1 do
        ComboBox1.Items.Add (Mixer.Destinations[A].Data.szName);
      If ComboBox1.Items.Count>0 then
      begin
        ComboBox1.ItemIndex:=0;    ComboBox1Change (Self);
      end;
    end
  else
  begin
    ComboBox1.OnChange:=nil;    ComboBox2.OnChange:=nil;
    TrackBar.OnChange:=nil;    CheckBox.OnClick:=nil;
    MessageDlg ('No mixer present in the system !',mtError,[mbOK],0);
  end;
  Setting:=False;
end;
procedure TForm0.Button1Click(Sender: TObject);
begin

```

```
Form0.hide; Form1.show;  
end;  
procedure TForm0.FormClose(Sender: TObject; var Action: TCloseAction);  
begin  
Form0.hide; Form1.show;  
end;  
end.
```

К6П3_2023

```
    Підпрограма виклику вікна налагодження звукової карти AudioSettings
unit Unit2;

interface

uses
    Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
    jpeg, ExtCtrls, StdCtrls, ComCtrls;

type
    TForm2 = class(TForm)
        Image1: TImage;
        RichEdit1: TRichEdit;
        ProgressBar1: TProgressBar;
    private
        { Private declarations }
    public
        { Public declarations }
    end;

var
    Form2: TForm2;

implementation

{$R *.DFM}

end.
```

КБПЗ_2023

Підпрограма селективного зв'язку

```
unit Unit3;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, ExtCtrls, jpeg, ComCtrls;

type
  TForm3 = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    ListBox1: TListBox;
    Animatel: TAnimate;
    Panel4: TPanel;
    BitBtn2: TBitBtn;
    BitBtn1: TBitBtn;
    BitBtn3: TBitBtn;
    Image1: TImage;
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn1Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form3: TForm3;

implementation

uses usimple;

{$R *.DFM}

procedure TForm3.BitBtn2Click(Sender: TObject);
begin
  form3.hide;
end;

procedure TForm3.BitBtn1Click(Sender: TObject);
begin
  Animatel.Active:=true;
  form1.telefoon1.placecall(form1.edit1.text);
end;

procedure TForm3.BitBtn3Click(Sender: TObject);
begin
  Form1.Telefoon1.calling:=false;
  Animatel.Active:=false;
end;

end.
```

Підпрограма парольного захисту

```

unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask, ExtCtrls;
type
  TForm4 = class(TForm)
    Panell: TPanel;
    M: TMaskEdit;
    BitBtn1: TBitBtn;
    procedure BitBtn1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
var
  Form4: TForm4;
implementation
uses usimple;
VAR
  KEY:string='2#$T%(*qwrdas@@@#45';
  DAT:string='VUW';
{$R *.DFM}
  function Crypt(Text,Key: String; Encode: boolean): String;
  var
    i, KeyLength: integer;
    Sign: ShortInt;
  begin
    KeyLength:=Length(Key);
    if Encode then Sign :=-1 else Sign:=1;
    for i:=1 to Length(Text) do
      Text[i]:=chr(ord(Text[i])+Sign*ord(Key[i mod KeyLength+1]));
    Result:=Text;
  end;
procedure TForm4.BitBtn1Click(Sender: TObject);
var
  F: TextFile;
  i:integer;
  s:string;
  Y:string;
  UUU:string;
begin
if FileExists('main.dat')=TRUE then
  begin
    AssignFile(F, 'main.dat');
    Reset(F);
    Readln(F, S);
    CloseFile(F);
    UUU:=Crypt(M.text,KEY,false);
    if UUU=S then
      begin
        Form4.hide;
        Form1.show;
        MessageDlg('Дякую, пароль вірний!',mtInformation,[mbOK],0);
      end
      else MessageDlg('Введіть пароль!',mtInformation,[mbOK],0);
    end
    else
      begin
        MessageDlg('Файл main.dat не знайдено! завершення
програми',mtInformation,[mbOK],0);
        Application.Terminate;
      end;
end;
end.

```

Розроблений компонент налагодження звука

```

unit AMixer;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms,
  MMSystem;

type
  TAudioMixer=class;

  TPListFreeItemNotify=procedure (Pntr:Pointer) of object;
  TMixerChange=procedure (Sender:TObject;MixerH:HMixer;ID:Integer) of object;
    {MixerH is handle of mixer, which sent this message.
     ID is ID of changed item (line or control).}

  TPointerList=class(TObject)
  private
    FOnFreeItem:TPListFreeItemNotify;
    Items:Tlist;
  protected
    function GetPointer (Ind:Integer):Pointer;
    function GetCount :integer;
  public
    constructor Create;
    destructor Destroy; override;
    procedure Clear;
    procedure Add (Pntr:Pointer);
    property Count:Integer read GetCount;
    property Pointer[Ind:Integer]:Pointer read GetPointer; default;
    property OnFreeItem:TPListFreeItemNotify read FOnFreeItem write
FOnFreeItem;
  end;

