

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
« ____ » _____ 2024 р.

ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
за другим (магістерським) рівнем вищої освіти
на тему
“Дослідження та програмна реалізація системи віддаленого
керування мобільними пристроями під керуванням ОС
Android”

Виконав здобувач вищої освіти
II курсу, групи КІ-23М
ОПП «Комп’ютерна інженерія»
спеціальності 123 «Комп’ютерна інженерія»
_____ Коптєв Р.В.
« ____ » _____ 2024 р.

Керівник проекту
кандидат технічних наук
_____ Улічев О.С.
« ____ » _____ 2024 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 2024 року

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

Коптєву Руслану Валерійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи *Дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android*

2. Керівник роботи *Улічев Олександр Сергійович, канд. техн. наук*

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

затверджені наказом вищого навчального закладу № 19-13 від 07.08.2024 року

3. Строк подання студентом роботи до захисту *2.12.2024 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою розробки є дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android*

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

- | | |
|--|--|
| <i>1. Призначення та область використання.</i> | <i>6. Наукова новизна.</i> |
| <i>2. Перегляд аналогічних існуючих систем.</i> | <i>7. Маркетингове та економічне обґрунтування IT-проєкту.</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	Доренська А.О.	05.10.2024	14.11.2024
Охорона праці	Марченко К.М., к.т.н., доцент	06.10.2024	16.11.2024

7. Дата видачі завдання « 6 » вересня 2024 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	10.10.2024 р.	
2.	Постановка задачі, оформлення ТЗ	15.10.2024 р.	
3.	Розробка моделі компонента	20.10.2024 р.	
4.	Розробка структур даних	25.10.2024 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.10.2024 р.	
6.	Програмування алгоритмів	10.11.2024 р.	
7.	Розрахунок економічної ефективності	13.11.2024 р.	
8.	Розрахунки з охорони праці та техніки безпеки	15.11.2024 р.	
9.	Оформлення ПЗ	17.11.2024 р.	
10.	Попередній захист роботи	2.12.2024 р.	

Дата видачі завдання
« 6 » вересня 2024 р.

Підпис керівника

(прізвище та ініціали)Завдання прийнято до виконання
« 6 » вересня 2024 р.

Підпис здобувача

(прізвище та ініціали)

АНОТАЦІЯ

Коптєв Р.В. Дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android. 123 Комп'ютерна інженерія. Центральнoукраїнський національний технічний університет. Кропивницький. 2024.

В даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи віддаленого керування мобільними пристроями під керуванням ОС Android.

Метою розробки є дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android.

Об'єктом дослідження є процес віддаленого керування мобільними пристроями під керуванням ОС Android.

Предметом дослідження є методи віддаленого керування мобільними пристроями під керуванням ОС Android.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

Результат роботи – програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android.

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 під керуванням ОС Android.

Програму розроблено в середовищі Python.

Ключові слова: комп'ютерна інженерія, віддалене керування, мобільні пристрої, ОС Android

ABSTRACT

Koptev R.V. Research and software implementation of a remote control system for mobile devices running the OS Android. 123 Computer engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2024.

In this graduation thesis for the second (master's) level of higher education, software is developed, which is intended for the remote control system of mobile devices running the OS Android.

The purpose of the development is the research and software implementation of the remote control system for mobile devices running the OS Android.

The object of research is the process of remote control of mobile devices running the OS Android.

The subject of research is methods of remote control of mobile devices running the OS Android.

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

The result of the work is a software implementation of the remote control system for mobile devices running the OS Android.

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 running the OS Android.

The program was developed in the Python environment.

Keywords: computer engineering, remote control, mobile devices, OS Android

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ	3
ВСТУП.....	4
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ	6
1.1 Призначення системи.....	6
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ	8
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти.....	8
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	18
2.3 Розгорнута постановка завдання	20
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ	22
3.1 Опис функціонування системи	22
3.2 Розробка структурної схеми.....	27
3.3 Розробка функціональної схеми	33
3.4 Розробка діаграми процесів.....	35
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	37
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	37
4.2 Захист розробленого програмного забезпечення.....	50
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ	55
6 НАУКОВА НОВИЗНА	59

					ВКРМ-123.24.0017.00.00.ПЗ			
Вим.	Арк.	№ докум.	Підп.	Дата	<i>Дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android</i>	Літ.	Аркуш	Аркушів
<i>Розроб.</i>	<i>Коптєв Р.В.</i>					М	1	86
<i>Перев.</i>	<i>Улічев О.С.</i>							
<i>Н.контр.</i>	<i>Коваленко А.С.</i>							
<i>Затв.</i>	<i>Смірнов О.А.</i>						ЦНТУ КІ-23М	

7	МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ ІТ-ПРОЄКТУ	60
7.1	Визначення цільової аудиторії кінцевого готового продукту	60
7.2	Оцінка привабливості шляхом застосування методів експертних оцінок ...	61
7.3	Вибір методу оцінки вартості ПЗ	62
7.4	Розрахунок економічної ефективності від впровадження реалізованого ПЗ як фактору його привабливості.....	63
7.5	Пропозиція алгоритму просування проєкту розробки ПЗ	64
7.6	Оптимізація каналів збуту та шляхів реалізації ПЗ	65
7.7	Визначення ключових факторів успіху конкретного проєкту.....	66
8	ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ	68
8.1	Вступ.....	68
8.2	Пожежна безпека.....	69
8.3	Характеристика умов праці програміста	70
8.4	Розробка заходів з умов поліпшення охорони праці	73
8.5	Розрахункова частина	74
9	ОСНОВНІ ВИСНОВКИ.....	78
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	80

КБПЗ-2024

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		2

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ABH	–	автоматичний визначник номера
ATM	–	автоматична телефонна мережа
ЛОМ	–	локальна обчислювальна мережа
ОП	–	оперативна пам'ять
ОС	–	операційна система
ПК	–	персональний комп'ютер
ЦП	–	центральний процесор
AES	–	алгоритм шифрування
DNS	–	сервер доменних імен
IP	–	Internet Protocol
LAN	–	локальна мережа
MAC	–	Media Access Control – управління доступом до носія
NAT	–	Network Address Translation
TCP	–	Transmission Control Protocol
UDP	–	User Datagram Protocol – протокол користувальницьких дейтаграмм
UPS	–	блок безперебійного живлення

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

ВСТУП

Актуальність теми. Є багато способів контролю й доступу до вашому ПК або Mac за допомогою пристрою Android, але дотепер, на жаль, одержання доступу до Android пристроїв за допомогою комп'ютера, іноді сполучено з деякими труднощами. За суттю, програма віддаленого керування мобільним пристроєм перетворює ваш телефон у локальний HTTP сервер з можливістю доступу до нього будь-якого пристрою, підключеного до мережі. Іншими словами – вам досить просто запустити додаток на Android пристрої, підключитися до Wi-Fi мережі й ви можете за допомогою будь-якого веб-браузера ввійти на своєрідний сайт, що відкриває вам доступ до ресурсів пристрою.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android.

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

- Огляд існуючих систем віддаленого керування мобільними пристроями під керуванням ОС Android.
- Дослідження системи віддаленого керування мобільними пристроями під керуванням ОС Android.
- Програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android.

Об'єктом дослідження є процес віддаленого керування мобільними пристроями під керуванням ОС Android.

Предметом дослідження є методи віддаленого керування мобільними пристроями під керуванням ОС Android.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

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

– Удосконалено метод віддаленого керування мобільними пристроями під керуванням ОС Android.

– Розроблено вітчизняний продукт віддаленого керування мобільними пристроями під керуванням ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

Практична цінність отриманих результатів полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі віддаленого керування мобільними пристроями під керуванням ОС Android.

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 LV науково-технічної конференції «Наука в ЦНТУ: основні досягнення та перспективи розвитку» (2024 р.), основні положення випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти надруковані у статті збірника праць молодих науковців ЦНТУ, випуск №15.

Таким чином, виходячи з вищеперахованого, дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

1.1 Призначення системи

Програма надасть вам доступ до всіх файлів, що зберігається в цей час у телефоні, у тому числі ви зможете переглядати зображення й прослуховувати музику за допомогою музичного плеєра. Ви також одержите можливість, установки й створення резервних копій всіх додатків вашого пристрою. Програма відкриває доступ навіть до системних додатків, які звичайно недоступні користувачам без root прав.

