

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

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

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи розширеного
набору телекомунікаційних сервісів RCS на мобільних
пристроях”

Виконав здобувач вищої освіти
II курсу, групи КІ-22М-2
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Ключкін Є.Ю.
« ____ » _____ 2023 р.

Керівник проекту
доктор технічних наук, доцент
_____ Коваленко О.В.
« ____ » _____ 2023 р.

Рецензент _____

Центральноукраїнський національний технічний університет
Факультет Механіко-технологічний
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

Ключкіну Євгену Юрійовичу

(прізвище, ім'я, по батькові)

- | | | | | | | | | | | | |
|--|--|--|----------------------------|---|---|--|--|--|---------------------|--|--|
| 1. Тема роботи | <i>Дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях</i> | | | | | | | | | | |
| 2. Керівник роботи | <i>Коваленко Олександр Володимирович, докт. техн. наук, доцент</i>
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом вищого навчального закладу № 35-13 від 04.08.2023 року | | | | | | | | | | |
| 3. Строк подання студентом роботи до захисту | <i>10.12.2023 р.</i> | | | | | | | | | | |
| 4. Мета та завдання випускної кваліфікаційної роботи: | <i>Метою розробки є дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях</i> | | | | | | | | | | |
| 5. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) | <table border="1"><tr><td><i>1. Призначення та область використання.</i></td><td><i>6. Наукова новизна.</i></td></tr><tr><td><i>2. Перегляд аналогічних існуючих систем.</i></td><td><i>7. Економічна ефективність розробленої програми.</i></td></tr><tr><td><i>3. Опис і обґрунтування проектних рішень.</i></td><td><i>8. Заходи з охорони праці та техніки безпеки.</i></td></tr><tr><td><i>4. Етапи програмування системи.</i></td><td><i>9. Висновки.</i></td></tr><tr><td><i>5. Впровадження системи в промислову експлуатацію</i></td><td></td></tr></table> | <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | <i>5. Впровадження системи в промислову експлуатацію</i> | |
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> | | | | | | | | | | |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Економічна ефективність розробленої програми.</i> | | | | | | | | | | |
| <i>3. Опис і обґрунтування проектних рішень.</i> | <i>8. Заходи з охорони праці та техніки безпеки.</i> | | | | | | | | | | |
| <i>4. Етапи програмування системи.</i> | <i>9. Висновки.</i> | | | | | | | | | | |
| <i>5. Впровадження системи в промислову експлуатацію</i> | | | | | | | | | | | |
| 6. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) | | | | | | | | | | | |
| <i>Наукова новизна</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Структурна схема системи</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Функціональна схема системи</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Діаграма процесів</i> | <i>1 аркуш</i> | | | | | | | | | | |
| <i>Блок-схема алгоритму роботи додатку</i> | <i>2 аркуша</i> | | | | | | | | | | |
| <i>Показники економічної ефективності</i> | <i>1 аркуш</i> | | | | | | | | | | |

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

Ключкін Є.Ю. Дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях. 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 розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Метою розробки є дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Об'єктом дослідження є процес розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Предметом дослідження є методи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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

Результат роботи – програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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 мобільному пристрої з ОС Android/iOS.

Програму розроблено в середовищі RAD Studio Delphi.

Ключові слова: комп'ютерна інженерія, телекомунікаційні сервіси

ABSTRACT

Kliuchkin E.Yu. Research and software implementation of the system of an extended set of telecommunication services RCS on mobile devices. 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 system of an extended set of telecommunication services RCS on mobile devices.

The purpose of the development is the research and software implementation of the system of an extended set of telecommunication services RCS on mobile devices.

The object of the study is the process of an expanded set of telecommunication services RCS on mobile devices.

The subject of the study is methods of an expanded set of RCS telecommunication services on mobile devices.

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 system of an extended set of telecommunication services RCS on mobile devices.

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 a mobile device with Android/iOS OS.

The program was developed in the RAD Studio Delphi environment.

Keywords: computer engineering, telecommunication services

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	7
1.1 Призначення системи.....	7
1.2 Область застосування.....	8
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	13
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	13
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	20
2.3 Розгорнута постановка завдання	25
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	27
3.1 Опис функціонування системи	27
3.2 Розробка структурної схеми.....	33
3.3 Розробка функціональної схеми	38
3.4 Розробка діаграми процесів.....	57
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	59
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	59
4.2 Захист розробленого програмного забезпечення.....	69
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	72
6 НАУКОВА НОВИЗНА	74

					ВКРМ-123.23.0037.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Ключкін Є.Ю.</i>					М	1	114
<i>Перев.</i>	<i>Коваленко О.В.</i>					ЦНТУ КІ-22М-2		
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>							

7 ЕКОНОМІЧНА ЕФЕКТИВНІСТЬ РОЗРОБЛЕНОЇ ПРОГРАМИ.....	75
7.1 Техніко економічне обґрунтування теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	75
7.2 Розрахунок трудомісткості розробки програмної продукції.....	77
7.3 Визначення чисельності виконавців і планового фонду зарплати.....	79
7.4 Розрахунок капітальних вкладень та амортизаційних відрахувань у розробника.....	83
7.5 Визначення собівартості розробки та ціни програмної продукції.....	87
7.6 Визначення об'єму капітальних вкладень та експлуатаційних витрат у споживача програмної продукції.....	91
7.7 Визначення експлуатаційних витрат.....	92
7.8 Визначення економічної ефективності програмної продукції.....	93
7.9 Висновок.....	95
8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	96
8.1 Вступ	96
8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером.....	98
8.3 Аналіз санітарно-гігієнічних умов праці на робочому місці програміста ...	99
8.4 Розробка заходів з умов поліпшення охорони праці.....	102
8.5 Розрахункова частина	102
9 ОСНОВНІ ВИСНОВКИ.....	106
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	108

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

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

ВСТУП

Актуальність теми. З поширенням смартфонів відео стає наступним логічним кроком в еволюції мобільних сервісів. Однак для дійсно широкого поширення відео на мобільних пристроях треба буде розв'язати безліч проблем самого різного характеру. Уже зараз на відео доводиться половина мобільного трафіку, а до 2025 року, за прогнозом Cisco, його частка складе як мінімум дві третини. При цьому, згідно Cisco Visual Networking Index, обсяг переданих відеоданих буде рости швидше, ніж всі інші види мобільного трафіку – на 75% щорічно.

Як і будь-який прогноз, наведені цифри варто сприймати критично. Так, в 2010 році Cisco уже пророкувала досягнення вищезгаданого рубежу – дві третини від усього мобільного трафіку відео повинне було скласти в 2022 році, але, як бачимо, цю дату довелося скорегувати й перенести на більше пізній строк. Та й у цілому обсяг переданих даних росте не так швидко, як очікувалося. Наприклад, замість очікуваних 118% його ріст у США в 2023 році склав 64%. І всі такі темпи росту досить високі, а користувачі, як свідчать самі різні опитування, проявляють зацікавленість у застосуванні відеододатків. За даними дослідження Rovi's Consumer Study Survey, оприлюдненого в ході проведення Mobile World Congress 2023 у Барселоні, європейські користувачі планшетів і смартфонів дивляться потокове відео два-три рази в тиждень, при цьому середня тривалість сеансу становить 30 хв. Однак для дійсно широкого поширення відео на мобільних пристроях треба буде розв'язати безліч проблем самого різного порядку: від виділення спектра частот до монетизації сервісів, від оптимізації доставки контенту до забезпечення сумісності.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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

– Огляд існуючих систем розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

– Дослідження системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

– Програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Об'єктом дослідження є процес розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Предметом дослідження є методи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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 розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

– Розроблено вітчизняний продукт розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

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 розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

КБГІЗ-2023

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Стільникові оператори продовжують шукати найкращу лінію поведінки в умовах втрати доходів від SMS і голосового зв'язку, що стало наслідком доступності безкоштовних альтернатив, таких як Skype, Viber, Facebook Messenger, WhatsApp і ін. Властивоосновних можливостей дві: пропозиції власних привабливих сервісів і співробітництво з контент-провайдерами.

Розширений набір комунікаційних сервісів Reach Communications Services (RCS) дозволяє запропонувати абонентам (поряд з голосовим зв'язком і SMS) індикацію присутності, миттєвий обмін повідомленнями, передачу файлів і відеодзвінки – і все це незалежно від виду використовуваного пристрою й від того, до мережі якого оператора він підключений. Тим часом із впровадженням RCS оператори явно запізнюються, не говорячи вже про те, що така ключова функціональність, як підтримка відео, з'явилася лише в специфікації RCS 5.1, прийнятої торік.

З одного боку, оператори продовжують розглядати RCS як спосіб розширення комунікаційних можливостей і платформи для надання сторонніх сервісів. З іншого боку, до широкої доступності відповідних сервісів ще далеко. Так, напередодні MWC 2023 оператор Deutsche Telekom, один з п'ятірки ключових європейських пропонентів RCS, заявив, що він «відкладає запуск власних сервісів joyn (бренд, під яким GSMA просуває RCS) на невизначений час внаслідок технічних проблем».

Крім того, базовий набір комунікаційних сервісів RCS не пропонує нічого нового в порівнянні із уже реалізованими провайдерами OTT.

До того ж всі підтримувані RCS комунікаційні можливості незабаром, напевно, стануть доступні в будь-якому браузері без установки яких-небудь

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

модулів, що підключаються, (plug-in). На це націлено проект розробки відкритого вихідного коду Real Time Communication (або Web RTC). Відкривши вікно відеочату, абоненти зможуть легко продемонструвати фото й відео або ж обмінятися посиланнями, документами й іншою інформацією, просто перетягнувши відповідний об'єкт у вікно.

Всі необхідні компоненти – WebRTC, getUserMedia, PeerConnection, DataChannels – уже реалізовані в браузері Firefox. getUserMedia дозволяє перехоплювати дані з відеокамери або мікрофона (при наявності дозволу, природно), PeerConnection – здійснювати захищені аудіо- і відеодзвінки, а DataChannels – обмінюватися даними. До слова, комунікації шифруються, ніж нівелюється ще одне заявляється переваження, ЩО, RCS – захищеність сервісів.

Mozilla, при сприянні Ericsson і AT&T, продемонструвала на доказ даної концепції на MWC можливість здійснення відеовикликів без яких-небудь плагінів по номеру в телефонній книзі смартфона. Таким чином, як було показано, браузер може виконувати багато функцій мобільного пристрою, такі як голосові й відеодзвінки, обмін SMS і MMS. Крім Mozilla, у робочу групу W3C WebRTC входить, зокрема, Google. Одним з ініціаторів розробки WebRTC є також Opera. І кому знадобиться RCS при такій широкій підтримці з боку браузерів (Microsoft, як звичайно, пропонує свій варіант – CU-WebRTC).

1.2 Область застосування

Із всіх можливих відеосервісів явно був виділений один напрямок – мобільне ТБ. Зважаючи на все, саме його стільникові оператори розглядають як найбільш привабливе з погляду можливості добування доходу. Відповідно до опитування Mobile Content Venture, більше 50% жителів США виявляють цікавість до перегляду телевізійних програм на своїх смартфонах і планшетах. 68% респондентів указали, що вони дивилися б ТБ частіше, якби потрібні їм

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

канали були доступні на їхніх мобільних пристроях. А 61% навіть затверджує, що вони готові перемінити оператора для одержання доступу до сервісів мобільного ТБ.

Опитування опитуваннями, але в реальності мобільне ТБ поки не робить погоди ні для користувачів, ні для операторів. Як показує інше дослідження, англійського агентства Thinkbox, у Великобританії середньостатистичний глядач проводить у телевізора 4 год 4 хв у день, і тільки 3 хв із сумарного часу перегляду доводиться на смартфони, планшети й ноутбуки. До того ж при мобільному доступі пріоритет віддається відео за запитом, а не телемовленню. Із цього погляду перспективи в мобільного ТБ безрадісні. Але не все так похмуро. Бажаючих дивитися ТБ на мобільних пристроях чимало.

Уже зараз на відео доводиться половина всього трафіку в стільникових мережах. Збільшення завантаження мережі вимагає більше ефективних способів доставки контенту. І отут як не можна до речі може виявитися прийняття нового стандарту H.265, що був схвалений у січні цього року. Цей стандарт пропонує високоефективне кодування відео (High Efficiency Video Coding, HEVC) – у порівнянні зі своїм попередником кодеком H.264 AVC йому необхідна у два рази менша пропускна здатність при тій же якості.

Зниження вимог до каналу досягається за рахунок ряду мер. Зокрема, в H.265 використовується більше гнучкий підхід до формування блоків. В H.264 зображення розбивається на блоки розміром 16x16 пікселів, тоді як в HEVC максимальний розмір блоку може досягати 64x64 пікселів. Блоки більшого розміру дозволяють домогтися набагато більшої ефективності кодування, особливо у випадку відео високого дозволу (HD). Завдяки гнучкості у виборі розміру блоків і збільшенню їхнього максимального розміру, для кодування руху потрібно менше даних.

Природно, кодек HEVC розроблявся не тільки розраховуючи на трансляцію ТБ-каналів, але саме ця область застосування розглядається як найбільш перспективна на першому етапі його поширення.

						ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			9

З'явившись дуже вчасно, HEVC негайно був включений поруч виробників у рішення для доставки відеоконтенту в мережах LTE (H.265 найбільш ефективний у випадку відеопотоку з HD-Якістю, для чого в кожному разі потрібні мегабітні швидкості). На MWC 2023 повні рішення для реалізації віщання в мережах LTE представили компанії Ericsson і Huawei. Поряд з HEVC, необхідними елементами є eMBMS і MPEG-DASH.

Evolved Multimedia Broadcast Multicast Service (eMBMS), стандарт 3GPP, покликаний полегшити реалізацію в стільникових мережах широкомовних сервісів. Крім властиво віщання, він дозволяє оптимізувати також доставку контенту за запитом (у принципі, eMBMS може використовуватися для розсилання будь-яких файлів). При цьому для кодування віщального і потокового відео використовується один кодувальник, що скорочує витрати на інфраструктуру. MPEG Dynamic Adaptive Streaming over HTTP (DASH) спрощує й стандартизує адаптивну доставку відео на споживчі пристрої: потік розбивається на послідовність медіасегментів, які потім передаються як незалежні файли.

Таке рішення дозволяє забезпечити доставку користувачам відеоконтенту з високою якістю навіть у місцях масового скупчення людей, наприклад на стадіонах.

Досить імовірно, що відеотелефонії вдасться домогтися більших успіхів у мобільних мережах, ніж у випадку стаціонарного телефонного зв'язку. Дійсно, сучасні моделі смартфонів за замовчуванням оснащуються фронтальними камерами, а по обчислювальній потужності й функціональності вони порівнянні з комп'ютерами, так що спеціальних апаратів для здійснення відеовикликів здобувати не треба. Поліпшення користувацьких інтерфейсів і спеціалізованих технологій сприяє підвищенню привабливості відеодзвінків. По оцінці Juniper Networks, до 2025 року відповідні можливості будуть затребувані 160 млн чоловік (причому більшість із них у робочих цілях). Як прогнозує InStat, в 2023 році дохід від мобільних відеодзвінків перевищить 1 млрд доларів.

Проте провайдерам і операторам ще має бути вишукати спосіб монетизації відеодзвінків. Для корпоративного використання важлива якість зв'язку, і цієї

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

категорії замовників оператори можуть запропонувати платні відеодзвінки з гарантованими параметрами. При великій кількості викликів використання індикатора пріоритету потоку (Flow Priority Indicator, FPI) дозволяє надати відеодзвінкам перевага – мережні ресурси виділяються в першу чергу тим, хто придбав передплату на преміальні сервіси.

Природним розширенням відеодзвінків є відеоконференції. Як уважають в Wainhouse Research, по охопту аудиторії мобільні відеододатки незабаром обженуть всі інші пропозиції відеоконференцій. Тенденція ринку складається у відмові від дорогих систем телеприсутності на користь мобільних додатків (що підтверджується зниженням продажів систем Telepresence). Дійсно, мобільні співробітники проводять в офісі менше чверті свого робочого часу, а мобільні відеоконференції дозволяють організовувати зустрічі в будь-яке зручне для всіх час.

На ринку з'являється усе більше рішень для реалізації мобільних відеоконференцій, які поряд зі своєю основною функціональністю підтримують інструменти для спільної роботи, як Cisco WebEx. Так, вихідці з Cisco заснували компанію Zoom.us. Особливим предметом гордості розроблювачів є підтримка конференції до 25 учасників на мобільній пристрій. Для використання Web-Ex подібного інструмента Meet.FM від YouDazzle досить браузера. Можливість спільного використання контенту на планшеті безпосередньо в ході відеоконференції була додана й у нову версію додатка Polycom RealPresence Mobile 2.2, що компанія представила на MWC 2023. Цей контент можна відправляти всім іншим учасникам відеонаради, поза залежністю від того, до якого пристрою вони підключені.

Крім того, у версію 2.2 додана підтримка масштабованого кодеку (Scalable Video Coding, SVC), що, як затверджується, дозволяє втриє збільшити ємність HD-Відеозв'язку. Відповідно до оцінки Wainhouse Research, завдяки використанню SVC можна на порядок зменшити витрати на з'єднання при проведенні конференцій з більшим числом учасників. Більше ефективний у частині необхідної пропускну здатності H.265 поки не має масштабованого варіанта, але, як заявили Vidyo і Samsung, вони мають намір підготувати й

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

представити відповідне розширення для стандартизації, що буде сприяти ще більшому підвищенню гнучкості й ефективності відеокommунікацій по мобільних мережах.