  TMixerControls=class(TObject)
  private
    heap:pointer;
    FControls:TPointerList;
  protected
    function GetControl (Ind:Integer):PMixerControl;
    function GetCount:Integer;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Control[Ind:Integer]:PMixerControl read GetControl; default;
    property Count:Integer read GetCount;
  end;

  TMixerConnection=class(TObject)
  private
    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
  public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;
  end;

  TMixerConnections=class(TObject)
  private
    XMixer:TAudioMixer;
    FConnections:TPointerList;
  protected
    procedure DoFreeItem (Pntr:Pointer);
    function GetConnection (Ind:Integer):TMixerConnection;
    function GetCount:Integer;

```

```

public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connection[Ind:Integer]:TMixerConnection read GetConnection;
default;
    property Count:Integer read GetCount;
end;

TMixerDestination=class(TObject)
private
    XMixer:TAudioMixer;
    FData:TMixerLine;
    FControls:TMixerControls;
    FConnections:TMixerConnections;
public
    constructor Create (AMixer:TAudioMixer; AData:TMixerLine);
    destructor Destroy; override;
    property Connections:TMixerConnections read FConnections;
    property Controls:TMixerControls read FControls;
    property Data:TMixerLine read FData;
end;

TMixerDestinations=class(TObject)
private
    FDestinations:TPointerList;
protected
    function GetDestination (Ind:Integer):TMixerDestination;
    procedure DoFreeItem (Pntr:Pointer);
    function GetCount:Integer;
public
    constructor Create (AMixer:TAudioMixer);
    destructor Destroy; override;
    property Count:Integer read GetCount;
    property Destination[Ind:Integer]:TMixerDestination read GetDestination;
default;
end;

TAudioMixer = class(TComponent)
private
    XWndHandle:HWND;

    FDestinations:TMixerDestinations;
    FMixersCount:Integer;
    FMixerHandle:HMixer;
    FMixerId:Integer;
    FMixerCaps:TMixerCaps;
    FDriverVersion: MMVERSION;
    FManufacturer: String;
    FProductId: Word;
    FNumberOfLine: Integer;
    FProductName: String;
    FOnLineChange:TMixerChange;
    FOnControlChange:TMixerChange;
protected
    procedure SetMixerId (Value:Integer);
    procedure MixerCallBack (var Msg:TMessage);
    procedure CloseMixer;
published
    constructor Create (AOwner:TComponent); override;
    destructor Destroy; override;
    property DriverVersion: MMVERSION read FDriverVersion;
    property ProductId: WORD read FProductId;
    property NumberOfLine: Integer read FNumberOfLine;
    property Manufacturer: string read FManufacturer;
    property ProductName: string read FProductName;
    property MixerId:Integer read FMixerId write SetMixerId;
    {Opened mixer - value must be in range 0..MixersCount-1
     If no mixer is opened this value is -1}
    property OnLineChange:TMixerChange read FOnLineChange write FOnLineChange;

```

```

property OnControlChange:TMixerChange read FOnControlChange write
FOnControlChange;
public
function GetVolume (ADestination, AConnection:Integer; var LeftVol,
RightVol, Mute:Integer; var Stereo, VolDisabled, MuteDisabled,
MuteIsSelect:Boolean):Boolean;
{This function return volume of selected Destination and Connection.
ADestination must be from range 0..Destinations.Count-1
AConnection must be in range
0..Destinations[ADestination].Connections.Count-1
If you want to read master volume of some Destination, you have to
set AConnection to -1.
If LeftVol, RightVol or Mute is not supported by queried connection,
it's return value will be -1.

LeftVol and RightVol are in range 0..65536

If Mute is non-zero then the connection is silent (or vice-versa - see
MuteIsSelect parameter)
If specified line is recording source then Mute specifies if programs
will
record from this connection (it is copy of "Select" Checkbox in
standard Windows Volume Control program)
Stereo is true, then this control is stereo.
VolDisabled or MuteDisabled is True when you cannot apply settings to
this
control (but can read it).
MuteIsSelect returns True is "mute" work here as select - opposite of
mute.

Return value of the function is True if no error has ocured,
otherwise it returns False.}
function SetVolume (ADestination, AConnection:Integer; LeftVol, RightVol,
Mute:Integer):Boolean;
{This function sets volume.
If you set RightVol to -1 and connection is stereo then LeftVol will be
copied to RightVol.
If LeftVol or Mute is -1 then this value will not be set.
Note that "Mute" can be "select" (which is reversed mute) - see
function
GetVolume, parameter MuteIsSelect.