Крім того, ви одержите повний доступ до контактів вашого Android телефону, журналів викликів і SMS-повідомлень. І все це за допомогою зручного й інтуїтивно зрозумілого інтерфейсу у вигляді робочого стола Android

Звичайно, деякі речі, які надає вам програма можна зробити за допомогою USB кабелю й спеціалізованого програмного забезпечення, встановленого на комп'ютер, однак як бути, коли ви перебуваєте в гостях або в офісі партнера, а кабелю під рукою немає й на комп'ютері не встановлені ці спеціалізовані програми? От отут й застосуєте розробляємо у даній роботі програму, адже веб-браузер є на будь-якому комп'ютері, і всі що вам буде потрібно – це підключити свій пристрій до WiFi мережі.

1.2 Область застосування

Чи можна дистанційно керувати телефоном Android? Так, ви можете дистанційно керувати телефоном Android за допомогою різних програм дистанційного керування. Ці програми дозволяють отримати доступ до пристрою Android і керувати ним з іншого пристрою, наприклад комп'ютера чи іншого

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

мобільного пристрою. У сучасному швидкоплинному взаємопов'язаному світі дистанційне керування пристроями стало необхідністю.

Програмне забезпечення для дистанційного керування дозволяє керувати телефоном Android з іншого пристрою. Це може бути підключення ПК до телефону або з телефону на телефон. Незалежно від того, чи потрібно вам отримати доступ до файлів, керувати програмами чи вирішувати проблеми, програми дистанційного керування дають вам змогу керувати своїм Android так, ніби він у вас у руках.

Незалежно від того, чи ви ентузіаст техніки чи бізнес-професіонал, програмне забезпечення для дистанційного керування дозволяє отримати доступ до екрана, програм і файлів вашого телефону Android практично з будь-якого місця. Це як носити панель управління телефону в кишені!

Вивчення методів віддаленого керування телефоном Android пропонує різноманітні варіанти. У цій роботі реалізована програма дистанційного керування для телефонів Android, яка спрощують процес дистанційного керування пристроями Android.

Таким чином, виходячи з вищеперерахованого, дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7

2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Наведемо нижче огляд дев'яти найпопулярніших програм дистанційного керування для телефонів Android із покроковими інструкціями для безпроблемного впровадження.

AnyViewer

Щоб знайти найкращу програму дистанційного керування для телефону Android AnyViewer є найкращим вибором завдяки його продуктивності, простоті використання та потужним функціям, які роблять його чудовим вибором для віддаленого керування телефоном Android з комп'ютера. На відміну від багатьох інших програм, AnyViewer розроблено для забезпечення швидкого та безперебійного дистанційного керування, забезпечуючи плавне з'єднання з низькою затримкою навіть на великих відстанях. Якщо ви керуєте файлами, надаєте віддалену технічну підтримку чи просто керуєте телефоном із комп'ютера, інтуїтивно зрозумілий інтерфейс AnyViewer полегшить це.

Програмне забезпечення також має надійне шифрування та безпечний доступ для збереження даних під час віддалених сеансів. Завдяки міжплатформній сумісності та миттєвому підключенню AnyViewer змінює ваш спосіб взаємодії з пристроєм Android, роблячи його незамінним інструментом для особистого та професійного використання. Крім того, основна функціональність AnyViewer не обмежується прямим дистанційним керуванням, а забезпечує віддалений доступ без нагляду, безперебійна передача файлів, текстовий чат у реальному часі та скоординоване виконання завдань.

Його універсальність робить його всеохоплюючим рішенням для

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		8

різноманітних сценаріїв, включаючи ситуації віддаленої роботи та надання віддаленої допомоги. Незалежно від того, чи ви ентузіаст техніки чи бізнес-професіонал, AnyViewer є ідеальним рішенням для безперервного, надійного та безпечного дистанційного керування.

Крок 1. Почніть із завантаження та встановлення AnyViewer на вашому пристрої Android. Після встановлення відкрийте програму та натисніть «Зареєструватися зараз», щоб створити новий обліковий запис.

Крок 2. Далі встановіть AnyViewer на свій комп'ютер і увійдіть, використовуючи ті самі дані облікового запису.

Крок 3. Перейдіть до розділу «Пристрої» в AnyViewer, знайдіть свій пристрій Android і виберіть «Віддалене керування».

Крок 4. На своєму пристрої Android увімкніть «AnyViewer Auxiliary Service», натиснувши «Allow» у меню Installed Services.

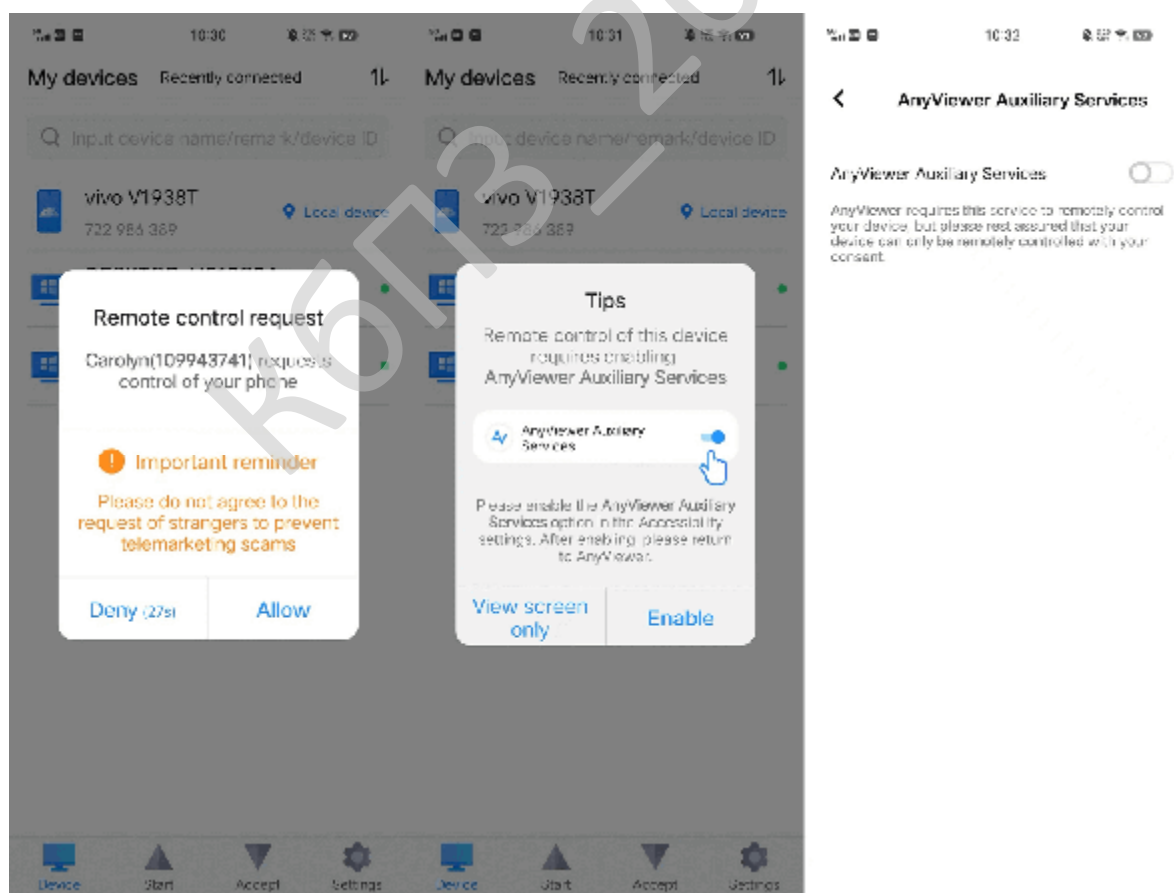


Рисунок 2.1 – Інтерфейс користувача AnyViewer

Крок 5. Тепер ви готові дистанційно керувати своїм пристроєм Android з комп'ютера за допомогою AnyViewer.

Поради. Для ще більшої ефективності оновіть обліковий запис AnyViewer до Professional or Enterprise plan. Оновлення дозволяє керувати декількома мобільними пристроями одночасно, розширити можливості дистанційного керування екраном і насолоджуватися неперевершеним зображенням у режимі Ultra HD, режимі 4:4:4 True Color.

AirDroid

AirDroid виділяється як зручна та комплексна програма, розроблена для оптимізації керування пристроями Android. Цей універсальний інструмент дозволяє користувачам легко керувати своїми пристроями Android з комп'ютерів або інших пристроїв, надаючи набір функцій для покращення зв'язку. Якщо ви шукаєте ефективне рішення для керування пристроєм, AirDroid – популярний і надійний вибір.

Крок 1. Розпочніть процес, завантаживши та встановивши AirDroid Personal на пристрої Android і комп'ютері.

Крок 2. Відкрийте програму на своєму пристрої Android, увійдіть і переконайтеся, що ви надали необхідні дозволи на дистанційне керування.