У деяких ситуаціях передача відео по стільниковій мережі виявляється єдиним можливим рішенням, наприклад при організації відеоспостереження на транспорті (справді, не супутникову ж тарілку встановлювати на трейлер або поліцейську машину). Рішення ізраїльської компанії SerVision дозволяє організувати відеоспостереження на поїздах, патрульних автомобілях, таксі, каретах швидкої допомоги й т.д.

Завдяки використанню власного алгоритму стиску, рішення оптимізоване для мереж з низькою пропускнуою здатністю – спостереження можливо по мережі 2G при швидкості каналу всього 9 Кбіт/с. Звичайно, чудес не буває, і на якісну картинку сподіватися не коштує. Однак зображення не завмирає (частота кадрів скорочується до трьох у секунду), а що загалом відбувається розглянути цілком вдається.

Цифрові вивіски (Digital Signage) зі способу інформування перетворюються в засіб взаємодії з потенційним покупцем. Для цього французька компанія Think&Go запропонувала використовувати можливості технології ближнього зв'язку (Near Field Communication), що активно підтримується GSMA і вбудовується в усі більше число смартфонів. Представлений на виставці «динамічний NFC екран» Dynamic NFC-Screen дозволяє споживачам взаємодіяти «крізь» нього за допомогою смартфона з NFC. Покупці можуть зчитувати інформацію про продукт, одержувати купон на знижку й переміщатися по різних інформаційним меню. При цьому контент NFC міняється динамічно синхронно зі зміною відеоконтенту.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		12

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

З будь-якого смартфона можна подзвонити двома способами: через звичайну телефонну лінію й через Інтернет. Другий спосіб стає все популярніше – адже швидкість мобільних інтернет-підключень останнім часом різко виросла, а плата за них стала, по колишніх мірках, символічної.

Отже, установіть на смартфон обраний додаток – і без обмежень спілкуйтеся з усіма його користувачами по усьому світі, оплачуючи тільки трафік. А деякі сервіси дозволять дзвонити й на звичайні телефони, причому дешево. iBusiness представляє огляд кращих безкоштовних програм для голосового спілкування.

Skype

Платформи: Android, iOS, Windows Phone, BlackBerry, Windows, Mac OS X, Linux.

Перша версія Skype з'явилася в 2003 році. Із цього часу програма, що швидко набрала популярність, продовжує залишатися однією з найвідоміших у своїй категорії. А коли її за гігантські гроші купила Microsoft, стало ясно, що Skype буде таким же вічним, як Windows.

Skype володіє, мабуть, самою широкою функціональністю із всіх аналогічних програм. Крім безкоштовних аудіо- або відеодзвінків іншим користувачам, він уміє зв'язуватися зі стаціонарними й мобільними телефонами, проводити групові відеозустрічі й навіть улаштовувати повноцінні телеконференції! У деяких країнах користувачі можуть зв'язати свій обліковий

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

запис Skype з місцевим номером, що дає можливість приймати на Skype дзвінки зі звичайних телефонів.

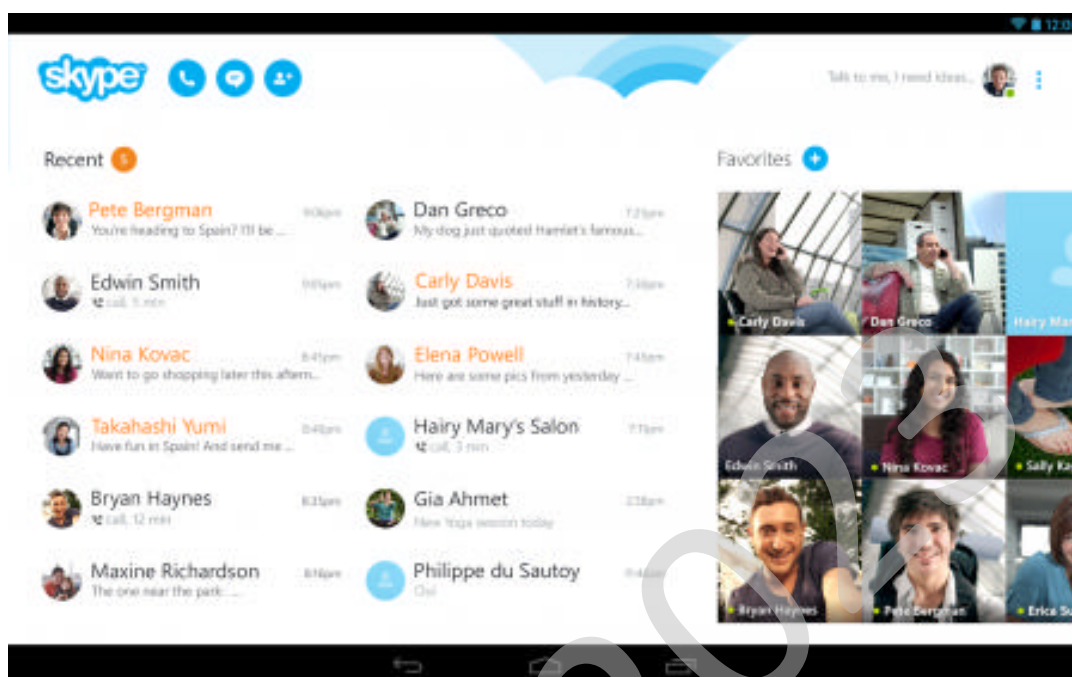


Рисунок 2.1 – Інтерфейс користувача Skype

Не дивно, що при такій розмаїтості функцій програма використовується не тільки на комп'ютерах і мобільних пристроях: її можна знайти на новітніх домашніх телефонах, телевізорах і ігрових приставках. Продаються навіть компактні пристрої, що представляють собою відеокамеру з убудованим Skype-клієнтом, які підключаються до будь-якого типу дисплея.

Viber

Платформи: Android, iOS, Windows Phone, BlackBerry, Windows, Bada, Symbian.

Мобільний додаток Viber всі частіше називають «убивцею Skype». Ця програма – месенджер нового покоління, у якому обліковий запис прив'язується до телефонного номера й інтегрується зі списком контактів, автоматично визначаючи інших користувачів із установленим Viber. Можна безкоштовно дзвонити їм через інтернет і обмінюватися безкоштовними повідомленнями, а за

Hangouts особливо зручний своєю інтеграцією з Android і адресною книгою Google. У додатку доступні всі контакти з телефонної книги смартфона.

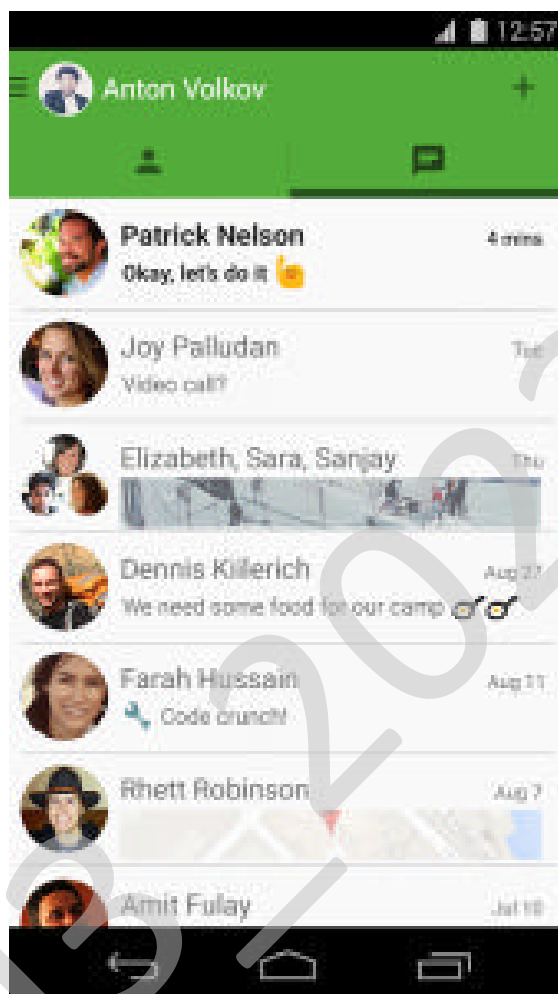


Рисунок 2.3 – Інтерфейс користувача Google Hangouts

Сервіс робить основний акцент на відеодзвінки й відеочати, що дозволяють поєднувати до дев'яти користувачів в одну відеоконференцію. Функція текстових чатів дозволяє продовжити спілкування з будь-якого місця, тому що історія повідомлень ведеться на всіх пристроях одночасно – будь те телефони, планшети або настільні комп'ютери. Якщо абонента немає в мережі, Hangouts пропонує подзвонити йому на мобільний або міський номер. Для цього треба тільки поповнити баланс, а додаток знайде потрібний номер у єдиному списку контактів.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

Facebook Messenger

Платформи: Android, iOS, Windows Phone, BlackBerry, Windows.

Facebook, користуючись статусом найбільшої соцмережі у світі, змусив своїх користувачів звернути увагу на новий мобільний додаток для спілкування через Інтернет – Facebook Messenger. Не звернути увагу було важко: можливість відправлення повідомлень у звичайному мобільному додатку Facebook просто відключили.

Незважаючи на такі методи «залучення користувачів», новий сервіс має важлива перевага – він включає всіх користувачів соціальної мережі. А оскільки до неї добровільно приєдналися вже більше мільярда чоловік, можна безкоштовно подзвонити кожному з них – у всякому разі, якщо він у цей момент перебуває в мережі.

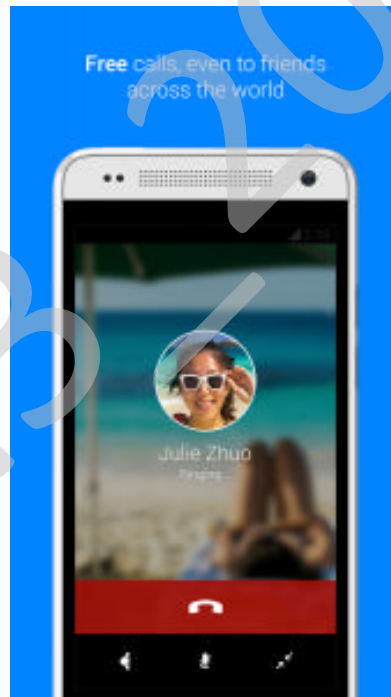


Рисунок 2.4 – Інтерфейс користувача Facebook Messenger

Додаток доступний для більшості платформ і має широкий функціонал для пересилання текстових повідомлень і фотографій, для групових чатів і багато чого іншого. Але при рекордному числі завантажувачів Facebook Messenger має

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

настільки ж рекордне число негативних відкликань. Людям не подобається й політика компанії, і зайвий збір інформації про клієнтів, і швидка витрата акумулятора.

Nimbuzz

Платформи: Android, iOS, Windows Phone, BlackBerry, Windows, Mac OS X.



Рисунок 2.5 – Інтерфейс користувача Nimbuzz

Крос-платформний сервіс Nimbuzz унікальний своєю універсальністю. Розроблювачі подбали про користувачів всіх основних платформ, написавши додатка для Android, iOS, BlackBerry, Windows Phone, Kindle, старих смартфонів

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

Nokia, а також Java-Версію й програми для настільних комп'ютерів. Якщо власник телефону не впевнений, на який ОС працює його апарат, то гід на вибір платформи попросить указати бренд і модель, після чого підкаже, де скачати потрібну версію.

Серед можливостей Nimbuzz – безкоштовні дзвінки усередині додатка; обмін повідомленнями; групові й тематичні чати, де можна зустріти нових друзів; стікери; обмін фотографіями, музикою й відеороликами. Щоб охопити максимальну кількість друзів, до Nimbuzz можна підключити аккаунти Facebook і Google+. Сервіс підтримує функцію NimbuzzOut, що дозволяє робити платні дзвінки на будь-які мобільні й міські телефони.

FaceTime

Платформи: iOS, Mac OS X.



Рисунок 2.6 – Інтерфейс користувача FaceTime

Система відеодзвінків FaceTime входить у базовий програмний пакет для всіх пристроїв компанії Apple – як мобільних, так і стаціонарних. Власники будь-якої техніки Apple прямо «з коробки» одержують можливість для

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

відеоспілкування зі своїми друзями й близькими – за умови, що ті теж купили «яблучну» продукцію.

Раніше в FaceTime не було функції повного відключення відеокамери, але така можливість з'явилася у версії iOS 7. Крім того, користувачі одержали можливість робити голосові дзвінки без використання відеопотоку, що знижує кількість даних, що пересилаються, і збільшує якість зв'язку при роботі в мобільних мережах.

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,

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

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.

– 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

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

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 звільнюються з допомогу вашого коду, який буде виконуватися у відповідний момент.

Це розширює потужність конструкцій records в Delphi, які використовуються щоб одержати більшу ефективність у порівнянні із класами.

Єдине керування пам'яттю

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

У порівнянні з Automatic Reference Counting (ARC), це дає кращу сумісність із існуючим кодом і спрощує написання компонентів, бібліотек і застосунків.

ARC модель керування пам'яттю model залишилася для керування рядками й посиланнями на тип інтерфейсу на всіх платформах. Для C++ це означає, що при створенні й звільненні Delphi-style класів в C++ використовується звичайне керування пам'яттю, як у будь-якого heap-allocated класу C++, що значно знижує складність коду.

Розширена підтримка бібліотек C++

В 10.4 ми портували багато популярних бібліотек C++ у C++Builder.

Забезпечивши оптимізовану підтримку бібліотек ZeroMQ, SDL2, SOCl, libSIMDpp і Nematode, поряд із уже підтримуваними Boost і Eigen, які можуть бути додані за допомогою менеджера пакетів Getit.

Win 64-відладник і збирач для C++

В 10.4 з'явився новий відладник C++ для Windows 64-bit. Відладник заснований на LLDB і показує значне збільшення стабільності при налагодженні 64-bit застосунків поряд з новими відладочними можливостями, такими як

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

гнучкості використання стилів, це дозволяє використовувати нестилізуємі компоненти із зовнішніх бібліотек в VCL застосунках, що використовують стиль.

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 стилізуємі 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.

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випуск кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

забезпечення, яке призначено для системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Основною метою концепції «розширених комунікацій» RCS (Rich Communication Services), що просувається Асоціацією GSMA, є вивід операторами мобільного зв'язку на ринок мультимедійних послуг на основі протоколу SIP. Міжоператорські послуги RCS розвиваються під брендом JOYN як альтернатива сервісам IP-комунікацій, які пропонують сторонні компанії (OTT) і «ведуть» істотну частину доходів від базових послуг (голос і передача повідомлень) у традиційних операторів. Так, по оцінці Informa Telecoms & Media, сукупне зниження доходів стільникових операторів від використання їхніми абонентами сторонніх IP-сервісів у Західній і Східній Європі, досягло в 2023 р. 3,9 і 1,6%, відповідно.

Концепція RCS з'явилася ще в 2008 р., перші пілотні проекти були випробувані в 2010 р., однак комерційний запуск послуг відбувся тільки до кінця 2014 р. До березня 2014 р. проект JOYN запустили 9 операторів в 5 країнах.

До операторів, які беруть участь у створенні RCS, відносяться AT&T, Bell Mobility, Bharti Airtel, Deutsche Telekom, KPN, KT Corporation, LG Uplus (LGU+), Orange, Orascom Telecom, Rogers Communications, SFR, SK Telecom, Telecom Italia, Telefonica, TeliaSonera, Telus, Verizon і Vodafone. Для того щоб почати надавати абонентам послугу RCS під брендом JOYN, операторам необхідно одержати акредитацію від GSMA.

Набір сервісів

Як планується, передплатникам RCS-послуг буде доступний наступний набір сервісів (Список потенційно можливих послуг, не є обов'язковим для всіх мереж):

– розширена телефонна книга з деталізованою інформацією про контакт (Rich Address Book);

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

– розширений функціонал сервісу передачі повідомлень (Rich Messaging) – містить у собі службу миттєвих повідомлень (IM, instant messaging), чат з декількома користувачами одночасно, історію переписки, можливість обміну документами, фото- і відеоконтент;

– розширений функціонал голосових викликів (Rich Calls), що дозволяє абонентам під час голосового дзвінка ділитися мультимедійним контентом (у тому числі доступна опція «see what I see» для обміну навколишнього відеоряду), робити відеодзвінки (у тому числі проводити відеоконференцзв'язок) та ін.;

– статус готовності абонента до комунікацій (Presence);

– SNS Services (Social Network Services) – агрегування інформації, яка надходить від соціальних мереж;

– можливість бачити властивості мережі співрозмовника (які саме опції можуть бути їм підтримані – наприклад, можливість передачі відео буде доступна тільки при роботі в мережі 3G або 4G).

Пристрою з підтримкою RCS

У якості передвстановленого додатку RCS є в телефонах HTC, Huawei, LG, Nokia, Samsung, Sony. Також про наміри почати виробництво мобільних пристроїв, що підтримують RCS, заявили компанії RIM (Blackberry), Motorola, Panasonic і ZTE. Крім того, завантажувемі додатка RCS підтримуються смартфонами iPhone (починаючи з 4-й версії) і планшетами iPad.