Return value is True if ADestination and AConnection are correct,
otherwise False.}

function GetPeak(ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer):Boolean;
function GetMute(ADestination, AConnection:Integer; var
Mute:Boolean):Boolean;
function SetMute(ADestination, AConnection:Integer; Mute:Boolean):Boolean;

property Destinations:TMixerDestinations read FDestinations;
{Ind must be in range 0..DestinationsCount-1}
property MixerCaps:TMixerCaps read FMixerCaps;
property MixerCount:Integer read FMixersCount;
{Number of mixers present in system; mostly 1}
property MixerHandle:HMixer read FMixerHandle;
{Handle of opened mixer}
end;

procedure Register;

implementation

{-----}
{TPointerList}
{-----}

constructor TPointerList.Create;
```

```

begin
  Items := TList.Create;
end;

destructor TPointerList.Destroy;
begin
  Clear;
  Items.Free;
end;

procedure TPointerList.Add (Pntr:Pointer);
begin
  Items.Add (Pntr);
end;

function TPointerList.GetPointer (Ind:Integer):Pointer;
begin
  Result := nil;
  If (Ind < Count) then
    Result := Items[Ind];
end;

procedure TPointerList.Clear;
var I:Integer;
begin
  for I := 0 to Items.Count-1 do begin
    If Assigned (FOnFreeItem) then
      FOnFreeItem (Items[I])
    end;
  Items.Clear;
end;

function TPointerList.GetCount:Integer;
begin
  Result := Items.Count;
end;

{-----}
{TMixerControls}
{-----}
constructor TMixerControls.Create (AMixer:TAudioMixer; AData:TMixerLine);
var MLC:TMixerLineControls;
    A,B:Integer;
    P:PMixerControl;
begin
  FControls := TPointerList.Create;
  GetMem (P, SizeOf(TMixerControl)*AData.cControls);
  heap := P;
  MLC.cbStruct := SizeOf(MLC);
  MLC.dwLineID := AData.dwLineID;
  MLC.cbmxcctrl := SizeOf(TMixerControl);
  MLC.cControls := AData.cControls;
  MLC.pamxcctrl := P;
  A := MixerGetLineControls(AMixer.MixerHandle, @MLC,
MIXER_GETLINECONTROLSF_ALL);
  If A = MMSYSERR_NOERROR then
  begin
    For B := 0 to AData.cControls-1 do
    begin
      FControls.Add (P);
      P := PMixerControl (DWORD(P) + sizeof (TMixerControl));
    end;
  end;
end;

destructor TMixerControls.Destroy;
begin
  FControls.free;
  freemem(heap);
end;

```

```

    inherited;
end;

function TMixerControls.GetControl (Ind:Integer):PMixerControl;
begin
    Result := FControls.Pointer[Ind];
end;

function TMixerControls.GetCount:Integer;
begin
    Result := FControls.Count;
end;

{-----}
{TMixerConnection}
{-----}

constructor TMixerConnection.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
    FData := AData;
    XMixer := AMixer;
    FControls := TMixerControls.Create (AMixer, AData);
end;

destructor TMixerConnection.Destroy;
begin
    FControls.Free;
    inherited;
end;

{-----}
{TMixerConnections}
{-----}

constructor TMixerConnections.Create (AMixer:TAudioMixer; AData:TMixerLine);
var A,B:Integer;
    ML:TMixerLine;
begin
    XMixer := AMixer;
    FConnections := TPointerList.Create;
    FConnections.OnFreeItem := Dofreeitem;
    ML.cbStruct := SizeOf(TMixerLine);
    ML.dwDestination := AData.dwDestination;
    For A := 0 to AData.cConnections-1 do
        begin
            ML.dwSource := A;
            B := MixerGetLineInfo (AMixer.MixerHandle, @ML,
MIXER_GETLINEINFOF_SOURCE);
            If B = MMSYSERR_NOERROR then
                FConnections.Add (Pointer(TMixerConnection.Create (XMixer, ML)));
        end;
    end;
end;

destructor TMixerConnections.Destroy;
begin
    FConnections.Free;
    inherited;
end;

procedure TMixerConnections.DoFreeItem (Pntr:Pointer);
begin
    TMixerConnection(Pntr).Free;
end;

function TMixerConnections.GetConnection (Ind:Integer):TMixerConnection;
begin
    Result := FConnections.Pointer[Ind];
end;

```

```

function TMixerConnections.GetCount:Integer;
begin
  Result := FConnections.Count;
end;

{-----}
{TMixerDestination}
{-----}

constructor TMixerDestination.Create (AMixer:TAudioMixer; AData:TMixerLine);
begin
  FData := AData;
  XMixer := AMixer;
  FConnections := TMixerConnections.Create (XMixer, FData);
  FControls := TMixerControls.Create (XMixer, AData);
end;

destructor TMixerDestination.Destroy;
begin
  Fcontrols.Free;
  FConnections.Free;
  inherited;
end;