Крок 3. Увійдіть у AirDroid Personal на своєму комп'ютері, використовуючи той самий ідентифікатор, пов'язаний із вашим пристроєм Android.

Крок 4. У меню робочого столу знайдіть і натисніть кнопку «Віддалене керування». Визначте свій Android-пристрій зі списку та виберіть функцію «Віддалене керування», розташовану внизу праворуч.

Крок 5. Переконайтеся, що екран вашого Android плавно віддзеркалюється на вашому ПК, надаючи вам можливості легкого керування.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		10

забезпечення TeamViewer на своєму комп'ютері, встановлюючи безпечне з'єднання.

Крок 6. На пристрої Android натисніть «Дозволити».

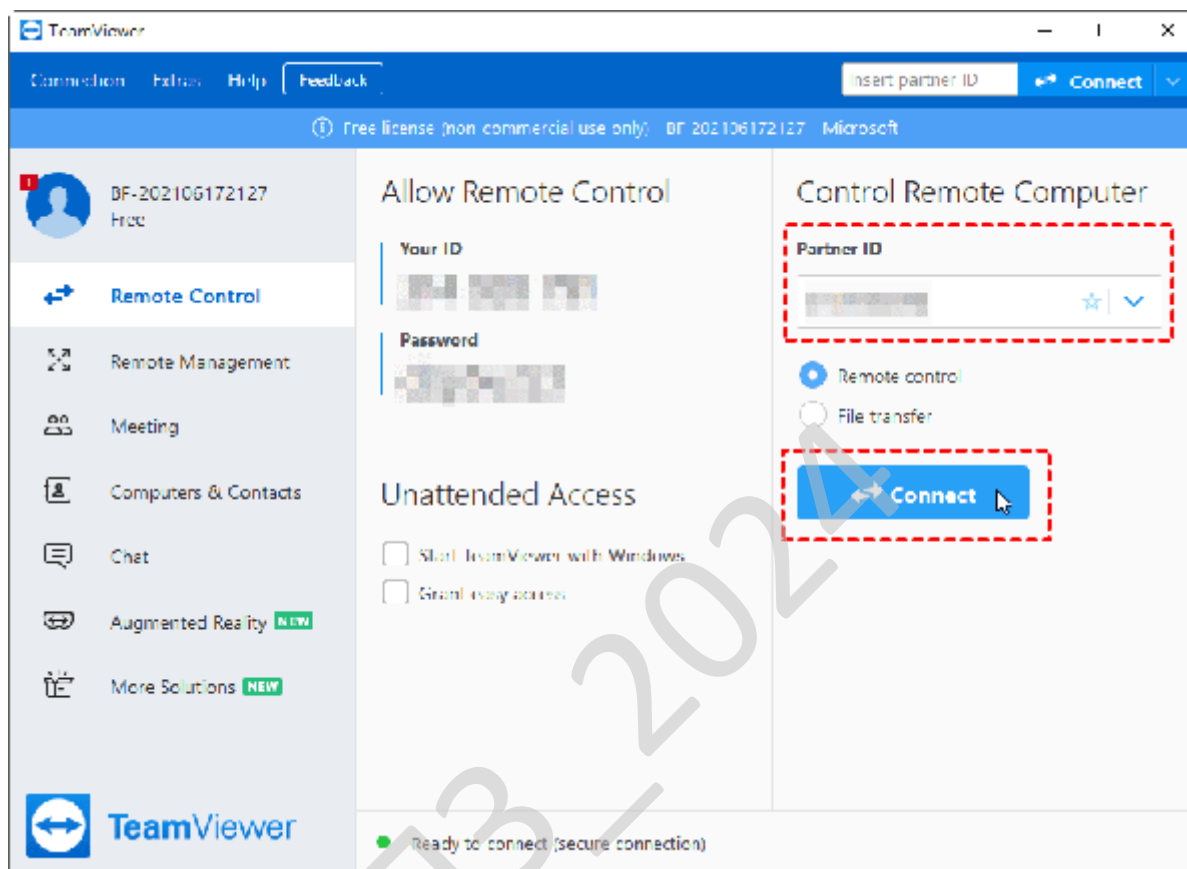


Рисунок 2.3 – Інтерфейс користувача TeamViewer QuickSupport

Vysor

Vysor, зручна програма для віддзеркалення екрана та дистанційного керування, виходить у центр уваги, забезпечуючи безперебійне відображення та керування екраном пристрою Android безпосередньо з комп'ютера. Цей універсальний інструмент обслуговує як USB, так і бездротові підключення, надаючи користувачам гнучкість вибору методу, який найкраще відповідає їхнім уподобанням і потребам.

Крок 1. На своєму пристрої Android перейдіть у налаштування. Виберіть «Про телефон» і натисніть «Номер збірки» сім разів, відкриваючи приховані

						ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата			12

параметри розробника.

Крок 2. У «Параметрах розробника» перемістіть перемикач на «**USB Debugging**».

Крок 3. Перейдіть до Google Play Store на своєму Android, знайдіть «**Vysor**» і встановіть цю програму.

Крок 4. На ПК перейдіть на офіційний веб-сайт Vysor. Завантажте версію Windows і виконайте процес встановлення.

Крок 5. Відкрийте Vysor на своєму ПК та увійдіть, використовуючи облікові дані свого облікового запису Google.

Крок 6. Встановіть фізичне з'єднання між пристроєм Android і ПК за допомогою кабелю USB, а потім запусіть програму Vysor на своєму Android.

Крок 7. Натисніть «**Переглянути**», щоб переглянути дзеркальний екран.

Крок 8. Станьте свідками магії, коли екран вашого Android відображається на вашому комп'ютері, безперервно взаємодіючи з вашим Android за допомогою миші та клавіатури комп'ютера.

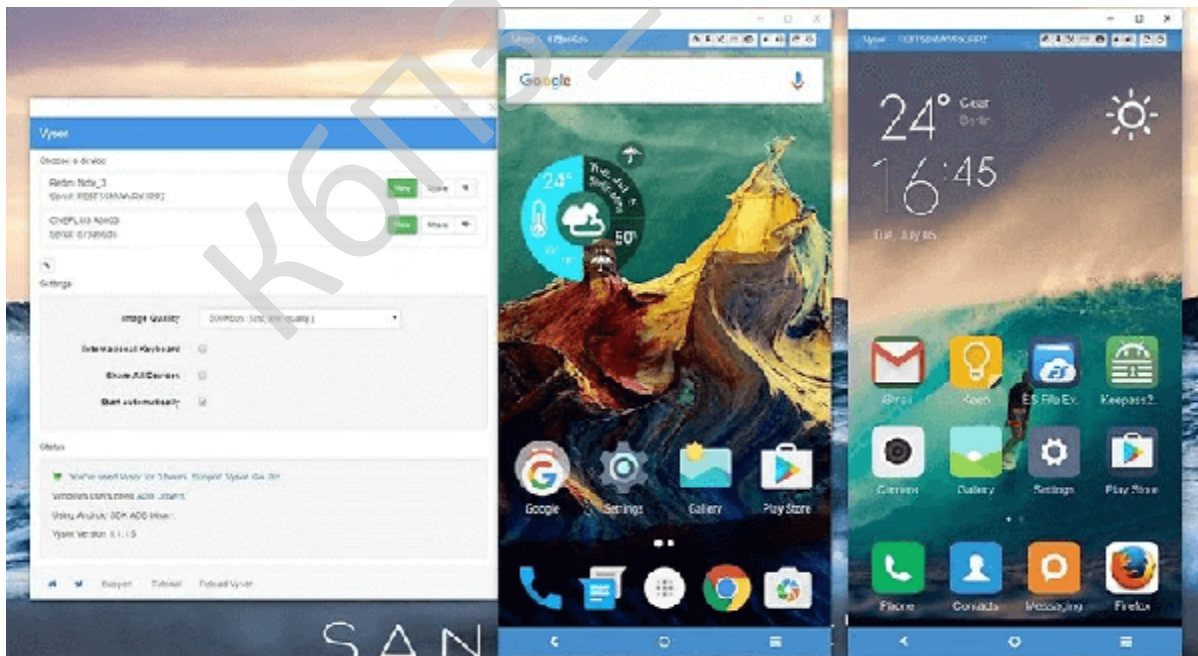


Рисунок 2.4 – Інтерфейс користувача Vysor

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		13

ApowerMirror

ApowerMirror виступає як програма дистанційного керування Android, яка доводить перевагу для віддзеркалення екрана вашого пристрою Android на більший дисплей, наприклад настільний комп'ютер або ноутбук. Це позбавляє від необхідності встановлення емуляторів для роботи програм Android або відображення вмісту на великому екрані.