Додаток JOYN також може бути завантажено з Android Market або з Apple Store. Постачальниками додатків виступають 6 компаній: всі вони розробляють додатка RCS для ОС Android, і половина з них – також для iOS.

RCS і IMS

Асоціація GSMA активно працює з операторами зв'язку, роз'яснюючи переваги наявності IMS-мережі для розгортання послуг RCS і VoLTE. Останні можуть пропонуватися як окремо (RCS можуть бути реалізовані на будь-яких мережах поза залежністю від наявності/відсутності LTE-мережі), так і спільно. RCS є міжоператорською послугою: компанії-провайдери послуг забезпечують безшовне з'єднання в мережах різних операторів.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

Незважаючи на те, що специфікації RCS вимагають розгортання IMS-мережі (що є комплексним і витратним процесом), існують операторські рішення для забезпечення RCS-послуг поза залежністю від того, володіє оператор платформою IMS чи ні. Дані пропозиції з'явилися на ринку вже в 2010 р., у тому ж році була запущена перша пілотна RCS-послуга. Завдяки таким рішенням оператор має можливість швидко вивести послугу на ринок – приблизно протягом 10 тижнів після придбання відповідного встаткування.

Поширення RCS (JOYN) у світі

Послуги RCS запущені в 5 країнах, найбільше поширення вони одержали в Іспанії й Південній Кореї – у кожній із цих країн по трьох оператора забезпечують міжоператорський сервіс RCS. До середини лютого 2014 р. (50 днів в експлуатації) кількість передплатників JOYN.T південнокорейського оператора SK Telecom, за даними компанії, досягло 1 млн. Під час тестового періоду, до кінця травня 2014 р., користування послугами RCS було безкоштовним. Анонсуєма вартість послуг у довгостроковому плані: вартість передачі повідомлення – \$0,02, відеодзвінок – \$0,03 за хвилину. При цьому оператори відзначають, що споконвічно сервіс не буде приносити прибуток. Основним конкурентом RCS у Кореї є Kakao Talk – мобільний додаток, що використовують як мінімум 50 млн корейських абонентів.

Американський оператор MetroPCS крім стандартного пакета послуг пропонує голосові дзвінки поперх Wi-Fi (Voice over Wi-Fi) і відеодзвінки (Video calling). Для забезпечення росту абонентської бази оператор планує через спеціальний додаток забезпечити можливість запрошувати в мережу контакти з адресної книги абонента, які обслуговуються в мережах інших операторів Канади, Мексики й США.

До серпня 2014 р. JOYN був доступний безкоштовно для всіх абонентів Deutsche Telekom у Німеччині, що використовують передоплатну систему або контракт із фіксованими умовами.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

На момент написання звіту практично всі оператори, що запустили RCS, надавали сервіс у безкоштовному або умовно-безкоштовному режимі. Відповідь на питання про те, наскільки буде успішний проект JOYN з економічної точки зору, можна буде довідатися після появи перших підсумків його реалізації. На сучасний момент найбільших успіхів по наборі абонентів RCS досяг південнокорейський оператор SK Telecom (близько 1 млн).

Бізнес-модель і переваги RCS

Можна позначити кілька джерел генерування нових доходів операторів і скорочення витрат:

– Завдяки тому, що послуга RCS є міжоператорський, значно скорочуються витрати на маркетинг послуги (по суті, ним займається асоціація GSMA).

– Послуги RCS дозволяють еволюціонувати мережам оператора – для надання послуг RCS впровадження IMS-платформи не є обов'язковим, однак перехід до даних послуг може стати першим кроком у розвитку IMS-мережі.

– Можливість створення нових видів послуг на базі платформи RCS. Прикладами є додатки для спільного редагування зображень, конференц-дзвінків (до 10 користувачів одночасно) і т.п.

Очевидно, що далеко не всі користувачі готові платити за RCS-послугу. Якимось проміжним варіантом є модель Freemium, в основі якої – безкоштовна пропозиція самого сервісу, його опцій – за гроші; прикладом може бути безкоштовний продукт із відображенням рекламних оголошень.

Збільшення доходів операторів і/або зниження витрат у рамках проекту JOYN носять довгостроковий характер. У короткостроковому періоді послуга RCS, швидше за все, буде збитковою – позначається технологічна складність і тривалість впровадження IMS-мережі як ядра платформи для надання RCS-послуг, а також конкуренція з боку OTT-сервісів.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

Основними перевагами проекту JOYN у порівнянні з OTT-сервісами є:

1. Надання послуг у рамках єдиного стандарту й взаємопов'язаних мереж. RCS дозволяє працювати в різних мережах і на різних пристроях. Така інтегрованість OTT-сервісам недоступна: не можна, наприклад, відправити повідомлення з Skype на Windows Messenger.

2. Наявність унікального номера абонента (MSISDN). Унікальний номер абонента побудований по єдиному для операторів усього миру стандарту E.164. Відповідно, стандарт RCS-е дозволяє користувачеві створити персональне ком'юніті, використовуючи свій глобальний ідентифікаційний номер (global user ID), тобто номер телефону.

3. Труднощі зі створенням моделі монетизації OTT-сервісів. Незважаючи на те, що, по деяких прогнозах, до 2017 р. число мобільних користувачів OTT-послуг наблизиться до 1 млрд, такі сервіси як і раніше є слабо монетизованими. Багато передплатників OTT-послуг (Skype і т.п.) споконвічно не планують платити за них.

Основною можливістю в рамках моделі RCS є розвиток міжоператорського підходу – саме він дозволяє збільшувати чисельність користувачів послуги. Основними погрозами є незатребуваність RCS і відтік абонентів у сегмент безкоштовних OTT-послуг, і відсутність апробованої бізнес-моделі монетизації RCS, а також розвиток власних послуг під брендом оператора зв'язку (RCS-like).

Основними факторами, що гальмують впровадження RCS, є:

- технологічна складність рішення;
- недостатньо широкий ряд мобільних пристроїв з підтримкою RCS;
- залежність від дорогого ядра мережі IMS;
- споживчі переваги – далеко не кожний користувач безкоштовних OTT-послуг готовий платити за RCS.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		31

Основні тенденції й перспективи розвитку й поширення RCS

Можна виділити наступні фактори можливого успіху впровадження RCS, які допоможуть поширити цю послугу на масовий ринок і забезпечити висока якість обслуговування:

- тісна інтеграція послуг в адресній книзі й простота використання;
- міжоператорський підхід;
- стандартизація технічних умов, поділювана всіма гравцями галузі;
- зростаюча популярність відеодзвінків;
- зсув попиту у бік комунікацій у соціальних мережах;
- швидкість розгортання мереж IMS;
- поширення хмарних рішень, які дозволяють розвивати послуги RCS при відсутності IMS-мережі;
- розвиток ринку мобільних пристроїв, що підтримують мережу LTE.

Ринки, на яких послуга RCS уже представлена, характеризуються високою швидкістю впровадження LTE-мереж, а також гарною переговорною здатністю операторів – послуга з'являється відразу в декількох провайдерів, що дозволяє реалізовувати міжоператорський підхід. Відповідно до даної тенденції цілком імовірний послідовний запуск RCS операторами тих країн, де послуга тільки з'явилася (США, Німеччина) з розширенням її охопту на всі країни присутності її оператора, що запусить.

На ринках, що розвиваються, перспективи RCS менш очевидні. Основні стримуючі фактори – це низьке проникнення смартфонів і слабе покриття мереж LTE.

За прогнозами Infonetics, мобільні оператори по усьому світі за період 2024-2026 рр. сумарно одержать \$1,6 млрд за надання послуг RCS. В 2024 р. користувачами послуги стануть порядку 15 млн чоловік (Infonetics), до 2026 р., за прогнозами Juniper Research, цей показник збільшиться до 74 млн., тобто проникнення RCS складе менш 1% від абонентської бази мобільного зв'язку.

Основною проблемою при ухваленні рішення про запуск (для тих операторів, які сумніваються в затребуваності послуги) є відсутність даних про кількість користувачів і доходів.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32

3.2 Розробка структурної схеми

У даній роботі у якості RCS-послуг обрана IP-телефонія.

Опис алгоритму IP-телефонії

IP-телефонія – це технологія, що дозволяє використовувати Інтернет або будь-яку іншу IP-мережу для ведення телефонних розмов і передачі факсів у режимі реального часу. Особливо актуально, з економічної точки зору, використання даної технології для здійснення міжнародних і міжміських телефонних розмов або для створення розподілених телекомунікаційних телефонних мереж.

Устаткування для IP-телефонії

Для організації телефонного зв'язку по IP-мережах використовується спеціальне устаткування – шлюзи IP-телефонії. Загальний принцип дії телефонних шлюзів IP-телефонії такий: з однієї сторони шлюз підключається до телефонних ліній – і може з'єднатися з будь-яким телефоном миру. З іншої сторони шлюз підключений до IP-мережі – і може зв'язатися з будь-яким комп'ютером у світі. Шлюз приймає телефонний сигнал, оцифровує його (якщо він вихідно не цифровий), значно стискає, розбиває на пакети й відправляє через IP-мережу по призначенню з використанням протоколу IP. Для пакетів, що приходять із IP-мережі на шлюз і, що направляються в телефонну лінію, операція відбувається у зворотному порядку. Обидві складові процесу зв'язку (вхід сигналу в телефонну мережу і його вихід з телефонної мережі) відбуваються практично одночасно, що дозволяє забезпечити повнодуплексну розмову. На основі цих базових операцій можна побудувати багато різних конфігурацій.

У цей час, все більшу популярність здобувають IP-АТМ, які крім функцій шлюзу IP-телефонії виконують також традиційні функції звичайних офісних АТМ. Таким чином, при організації телефонного зв'язку через IP-мережі з використанням IP-АТМ можна цілком обійтися без офісної АТМ, тобто заощадити на додатковому встаткуванні.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		33

IP-телефонія опирається на дві основних операції: перетворення двонаправленої аналогової мови в цифрову форму усередині що кодує/декодувального пристрою (кодека) і впакування в пакети для передачі по IP-мережі. В IP-телефонії використовується особлива система передачі пакетів зі звуковою інформацією, що обумовлено специфікою передачі даних по IP-мережах.

У традиційних телефонних лініях між абонентами під час розмови створюється електричне коло, і цим забезпечується фіксована пропускна здатність для передачі сигналу. У той час як IP-мережа являє собою систему, що реалізує принцип комутації й маршрутизації пакетів, і не надає гарантованого шляху між точками зв'язку. Вся інформація, передана через IP (голос, текст, зображення, і т.п.) розділяється на пакети даних, що мають у своєму складі адреси точок призначення (прийому й передачі) і порядковий номер. Вузли IP направляють ці пакети по мережі до закінчення маршруту доставки.

Після прибуття пакетів до точки призначення, для відновлення вихідного обсягу впорядкованих даних використовуються порядкові номери пакетів. Для додатків, де не важливі порядок і інтервал приходу пакетів, таких як e-mail, час затримок між окремими пакетами не має вирішального значення. IP-телефонія є однією з областей передачі даних, де важлива динаміка передачі сигналу, що забезпечується сучасними методами кодування й передачі інформації. Для забезпечення стабільного телефонного зв'язку по IP-мережах введені спеціальні протоколи передачі даних, наприклад, H.323 і SIP.

Основні протоколи IP-телефонії

H.323 – основний стандарт, прийнятий ІТУ-Т, де описується, яким образом чутливий до затримки трафік, зокрема голос і відео, одержує пріоритет у локальних і глобальних мережах. Він складається з ряду рекомендацій (стека протоколів) по суміжних технічних питаннях, таким, як якість мови, стандарти кодування звуковий і відеоінформації та ін. Протокол SIP (Session Initiation Protocol) прийнятий у березні 2000 року організацією ІETF як стандарт RFC 2543. SIP більшою мірою відповідає ідеології TCP/IP, ніж стек протоколів H.323. Про

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

підтримку цього протоколу заявили такі виробники як 3Com, Cisco, Ericsson, Siemens і ін. Однозначність стандарту SIP дозволяє із упевненістю говорити про сумісність IP-шлюзів різних виробників.

При передачі в режимі реального часу до 30% пакетів можуть бути загублені або отримані із запізненням (що в режимі реального часу те саме). Гарний додаток IP-телефонії повинен відшкодувати недостачу пакетів, відновивши загублені дані. Сам алгоритм кодування мови також впливає на відновлення даних.

На рисунку 3.1 зображена структурна схема розробленого програмного забезпечення розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Розглянемо основні структурні елементи, які застосовані у розробленій схемі розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях:

- GateKeeper – сервер, що перетворює імена користувачів у динамічні адреси IP.

- ТфОП – телефонний оператор, який надає послуги традиційного телефонного зв'язку від шлюзу до телефонної станції.

- IP-phone – спеціальний телефон, який можливо під'єднати до IP-мережі напряму. Повнофункціональний IP-телефон дає можливість користувачеві ініціювати й приймати дзвінки, взаємодіючи з телекомунікаційною VoIP-системою або інфраструктурою постачальника послуг IP-телефонії

- Soft phone – програмне забезпечення, яке встановлене на ЕОМ та дозволяє спілкуватися використовуючи мікрофон та WEB-камеру, під'єднані до ЕОМ.

- Analog phone – звичайний телефон.

- Gateway – комп'ютери з'єднання, “перекладачі”. Вони використовуються для того, щоб утворити з'єднання зі звичайною телефонною мережею. Ці Gateways з'єднані як з комп'ютерною мережею, так і з нормальною телефонною мережею й передають запити в обоє напрямків. При цьому IP-пакети з однієї

цьому вибір конкретної моделі шлюзу залежить від типу й кількості інтерфейсів, застосовуваних для стикування із ТфОП (можливе використання як аналогових, так і цифрових інтерфейсів). Шлюзи також потрібні якщо буде потреба підключення системи IP-телефонії до встановленого раніше офісної АТМ.

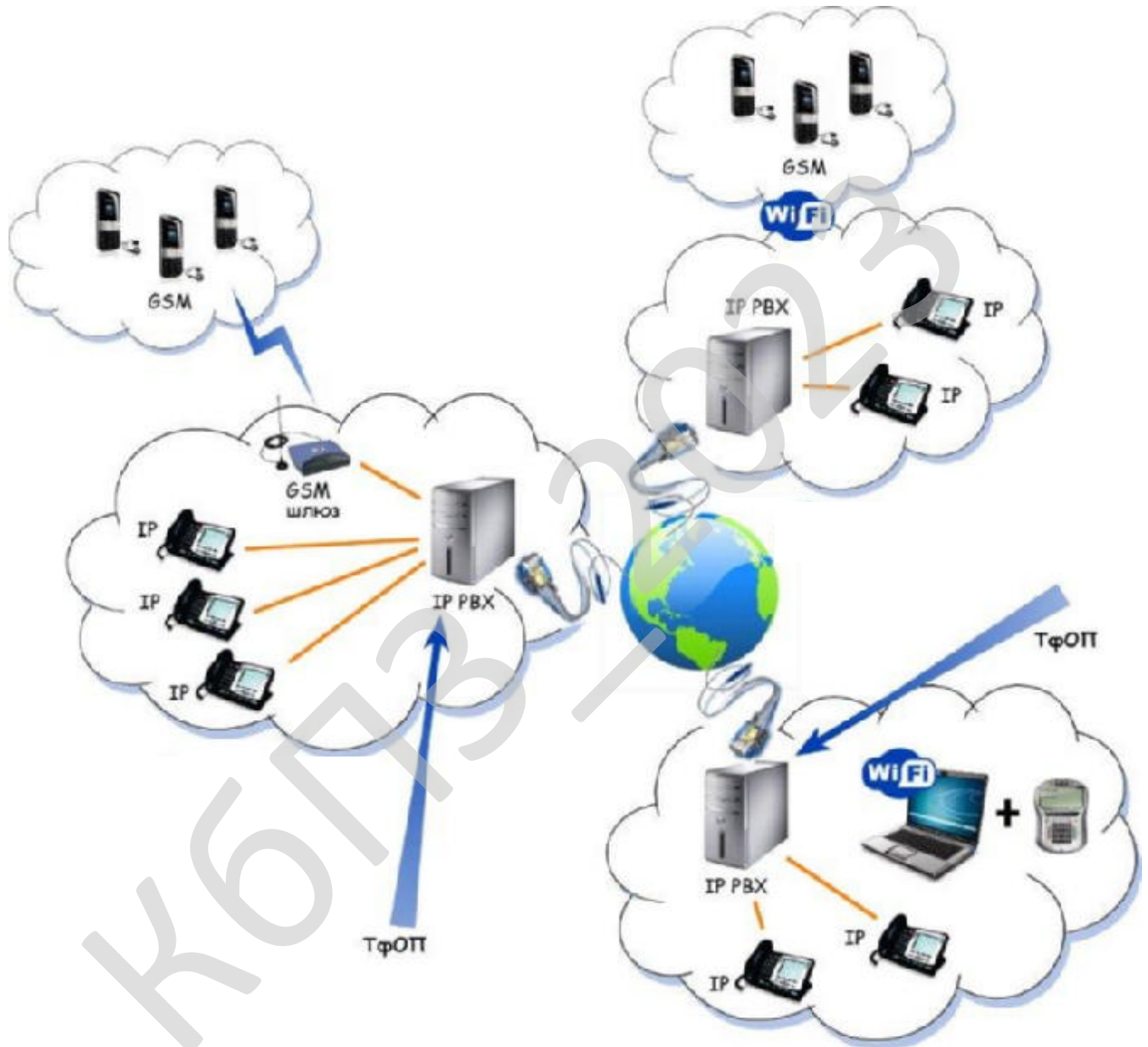


Рисунок 3.1 – Структурна схема розробленої системи

Основні характеристики моделі побудови мережі IP-телефонії для одного будинку або кампуса (декількох будинків, об'єднаних високошвидкісною локальною мережею):

– для організації системи IP-телефонії може використовуватися D-Link DVX-7090, що виконує функції голосового маршрутизатора й рекомендується для застосування в мережах середніх і великих офісів;

– для подальшого масштабування мережі можливе використання декількох подібних пристроїв;

– для підключення до телефонної мережі загального користування (ТфОП) аналогових телефонів і факсових апаратів, стикування з існуючими УАТМ застосовуються голосові шлюзи;

– ресурси голосових сервісних модулів використовуються для організації аудіоконференцій;

– для забезпечення якісної роботи різних додатків рекомендується застосування комутаторів, що підтримують необхідні засоби забезпечення якості сервісу (QoS).