{-----}
{TMixerDestinations}
{-----}

constructor TMixerDestinations.Create (AMixer:TAudioMixer);
var A,B:Integer;
    ML:TMixerLine;
begin
  FDestinations := TPointerList.Create;
  FDestinations.OnFreeItem := DoFreeItem;
  if (AMixer = nil) then
    Exit;
  For A := 0 to AMixer.MixerCaps.cDestinations-1 do
  begin
    ML.cbStruct := SizeOf(TMixerLine);
    ML.dwDestination := A;
    B := MixerGetLineInfo (AMixer.MixerHandle, @ML,
MIXER_GETLINEINFOF_DESTINATION);
    If B = MMSYSERR_NOERROR then
      FDestinations.Add (Pointer(TMixerDestination.Create (AMixer, ML)));
    end;
  end;
end;

procedure TMixerDestinations.DoFreeItem (Pntr:Pointer);
begin
  TMixerDestination(Pntr).Free;
end;

destructor TMixerDestinations.Destroy;
begin
  FDestinations.Free;
  inherited;
end;

function TMixerDestinations.GetDestination (Ind:Integer):TMixerDestination;
begin
  Result := nil;
  If (Assigned (FDestinations)) then
    Result := FDestinations.Pointer[Ind];
end;

function TMixerDestinations.GetCount:Integer;
begin
  Result := FDestinations.Count;
end;

```

```

{-----}
{TAudioMixer}
{-----}

constructor TAudioMixer.Create (AOwner:TComponent);
begin
  inherited Create (AOwner);
  XWndHandle := AllocateHwnd (MixerCallBack);
  FMixersCount := mixerGetNumDevs;
  FMixerId := -1;
  if (FMixersCount = 0) then
    FDestinations := TMixerDestinations.Create (nil)
  else
    begin
      FDestinations := nil;
      SetMixerId (0);
    end;
end;

destructor TAudioMixer.Destroy;
begin
  CloseMixer;
  if XWndHandle <> 0 then
    DeAllocateHwnd (XWndHandle);
  inherited;
end;

procedure TAudioMixer.CloseMixer;
begin
  If FMixerId >= 0 then
    begin
      mixerClose (FMixerHandle);
      FMixerId := -1;
    end;
  FDestinations.Free;
  FDestinations := nil;
end;

procedure TAudioMixer.SetMixerId (Value:Integer);
label AllOK;
begin
  If (Value < 0) OR (Value >= FMixersCount) then
    Exit;
  CloseMixer;

  If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW OR
MIXER_OBJECTF_MIXER) = MMSYSERR_NOERROR then
    goto AllOK;

  // we will go here very rarely, but sometimes it could help
  If mixerOpen (@FMixerHandle, Value, XWndHandle, 0, CALLBACK_WINDOW) =
MMSYSERR_NOERROR then
    goto AllOK;
  If mixerOpen (@FMixerHandle, Value, 0, 0, 0) = MMSYSERR_NOERROR then
    goto AllOK;

  // an error has occurred
  FMixerId := -1;
  FDestinations := TMixerDestinations.Create (nil);

  Exit;

AllOK:
  FMixerId := Value;
  mixerGetDevCaps (MixerId, @FMixerCaps, SizeOf (TMixerCaps));

  if FMixerCaps.wMid = MM_MICROSOFT then
    FManufacturer := 'Microsoft'

```

```

else
  FManufacturer := IntToStr(FMixerCaps.wMid) + ' = Unknown';
  FDriverVersion := FMixerCaps.vDriverVersion;
  FProductId := FMixerCaps.wPid;
  FProductName := StrPas(FMixerCaps.szPName);
  FNumberOfLine := FMixerCaps.cDestinations;

  FDestinations := TMixerDestinations.Create (Self);
end;

procedure TAudioMixer.MixerCallBack (var Msg:TMessage);
begin
  case Msg.Msg of
    MM_MIXM_LINE_CHANGE:
      If Assigned (OnLineChange) then
        OnLineChange (Self, Msg.wParam, Msg.lParam);
    MM_MIXM_CONTROL_CHANGE:
      If Assigned (OnControlChange) then
        OnControlChange (Self, Msg.wParam, Msg.lParam);
  else
    Msg.Result := DefWindowProc (XWndHandle, Msg.Msg, Msg.WParam,
Msg.LParam);
  end;
end;

const MIXER_LONG_NAME_CHARS = 64;

type MIXERCONTROLDETAILS_LISTTEXT = record
  dwParam1:DWORD;
  dwParam2:DWORD;
  szName:Array [0..MIXER_LONG_NAME_CHARS-1] of Char;
end;

type ListTextArray = array [0..1000] of MIXERCONTROLDETAILS_LISTTEXT;

function TAudioMixer.GetVolume (ADestination,AConnection:Integer;var LeftVol,
RightVol, Mute:Integer;var Stereo, VolDisabled, MuteDisabled,
MuteIsSelect:Boolean):Boolean;
var MD:TMixerDestination;
    MC:TMixerConnection;
    Cntrl:TMixerControls;
    MCD:TMixerControlDetails;
    Cntrl:PMixerControl;
    A,B:Integer;
    ML:TMixerLine;
    details:array [0..100] of Integer;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;

begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  Stereo := False;
  MuteDisabled := True;
  VolDisabled := True;
  LeftVol := -1;
  RightVol := -1;
  Mute := -1;
  MuteIsSelect := False;
  MD := Destinations[ADestination];
  MC := nil;
  If AConnection <> -1 then
    MC := MD.Connections[AConnection];