Крок 1. Завантажте та встановіть програму з магазину Google Play на свій мобільний пристрій і відповідну версію Windows на ПК.

Крок 2. Переконайтеся, що ваш телефон і ПК підключені до однієї мережі Wi-Fi.

Крок 3. Запустіть програму на своєму пристрої Android і натисніть піктограму дзеркала, щоб знайти доступні пристрої.

Крок 4. Знайдіть ім'я свого ПК, а потім виберіть «Дзеркало телефону на ПК».

Крок 5. Натисніть «LocalCast».

Крок 6. Розпочніть процес дзеркального відображення, натиснувши «Почати зараз», без особливих зусиль проєктуючи екран свого Android на ПК.

Scrcpy

Scrcpy – це програмне забезпечення для віддаленого доступу та керування Android, створене на платформі з відкритим кодом. Ця програма дзеркального відображення екрана дає змогу отримати віддалений доступ до вашого пристрою Android із комп'ютера чи робочого столу. Scrcpy підтримує різні функції, дозволяючи відкривати програми на віддаленому Android, обмінюватися файлами тощо на кількох пристроях.

Крок 1. На своєму пристрої Android перейдіть у «Налаштування», виберіть «Про телефон» і натисніть «Номер збірки» сім разів, щоб відкрити параметри розробника. У параметрах розробника ввімкніть «Налагодження USB».

Крок 2. Отримайте Інструменти платформи Android SDK які відповідають

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		14

вашій операційній системі, завантаживши їх.

Крок 3. Встановіть фізичне з'єднання між пристроєм Android і ПК за допомогою кабелю USB.

Крок 4. Завантажте Scrcpy з офіційного репозиторію GitHub. Розпакуйте завантажений файл ZIP і запустіть файл scrcpy.exe, щоб розпочати встановлення.

Крок 5. Відкрийте командний рядок і введіть «**scrcpy**», а потім натисніть Enter.

Крок 6. Екран комп'ютера тепер безперешкодно віддзеркалюватиме ваш пристрій Android, надаючи вам пряме керування ним із комп'ютера.

Web Key

Webkey виділяється як універсальна платформа дистанційного керування та доступу Android, яка пропонує можливості для керування, віддзеркалення екранів і безперешкодної передачі даних через Інтернет. Примітно, що для доступу до телефону Android не потрібно встановлювати програмне забезпечення для ПК, що робить його зручним вибором. Webkey, особливо корисний для рутованих пристроїв, надає низку функцій, включаючи віддалені знімки екрана, запис екрана, копіювання та можливість перегляду екрана телефону безпосередньо на робочому столі, серед інших функцій.

Крок 1. Відвідайте офіційний веб-сайт і натисніть «**Спробувати безкоштовно**». Зареєструйте свої дані, щоб отримати доступ до інструменту.

Крок 2. Дотримуйтеся вказівок на екрані, щоб створити пару між інструментом і пристроєм Android. Для цього потрібно завантажити програму з Google Play і ввести наданий код сполучення. Крім того, ви можете скористатися наданим посиланням для сполучення та надіслати його електронною поштою, щоб легко підключити обидва пристрої.

Крок 3. Встановіть підключення та насолоджуйтеся зручністю віддаленого керування своїми пристроями.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		15

DeskDock

DeskDock надає перевагу простоті над віддзеркаленням екрана, надаючи унікальну функціональність. Ця програма дозволяє інтегрувати вашу комп'ютерну мишу безпосередньо на ваш пристрій Android, пропонуючи повний контроль. Особливо корисний на етапі тестування, DeskDock стає ідеальним інструментом для розробника, що полегшує швидке тестування різноманітних функцій завдяки легкому використанню комп'ютерної миші.

Крок 1. Почніть із завантаження, встановлення та відкриття DeskDock на комп'ютері та пристрої Android.

Крок 2. Використовуйте USB, щоб підключити телефон Android до ПК. Переконайтеся, що необхідні дозволи надано, перейшовши до «**Номер збірки**» > «**Параметри розробника**» > «**Налагодження USB**» на своєму пристрої Android.

Крок 3. На сторінці стану натисніть «**Підключити**», щоб встановити з'єднання.

Крок 4. Після підключення насолоджуйтеся плавним переміщенням курсора миші комп'ютера по екрану пристрою Android. Крім того, клавіатура вашого комп'ютера має бездоганно працювати з вашим пристроєм Android.

Samsung Flow

Samsung Flow виділяється як оптимальний вибір для комплексного керування вашим телефоном Samsung з ПК. Цей інтуїтивно зрозумілий інструмент плавно інтегрує всі ваші продукти Samsung, спрощуючи легкий обмін повідомленнями та налаштуваннями. Незалежно від того, передаєте ви тексти на свій Galaxy Tab чи віддзеркалюєте повідомлення на своєму ноутбучі чи Galaxy Chromebook, Flow спрощує ці процеси за лічені секунди. Важливо те, що для Flow не потрібен ноутбук Samsung, що забезпечує сумісність із різноманітними пристроями.

Крок 1. Переконайтеся, що ваш смартфон і комп'ютер підключено до однієї мережі.

Крок 2. Завантажте та встановіть програму Samsung Flow на свій

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16

смартфон Samsung з офіційного магазину програм.

Крок 3. Надайте необхідні права, дотримуючись інструкцій на екрані.

Крок 4. Завантажте та встановіть програму Samsung Flow на свій комп'ютер.

Крок 5. Відкрийте програму та натисніть «Пуск».

Крок 6. Ваш комп'ютер автоматично ідентифікує ваш телефон у тій же мережі. Натисніть на розпізнаний пристрій.

Крок 7. Підтвердьте відповідний ключ доступу на телефоні та комп'ютері, а потім натисніть «ОК» на обох пристроях.

Крок 8. Виконавши ці кроки, тепер ви готові легко керувати своїм телефоном Android із ПК за допомогою Samsung Flow.

Висновок

Підсумовуючи, пошук потрібного програмного забезпечення дистанційного керування для телефону Android має важливе значення для підвищення вашої продуктивності та зручності. У цьому розділі розглядаються дев'ять різних підходів, починаючи від настійно рекомендованого AnyViewer і закінчуючи універсальним ApowerMirror, пропонуючи детальні вказівки щодо дистанційного керування Android. Незалежно від того, чи є ви ентузіастом технологій чи вам потрібно керувати своїм пристроєм Android для професійних цілей, ці рішення пропонують безпечні та зручні способи дистанційного керування телефоном. AnyViewer займає центр уваги завдяки своїй безкоштовній природі та адаптивності. Окрім програмного забезпечення дистанційного керування, воно розширює свої можливості до віддаленого доступу без нагляду, передачі файлів, підтримки кількох моніторів тощо. Для тих, хто шукає всеосяжне програмне забезпечення, AnyViewer є оптимальним вибором.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		17

2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

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

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

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

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

- у мови логічний синтаксис, завдяки чому вихідний код легко читати та розуміти;
- гнучкість та масштабованість Python дозволяє адаптувати високорівневу логіку та розширяти складні застосунки, як тільки виникне така необхідність;

- розробка на Python у більшості випадків проходить швидше, ніж на інших мовах програмування;
- Python – інтерпретована мова програмування. Це значить, що код можна написати у будь-якому текстовому файлі на будь-якій платформі, і потім успішно запустити;
- у Python – колосальна спільнота однодумців. Тож будь-які складнощі конкретних розробників вирішуються колективно.

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

Мова програмування Python використовується у наступних сферах:

1. Розробка програмних застосунків будь-якого напрямку.
2. Розробка серверної частини мобільних застосунків (найпопулярніший напрямок).
3. Ігри. Багато сучасних ігор для комп’ютерів (наприклад, World of Tanks) частково чи повністю написані на Python.
4. Вбудовані системи для різних пристроїв. Дуже часто Python використовують для написання внутрішніх платформ управління банкоматами.
5. Скрипти та плагіни до уже реалізованих програм для автоматизації процесів чи створення інших рішень.
6. Тестування (автоматизація цього процесу).
7. Машинне навчання. – основна мова для написання алгоритмів і аналітичних застосунків у сфері Machine Learning.

Бібліотеки Python

Різні бібліотеки Python використовують для виконання конкретних завдань. Наприклад, Matplotlib підходить для відображення даних у двовимірній та тривимірній графіці. Pandas підходить для зручної роботи з даними. NumPy дозволяє створювати масиви та керувати ними. Requests використовується для

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

веброзробки. OpenCV-Python відкриває можливості для обробки зображень з метою оптимізації систем «машинного зору».