3.3 Розробка функціональної схеми

Н.323

Н.323, містить описи термінальних пристроїв, устаткування й мережних служб, призначених для здійснення мультимедійного зв'язку в мережах з комутацією пакетів (наприклад, Intranet або Інтернет). Термінальні пристрої й мережне встаткування стандарту Н.323 можуть передавати дані, мову й відеоінформацію в масштабі реального часу. У рекомендації Н.323 не визначені: мережний інтерфейс, фізичне середовище передачі інформації й транспортний протокол, використовуваний у мережі. Мережа, через яку здійснюється зв'язок між терміналами Н.323, може являти собою сегмент або безліч сегментів зі складною топологією. Термінали Н.323 можуть бути інтегровані в персональні

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		38

комп'ютери або реалізовані як автономні пристрої. Але підтримка мовного обміну – обов'язкова функція для будь-якого пристрою стандарту H.323.

Рекомендації H.323 передбачають:

– Керування смугою пропускання. Передача аудіо- і відеоінформації досить інтенсивно навантажує канали зв'язку, і, якщо не стежити за ростом цього навантаження, працездатність критично важливих мережних сервісів може бути порушена. Тому рекомендації H.323 передбачають керування смугою пропускання. Можна обмежити як число одночасних з'єднань, так і сумарну смугу пропускання для всіх додатків H.323. Ці обмеження допомагають зберегти необхідні ресурси для роботи інших мережних додатків. Кожний термінал H.323 може управляти своєю смугою пропускання в конкретній сесії конференції.

– Міжмережні конференції. Рекомендації H.323 пропонують засоби з'єднання учасників відеоконференції в різнорідних мережах (наприклад, IP і ISDN, IP і PSTN).

– Платформна незалежність. H.323 "не прив'язаний" до яких-небудь технологічних рішень, пов'язаних з устаткуванням або програмним забезпеченням. Взаємодіючі між собою додатки можуть створюватися на основі різних платформ, з різними операційними системами.

– Підтримка багатоточкових конференцій. Рекомендації H.323 дозволяють організовувати конференцію із трьома або більше учасниками. Багатоточкові конференції можуть проводитися як з використанням центрального контролера – MCU (пристрою багатоточкової конференції), так і без нього.

– Підтримка багатоадресної передачі. H.323 підтримує багатоадресну передачу в багатоточковій конференції, якщо мережа підтримує протокол керування груповою адресацією. При багатоадресній передачі один пакет інформації відправляється всім необхідним адресатам без зайвого дублювання. багатоадресна передача використовує смугу пропускання набагато більш ефективно, оскільки всім адресатам – учасникам списку розсилання відправляється рівно один потік.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		39

– Стандарти для кодеків. H.323 установлює стандарти для кодування й декодування аудіо- і відеопотоків з метою забезпечення сумісності устаткування різних виробників. Разом з тим стандарт досить гнучкий. Сформульовано вимоги, виконання яких обов'язково, і існують опціональні можливості, у випадку використання яких також необхідно строго дотримуватися стандарту. Крім цього, виробник може включати в мультимедійні продукти й додатки додаткові можливості, якщо вони не суперечать обов'язковим і опціональним вимогам стандарту.

– Сумісність. Можливі випадки, коли учасники конференції хочуть спілкуватися один з одним, не піклуючись про питання сумісності між собою. Рекомендації H.323 підтримують з'ясування загальних можливостей устаткування кінцевих користувачів і встановлюють найкращі із загальних для учасників конференції протоколів кодування, виклику й керування.

– Гнучкість. H.323 конференція може включати учасників, кінцеве устаткування яких має різні можливості. Наприклад, один з учасників може використовувати термінал тільки з аудіо можливостями, у той час як інші учасники конференції можуть мати можливості передачі/прийому також відео й даних.

Архітектура стандарту H.323

У рекомендації H.323 установлюється чотири основних компоненти VoIP-з'єднання.

- термінал;
- контролер зони;
- шлюз (gateway);
- пристрій керування багатоточковою конференцією (MCU).

Термінал (Terminal) – окінцевий мультимедійне (голос, відео, дані) пристрій, призначений для участі в конференції. Під терміналом стандарт розуміє устаткування кінцевих точок мережі, що дозволяє користувачам спілкуватися один з одним у реальному часі. H.323-термінал повинен забезпечувати підтримку наступних протоколів:

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		40

1. H.245 для встановлення можливостей терміналів і створення каналу обміну аудіоінформацією.
2. H.225 для сигналізації виклику й установки параметрів зв'язку.
3. RAS для реєстрації терміналу користувача й установки додаткових параметрів керування контролером зони.
4. RTP/RTCP для упорядкування звукових і відеопакетів.

H.323-термінал повинен також підтримувати звуковий кодер-декодер відповідно до G.711.

Протоколи H.225 і RAS використовуються між H.323-окінцевими точками (терміналами й шлюзами) і контролером зони для забезпечення:

- виявлення контролера зони (GRQ);
- реєстрації окінцевої точки;
- визначення розташування окінцевої точки;
- керування автентифікацією;
- завдання маркера доступу.

RAS-повідомлення передаються через ненадійні RAS-канали, тому при обміні повідомленнями можливі втрати, затримки й повторні передачі.

Принципи побудови протоколу SIP

Протокол ініціювання сеансів (Session Initiation Protocol – SIP) є протоколом прикладного рівня й призначається для організації, модифікації й завершення сеансів зв'язку (наприклад, мультимедійних конференцій, телефонних з'єднань). Користувачі можуть брати участь в існуючих сеансах зв'язку, запрошувати інших користувачів і бути запрошеними ними до нового сеансу зв'язку.

Протокол SIP розроблений групою MMUSIC комітету IETF, а специфікації протоколу представлені в документі RFC 2543. В основу протоколу закладені наступні принципи:

1. Персональна мобільність користувачів. Користувачі можуть переміщатися без обмежень у межах мережі. Користувачеві привласнюється

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

унікальний ідентифікатор, а мережа надає йому послуги зв'язку поза залежністю від того, де він перебуває.

2. Масштабованість мережі. Вона характеризується, у першу чергу, можливістю збільшення кількості елементів мережі при її розширенні. Серверна структура мережі, побудована на базі протоколу SIP, відповідає цій вимозі.

3. Розширюваність протоколу. Вона характеризується можливістю доповнення протоколу новими функціями при введенні нових послуг і його адаптації до роботи з різними додатками.

Розширення функцій протоколу SIP може бути зроблене за рахунок введення нових заголовків повідомлень, які повинні бути зареєстровані в організації IANA. При цьому якщо SIP-сервер приймає повідомлення з невідомими йому атрибутами, то він просто ігнорує їх.

Для розширення можливостей протоколу SIP можуть бути також додані й нові типи повідомлень.

4. Інтеграція в стек існуючих протоколів Інтернету, розроблених IETF. Протокол SIP є частиною глобальної архітектури мультимедіа, розробленої IETF. Ця архітектура містить у собі також і інші протоколи: резервування ресурсів (Resource Reservation Protocol – RSVP, RFC 2205), транспортний протокол реального часу (Real-Time Transport Protocol – RTP, RFC 1889), протокол передачі потокової інформації в реальному часі (Real-Time Streaming Protocol – RTSP, RFC 2326), протокол опису параметрів зв'язку (SDP, RFC 2327). Однак функції самого протоколу SIP не залежать від жодного із цих протоколів.

5. Взаємодія з іншими протоколами сигналізації. Протокол SIP може бути використаний разом із протоколом H.323.

Протокол керування шлюзами MGCP

Протокол запропонований робочою групою MEGACO (Media Gateway Control Protocol) комітету IETF.

Основна ідея MGCP дуже проста. Вона полягає в тому, що керування сигналізацією (Call Control) зосереджено на центральному керуючому пристрої,

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

називаному контролером сигналізацій (Call Agent, CA), і повністю відділено від медіа-потоків. Ці потоки обробляються шлюзами або абонентськими терміналами, які здатні виконувати лише обмежений набір команд, що виходять від керуючого пристрою. В архітектурі протоколу MGCP-мережі можна виділити два основних функціональних компоненти. Перший може бути представлений транспортним шлюзом (Media Gateway, MG) або IP-телефоном, а другий – пристроєм керування викликами, що може називатися контролером сигналізацій (CA), контролером шлюзу (Media Gateway Controller, MGC) або програмним контролером (Softswitch, SS). Іноді контролер сигналізацій представляють у вигляді двох компонентів – власно контролера (Call Agent), що виконує функції керування шлюзами, і шлюзу сигналізації (Signaling Gateway), що забезпечує обмін сигнальною інформацією й узгодження між традиційною телефонною мережею й мережею IP.

При розробці протоколу керування шлюзами робоча група MEGACO опиралася на принцип декомпозиції, відповідно до якого шлюз розбивається на окремі функціональні блоки:

– транспортний шлюз – Media Gateway, що виконує функції перетворення мовної інформації, що надходить із боку ТфОП з постійною швидкістю, у вид, придатний для передачі по мережах з маршрутизацією пакетів IP: кодування й упакування мовної інформації в пакети RTP/UDP/IP, а також зворотне перетворення;

– пристрій керування викликами – Call Agent, що виконує функції керування шлюзом;

– шлюз сигналізації – Signaling Gateway, що забезпечує доставку сигнальної інформації, що надходить із боку ТфОП, до пристрою керування шлюзом і перенос сигнальної інформації у зворотному напрямку.

Контролери обмінюються зі шлюзами (або IP-телефонами) аудіоданими в простому текстовому форматі, а функціональне призначення кожного шлюзу визначається набором команд, які він "розуміє". Маніпулюючи наборами команд, можна одержувати спеціалізовані шлюзи: транкові (Trunking gateways, TGW),

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

абонентські (Residential gateways, RGW), шлюзи доступу (Access gateways, AGW) і ін.

Контролер сигналізації СА сприймає мережа як набір двох логічних елементів – пристроїв (endpoints) і з'єднань (connections) між ними. Пристрої можуть бути фізичними (наприклад, IP-телефони або лінії на шлюзах) або віртуальними (наприклад, лінії до серверів голосових повідомлень). З'єднання можуть бути орієнтовані на передачу голосу, факс-повідомлень або даних. Керування цими елементами, тобто організація з'єднань між пристроями, відбувається шляхом посилки команд у вигляді текстових (ASCII) повідомлень за протоколом UDP – при цьому може використовуватися протокол SDP.

Найпростіший сценарій з'єднання буде виглядати в такий спосіб: користувач телефону, підключеного до MGCP-шлюзу, знімає трубку, після чого шлюз повідомляє контролер про цю подію, а СА дає команду шлюзу включити в телефонну лінію сигнал готовності (dial-ton). Тепер користувач чує в трубці безперервний гудок. Далі треба набір телефонного номера – теж послідовність подій для контролера. Аналізуючи ці події, СА може встановити з'єднання з іншим абонентом в IP-мережі або в телефонній мережі.

Класифікація шлюзів по області застосування

Застосовується наступна класифікація транспортних шлюзів (Media Gateways):

– Trunking Gateway – шлюз між ТфОП і мережею з маршрутизацією пакетів IP, орієнтований на підключення до телефонної мережі.

– Voice over ATM Gateway – шлюз між ТфОП і ATM-мережею, що також підключається до телефонної мережі за допомогою великої кількості цифрових трактів.

– Residential Gateway – шлюз, що підключає до IP-мережі аналогові, кабельні модеми, лінії xDSL і широкополосні пристрої бездротового доступу.

– Access Gateway – шлюз для підключення до мережі IP-телефонії невеликої ATM для установ через аналоговий або цифровий інтерфейс.

						ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			44

- Business Gateway – шлюз із цифровим інтерфейсом для підключення до мережі з маршрутизацією IP-пакетів АТМ установи.
- Network Access Server – сервер доступу до IP-мережі для передачі даних.
- Circuit switch або packet switch – комутаційні пристрої з інтерфейсом для керування від зовнішнього пристрою.

Основні алгоритми стиску звуку, використовувані в IP-телефонії

Для кодування звукової інформації звичайно використовуються наступні кодеки: G.711, G.722, GSM0610, G.723, G.723.1, G.728, і G.729. Для кодека G.711 потрібна ширина смуги частот в 64 Кбіт/с, тому він прийнятний не у всіх IP-мережах (наприклад, в Інтернет), тому що більшість користувачів Інтернету має канал свідомо меншої ширини. Кодеки з низькою шириною смуги частот – G.729 в 8 Кбіт/с і G.723.1 в 5.3/6.3 Кбіт/с – цілком підходять для використання в Інтернет. Зокрема, G.723.1 є одним з декількох "стандартних" кодеків для IP-телефонії, особливо після того, як Intel, Microsoft і Netscape оголосили про підтримку цього стандарту звукового кодування.

Взаємодія протоколів VoIP

При використанні протоколів, які безпосередньо мають справа з VoIP, важливо правильне розуміння специфікації, внесеної цими протоколами. Нижче наведений стек протоколів VoIP. Тут відсутній верхній рівень, що має на увазі в собі будь-яку розмовну мову. Даний рисунок характеризує винятково передачу голосових даних.

Стек протоколів VoIP включає в себе (по рівням OSI):

6. Рівень подань G.729/G.711.
5. Рівень сеансу H.323, шлюз SIP/SDP.
4. Транспортний рівень, протоколи RTP/UDP/RSVP.
3. Мережний рівень IP/LLQ.
2. Канальний рівень MLPPP/FR/ATM AAL5.
1. Фізичний рівень.

Технологія VoIP може працювати в будь-якому фізичному середовищі, що може використовуватися звичайним протоколом IP. Такі середовища можуть бути представлені у вигляді кабелю крученої пари (використовуваної в традиційному Ethernet), телефонних проводів, бездротових з'єднань (протокол IEEE 802.11) і ін.

Другий рівень цієї моделі – канальний рівень – указує, що протокол IP для створення фреймів може використовувати різні формати. Як показано на рисунку 3.1, він включає багатоканальний PPP (Multilink PPP), Frame Relay (FR) і ATM. При проектуванні мережі можливі й інші варіанти, оскільки передавати голос можуть також Ethernet, Wi-Fi і інші технології локальних мереж.

На третьому, мережному рівні використовується протокол IP як спосіб передачі голосу, однак звичайний IP повинен бути доповнений спеціальними засобами. Оскільки існують проблеми із затримкою, протоколу IP потрібно використовувати який-небудь спосіб установа черговості для того, щоб голосовим даним не довелося очікувати передачі в умовах конкуренції зі звичайними даними. На маршрутизаторах повинна бути використана черговість із малою затримкою (Low-Latency queuing – LLQ) або яка-небудь інша сучасна схема установки черговості, щоб голосові дані відправлялися раніше звичайних даних. Крім того, повинні використовуватися схеми маркування (marking) із завданням пріоритетів (coloring), називані IP-пріоритетами, для забезпечення того, щоб голосові дані розглядалися системою як більше важливі для першочергової передачі, чим звичайні дані.

Наступним рівнем є транспортний. Оскільки для передачі голосу використовується протокол UDP, системі не вистачає механізму установки черговості пакетів, щоб пакети доставлялися в необхідній послідовності. Транспортний протокол реального часу (Real-Time Transport Protocol – RTP) для виконання цієї вимоги додає номер пакета в послідовності передачі й механізм розміщення тимчасових міток. Також може використовуватися протокол резервування (Resource Reservation Protocol – RSVP) для резервування смуги пропускання уздовж шляхи проходження голосу по IP-мережі. Даний протокол

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

виключає застосування пакетами звичайних даних зарезервованої смуги пропусцення.

П'ятий рівень моделі – сеансовий. На сьогоднішній день мережі VoIP переходять зі стандарту ITU-T H.323 на інший протокол ініціювання сеансу (Session Initiation Protocol – SIP) і протокол опису сеансу (Session Description Protocol – SDP).

Шостим рівнем моделі є рівень подань. Як визначено в моделі OSI, рівень подань аналізує й інтерпретує формати даних. У термінах передачі голосу рівень подань забезпечує методи кодування й стиску, використовувані для передачі голоси.

Всі рівні стека протоколів спільно застосовуються для того, щоб вирішити проблеми мінімізації затримки й забезпечити необхідний порядок проходження пакетів.

Види з'єднань, взаємодія з комп'ютерною мережею

Можна виділити три найбільше часто використовуваних сценарії IP-телефонії:

- комп'ютер-комп'ютер;
- телефон-комп'ютер;
- телефон-телефон.