  If MD <> nil then
    begin

      Result := True;

```

```

//      If Mute = -1 then
begin
  If AConnection <> -1 then
  begin
    Cntrls := MD.Controls;
    ML := MD.Data;
    If (MC <> nil) AND (Cntrls <> nil) then
    begin
      A := 0;
      while (Mute = -1) AND (A < Cntrls.Count) do
      begin
        Cntrl := Cntrls[A];
//          If (Cntrl.dwControlType AND MIXERCONTROL_CONTROLTYPE_MIXER =
MIXERCONTROL_CONTROLTYPE_MIXER) OR
//          (Cntrl.dwControlType AND MIXERCONTROL_CONTROLTYPE_MUX =
MIXERCONTROL_CONTROLTYPE_MUX) then

//          If (Cntrl.dwControlType AND MIXERCONTROL_CT_CLASS_MASK =
MIXERCONTROL_CT_CLASS_LIST) then

          If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
            // Mux is similar to mixer, but only one line can be selected
at a time
            begin
              MCD.cbStruct := SizeOf(TMixerControlDetails);
              MCD.dwControlID := Cntrl.dwControlID;
              If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                MCD.cChannels := 1
              else
                MCD.cChannels := ML.cChannels;
              If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
                MCD.cMultipleItems := Cntrl.cMultipleItems
              else
                MCD.cMultipleItems := 0;
              MCD.cbDetails := 4;
              MCD.paDetails := @Details;
              B := mixerGetControlDetails
(FMixerHandle, @MCD, MIXER_GETCONTROLDETAILSF_VALUE);
              If B <> MMSYSERR_NOERROR then
              begin
                Inc (A);
                continue;
              end;

              MCDText.cbStruct := sizeof (MCDText);
              MCDText.dwControlID := Cntrl.dwControlID;
              If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                MCDText.cChannels := 1
              else
                MCDText.cChannels := ML.cChannels;
              If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
                MCDText.cMultipleItems := Cntrl.cMultipleItems
              else
                MCDText.cMultipleItems := 0;
              GetMem (ltext, MCDText.cChannels * MCDText.cMultipleItems *
sizeof (MIXERCONTROLDETAILS_LISTTEXT));
              MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
              MCDText.paDetails := ltext;
              B := mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
              If B <> MMSYSERR_NOERROR then
              begin
                FreeMem (ltext);
                Inc (A);
                continue;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

    B := MCD.cChannels - 1;
    while (B < integer(MCD.cMultipleItems)) do
    begin
        if (ltext[B].dwParam1 = MC.Data.dwLineID) then
            break;
        Inc (B, MCD.cChannels);
    end;
    FreeMem (ltext);

    If (B < integer (MCD.cMultipleItems)) then
    begin
        Mute := Details[B];
        MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
        MuteIsSelect := True;
        break;
    end;
    end;
    Inc (A);
end;
end;
end;
end;

If AConnection = -1 then
begin
    Cntrls := MD.Controls;
    ML := MD.Data;
end
else
begin
    If MC <> nil then
    begin
        Cntrls := MC.Controls;
        ML := MC.Data;
    end
    else
        Cntrls := nil;
    end;
    If Cntrls <> nil then
    begin
        A := 0;
        while ((LeftVol = -1) OR (Mute = -1)) AND (A < Cntrls.Count) do
        begin
            Cntrl := Cntrls[A];
            If Cntrl <> nil then
            begin
                If ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) OR
                    ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND (Mute
= -1))) AND
                    (Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE <>
MIXERCONTROL_CONTROLF_MULTIPLE)
                then
                begin
                    if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) then
                        MCD.cbStruct := SizeOf(TMixerControlDetails)
                    else
                        MCD.cbStruct := SizeOf(TMixerControlDetails);
                    MCD.dwControlID := Cntrl.dwControlID;
                    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                        MCD.cChannels := 1
                    else
                        MCD.cChannels := ML.cChannels;
                        MCD.cMultipleItems := 0;
                        MCD.cbDetails := SizeOf(Integer);
                        MCD.paDetails := @details;
                        B := mixerGetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
                        If B = MMSYSERR_NOERROR then

```

```

begin
  If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND
(LeftVol = -1) then
  begin
    VolDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
    LeftVol := details[0];
    If MCD.cChannels > 1 then
    begin
      RightVol := Details[1];
      Stereo := True;
    end
    else
      RightVol := LeftVol;
    end
  else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE)
AND (Mute = -1) then
  begin
    MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
    If Details[0] <> 0 then
      Mute := 1
    else
      Mute := 0;
    end
  // NEW ->
  (*
    else If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF)
AND (Mute = -1) then
  begin
    MuteDisabled := Cntrl.fdwControl AND
MIXERCONTROL_CONTROLF_DISABLED > 0;
    If Details[0] <> 0 then
      Mute := 1
    else
      Mute := 0;
      MuteIsSelect := True;
    end;*)
  // <- NEW
  end;
  end;
  end;
  Inc (A);
  end;