Найвідоміші фреймворки для мови програмування Python

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

Серед найпопулярніших фреймворків для Python:

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

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

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

Популярні Python IDE

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

2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за другим (магістерським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи віддаленого керування мобільними пристроями під керуванням ОС Android.

В процесі розробки випускної кваліфікаційної роботи за другим

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

(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-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		21

3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

3.1 Опис функціонування системи

Архітектура Android

Для поняття основних принципів взаємодії компонентів платформи досить глянути на схему нижче.

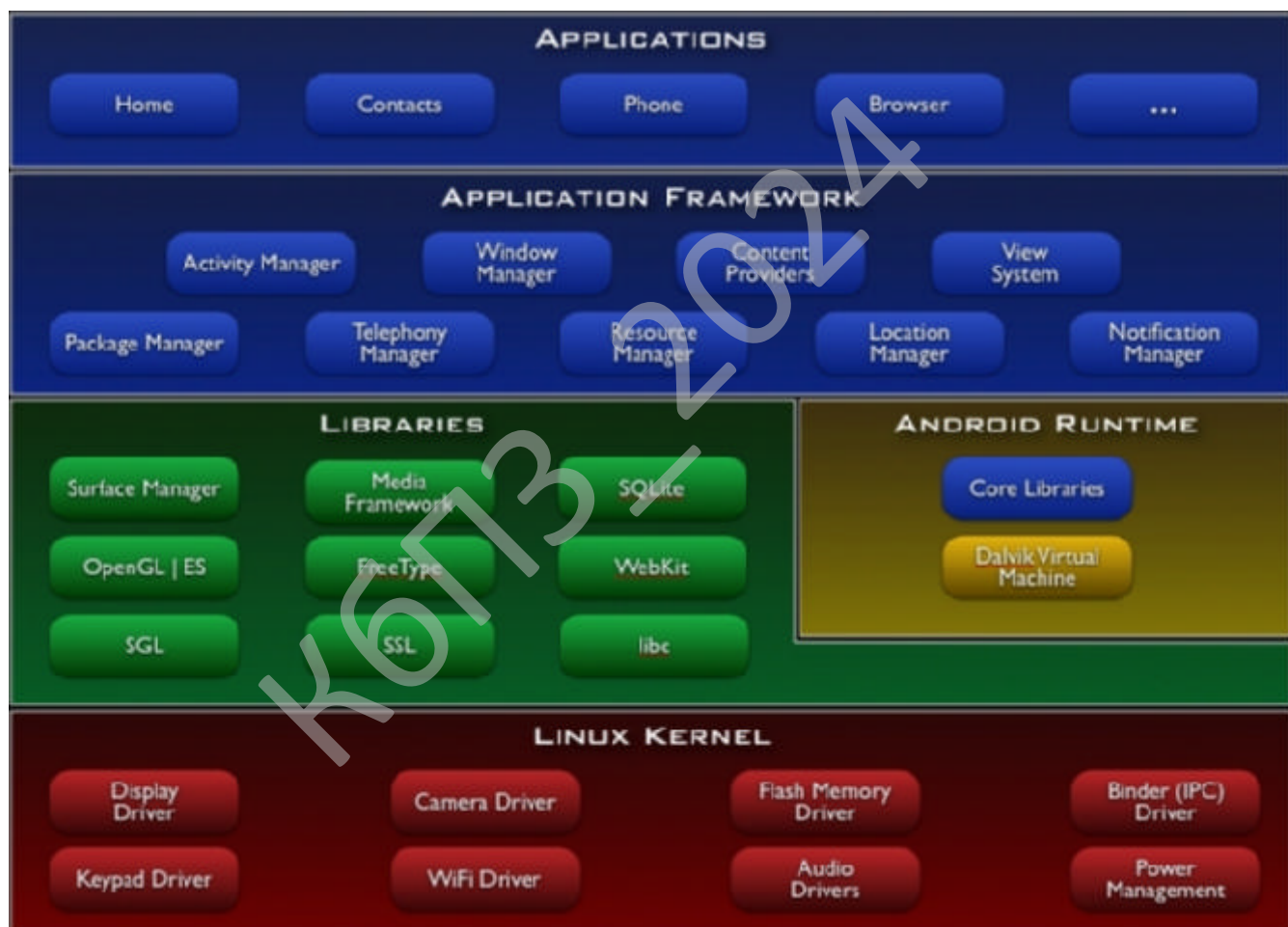


Рисунок 3.1 – Основні принципи взаємодії компонентів платформи

Додатки

Платформа Android поставляється з безліччю стандартних додатків, таких як SMS клієнт, Телефон, Календар, Карти (Google maps) і багато хто інші. Всі додатки написані, використовуючи язык програмування Java і запускаються в Dalvik – убудованій віртуальній машині.

Розроблювачі можуть створювати свої власні додатки, використовуючи Android SDK.

Функції Android

Контекстні підказки:

– Контекст від Now. Дізнавайтеся потрібну інформацію, не закриваючи додаток або сайт. Просто натисніть і втримуйте кнопку "Головний екран".

– Голосове керування: спілкуйтеся з додатками, які підтримують нові функції. Наприклад, якщо ви скажете "слухати музику", додаток уточнить, який жанр вас цікавить.

– Direct Share: можливість миттєво ділитися контентом із друзями в додатках.

Батарея:

– Коли ви не користуєтесь пристроєм, автоматично включається сплячий режим, щоб зберегти заряд батареї.

– Функція App Standby обмежує витрату батареї рідко використовуваними додатками, і ваш пристрій працює довше.

– Підтримка USB Type-C для швидкої зарядки й передачі даних по одному кабелю.

Особисті дані:

– Додатка для Android 6.0 запитують доступ до тих або інших даних і функціям тільки тоді, коли він потрібний. Крім того, ви зможете використовувати додаток, навіть якщо відхилите запит.

– Включайте й відключайте дозволу для всіх установлених додатків.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

– Якщо убудоване ПЗ й ОС Android відрізняються від вихідних заводських версій, то при запуску пристрою з'явиться попередження.

– Миттєво розблокуйте екран, підтверджуйте транзакції в додатках і оплачуйте покупки в Google Play або звичайних магазинах за допомогою сканера відбитка пальця*.

Android Runtime (ART):

– Робота в багатозадачному режимі стала ще швидше завдяки підвищенню продуктивності додатків і зниженню витрати пам'яті.

Робота:

– Підтримка стилуса Bluetooth™ із чутливістю до сили натискання й клавішами-модифікаторами.

– Поліпшене оформлення й відображення тексту.

– Поліпшена функція виділення тексту, зручне розташування кнопок скасування, повтору й форматування тексту.

– Дії з виділеним текстом, включаючи миттєвий переклад на іншу мову (необхідний додаток Google Перекладач).

– Економія паперу завдяки можливості двостороннього друку.

Зручність використання:

– Посилання в додатках: установлені додатки автоматично відкриваються замість мобільних сайтів при переході по посиланнях.

– Змінюйте параметри режиму "Не турбувати" прямо на панелі швидких налаштувань.

– Не пропускайте термінових повідомлень: дозвольте додзвонитися в режимі "Не турбувати" тим, хто намагався зв'язатися з вами двічі за 15 хвилин.

– Режим "Не турбувати" може включатися автоматично за вашими правилами або під час заходів з календаря.

– Управляйте гучністю музики, сигналу будильника й інших повідомлень або відключайте звук за допомогою кнопок регулювання гучності.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		24

– Єдиний розділ для налаштування всіх параметрів додатків – від використання пам'яті й витрати батареї до повідомлень і дозволів.

– У меню додатків Google Старт доданий пошук, швидке прокручування за алфавітом і розділ додатків, що рекомендуються.

Підключення:

– Сканування маячків і аксесуарів менше розряджає батарею завдяки застосуванню Bluetooth Low Energy (BLE).

– Швидке й надійне підключення до мереж Wi-Fi з технологією Hotspot 2.0

– Профіль Bluetooth SAP дозволяє дзвонити з автомобільного телефону, використовуючи SIM-Карту вашого пристрою.

– Для точки доступу Wi-Fi додана підтримка діапазонів частот 5 ГГц.

Розширюване сховище:

– Використовувати SD-Карти й зовнішні запам'ятовувальні пристрої як додаткові захищені сховища для додатків і ігор в Android 6.0 стало ще простіше.

Налаштування пристрою й перенесення даних:

– Легке копіювання аккаунтів, додатків і даних на новий пристрій.