Перші сценарій " комп'ютер-комп'ютер" реалізується на базі стандартних комп'ютерів, оснащених засобами мультимедіа й підключених до мережі Інтернет.

Компоненти сценарію " комп'ютер-комп'ютер"

У цьому сценарії аналогові мовні сигнали від мікрофона абонента А перетворюються в цифрову форму за допомогою аналого-цифрового перетворювача (АЦП). Відрахунки мовних даних у цифровій формі потім стискаються пристроєм, що кодує, для скорочення потрібної для їхньої передачі смуги у відношенні 4:1, 8:1 або 10:1. Вихідні дані після стиску формуються в пакети, до яких додаються заголовки протоколів, і потім пакети передаються через IP-мережу в систему IP-телефонії, що обслуговує абонента Б. Коли пакети

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		47

приймаються системою абонента Б, заголовки протоколу віддаляються, а стислі мовні дані надходять у пристрій, що розгортає їх у первісну форму, після чого мовні дані знову перетворюються в аналогову форму за допомогою цифро-аналогового перетворювача (ЦАП) і попадають у динамік телефону абонента Б. Для звичайного з'єднання між двома абонентами системи IP-телефонії на кожному кінці одночасно реалізують як функції передачі, так і функції прийому. Під IP-мережею, мається на увазі або глобальній мережі Інтернет, або телекомунікаційна мережа підприємства Intranet.

Для підтримки сценарію "комп'ютер-комп'ютер" постачальникові послуг Інтернет необхідно мати окремий сервер (GateKeeper), що перетворить імена користувачів у динамічні адреси IP. Сам сценарій орієнтований на користувача, якому мережа потрібна в основному для передачі даних, а програмне забезпечення IP-телефонії потрібно лише іноді для розмов з колегами. Ефективне використання телефонного зв'язку по сценарію "комп'ютер-комп'ютер" звичайно пов'язане з підвищенням продуктивності роботи великих компаній, наприклад, при організації віртуальної презентації в телекомунікаційній мережі з можливістю не тільки бачити документи на веб-сервері, але й обговорювати їхній зміст за допомогою IP-телефону.

Для проведення телефонних розмов один з одним абоненти А и Б повинні мати доступ до Інтернету або до іншої мережі із протоколом IP. Розберемо можливий алгоритм організації зв'язку між цими абонентами на прикладі протоколу H.323.

1. Абонент А запускає свій додаток IP-телефонії, що підтримує протокол H.323.
2. Абонент Б також задалегідь запустив свій додаток IP-телефонії, що підтримує протокол H.323.
3. Абонент А знає доменне ім'я абонента Б – Domain Name System (DNS), вводить це ім'я в розділ "кому подзвонити" у своєму додатку IP-телефонії й натискає кнопку Return.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

4. Додаток IP-телефонії звертається до DNS-сервера (який у даному прикладі реалізований безпосередньо в персональному комп'ютері абонента А) для того, щоб перетворити доменне ім'я абонента Б у IP-Адреса.

5. Сервер DNS повертає IP-адресу абонента Б.

6. Додаток IP-телефонії абонента А одержує IP-Адресу абонента Б і відправляє по цій адресі сигнальне повідомлення H.225 Setup.

7. При одержанні повідомлення H.225 Setup додаток Б сигналізує абонентові Б о вхідному виклику.

8. Абонент Б приймає виклик і додаток IP-телефонії відправляє відповідне повідомлення H.225 Connect.

9. Додаток IP-телефонії в абонента А починає взаємодія з додатком в абонента Б у відповідності з рекомендацією H.245.

10. Після закінчення взаємодії за протоколом H.245 і відкриття логічних каналів абоненти А и Б можуть розмовляти один з одним через IP-мережу.

При цьому блок "Керування й сигналізація" управляє пакетизацією і депакетизацією переданих фрагментів, а також здійснює контроль при їхній передачі.

У цьому прикладі не показані деякі службові деталі, які необхідні постачальникові послуг для розгортання мережі IP-телефонії.

Сценарій "телефон-комп'ютер"

За назвою "комп'ютер" у всіх сценаріях ми будемо розуміти термінал користувача, включений в IP-мережу, а за назвою "телефон" – термінал користувача, включений у мережу комутації каналів будь-якого типу: ТфОП, ISDN або GSM.

Наступний сценарій "телефон-комп'ютер" знаходить застосування в різного роду довідково-інформаційних службах Інтернету, у службах збуту товарів або в службах технічної підтримки. Користувач, що підключився до серверу WWW якої-небудь компанії, має можливість звернутися до оператора довідкової служби. Це цілком відповідає стилю життя сучасних споживачів, пов'язаному з потребою в додаткових зручностях і економії часу.

У другому сценарії "телефон-комп'ютер" з'єднання встановлюється між користувачем ТфОП і користувачем ІР-мережі. Передбачається, що встановлення з'єднання ініціює користувач мережі комутації каналів.

Шлюз для взаємодії мереж ТфОП і ІР може бути реалізований як окремим пристроєм, так і інтегрованим в існуюче встаткування ТфОП або ІР-мережі.

Можливий й інший різновид другого сценарію, коли з'єднання встановлюється між користувачем ІР-мережі й абонентом ТфОП, але ініціює його створення абонент ТфОП.

Розглянемо трохи докладніше архітектуру системи ІР-телефонії по сценарії "телефон-комп'ютер". При спробі викликати довідково-інформаційну службу, використовуючи послуги пакетної телефонії й звичайний телефон, на початковій фазі абонент А викликає прилеглий шлюз ІР-телефонії для мінімізації витрат на послуги зв'язку. Від шлюзу до абонента А надходить запит вводу номеру, до якого повинен бути спрямований виклик (наприклад, номер служби), і особистий ідентифікаційний номер (PIN) для автентифікації й наступного нарахування плати, якщо ця служба платна. Грунтуючись на викликуваному номері, шлюз визначає найбільш доступний шлях до даної служби. Крім того, шлюз активізує свої функції. Роз'єднання з будь-якої сторони передається протилежній стороні по протоколу сигналізації й викликає завершення встановлених з'єднань і звільнення ресурсів шлюзу для обслуговування наступного виклику.

Ефективність об'єднання послуг передачі мови й даних є основним стимулом використання ІР-телефонії по сценаріях "комп'ютер-комп'ютер" і "телефон-комп'ютер", не наносячи при цьому збитку інтересам операторів традиційних телефонних мереж.

Сценарій "телефон-телефон"

Третій сценарій "телефон-телефон" у значній мірі відрізняється від перших двох сценаріїв ІР-телефонії своєю соціальною значимістю, оскільки метою його застосування є надання звичайним абонентам ТфОП альтернативної можливості міжміського й міжнародного телефонного зв'язку.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Як правило, обслуговування викликів по такому сценарії IP-телефонії виглядає в такий спосіб. Постачальник послуг IP-телефонії підключає свій шлюз до комутаційного вузла або станції ТфОП по мережі Інтернет або по виділеному каналі до аналогічного шлюзу, що перебуває в іншому місті або іншій країні.

Типова послуга IP-телефонії по сценарії "телефон-телефон" використовує стандартний IP-телефон, а замість міжміського компонента ТфОП задіє або частну IP-мережу, або мережу Інтернет. Завдяки маршрутизації телефонного трафіка по IP-мережі стало можливим обходити мережі загального користування й, відповідно, не платити за міжміський/міжнародний зв'язок операторам цих мереж.

Постачальники послуг IP-телефонії надають послуги "телефон-телефон" шляхом установки шлюзів IP-телефонії на вході й виході IP-мереж. Абоненти підключаються до шлюзу постачальника послуг IP-телефонії через ТфОП, набираючи спеціальний номер доступу. Абонент одержує доступ до шлюзу, використовуючи персональний ідентифікаційний номер (PIN) або послугу ідентифікації номера зухвалого абонента (Calling Line Identification). Після цього шлюз просить ввести телефонний номер викликуваного абонента, аналізує цей номер і визначає, який шлюз має кращий доступ до потрібного телефону. Як тільки між вхідним і вихідним шлюзами встановлюється контакт, подальше встановлення з'єднання до викликуваного абонента виконується вихідним шлюзом через його місцеву телефонну мережу. Повна вартість такого зв'язку буде складатися для користувача з розцінок ТфОП на зв'язок із вхідним шлюзом, розцінок інтернет-провайдеру на транспортування даних і розцінок віддаленої ТфОП на зв'язок вихідного шлюзу з викликаним абонентом.

Одним з алгоритмів організації зв'язку по сценарії "телефон-телефон" є випуск постачальником послуги своїх телефонних карт. Маючи таку карту, користувач, що бажає подзвонити в інше місто, набирає номер постачальника даної послуги, потім у режимі донабору вводить свій ідентифікаційний номер і

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

PIN-код, зазначений на карті. Після процедури автентифікації він набирає телефонний номер адресата.

На рисунку 3.2 зображена функціональна схема розробленого програмного забезпечення.

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

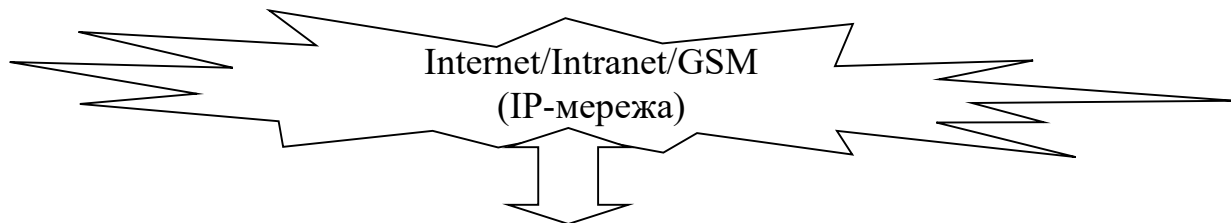
- IP-телефонія.
- Селективний зв'язок.
- Відеоконференція.
- Формування вхідної інформації.
- База даних абонентів телекомунікаційної мережі.
- Білінг.
- Виведення статусу ПЗ.
- Довідник.

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

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

Головне вікно програми призначений для запуску основних процедур програми і завершення роботи з програмою.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52



Н.323
Керування з'єднанням і сигналізація: H.225.0, H.225.0/RAS, H.245
Обробка звукових сигналів: G.711, G.722, G.723.1, G.728, G.729
Обробка відеосигналів: H.261, H.263
Конференц-зв'язок для передачі даних: T.120, T.123, T.124, T.125
Мультимедійна передача: RTP, RTCP
Забезпечення безпеки: H.235
Додаткові послуги: H.450

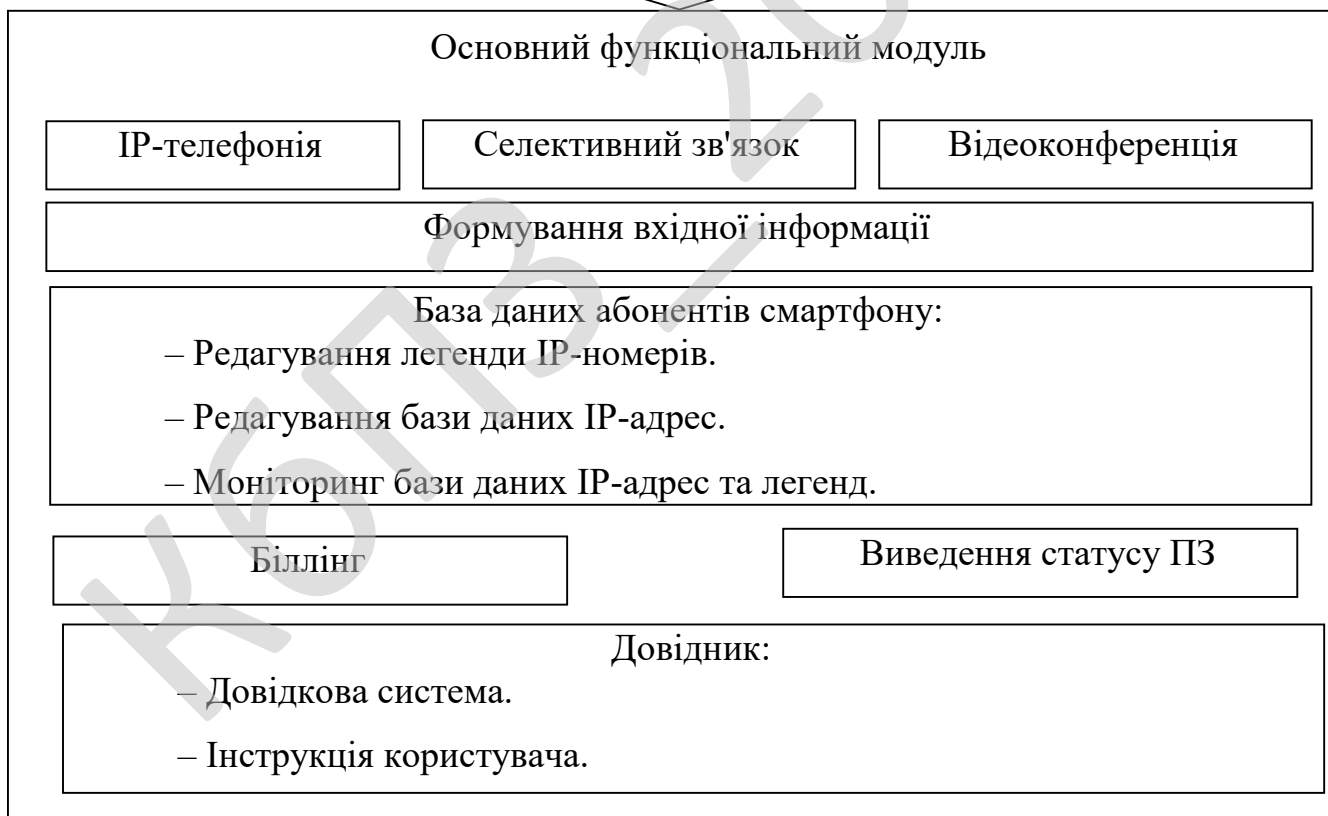
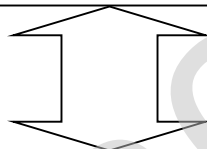


Рисунок 3.2 – Функціональна схема розробленої системи

Модуль роботи з довідниками містить у собі два довідники:

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

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

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

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

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

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

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.
- 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.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

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

Стек протоколів H.323

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

Стек H.323 складають 7 груп протоколів:

- керування й сигналізація;
- обробка звукових сигналів;
- обробка відеосигналів;
- конференц-зв'язок;
- передача мультимедійної інформації;
- забезпечення інформаційної безпеки;
- додаткові послуги;

1. Керування з'єднанням і сигналізація:

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

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

– H.245: протокол керування для мультимедіа.

2. Обробка звукових сигналів:

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

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

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

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

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

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

3. Обробка відеосигналів:

- Н.261: відеокодеки для аудіовізуальних послуг зі швидкістю 64 кбіт/с.
- Н.263: кодування відеосигналу для передачі з малою швидкістю.

4. Конференц-зв'язок для передачі даних:

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

5. Мультимедійна передача:

- RTP: транспортний протокол реального часу.
- RTCP: протокол керування передачею в реальному часі.

6. Забезпечення безпеки:

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

7. Додаткові послуги:

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

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

– Н.450.3: переадресація виклику.

– Н.450.4: утримання виклику.

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

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

– Н.450.7: індикація повідомлення, що очікує.

– Н.450.8: служба ідентифікації імен.

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

Розглянувши усі блоки функціональної схеми перейдемо до розгляду діаграми взаємодії процесів, які відбуваються у системі.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. Після початку роботи розробленого ПЗ ми потрапляємо до головного блоку системи звідки через ланку дій відбувається наступне:

- Інтерфейс ПЗ.
- Налаштування ПЗ.
- Сервіс RCS.