{
  If LeftVol = -1 then
    VolDisabled := True;
  If Mute = -1 then
    MuteDisabled := True;}
  end;
  end;
end;

function TAudioMixer.SetVolume (ADestination, AConnection:Integer; LeftVol,
RightVol, Mute:Integer):Boolean;
var MD:TMixerDestination;
    MC:TMixerConnection;
    Cntrls:TMixerControls;
    MCD:TMixerControlDetails;
    Cntrl:PMixerControl;
    A,B:Integer;
    ML:TMixerLine;
    details:array [0..100] of Integer;
    VolSet,MuteSet:Boolean;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;

```

```

MC := nil;
MD := Destinations[ADestination];
If MD <> nil then
begin
  If AConnection <> -1 then
    MC := MD.Connections[AConnection];

  VolSet := LeftVol = -1;
  MuteSet := Mute = -1;
  Result := True;

  If not MuteSet then
  begin
    If AConnection <> -1 then
    begin
      Cntrls := MD.Controls;
      ML := MD.Data;
      If (MC <> nil) AND (Cntrls <> nil) then
      begin
        A := 0;
        while not MuteSet AND (A < Cntrls.Count) do
        begin
          Cntrl := Cntrls[A];
          If (*(Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR*)
            (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
          begin
            MCD.cbStruct := SizeOf(TMixerControlDetails);
            MCD.dwControlID := Cntrl.dwControlID;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
              MCD.cChannels := 1
            else
              MCD.cChannels := ML.cChannels;
            If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
              MCD.cMultipleItems := Cntrl.cMultipleItems
            else
              MCD.cMultipleItems := 0;
            MCD.cbDetails := 4;
            MCD.paDetails := @Details;
            MuteSet := True;
            mixerGetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
            if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
              For B := 0 to Cntrl.cMultipleItems-1 do
                Details[B] := 0;

            GetMem (ltext, MCD.cChannels * MCD.cMultipleItems * sizeof
(MIXERCONTROLDETAILS_LISTTEXT));
            MCDText.cbStruct := sizeof (MCDText);
            MCDText.dwControlID := Cntrl.dwControlID;
            MCDText.cChannels := MCD.cChannels;
            MCDText.cMultipleItems := MCD.cMultipleItems;
            MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
            MCDText.paDetails := ltext;
            mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
            B := MCD.cChannels - 1;
            while (B < integer (MCD.cMultipleItems)) do
            begin
              if (ltext[B].dwParam1 = MC.Data.dwLineID) then
                break;
              Inc (B, MCD.cChannels);
            end;
            FreeMem (ltext);

            If (B < integer (MCD.cMultipleItems)) then
            begin
              Details[B] := Mute;

```

```

                mixerSetControlDetails          (FMixerHandle,          @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
                break;
            end;
        end;
        Inc (A);
    end;
end;
end;
end;

If AConnection = -1 then
begin
    Cntrl := MD.Controls;
    ML := MD.Data;
end
else
begin
    If MC <> nil then
    begin
        Cntrl := MC.Controls;
        ML := MC.Data;
    end
    else
        Cntrl := nil;
    end;
    If Cntrl <> nil then
    begin

        A := 0;
        while (not VolSet OR not MuteSet) AND (A < Cntrl.Count) do
        begin
            Cntrl := Cntrl[A];
            If Cntrl <> nil then
            begin
                If (((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) AND not
VolSet) OR
                    ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) AND not
MuteSet) (* NEW -> *) (*OR
                    ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) AND not
MuteSet)*) (* <- NEW *) AND
                    (Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE <>
MIXERCONTROL_CONTROLF_MULTIPLE)
                then
                begin
                    MCD.cbStruct := SizeOf(TMixerControlDetails);
                    MCD.dwControlID := Cntrl.dwControlID;
                    If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
                        MCD.cChannels := 1
                    else
                        MCD.cChannels := ML.cChannels;
                    MCD.cMultipleItems := 0;
                    MCD.cbDetails := SizeOf(Integer);
                    MCD.paDetails := @Details;
                    If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_VOLUME) then
                    begin
                        Details[0] := LeftVol;
                        If RightVol = -1 then
                            Details[1] := LeftVol
                        else
                            Details[1] := RightVol;
                        VolSet := True;
                    end
                    else if ((Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUTE) (*
NEW -> *) (* OR
                                (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_ONOFF) *)
                    (* <- NEW *) then

```