– При налаштуванні пристрою ви можете додати ще один особистий або корпоративний поштовий аккаунт, наприклад із протоколом IMAP.

– Автоматичне резервне копіювання даних додатків для їхнього швидкого відновлення.

– Резервне копіювання й відновлення додаткових системних налаштувань, у тому числі параметрів синхронізації, що воліються додатків, налаштувань режиму "Не турбувати", налаштувань спеціальних можливостей і включених способів уведення.

Мультимедіа:

– Підтримка MIDI: створюйте, слухайте й відтворюєте музику, використовуючи телефони й планшети Android разом з USB і програмними MIDI-пристроями, а також пристроями з підтримкою MIDI через Bluetooth LE.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25

Локалізація:

– Операційна система Android доступна більш ніж на 74 мовах, включаючи азербайджанський, албанський, гуджарати, казахський, узбецький і урду.

Android for Work:

– При вхідних викликах або при перегляді повідомлень вам доступні контактні дані з робочого профілю, навіть якщо ви в нього не ввійшли.

– Коли ви використовуєте додаток з робочого профілю, у рядку стану з'являється значок портфеля. Якщо розблокувати екран з додатком з робочого профілю, з'явиться попередження.

– Додатка для VPN перебувають у розділі "Налаштування > Ще > Налаштування VPN". Повідомлення VPN розрізняються при налаштуванні VPN для всього пристрою або окремого робочого профілю.

Функціонал додатка, що розробляється:

– Передача файлів між пристроями Android і комп'ютерами. Тепер з підтримкою drag&drop.

– Вирізання, копіювання, вставка, пошук, перейменування або видалення файлів на карті пам'яті.

– Одержання, відправлення, пересилання або видалення SMS повідомлень.

– Установка, видалення, резервне копіювання, пошук додатків і збереження їх у вигляді .apk – файлів.

– Попередній перегляд, видалення, завантаження, завантажування, установка як фоновий рисунок і слайд-шоу з фотографій з робочого стола.

– Редагування тексту в буфері обміну пристрою через комп'ютер.

– Угруповання, пошук, створення контактів, перегляд і видалення журналів викликів.

– Пошук, перегляд, завантаження, завантажування, видалення й налаштування мелодій викликів, повідомлень і будильника.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		26

- Відтворення, пошук, відновлення, видалення, або установка як мелодії викликів, повідомлень і будильника.
- Безпечна робота навіть у суспільної Wi-Fi мережі.
- Перегляд екрана Android у реальному часі (експериментальна функція, необхідні права root).
- Перегляд екрана вашого Android, зняття "знімків екрана" або запис відеороликів.
- Програма призначена для багатозадачності. Переміщайте додатки між робочими столами, щоб краще організувати свою роботу.
- Додано зображення для більшості пристроїв, різні віджети (календар, погода, відкриття посилань, буфер обміну й т.п.), часто використовуємі контакти.
- Перегляд стану убудованої пам'яті телефону, карти пам'яті, акумулятора, процесора й оперативної пам'яті в реальному часі й графіки для демонстрації доступних, використовуваних, загальних ресурсів і кнопкою для швидкої оптимізації.
- Зупинка або видалення працюючих додатків, підтримується пакетна обробка.
- Видалення, відправлення друзям і перегляд інформації про користувальницькі й системні додатки.
- Керування, вирізання, копіювання, перейменування, відправлення друзям, видалення, сортування й створення видимих або схованих папок і файлів.

3.2 Розробка структурної схеми

У сучасному швидкоплинному цифровому світі можливість віддаленого доступу, перегляду та керування телефонами та планшетами Android стає все більш важливою. Якщо ви вирішуєте проблеми, надаєте технічну підтримку чи просто дистанційно керуєте своїми пристроями, рішення віддаленого доступу пропонують неперевершену зручність і гнучкість.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

Дистанційне керування Android – це рішення, яке дозволяє користувачам отримувати доступ до своїх пристроїв Android і керувати ними з віддаленого місця. Ця технологія забезпечує повний контроль над пристроєм, включаючи перегляд екрана, навігацію програмами та виконання завдань, ніби пристрій у вас у руках.

Використовуючи розроблену систему, ви можете дистанційно переглядати екрани Android-пристроїв у режимі реального часу та віддалено отримувати доступ і керувати Android-пристроями. З розробленою системою ви можете:

- Віддалений доступ/перегляд телефонів і планшетів під керуванням Android.

- Дистанційно керуйте пристроями Android під управлінням Android 8.0 або новішої версії, такими як Asus, Essential, Google Pixel, OnePlus, HTC, Huawei, Honor, Motorola, Nokia, OPPO, Sony, Xiaomi та ZTE.

- Дистанційно керуйте деякими додатковими пристроями Android Samsung, Lenovo, навіть якщо на них не встановлено Android 8 або новішої версії.

- Віддалений доступ до комп'ютерів Windows і Mac зі своїх пристроїв Android.

Як віддалено переглядати екрани планшета Android і телефону Android

Зареєструйтеся в розробленій системі щоб дистанційно переглядати пристрій Android за допомогою комп'ютера з Windows або Mac або іншого пристрою Android. Ви можете користуватися послугою з безкоштовною пробною версією або за недорогою платною підпискою.

Розроблена система має велику цінність і коштує набагато дешевше, ніж інші інструменти підтримки з обслуговування. Він заснований на перевірених технологіях віддаленого доступу.

Функції доступу до мобільного пристрою не включені в безкоштовну некомерційну версію розробленої системи, оригінальну версію розробленої системи або віддалену підтримку розробленої системи. Ви повинні придбати

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

розроблену систему, якщо хочете мати віддалений доступ до мобільних пристроїв.

Завантаження програми розробленої системи на пристрій, який ви хочете переглядати або керувати

Попросіть вашого кінцевого користувача перейти за посиланням для завантаження програми розроблену систему (як показано у вашій програмі розроблену систему) зі свого пристрою Android, щоб завантажити, запустити та встановити програму розробленої системи.

Коли ви (або ваш клієнт) запускаєте програму розробленої системи на пристрої Android, вона відобразить код сеансу, який ви введете на екрані, щоб віддалено переглядати цей пристрій. Користувачам пристроїв з ОС Android 8 або новішої версії буде запропоновано увімкнути спеціальні можливості в Android, щоб дозволити віддалений доступ.

Користувачам певних пристроїв (Lenovo) може бути запропоновано також завантажити доповнення з Google Play Store, щоб додати віддалений доступ до певних сімейств пристроїв Android.

Введіть код у програмі розроблену систему і почніть дистанційний перегляд екрана Android

Запустіть програму розробленої системи на своєму комп'ютері та натисніть піктограму розробленої системи (знак плюс) у верхній частині програми, щоб відкрити екран підключення розроблену систему. Введіть дев'ятизначний код, який відображається в програмі розроблену систему на пристрої Android вашого клієнта. Після натискання стрілки розпочнеться сеанс.

На цьому етапі ви зможете наживо переглянути їхній екран Android у вікні на екрані свого комп'ютера.

Використання елементів керування переглядом

Натисніть стрілку у верхній частині вікна дзеркального екрана пристрою, щоб отримати доступ до елементів керування. Кнопки:

– Відключити.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

- Назад (керування пристроєм).
- Головна (керування пристроєм).
- Меню (керування пристроєм).
- Якість потокової передачі FPS.
- Перемикач масштабованого/оригінального розміру.
- Повний екран.
- Передача файлів

Ви також можете зробити друк екрану Android за допомогою стандартних комбінацій клавіш для друку екрану для комп'ютера Windows або Mac.

Відключення сеансу

Після завершення сеансу ви можете закрити вікно перегляду на своєму комп'ютері. Сеанс можна завершити на пристрої, який ви переглядаєте, натиснувши кнопку X відключення.

Поради щодо покращення роботи дистанційного керування Android

Покращення досвіду дистанційного керування Android може призвести до більш плавного та ефективнішого керування пристроєм. Нижче наведено кілька практичних порад, які допоможуть вам отримати максимальну віддачу від сеансів віддаленого доступу:

- Забезпечте стабільне підключення до Інтернету: надійне та стабільне підключення до Інтернету має вирішальне значення для безперебійного дистанційного керування. Переконайтеся, що ваш пристрій Android і пристрій дистанційного керування підключено до високошвидкісного Інтернету, щоб мінімізувати затримки та перебої.
- Оновлюйте свої пристрої: регулярно оновлюйте свій пристрій Android і розроблену систему до останніх версій.
- Використовуйте елементи керування клавіатурою та мишею: використовуйте елементи керування клавіатурою та мишею, які пропонує розроблену систему. Ці інструменти можуть значно покращити навігацію та взаємодію, полегшуючи дистанційне керування пристроєм Android.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		30

– Надайте пріоритет безпеці: застосуйте надійні заходи безпеки, щоб захистити свої віддалені сеанси. Увімкніть двофакторну автентифікацію, щоб захистити свій обліковий запис.