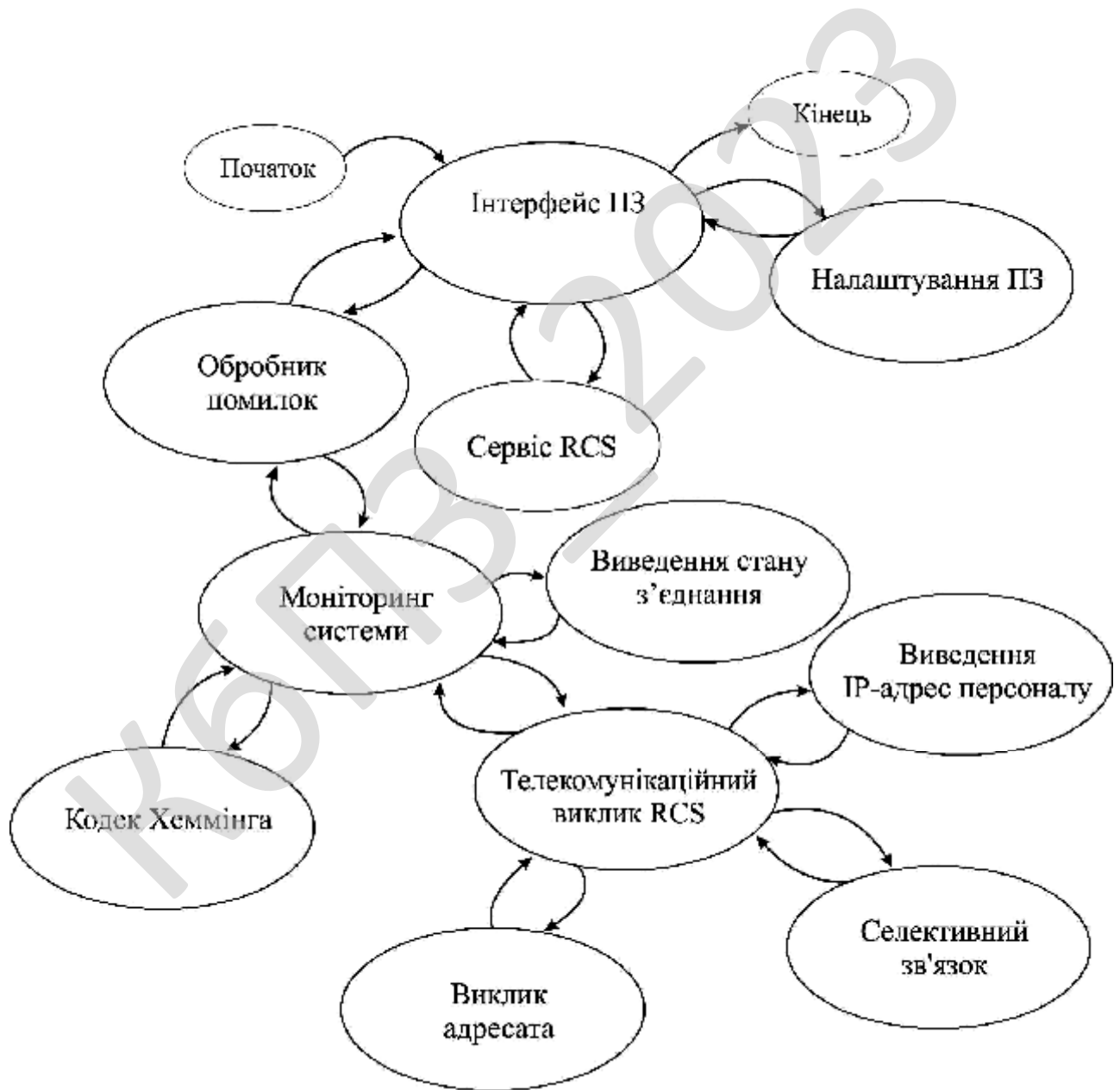


Рисунок 3.3 – Діаграма взаємодії процесів

- Обробник помилок.
- Моніторинг системи.
- Виведення стану з'єднання.
- Телекомунікаційний виклик RCS.
- Виведення IP-адрес персоналу.
- Селективний зв'язок.
- Виклик адресата.

Використовується модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма взаємодії процесів використовується для візуалізації процесів обробки даних (структурне проектування).

Для розробника вважається звичним спочатку креслити діаграму взаємодії процесів даних рівня контексту, завдяки чому буде показано взаємодію системи. Ця діаграма в подальшому підлягає уточненню шляхом деталізації процесів та потоків даних з метою показати систему що розробляється.

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.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем. На рисунку 4.1 зображена основна блок-схема програми, на рисунку 4.2 зображено роботу підпрограми.

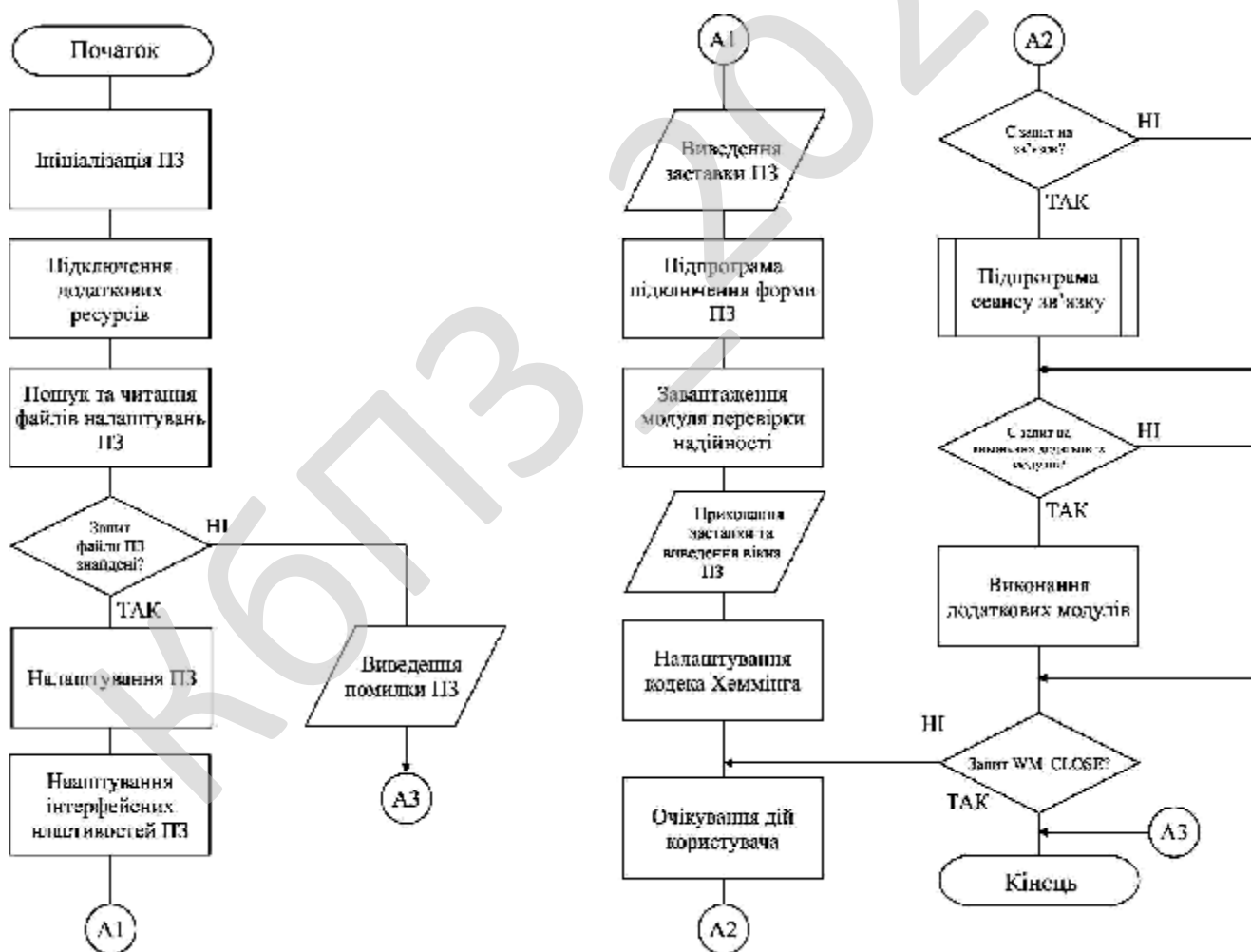


Рисунок 4.1 – Блок схема основної програми

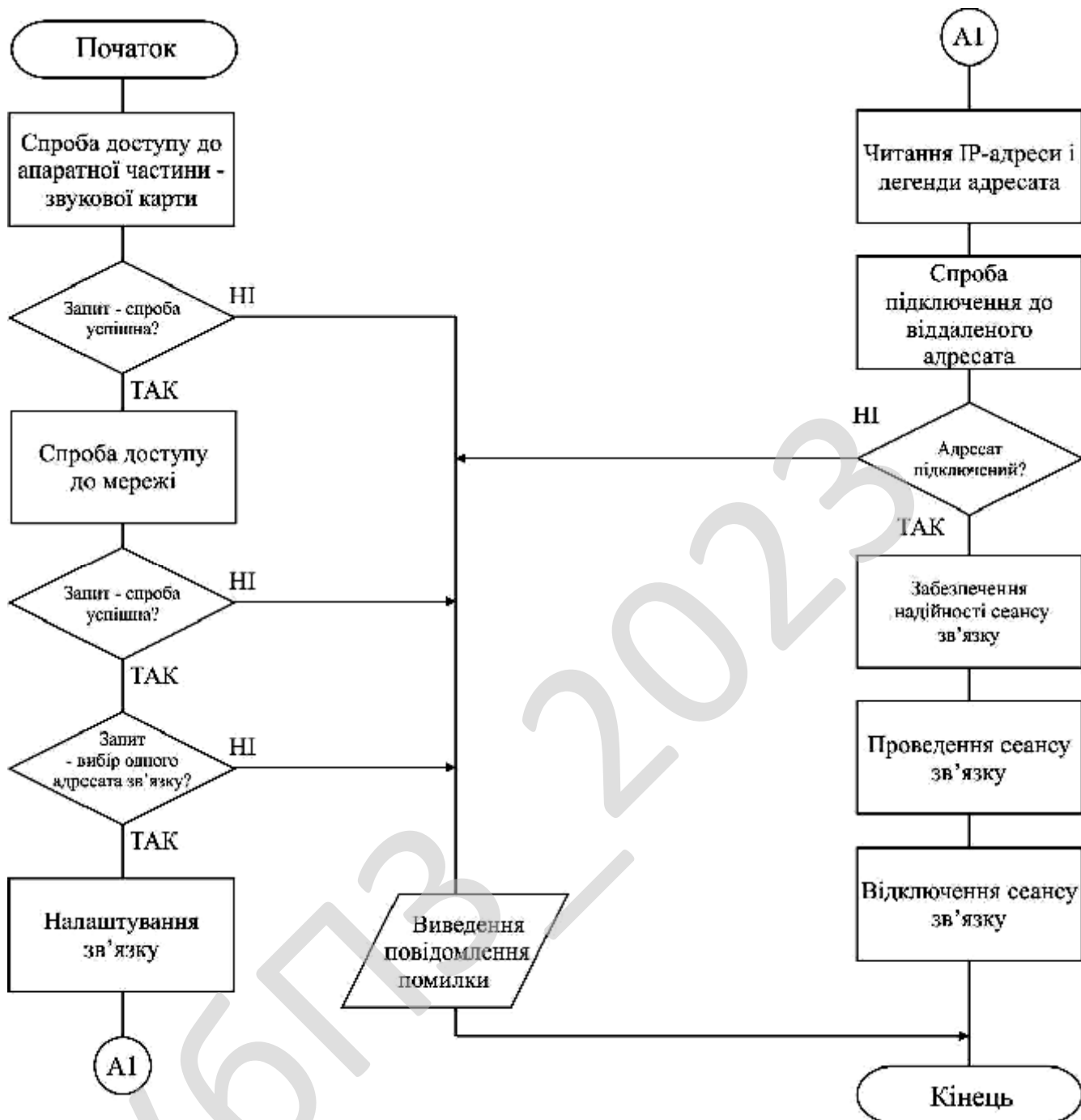


Рисунок 4.2 – Блок схема підпрограми

З рисунку видно що робота основної програми складається з початкових етапів ініціалізації ПЗ, перевірки наявності ресурсів системи, блоку початку основного циклу з чеканням запиту від користувача в якому відбувається виклик підпрограми та останньої стадії – перевірка поточного стану з завершенням роботи розробленого ПЗ.


```

if H<>' ' then
begin
    MessageDlg('Дякую пароль змінено',mtInformation,[mbOK],0);
    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;

```

4.2 Захист розробленого програмного забезпечення

Tiny Encryption Algorithm (TEA) [1] – блочний алгоритм шифрування типу «Мережі Фейстеля». Алгоритм був розроблений на факультеті комп'ютерних наук Кембриджського університету Девідом Вілером (David Wheeler) і Роджером Нідгемом (Roger Needham) та вперше представлений в 1994 році [2] на симпозіумі зі швидкими алгоритмами шифрування в Льовені (Бельгія).

Шифр не патентований, широко використовується в ряді криптографічних додатків і широкому спектрі апаратного забезпечення, завдяки вкрай низькими вимогами до пам'яті й простоті реалізації. Алгоритм має як програмну реалізацію на різних мовах програмування, так і апаратну реалізацію на інтегральних схемах типу FPGA.

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму TEA, який заснований на бітових операцій з 64-бітним блоком, має 128-бітний ключ шифрування. Стандартна кількість раундів мережі Фейстеля біля 64 (32 циклу), однак, для досягнення найкращої продуктивності або шифрування, число циклів можна варіювати від 8 (16 раундів) до 64 (128

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

раундів). Мережа Фейстеля несиметрична через використання в якості операції накладення додавання за модулем 232.

Перевагами шифру є його простота в реалізації, невеликий розмір коду й досить висока швидкість виконання, а також можливість оптимізації виконання на стандартних 32-бітних процесорах, так як в якості основних операцій використовуються операції виключна «АБО» (XOR), побітового зсуву й додавання за модулем 2³². Оскільки алгоритм не використовує таблиць підстановки і раундова функція досить проста, алгоритму потрібно не менше 16 циклів (32 раундів) для досягнення ефективної дифузії, хоча повна дифузія досягається вже через 6 циклів (12 раундів).

Алгоритм має відмінну стійкість до лінійного криптоаналізу і досить гарну до диференціального криптоаналізу. Головним недоліком цього алгоритму шифрування є його вразливість до атак «на пов'язаних ключах» (англ. Related-key attack). Через простий розклад ключів кожен ключ має 3 еквівалентних ключа. Це означає, що ефективна довжина ключа складає всього 126 біт [3] [4], тому даний алгоритм не слід використовувати в якості геш-функції.

Опис алгоритму

Вихідний текст розбивається на блоки по 64 біта кожен. 128-бітний ключ К ділиться на чотири 32-бітних підключа К [0], К [1], К [2] і К [3]. На цьому підготовчий процес закінчується, після чого кожен 64-бітний блок шифрується протягом 32 циклів (64 раундів) за нижченаведеним алгоритмом. [5]

Припустимо, що на вхід n-го раунду ($1 \leq n \leq 64$) надходять права й ліва частини (L_n, R_n), тоді на виході n-го раунду будуть ліва й права частини (L_{n+1}, R_{n+1}), які обчислюються за такими правилами:

$$L_{n+1} = R_n.$$

Якщо $n = 2 * i - 1$ для $1 \leq i \leq 32$ (непарні раунди), то:

$$R_{n+1} = L_n (\{ [R_n 4] K [0] \} \{ R_n i * \delta \} \{ [R_n 5] K [1] \}).$$

Якщо $n = 2 * i$ для $1 \leq i \leq 32$ (парні раунди), то:

$$R_{n+1} = L_n (\{ [R_n 4] K [2] \} \{ R_n i * \delta \} \{ [R_n 5] K [3] \}).$$

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Де

$X \oplus Y$ – операція додавання чисел X і Y за модулем 232.

$X \oplus Y$ – побітове виключне АБО» (XOR) чисел X і Y , яке в мові програмування Сі позначається як $X \wedge Y$

$X \ll Y$ і $X \gg Y$ – операції побітового зсуву числа X на Y біт вліво й вправо відповідно.

Константа δ була виведена з Золотого перерізу:

$$\delta = (-1) * 2^{31} = 2654435769 = 9E3779B9_{16}$$

У кожному раунді константа множиться на номер циклу i . Це було зроблено для запобігання простих атак, заснованих на симетрії раундів.

Також очевидно, що в алгоритмі шифрування TEA немає як такого алгоритму розкладу ключів. Замість цього в непарних раундах використовуються підключі $K [0]$ та $[1]$, у парних – $K [2]$ і $[3]$.

Так як це блочний шифроалгоритм, де довжина блоку 64-біт, а довжина даних може бути не кратна 64-біт, значення всіх байтів, які доповнюють блок до кратності в 64-біт, встановлюється в $0x01$.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		71

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

Розглянемо розроблене ПЗ яке зображено на рисунку 5.1. З рисунку можна побачити що інтерфейс головного вікна мобільного ПЗ розподілено на наступні розділи:

- Розділ налаштування.
- Розділ функціональних.
- Розділ випадаюче меню з наступним функціоналом: IP-адреси; Легенда IP-номерів; IP-виклик користувача; Вікно, у якому вказується, з ким спілкується користувач.
- Розділ обрання контактів.