```

        begin
            For B := 0 to MCD.cChannels - 1 do
                Details[B] := Mute;
                MuteSet := True;
            end;
            mixerSetControlDetails (FMixerHandle, @MCD,
MIXER_GETCONTROLDETAILSF_VALUE);
        end;
    end;
    Inc (A);
end;

end;
end;
end;

function TAudioMixer.GetMute(ADestination, AConnection: Integer; var Mute:
Boolean):Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mlcMixerLineControlsMute : TMIXERLINECONTROLS;
    mcdMixerDataMute : TMIXERCONTROLDETAILS;
    pmcMixerControlMute : PMIXERCONTROL;
    pmcdsMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
    mlMixerLine : TMixerLine;
    Cntrl:PMixerControl;
    Cntrls:TMixerControls;
    ML:TMixerLine;
    A,B:Integer;
    details:array [0..100] of Integer;
    ltext:^ListTextArray;
    MCDText:TMixerControlDetails;
begin
    Result := False;
    If (not Assigned (FDestinations)) then
        Exit;
    MC := nil;
    Mute := False;
    MD := Destinations[ADestination];
    if MD <> nil then
        begin
            if AConnection = -1 then
                mlMixerLine := MD.Data
            else
                begin
                    MC := MD.Connections[AConnection];
                    if MC <> nil then
                        mlMixerLine := MC.Data
                    else
                        Exit;
                    end;
                end;
            GetMem(pmcMixerControlMute, SizeOf(TMIXERCONTROL));
            GetMem(pmcdsMixerDataUnsignedMute, SizeOf(TMIXERCONTROLDETAILSBOOLEAN));

            with mlcMixerLineControlsMute do
                begin
                    cbStruct := SizeOf(TMIXERLINECONTROLS);
                    dwLineID := mlMixerLine.dwLineID;
                    dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
                    cControls := 1;
                    cbmxctrl := SizeOf(TMIXERCONTROL);
                    pamxctrl := pmcMixerControlMute;
                end;

                if (mixerGetLineControls(FMixerHandle, @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
                    begin

```

```

with mcdMixerDataMute do
begin
  cbStruct := SizeOf(TMIXERCONTROLDETAILS);
  dwControlID := pmcMixerControlMute^.dwControlID;
  cChannels := 1;
  cMultipleItems := 0;
  cbDetails := SizeOf(TMIXERCONTROLDETAILSBOOLEAN);
  paDetails := pmcdsMixerDataUnsignedMute;
end;

  if mixerGetControlDetails(FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR then
  begin
    Mute := pmcdsMixerDataUnsignedMute^.fValue = 1;
    Result := True;
  end;
end
else
begin
  If (AConnection <> -1) then
  begin
    Cntrl := MD.Controls;
    ML := MD.Data;
    If (MC <> nil) AND (Cntrl <> nil) then
    begin
      A := 0;
      while (Result = False) AND (A < Cntrl.Count) do
      begin
        Cntrl := Cntrl[A];
        If (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MIXER) OR
(Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
        begin
          mcdMixerDataMute.cbStruct := SizeOf(TMixerControlDetails);
          mcdMixerDataMute.dwControlID := Cntrl.dwControlID;
          If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_UNIFORM > 0 then
            mcdMixerDataMute.cChannels := 1
          else
            mcdMixerDataMute.cChannels := ML.cChannels;
          If Cntrl.fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE =
MIXERCONTROL_CONTROLF_MULTIPLE then
            mcdMixerDataMute.cMultipleItems := Cntrl.cMultipleItems
          else
            mcdMixerDataMute.cMultipleItems := 0;
          mcdMixerDataMute.cbDetails := 4;
          mcdMixerDataMute.paDetails := @Details;
          mixerGetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE);

          GetMem (ltext, mcdMixerDataMute.cChannels *
mcdMixerDataMute.cMultipleItems * sizeof (MIXERCONTROLDETAILS_LISTTEXT)); *
          MCDText.cbStruct := sizeof (MCDText);
          MCDText.dwControlID := Cntrl.dwControlID;
          MCDText.cChannels := mcdMixerDataMute.cChannels;
          MCDText.cMultipleItems := mcdMixerDataMute.cMultipleItems;
          MCDText.cbDetails := sizeof (MIXERCONTROLDETAILS_LISTTEXT);
          MCDText.paDetails := ltext;
          mixerGetControlDetails (FMixerHandle, @MCDText,
MIXER_GETCONTROLDETAILSF_LISTTEXT);
          B := mcdMixerDataMute.cChannels - 1;
          while (B < integer (mcdMixerDataMute.cMultipleItems)) do
          begin
            if (ltext[B].dwParam1 = MC.Data.dwLineID) then
              break;
            Inc (B, mcdMixerDataMute.cChannels);
          end;
          FreeMem (ltext);

          If (B < integer (mcdMixerDataMute.cMultipleItems)) then
          begin

```

```

        Result := True;
        Mute := Details[B] <> 0;
        break;
    end;
end;
end;
Inc (A);
end;
end;
end;
end;