– Ознайомтеся з комбінаціями клавіш: вивчіть і використовуйте комбінації клавіш і швидкі команди, доступні в розроблену систему

Дотримуючись цих порад, ви зможете покращити роботу дистанційного керування Android, забезпечуючи плавне, безпечне та ефективне керування пристроєм з будь-якого місця.

Вирішення питань безпеки та конфіденційності у віддаленому доступі Android:

– Використовуйте сувору автентифікацію: застосуйте багатофакторну автентифікацію (MFA) і надійні унікальні паролі, щоб гарантувати, що лише авторизовані користувачі можуть отримати віддалений доступ до ваших пристроїв Android.

– Контроль доступу: обмежте віддалений доступ лише необхідному персоналу та переконайтеся, що дозволи встановлено належним чином, щоб запобігти несанкціонованому доступу.

– Моніторинг активності: регулярно відстежуйте сеанси віддаленого доступу та ведіть журнали для виявлення будь-якої підозрілої активності або спроб неавторизованого доступу.

– Впровадьте політики безпеки: розробіть і запровадьте політики безпеки, які окреслюють найкращі практики для віддаленого доступу, включаючи керування паролями, протоколи автентифікації та вказівки щодо використання пристрою.

– Навчайте користувачів: навчайте користувачів щодо важливості безпеки та конфіденційності під час віддаленого доступу, переконавшись, що вони розуміють, як захистити свої пристрої та дані.

Дотримуючись цих найкращих практик, ви можете значно підвищити безпеку та конфіденційність своїх сеансів віддаленого доступу Android.

На рисунку 3.2 зображена структурна схема віддаленого управління ЕОМ у загальному випадку. З цієї схеми ми бачимо, що усі віддалені користувачі зв'язуються один з одним за допомогою RAS (серверів віддаленого управління доступом), модемів та маршрутизаторів.

Мобільний пристрій під керівництвом ОС Android, яким віддалено керують

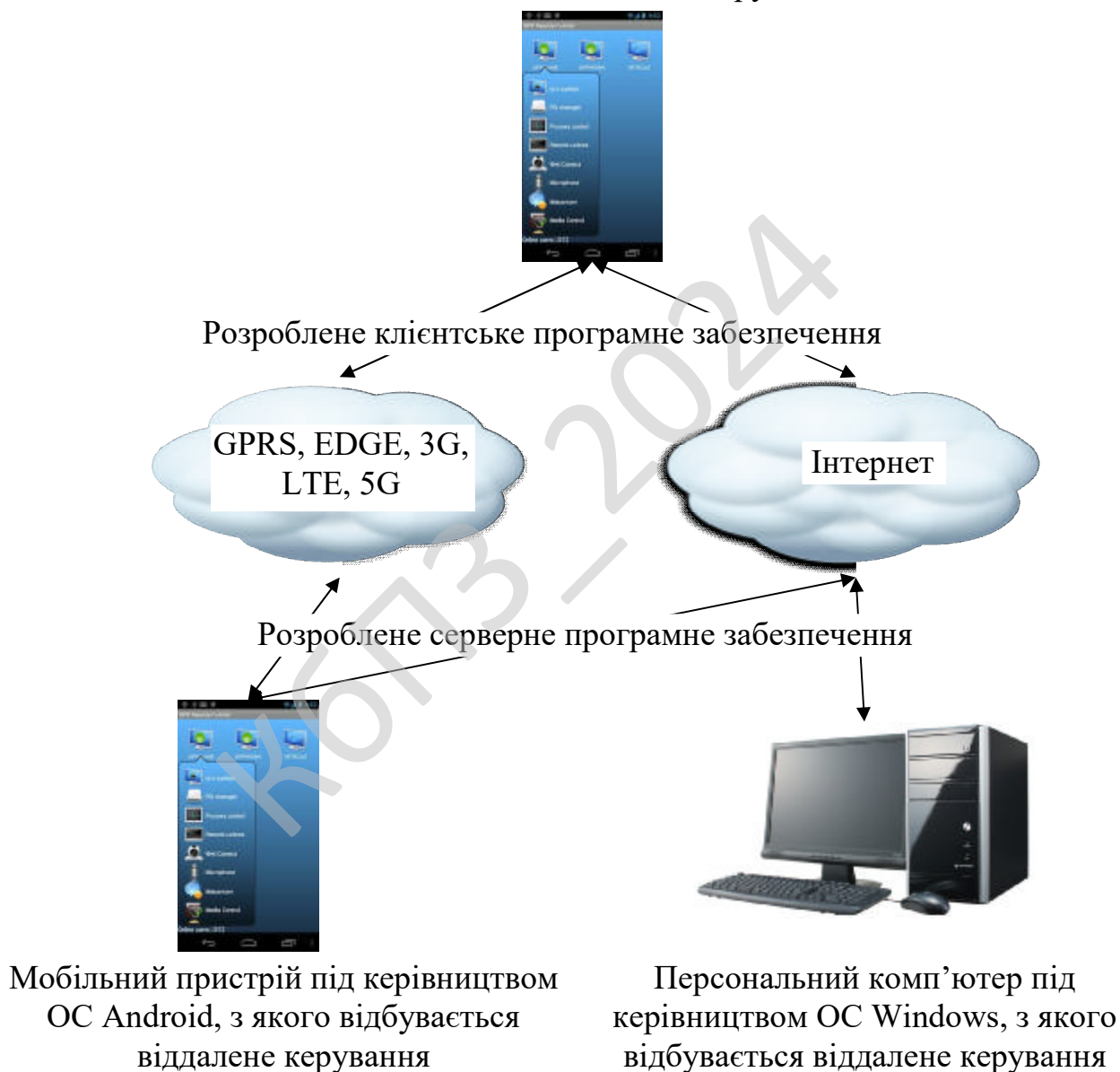


Рисунок 3.2 – Структурна схема системи

Схеми віддаленого керування наведені на рисунку 3.2, відрізняються типом взаємодіючих систем:

- термінал – мобільний пристрій під керівництвом ОС Android;
- мобільний пристрій під керівництвом ОС Android – мобільний пристрій під керівництвом ОС Android;
- мобільний пристрій під керівництвом ОС Android – мережа;
- мережа – мережа.

Відповідно при реалізації різних типів взаємодіючих систем використовується різне апаратне обладнання.