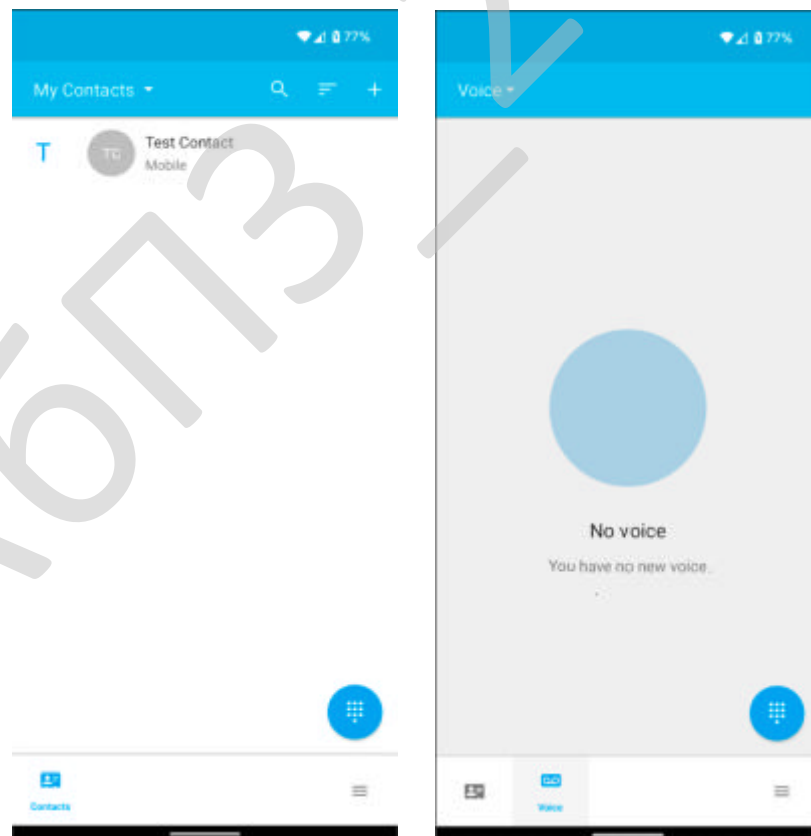


Рисунок 5.1 – Головне вікно ПЗ

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		72

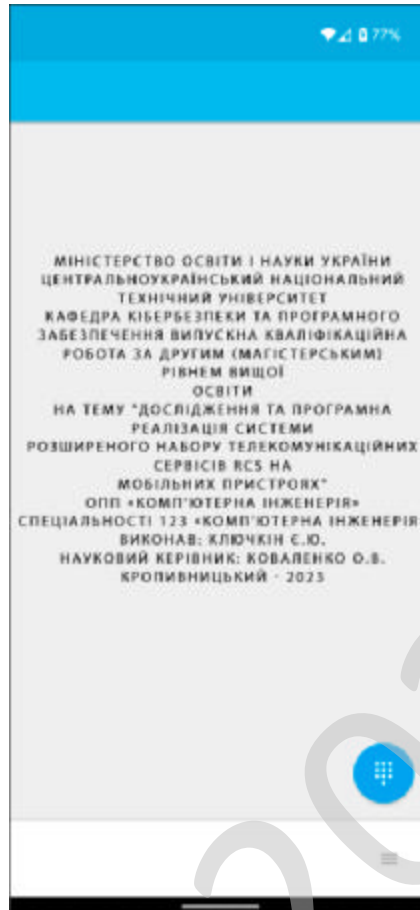


Рисунок 5.2 – Авторське право

На рисунку 5.2 зображено авторські дані розробленого програмного забезпечення. Обрано умови розповсюдження – Shareware. Під умовно-безплатним програмним забезпеченням можна розуміти спосіб або метод розповсюдження комерційного ПЗ на ринку (тобто на шляху до кінцевого користувача), при якому випробувачеві пропонується обмежена за можливостями (не повнофункціональна або демонстраційна версія), терміном дії (тріал версія) або версія з вбудованим набридливим нагадуванням про необхідність оплати використання програми. В іншому випадку користувач повинен припинити використання ПЗ та видалити його зі свого комп'ютера.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		73

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Метою розробки є дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Об'єктом дослідження є процес розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Предметом дослідження є методи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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 розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

– Розроблено вітчизняний продукт розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

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 проведене дослідження та виконана програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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	180
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)	–	В
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)	–	1
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.	–	18000
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	45
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-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

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)%.

$$T_{РП} = 0,3 \cdot 2,66 \cdot 12,9^{0,33+0,2(1,027-1,01)} \cdot 130 = 243 \text{ люд/день.}$$

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

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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	243	Ф 7.1-7.4
Впровадження	15	Д13
Всього	292	–

7.3 Визначення чисельності виконавців і планового фонду зарплати

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

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

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

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

$$Ч = \frac{292 \cdot 1}{24 \cdot 3} = 13,9 \text{ ставки.}$$

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	15	1350	22,5
Монітор	60	15	900	15
Клавіатура	30	15	450	7,5
Маніпулятор «мишка»	30	15	450	7,5
Принтер матричний	60	0	0	0,0
Принтер лазерний	120	3	360	6
Принтер струминний	60	1	60	1
Сканер	20	1	20	0,33
Концентратор-маршрутизатор	30	4	120	2
Кабельні господарства ЛОМ на 1 м.п.	2,5	650	1625	27,08
Копіювальний апарат	140	2	280	4,67
Усього за рік:			3 _ч	93,58

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:

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

$$\Phi_{\text{др}}^c = \frac{94 \cdot 1}{1,2} = 78 \text{ год.}$$

Визначаємо необхідну кількість ставок штатного персоналу сектора ТО:

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

Продовження таблиці 7.4

Посада	Вид роботи	Час	К-ть штатних одиниць
Продакт-менеджер	Презентації нової продукції, пошук каналів збуту	2	0,5
	Підтримка постійних клієнтів	1	
	Оформлення договорів, ведення тендерів	0,5	
	Контроль взаєморозрахунків з постачальниками	0,5	
Всього		4	
Дизайнер WEB	Розробка концепції оформлення та інтерфейсу сайту, оптимізація дизайну існуючих, проектує їх структуру та навігацію	2	0,5
	Створення графічних і стилістичних елементів сайту	1	
	Оформлення банерів і промо-сторінок	0,5	
	Розміщення графіки і контенту на Інтернет сторінках	0,5	
Всього		4	
Інженер верстальник	Розробка та верстка макетів рекламної продукції та технічної документації	1	0,25
	Верстка друкованих видань	0,5	
	Додрукова підготовка макетів	0,25	
	Розміщення графіки і контенту на Інтернет сторінках	0,25	
Всього		2	

Складемо штатний розклад виконавців.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		82

Таблиця 7.5 – Штатний розклад виконавців

Посада	Кількість ставок	Середньомісячний оклад, грн.	Всього за період розробки, грн.
Керівник (ІТ-менеджер)	1	20500	20500
Продакт-менеджер	0,5	17500	8750
Інженер-програміст	13,9	20000	278000
Інженер-електронщик	0,4	16000	6400
Інженер-системотехнік	0,25	16000	4000
Адміністратор мережі	0,5	16000	8000
Системний програміст	0,25	16000	4000
Дизайнер WEB	0,5	16000	8000
Інженер-верстальник	0,25	16000	4000
Бухгалтер-економіст	0,5	17000	8500
Всього за період розробки	$R_{cn} = 18,05$	-	$\Phi_{роб} = 350150$

Розрахуємо середньоденну зарплату одного виконавця:

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

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

$$z_{cd} = \frac{350150}{18,05 \cdot 24} = 808 \text{ грн.}$$

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:

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

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		83

де: R_{cn}^1 – кількість робочих місць виконавців, шт. Приймаємо 13 робочих місць;
 S_y – питома площа на одне робоче місце, m^2 ;
 $C_{пл}$ – вартість одного квадратного метра площі, грн.

Згідно даних ТОВ науково-дослідницького консалтингового підприємства «Пектораль» ціна одного квадратного метра площі новобудови, вік якої не перевищує 25 років, по місту складає 200...1600 у.о./ m^2 . Враховуючи, що курс складає 1 у.о. = 37 грн. приймаємо для розрахунку вартість одного метра квадратного рівною 20000 грн./ m^2 . На кожне робоче місце у середньому потрібно 8 m^2 . З урахуванням цього:

$$B_{y\partial} = 18 \cdot 8 \cdot 20000 = 2880000 \text{ грн.}$$

Вартість передавальних пристроїв складає 10% від вартості будівель, і у даному випадку вона складе: 288000 грн.

Балансова вартість інвентарю розраховується за нормою 15000 грн. на одне робоче місце. Тобто:

$$I_{нв} = R_{cn}^1 \cdot C_m, \quad (7.10)$$

де: C_m – ціна меблів для одного робочого місця, грн.

$$I_{нв} = 18 \cdot 15000 = 270000 \text{ грн.}$$

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 за прайсом фірми brain від 27.10.23 – джерело <https://brain.com.ua>.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

Продовження таблиці 7.6

Найменування комплектуючої або обладнання	Тип	Оптова ціна
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	2880000	-	-
2. Передавальні пристрої	288000	-	-
Всього по групі	3168000	5	158400
Група 4			
3. Обчислювальна техніка	185654	-	-
Всього по групі	185654	50	92827
Група 5, 6			
4. Вимірювальні пристрої	3999	25	999,75
5. Транспортні засоби	97500	20	19500
6. Господарський інвентар	270000	25	67500
Всього по групі	371499	-	87999,75
7. Нематеріальні активи	18000	10	1800
Разом	$K_p = 3995153$		$A_p = 406726,75$

Примітка: вартість автомобіля взята по даним з автосалону автотрейдинг, вкладки автобазар, джерело <http://www.auto-trading.com.ua/sale/lot20772.html>, складає 97500 грн.

Згідно виданих норм приймаємо 0,25 пачки паперу на місяць розробки. Тоді, враховуючи, що вартість пачки паперу складає $C_n = 160$ грн., визначаємо вартість паперу за період розробки $N_m = 1$ міс:

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

$$Z_{M1} = 160 \cdot 0,25 \cdot 1 = 40 \text{ грн.}$$

Згідно виданих норм до вартості запам'ятовуючих пристроїв входить вартість CD дисків в кількості 100 примірників та одного DVD диска для збереження резервної копії програми:

$$Z_{M2} = \sum C_d, \quad (7.17)$$

де: C_d – вартість дисків CD/DVD: CDR TDK 700Mb, 80Min, 52x Cake box – 16 грн./шт., DVD-R LG 4,7Gb, 16x speed Cake box – 32 грн./шт.

$$Z_{M2} = 100 \cdot 16 + 32 = 1632 \text{ грн.}$$

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

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

де: C_z – 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.

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

$$Z_M = (40 + 1632 + 1702) / 180 = 18,7 \text{ грн.}$$

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, %.

$$O_n = 1311 \cdot 15 \cdot 0,01 = 197 \text{ грн.}$$

Визначимо витрати на амортизацію основних фондів з урахуванням загальної річної суми амортизаційних відрахувань та кількості екземплярів програм ($N_e = 180$ прим.):

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		89

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

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

$$A_m = 406727 \cdot 1 / (180 \cdot 12) = 188 \text{ грн.}$$

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

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

$$C_n = 1311 + 131 + 317 + 197 + 18,7 + 197 + 188 = 2360 \text{ грн.}$$

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 45%.

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

де: P_n – рівень рентабельності, %.

$$P_p = 0,01 \cdot 45 \cdot 2360 = 1062 \text{ грн.}$$

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

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	1311
2. Додаткова зарплата виконавців	Z_d	131
3. Відрахування на соціальні потреби	C_{oc}	317
4. Загальногосподарські витрати	Γ_{ocn}	197
5. Витрати на матеріали	Z_m	18,7
6. Освоєння нових операційних систем, мов програмування	O_n	197

Продовження таблиці 7.9

1	2	3
7. Амортизація основних фондів	A_m	188
8. Повна собівартість програмного забезпечення	C_n	2360
9. Плановий прибуток	P_p	1062
10. Ціна підприємства $C_n = C_n + P_p$	C_n	3422
11. Податок на додану вартість $ПДВ = 0.01 \cdot H_{об} \cdot C_n$	$ПДВ$	684,4
12. Відпускна ціна програмної продукції $C = C_n + ПДВ$	C	4106,4

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	–	4106
Всього капітальних витрат	–	4106

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

$$Z_{ел\ баз} = 0,15 \cdot 1140 \cdot 3,8 = 650 \text{ грн.}$$

$$Z_{ел\ нов} = 0,15 \cdot 950 \cdot 3,8 = 542 \text{ грн.}$$

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

Таблиця 7.12 – Розрахунок амортизаційних відрахувань

Групи основних фондів	Норма амортизації %	Балансова вартість, грн., за варіантами		Сума відрахувань, грн за варіантами	
		Базовий	Новий	Базовий	Новий
Програмна продукція	25	–	4106	–	1026,5
Всього відрахувань	-	–	4106	–	1026,5

7.8 Визначення економічної ефективності програмної продукції

Економічна ефективність програмного забезпечення визначається для виготовлювача і споживача за такими показниками.

Величина економічного ефекту при виготовленні програмної продукції, розраховуємо за формулою:

$$E_e = (C_n - C_n) \cdot N_e - \sum_{i=1}^m E_{p_m} \cdot K_{p_m}, \quad (7.25)$$

де: K_p – балансова вартість основних фондів розробника, грн.; E_p – розрахунковий коефіцієнт капіталовкладень.

$$E_e = (3422 - 2360) \cdot 180 - (0,05 \cdot 3168000 + 0,5 \cdot 185654 + 0,25 \cdot 273999 + 0,2 \cdot 97500 + 0,1 \cdot 18000) \cdot 1/12 = 162741 \text{ грн.}$$

Визначимо період окупності додаткових капітальних вкладень у виробника програмної продукції:

$$T_e = \frac{K_p}{(C_n - C_n) \cdot N_e}, \quad (7.26)$$

де: K_p – балансова вартість основних фондів розробника.

$$T_g = \frac{3995153}{(3422 - 2360) \cdot 180 \cdot 12 / 1} = 1,7 \text{ року.}$$

Таблиця 7.13 – Показники економічної ефективності програмної продукції

Найменування показників	Одиниця виміру	Величина
1. Кількість екземплярів програми	Прим.	180
2. Повна собівартість розробленої програми	Грн.	2360
3. Ціна розробленої програми	Грн.	3422
4. Плановий прибуток від реалізації розробленої програми	Грн.	1062
5. Рентабельність програмної продукції	%	40
6. Об'єм додаткових капітальних вкладень у виробника програмної продукції	Грн.	3995153
7. Загальний прибуток від реалізації програмної продукції	Грн.	191160
8. Величина економічного ефекту при виготовлені програмної продукції	Грн.	162741
9. Період окупності додаткових капітальних вкладень у виробника програмної продукції	Років	1,7
10. Об'єм додаткових капітальних вкладень у споживача програмної продукції	Грн.	4106
11. Величина економічного ефекту у користувача програмної продукції	Грн.	11475
12. Період окупності додаткових капітальних вкладень у користувача програмної продукції	Років	0,33

Визначимо величину економічного ефекту у користувача програмної продукції за формулою:

$$E_{cn} = (I_{\bar{o}} - I_n) - E_n(K_n - K_{\bar{o}}), \quad (7.27)$$

де: $I_{\bar{o}}$, I_n – величина експлуатаційних витрат за базовим и новим варіантом відповідно;

$K_{\bar{o}}$, K_n – об'єм капітальних вкладень за варіантами, що порівнюються.

$$E_{cn} = (34200 - 21699) - 0,25 \cdot 4106 = 11475 \text{ грн.}$$

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

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

$$T_{cn} = \frac{4106}{34200 - 21699} = 0,33 \text{ року.}$$

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-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		95

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Загальна комп'ютеризація суспільства призвела до того, що використання комп'ютерів стало повсюдним у всіх сферах економіки та народного господарства. Застосування персональних комп'ютерів і ЕОМ дозволило значно підвищити продуктивність праці, змінити характер і зміст праці. Комп'ютеризація, поряд з незаперечними перевагами, тягне за собою і багато проблем. Для того, щоб активне застосування комп'ютерних технологій не стало додатковим чинником погіршення здоров'я, вкрай необхідно щоб робоче місце відповідало гігієнічним вимогам. Темою дипломного проекту є розробка та дослідження та реалізація програмного продукту, тому актуально буде розгляд умов праці програміста.

В охорону праці включають санітарно-гігієнічні, лікувально-профілактичні та організаційно-технічні системи правових і соціально-економічних заходів.

В кожній ІТ компанії є трудові відносини з працівниками. Згідно закону України “Про охорону праці” [1] кожна компанія впроваджує заходи з охорони праці. Реалізується трудові відносини з вживанням необхідних засобів з охорони праці та розробки відповідних документів:

- Інструкцій з охорони праці по кожній професії і загальні;
- Положення про охорону праці;
- Накази з охорони праці;
- Журнали реєстрації та інструктажу.

Роботодавець створює відділ який працює відповідно до типового положення, яку затверджується центральним органом виконавчої влади і забезпечує виконання вимог державної політики у сфері охорони праці.

За недотриманням вимог, керівники ІТ компаній можуть бути притягнуті до відповідальності, яка виглядає у виді накладання штрафу. Якщо в результаті

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		96

порушення умов охорони праці є постраждали працівники то керівні особи ІТ компаній притягуються до кримінальної відповідальності.

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, який зареєстровано в Міністерстві юстиції України 25 квітня 2018 р. за №508/31960 «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» [5], яким затверджено нормативно-правовий акт з охорони праці НПАОП 0.00-7.15-18, «Правила охорони праці під час експлуатації електронно-обчислювальних машин», та «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98.

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 навантаження. До шкідливих факторів, які впливають на робітників 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.

Розглянемо шкідливі чинники роботи програмістів керуючись наступними нормативно-правовими актами: «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПіН 3.3.2-007-98, та «Вимоги щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» НПАОП 0.00-7.15-18.

Умови праці програміста включають наступні фактори:

- параметри повітряного середовища в приміщенні;
- вентиляція приміщення;

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		97

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

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

8.2 Шкідливі і небезпечні фактори при роботі з комп'ютером

Електронно-обчислювальна машин (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;

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		98

обчислювальних машин» [2] та НПАОП 0.00-1.28-10 «Правила охорони праці під час експлуатації електронно-обчислювальних машин»).