FreeMem (pmcdsMixerDataUnsignedMute);
FreeMem (pmcMixerControlMute);
end;
end;

function TAudioMixer.SetMute (ADestination, AConnection: Integer; Mute:
Boolean): Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mlcMixerLineControlsMute : TMIXERLINECONTROLS;
    mcdMixerDataMute : TMIXERCONTROLDETAILS;
    pmcMixerControlMute : PMIXERCONTROL;
    pmcdsMixerDataUnsignedMute : PMIXERCONTROLDETAILSBOOLEAN;
    mlMixerLine : TMixerLine;
    Cntrl: PMixerControl;
    Cntrls: TMixerControls;
    ML: TMixerLine;
    A, B: Integer;
    details: array [0..100] of Integer;
    ltext: ^ListTextArray;
    MCDText: TMixerControlDetails;
begin
    Result := False;
    If (not Assigned (FDestinations)) then
        Exit;
    MC := nil;
    MD := Destinations[ADestination];
    if MD <> nil then
        begin
            if AConnection = -1 then
                mlMixerLine := MD.Data
            else
                begin
                    MC := MD.Connections[AConnection];
                    if MC <> nil then
                        mlMixerLine := MC.Data
                    else
                        Exit;
                    end;
                end;
        end;

    GetMem (pmcMixerControlMute, SizeOf (TMIXERCONTROL));
    GetMem (pmcdsMixerDataUnsignedMute, SizeOf (TMIXERCONTROLDETAILSBOOLEAN));

    with mlcMixerLineControlsMute do
        begin
            cbStruct := SizeOf (TMIXERLINECONTROLS);
            dwLineID := mlMixerLine.dwLineID;
            dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
            cControls := 0;
            cbmxctrl := SizeOf (TMIXERCONTROL);
            pamxctrl := pmcMixerControlMute;
        end;

        if (mixerGetLineControls (FMixerHandle, @mlcMixerLineControlsMute,
MIXER_GETLINECONTROLSF_ONEBYTYPE) = MMSYSERR_NOERROR) then
            begin
                with mcdMixerDataMute do

```



```

while (B < integer (mcdMixerDataMute.cMultipleItems)) do
begin
  if (ltext[B].dwParam1 = MC.Data.dwLineID) then
    break;
  Inc (B, mcdMixerDataMute.cChannels);
end;
FreeMem (ltext);

If (B < integer (mcdMixerDataMute.cMultipleItems)) then
  Details[B] := 1;
end;
if (mixerSetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) = MMSYSERR_NOERROR) then
begin
  Result := True;
  break;
end;
end;
Inc (A);
end;
end;
end;
end;

FreeMem (pmcMixerDataUnsignedMute);
FreeMem (pmcMixerControlMute);
end;
end;

function TAudioMixer.GetPeak (ADestination, AConnection:Integer; var LeftPeak,
RightPeak:Integer): Boolean;
var
  MD : TMixerDestination;
  MC : TMixerConnection;
  mcdMixerDataPeak : TMIXERCONTROLDETAILS;
  pmcMixerControlPeak : PMIXERCONTROL;
  { pmcMixerDataSignedPeak : PMIXERCONTROLDETAILSSIGNED;}
  mlMixerLine : TMixerLine;
  A:Integer;
  Cntrl:TMixerControls;
  Details:Array [1..100] of Integer;
begin
  Result := False;
  If (not Assigned (FDestinations)) then
    Exit;
  LeftPeak := 0;
  RightPeak := 0;
  MD := Destinations[ADestination];
  if MD <> nil then
  begin
    if AConnection = -1 then
    begin
      mlMixerLine := MD.Data;
      Cntrl := MD.Controls;
    end
    else
    begin
      MC := MD.Connections[AConnection];
      if MC <> nil then
      begin
        mlMixerLine := MC.Data;
        Cntrl := MC.Controls;
      end
      else
        Exit;
    end;
  end;
  GetMem (pmcMixerControlPeak, SizeOf (TMIXERCONTROL));

  A := 0;

```

```

while (A < Cntrls.Count) do
begin
  If (Cntrls[A].dwControlType AND MIXERCONTROL_CT_CLASS_MASK) =
MIXERCONTROL_CT_CLASS_METER then
    break;
  Inc (A);
end;
If A = Cntrls.Count then
begin
  FreeMem(pmcMixerControlPeak);
  Exit;
end;

with mcdMixerDataPeak do
begin
  cbStruct := SizeOf(TMIXERCONTROLDETAILS);
  dwControlID := Cntrls[A].dwControlID;
  cChannels := mlMixerLine.cChannels;
  If (Cntrls[A].fdwControl AND MIXERCONTROL_CONTROLF_MULTIPLE) =
MIXERCONTROL_CONTROLF_MULTIPLE then
    cMultipleItems:=Cntrls[A].cMultipleItems
  else
    cMultipleItems:=0;
  cbDetails := SizeOf(TMIXERCONTROLDETAILSSIGNED);
  paDetails := @Details;
end;
if
(mixerGetControlDetails(FMixerHandle,@mcdMixerDataPeak,MIXER_GETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
begin
  LeftPeak := Details[1];
  if mlMixerLine.cChannels = 2 then
    RightPeak := Details[2]
  else
    RightPeak := LeftPeak;
  Result := True;
end;
FreeMem(pmcMixerControlPeak);
end;
end;
procedure Register;
begin
  RegisterComponents('Samples', [TAudioMixer]);
end;
end.

```