Температура повітря в приміщенні визначається впливом температури зовнішнього повітря і тепловою енергією, яка виділяється всередині приміщення. Джерелами виділення теплоти в даному приміщенні є електроустаткування, освітлювальні прилади, а також люди. У світлий час доби джерелом надлишкового тепла є сонячна радіація. Згідно Постанови Головного державного санітарного лікаря України [5], робота, яка виконується в даному приміщенні, відноситься до категорії Іа. В цьому випадку людина витрачає енергії до 120 ккал у годину. Вологість повітря у приміщенні визначається впливом багатьох факторів, серед яких: вологість атмосферного повітря, виділення вологи людьми (при диханні та випарами з поверхні шкіри).

Мікроклімат повітряного середовища в приміщенні характеризується запиленістю та загазованістю повітря. Мікроклімат приміщення визначається діючим на організм людини поєднанням, вологості, температури, швидкості руху повітря та інтенсивності теплового випромінювання. Аналіз мікроклімату складається з визначення зазначених вище факторів і порівняння результатів із встановленими нормами.

У таблиці 8.3 наведено оптимальні та фактичні значення параметрів мікроклімату як для категорії ваги робіт Іа, так і розглянутого приміщення. У приміщеннях, де встановлено ЕОМ, рекомендується застосування тільки оптимальних значень показників мікроклімату.

Таблиця 8.3 – Оптимальні і фактичні значення параметрів мікроклімату

Пора року	Оптимальні для Іа			Фактичні		
	Температура, °С	Воло- гість,%	Швидкість повітря, м/с	Температура, °С	Воло- гість%	Швидкість повітря, м/с
Холодна	22-24	40-60	0,1	22-23,5	45-60	0,1
Тепла	23-25	50-70	0,1	23-25	52-70	0,12

Проведений аналіз показує, що показники мікроклімату в приміщенні відповідають установленим нормам. Штучне опалення застосовується у холодний період року.

В літню пору застосовується кондиціонер.

Для боротьби з пилом робляться регулярні провітрювання та вологі прибирання приміщенні.

У приміщенні знаходяться наступні джерела шуму: принтер HP Laser 107a, електродвигуни вентиляторів ЕОМ.

Одним з найважливіших факторів, які впливають на ефективність трудової діяльності людини, та попереджають травматизм і професійні захворювання програмістів є освітлення на робочому місці.

Працю працівника, який постійно працює за комп'ютером, згідно ДБН В.2.5 – 28 – 2006 р можна віднести до роботи з малою точністю (найменший розмір об'єкта розрізнення від 1 до 5 мм) V-го розряду зорової роботи, з великою контрастністю об'єкта розрізнення (символів на екрані дисплея), з темним тлом (під розряд зорової роботи В). Приміщення можна віднести до 1-ої групи приміщень, у яких проводиться розрізнення об'єктів зорової роботи при фіксованому напрямку лінії зору того, що працює на робочу поверхню. Для такого типу приміщень і розряду зорової роботи нормоване значення коефіцієнта природної освітленості (КПО) робочої поверхні (при поєднаному, спільному освітленні), повинен становити не більше 1,5%, освітленість при штучному висвітленні повинна становити 300 лк. Крім того все поле зору повинне бути освітлено достатньо рівномірно – ця основна гігієнічна вимога. Так як яскраве світло на ділянці периферійного зору значно збільшує напруженість очей і, як наслідок, призводить до їх швидкої стомлюваності, ступінь освітлення приміщення і яскравість екрану комп'ютера повинні бути приблизно однаковими.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		101

8.4 Розробка заходів з умов поліпшення охорони праці

Згідно аналізу умов праці в розглянутому приміщенні, ми одержали наступні результати:

- розмірі приміщення, у розрахунку на одному працюючого, відповідають нормативам;
- мікроклімат відповідає нормативному значенню;
- акустичні умови роботи не перевищують нормативних значень;

Таким чином можна припустити, що основною причиною можливого зниження працездатності програміста є психофізіологічний фактор, тому основна пропозиція буде така: дотримання позитивної психологічної атмосфери в колективі та регламентованого режиму праці та відпочинку, організація робочого місця з урахуванням ергономічних вимог.

Рекомендовані заходи: регулярні періодичні наочні огляди персоналом шляхів для евакуації людей із приміщення, відповідно до плану евакуації (який повинен розташовуватись на видному місці у приміщенні), включення до колективного договору мінімально можливого вмісту аптечок з обов'язково наявністю масок-клапанів, або іншого спорядження для штучного дихання. Регулярна періодична перевірка параметрів заземлення та занулення (вимірювання опору).

8.5 Розрахункова частина

Завдання: розрахувати штучне освітлення робочого приміщення.

Початкові дані: ширина робочого приміщення: 4 м.; довжина – 5 м.; висота – 3 м.

Розрахунок штучного освітлення проведемо за методом коефіцієнта використання світлового потоку.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		102

4. 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>

5. Постанова № 42 від 01.12.1999 Головного державного санітарного лікаря 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>

6. Оришака, О. В. Основи охорони праці: навч. посіб. / О. В. Оришака, Г. П. Горбачова, К. М. Марченко; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т. – Кропивницький: ЦНТУ, 2022. – 175 с. – Режим доступу до ресурсу: <http://dspace.kntu.kr.ua/jspui/handle/123456789/12161> (дата звернення: 16.06.2023).

КБГПЗ-2023

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		105

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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 розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.
- Досліджена система розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.
- На основі отриманих результатів досліджень створена програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		106

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

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

Програма реалізована на мові високого рівня RAD Studio Delphi. 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 Android/iOS.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм TEA.

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 11475 грн. З урахуванням вартості розробки програми та обладнання, строк окуплення становить 0,33 роки.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		107

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Ключкін Є.Ю. Дослідження та програмна реалізація системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2023.
2. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 1». Cisco Press. 2020. – 848 p.
3. Wendell Odom. «CCNA 200-301 Official Cert Guide, Volume 2 Premium Edition eBook and Practice Test». Cisco Press. 2020. – 624 p.
4. Scott Jernigan «CompTIA Network+ Certification All-in-One Exam Guide, Eighth Edition». 2022. – 976 p.
5. Doug Lowe «Networking For Dummies 12th Edition». 2020. – 480 p.
6. Ramon Nastase «Computer Networking: The Beginner's guide for Mastering Computer Networking, the Internet and the OSI Model». 2018. – 186 p.
7. Russ White & Ethan Banks «Computer Networking Problems and Solutions: An Innovative Approach to Building Resilient, Modern Networks». 2017. – 832 p.
8. 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.
9. Smirnova, T., Gnatyuk, S., Yudin, O., Sydorenko, V., Polozhentsev, A., «The Model for Calculating the Quantitative Criteria for Assessing the Security Level of Information and Telecommunication Systems». *CEUR Workshop Proceedings Volume 3156*, 2022, Pages 390-399.
10. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools».

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		108

Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

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

12. 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.

13. 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.

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

15. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings* Volume 2616, 2020, Pages 125-136.

16. 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.

17. 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.

					БКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		109

18. 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.

19. 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.

20. 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.

21. Smirnov, O., Odarchenko, R., Abakumova, A., Usik, P., Kundyz, M., «QoE optimization technique for media delivery in 5G networks». *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, Kyiv, Ukraine, 8 – 11 October 2019. P.597-601.

22. 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.

23. 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.

24. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

					БКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		110

25. 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.

26. 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.

27. 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.

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

29. Смірнова Т.В., Гнатюк С.О., Сидоренко В.М., Юдін О.Ю., Сидоренко С.Ю., «Модель визначення критичності галузевих інформаційно-телекомунікаційних систем». *Проблеми інформатизації та управління*, № 2(70). 2022. С. 28-37.

30. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Смірнов С.А., Поліщук Л.І., «Дослідження стійкості до диференціального криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку*, 2022, № 3(69). С. 93-98.

31. Смірнов О.А., Смірнова Т.В., Якименко Н.М., Поліщук Л.І., Смірнов С.А. «Дослідження статистичної стійкості та швидкісних характеристик запропонованої функції гешування удосконаленого модуля криптографічного

захисту в інформаційно-комунікаційних системах» *Вісник Хмельницького національного університету. Серія: «Технічні науки», № 2 (307). С. 46-52. 2022.*

32. Смірнов О.А., Смірнова Т.В., Константинова Л.В., Смірнов С.А., Якименко Н.М., «Дослідження стійкості до лінійного криптоаналізу запропонованої функції гешування удосконаленого модуля криптографічного захисту в інформаційно-комунікаційних системах» *Системи управління, навігації та зв'язку, 2022, № 1(67). С. 84-89.*

33. Смірнов О.А., Смірнова Т.В., Буравченко К.О., Кравченко С.С., Горбов В.О., «Хмарна система підтримки прийняття рішень технологічного процесу відновлення поверхонь конструкцій і деталей машин». *Сучасні інформаційні системи. 2021. Т. 5, № 4. С. 79-95*

34. Смірнов О.А., Усік П.С., Миронець І.В., Буравченко К.О., Якименко Н.М. «Метод підвищення ефективності розподіленої обробки даних у комп'ютерних системах операторів стільникового зв'язку» *Вісник Черкаського державного технологічного університету. Технічні науки. №4. С. 103-110. 2020.*

35. 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.*

36. Смірнов О.А., Коноплицька-Слободенюк О.К., Смірнов С.А., Буравченко К.О., Смірнова Т.В., Поліщук Л.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2020. – 294 с.

37. 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.*

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

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		112

39. 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.

40. 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).

41. 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

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

43. Смірнова Т.В., Солових Є.К., Смірнов О.А., Дреєв О.М. Побудова хмарних інформаційних технологій оптимізації технологічного процесу відновлення та зміцнення поверхонь деталей. Центральноукраїнський науковий вісник. Технічні науки. № 1(32). с. 184-194, 2019.

44. Смірнов О.А., Смірнов С.А., Поліщук Л.І., Смірнова Т.В., Коноплицька-Слободенюк О.К. Метод формування антивірусного захисту даних з використанням безпечної маршрутизації метаданих. Кібербезпека: освіта, наука, техніка. – Том 3 № 3. – Київ: КУ ім. Бориса Грінченка. – 2019. – С. 63-87.

45. Смірнов О.А., Гнатюк С.О., Кавун С.В., Терейковський І.А., Жмурко Т.О., Смірнов С.А., Коваленко А.С. Основи безпеки в комп'ютерних мережах. Навчальний посібник – Кропивницький: вид. Лисенко В.Ф. 2018. – 177 с.

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		113

46. Смірнов О.А., Котелянець В.В. Стійкі до колізій стохастичні моделі функціонування безпроводових сенсорних мереж. Вісник інженерної академії України, №3, с. 145-152, 2018

47. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Алгоритми формування безлічі маршрутів передачі метаданих у антивірусні хмарні системи. Збірник наукових праць "Системи обробки інформації". – Випуск 5 (142). – Х.: ХУПС – 2016. – С. 148-152.

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

49. Смірнов О.А., Смірнов С.А., Дідик А.К., Дреєв А.М. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Спосіб контролю ліній зв'язку телекомунікаційної системи антивірусу. Збірник наукових праць Харківського університету Повітряних Сил. Випуск 2 (47). – Харків: ХУПС. – 2016. – С. 121-127.

50. Смірнов О.А., Смірнов С.А., Дідик А.К. Метод безпечної маршрутизації метаданих у хмарні антивірусні системи. Системи озброєння та військова техніка. – Випуск 2 (46) – Х.: ХУПС – 2016. – С. 146-149.

51. Смірнов О.А., Кавун С.В., Доренський О.П., Вялкова В.І. Інформаційна безпека в комп'ютерних мережах. Навчальний посібник – Кіровоград: РВЛ КНТУ, 2016. – 151 с.

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

					ВКРМ-123.23.0037.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		114

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-123.23.0037.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Клочкін Є.Ю.				Літ.	Аркуш	Аркушів
Перевірів	Коваленко О.В.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-22М-2		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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. № 35-13 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 розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях.

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-123.23.0037.00.00.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 розширеного набору телекомунікаційних сервісів RCS на мобільних пристроях;
- 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-123.23.0037.00.00.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 мобільному пристрої з ОС Android/iOS 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 Android/iOS.

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 RAD Studio Delphi.

					ВКРМ-123.23.0037.00.00.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 повинен бути розглянутий аналіз санітарно-гігієнічних умов праці на робочому місці програміста.

					ВКРМ-123.23.0037.00.00.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 – 114 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-123.23.0037.00.00.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
_____ Коваленко О.В.

*Дослідження та програмна реалізація
системи розширеного набору телекомунікаційних сервісів RCS на мобільних
пристроях*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 35

Літера: РП

Кропивницький – 2023 року

RCS_services.pas - Програма організації зв'язку

```

program RCS_services;
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 := 'RCS_services';
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

U_SPLASH.pas - Підпрограма завантаження заставки

```
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.
```

usimple.pas - 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;
    Mem1: 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;
    Image1: TImage;
```

```

    Timer1: TTimer;
    procedure Button1Click(Sender: TObject);
    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);
        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 Memo1.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.
```

K6П3 - 2023

Unit1.pas - 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].Dat
a.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 ('У системі немає звукового міксера!', 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

**Unit2.pas - 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.
```

Unit3.pas - 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.
```

Unit4.pas - Підпрограма парольного захисту

```

unit Unit4;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, Buttons, Mask, ExtCtrls;
type
  TForm4 = class(TForm)
    Panel1: 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#$T%&(*qwrda@@#@45';
  DAT:string='VUU';
{$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.

```

AMixer.pas - 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;
    {Ця функція повертає значення 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);
    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;

```

```

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;

    // відбулась помилка
    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);
                end;
            end;
          end;
        end;
      end;
    end;
  end;
end;

```

```

        continue;
    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;
                end;
                Inc (A);
                end;
            {
                If LeftVol = -1 then
                    VolDisabled := True;
                If Mute = -1 then
                    MuteDisabled := True;}
            end;
        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;
    end;
    Inc (A);
end;
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 (not VolSet OR not MuteSet) AND (A < Cntrls.Count) do
        begin
            Cntrl := Cntrls[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;
  pmcMixerDataUnsignedMute : 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 (pmcMixerDataUnsignedMute, SizeOf (TMIXERCONTROLDETAILSBOOLEAN));

  with mlcMixerLineControlsMute do
    begin
      cbStruct := SizeOf (TMIXERLINECONTROLS);
      dwLineID := mlMixerLine.dwLineID;
      dwControlType := MIXERCONTROL_CONTROLTYPE_MUTE;
      cControls := 1;
      cbmxcntrl := SizeOf (TMIXERCONTROL);
      pamxcntrl := 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
Cntrls := MD.Controls;
ML := MD.Data;
If (MC <> nil) AND (Cntrls <> nil) then
begin
A := 0;
while (Result = False) AND (A < Cntrls.Count) do
begin
Cntrl := Cntrls[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 (pmcMixerDataUnsignedMute);
FreeMem (pmcMixerControlMute);
end;
end;

function TAudioMixer.SetMute (ADestination, AConnection: Integer; Mute:
Boolean): Boolean;
var
    MD : TMixerDestination;
    MC : TMixerConnection;
    mlcMixerLineControlsMute : TMIXERLINECONTROLS;
    mcdMixerDataMute : TMIXERCONTROLDETAILS;
    pmcMixerControlMute : PMIXERCONTROL;
    pmcMixerDataUnsignedMute : 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;

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

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

```

```

if (mixerGetLineControls(FMixerHandle, @mclMixerLineControlsMute,
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 Mute then
pmcdsMixerDataUnsignedMute^.fValue := 1
else
pmcdsMixerDataUnsignedMute^.fValue := 0;

if
(mixerSetControlDetails(FMixerHandle,@mcdMixerDataMute,MIXER_SETCONTROLDETAILS
F_VALUE) = MMSYSERR_NOERROR) then
Result := True;
end
else
begin
If (AConnection <> -1) then
begin
Cntrls := MD.Controls;
ML := MD.Data;
If (MC <> nil) AND (Cntrls <> nil) then
begin
A := 0;
while (Result = False) AND (A < Cntrls.Count) do
begin
Cntrl := Cntrls[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;
if (mixerGetControlDetails (FMixerHandle, @mcdMixerDataMute,
MIXER_GETCONTROLDETAILSF_VALUE) <> MMSYSERR_NOERROR) then
begin
Inc (A);
continue;
end;
if (Cntrl.dwControlType = MIXERCONTROL_CONTROLTYPE_MUX) then
For B := 0 to Cntrl.cMultipleItems-1 do
Details[B] := 0;
If Mute then
begin
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
            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;
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;
            end;
        end;
    end;
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.

```

AMixer.pas- Довідка

```

unit about;

interface

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

type
  TForm5 = class(TForm)
    Mem1: TMemo;
    Button1: TButton;
    Image1: TImage;
    procedure FormCreate(Sender: TObject);
    procedure Button1Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form5: TForm5;

implementation

{$R *.dfm}

procedure TForm5.FormCreate(Sender: TObject);
begin
  Mem1.Clear;
  Mem1.Lines.Add('МАГІСТЕРСЬКА РОБОТА');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('на тему:');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('Дослідження та програмна реалізація системи розширеного
набору');
  Mem1.Lines.Add('телекомунікаційних сервісів RCS на мобільних пристроях');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('Керівник: Коваленко О.В. ');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('Розробив: студент Ключкін Євген Юрійович');
  Mem1.Lines.Add('гр. KI-22М-2');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('м. Кропивницький 2023');
  Mem1.Lines.Add('');
  Mem1.Lines.Add('');
end;

procedure TForm5.Button1Click(Sender: TObject);
begin
  Form5.Close;
end;

end.

```