3.3 Розробка функціональної схеми

На рисунку 3.3 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

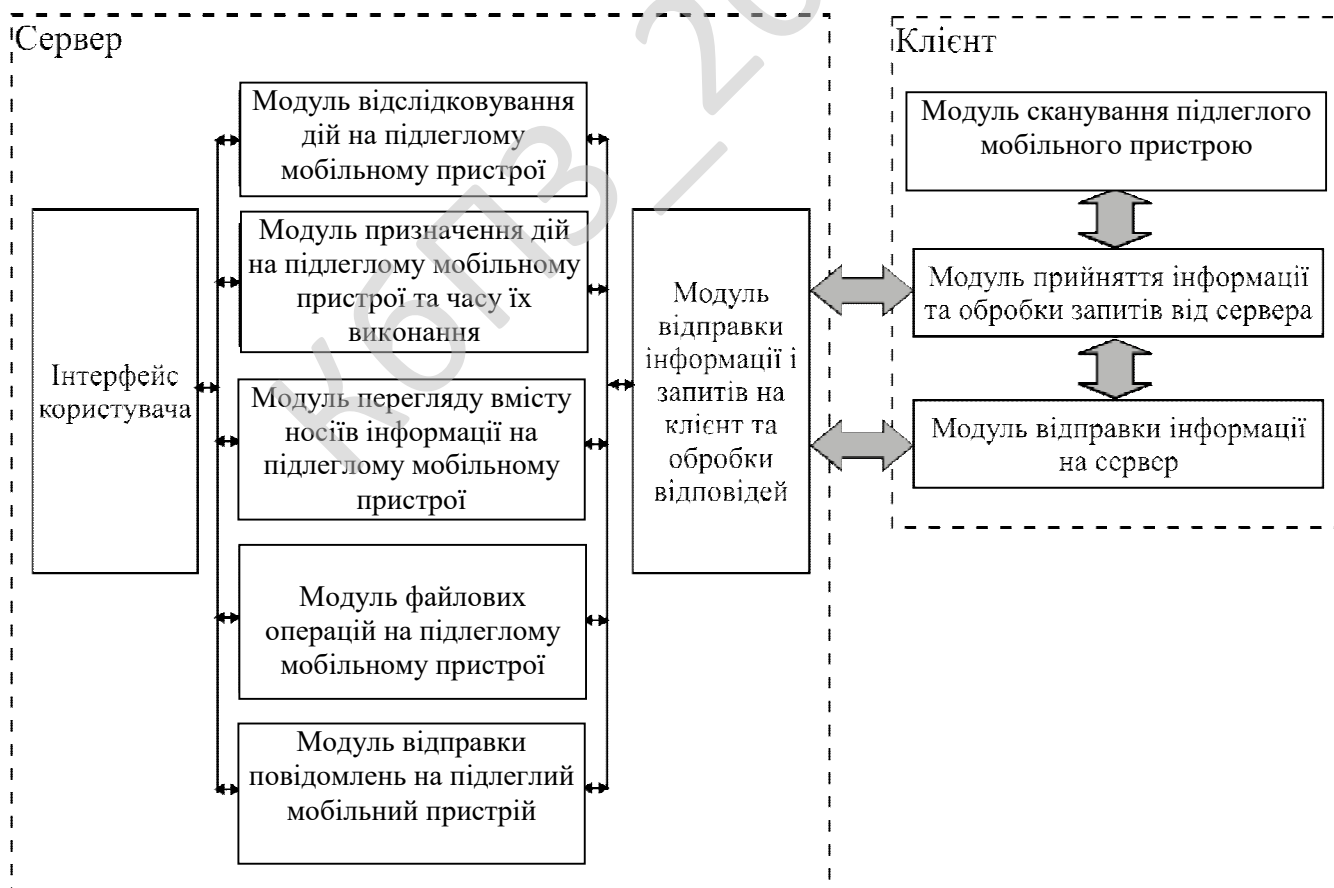


Рисунок 3.3 – Функціональна схема системи

З рисунка ми бачимо, що розроблена система функціонально складається з двох основних блоків:

- серверної частини;
- клієнтської частини.

Розглянемо ці функціональні частини більш докладно.

Серверна частина включає в себе наступні модулі.

Інтерфейс користувача, який дозволяє користувачу наочно працювати з програмою використовуючи, ті або інші функціональні модулі. До цих модулів відносяться наступні:

- Модуль відслідковування подій на підлеглому мобільному пристрої під керівництвом ОС Android.
- Модуль призначення дій над підлеглим мобільним пристроєм під керівництвом ОС Android та часу їх виконання.
- Модуль перегляду вмісту дисків підлеглому мобільного пристрою під керівництвом ОС Android.
- Модуль виконання файлових операцій (завантаження, відправка, видалення файлів).
- Модуль відправки повідомлень на підлеглий мобільний пристрій під керівництвом ОС Android.

Усі вищеперераховані модулі, взаємодіють з однієї сторони з інтерфейсом користувача, а з іншої сторони з модулем відправки інформації запитів на клієнт та обробки відповідей. Цей модуль взаємодіє вже безпосередньо з клієнтською частиною розробленої системи.

Відповідно перейдемо до розгляду клієнтської частини розробленої програми віддаленого управління мобільним пристроєм під керівництвом ОС Android через мережу.

Клієнтська частина складається з наступних функціональних блоків:

- Модуль прийняття інформації та обробки запитів від серверу.
- Модуль відправки інформації на сервер.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

– Модуль сканування підлеглого мобільного пристрою під керівництвом ОС Android.

Перший та другий модуль безпосередньо взаємодіють з серверною частиною, а точніше з таким її функціональним блоком, як модуль відправки інформації і запитів на клієнт та обробки відповідей.

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

3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.4. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій системі.

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.

Діаграми потоків даних містять чотири типи елементів:

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

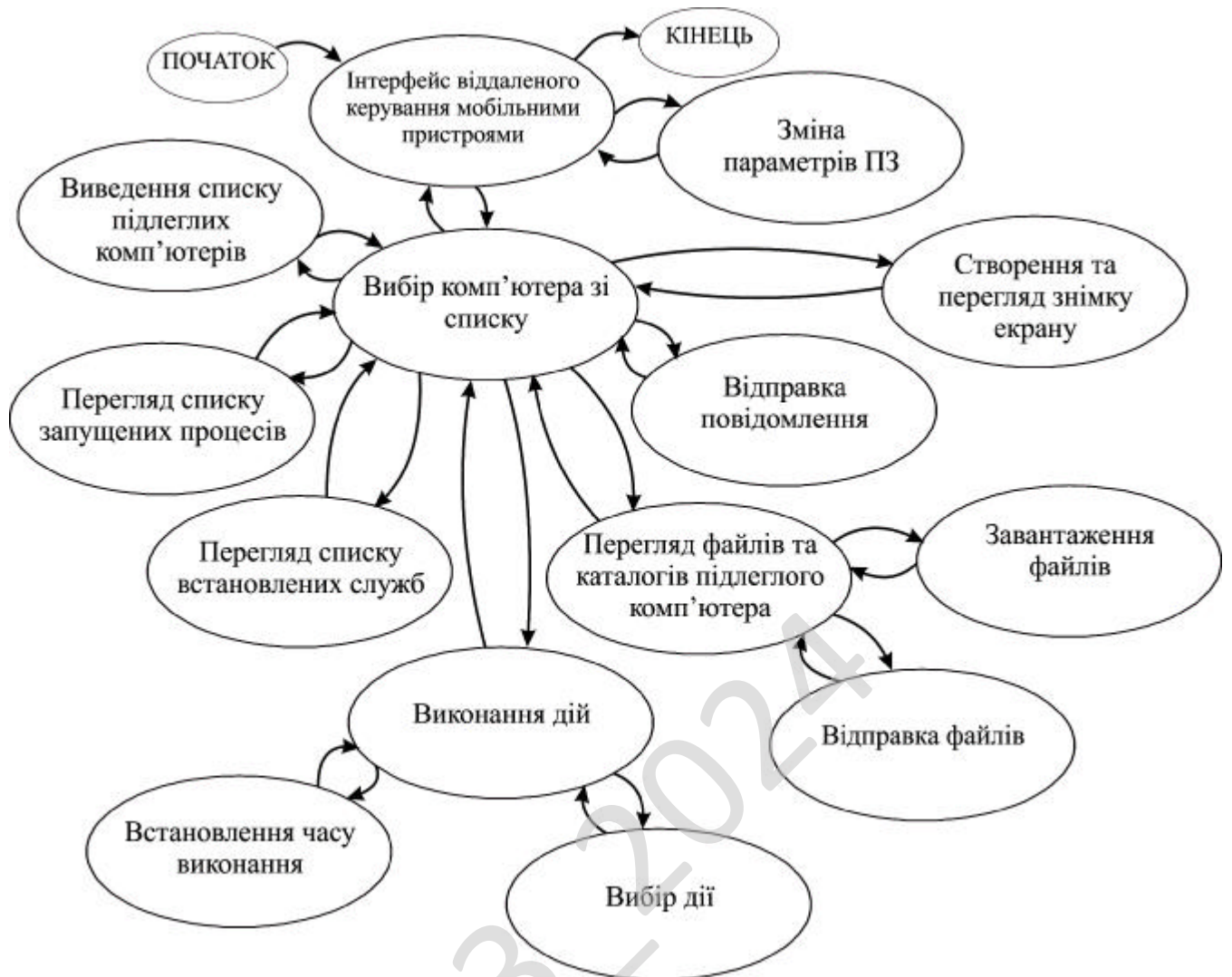


Рисунок 3.4 – Діаграма взаємодії процесів

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

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

4.1 Блок-схеми та опис алгоритмів функціонування системи

Під час роботи над магістерською дипломною роботою було створено блок-схеми. Перед їх розглядом необхідно провести роз'яснення який саме тип блок-схем використовується.

Блок-схема це представлення задачі для її аналізу або розв'язування за допомогою спеціальних символів (геометричних образів), які позначають такі елементи, як операції, потік, дані тощо. Блок вхідних та вихідних даних прийнято позначати паралелограмом, блок обчислень (обробки) даних – прямокутником, блок прийняття рішень – ромбом, еліпсом – початок та кінець алгоритму.

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

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

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

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

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		37

Термінатор це елемент відображає вхід із зовнішнього середовища або вихід з неї (найчастіше застосування – початок і кінець програми). Всередині фігури записується відповідна дія.

Процес це виконання однієї або кількох операцій, обробка даних будь-якого виду (зміна значення даних, форми подання, розташування). Всередині фігури записують безпосередньо самі операції.

Рішення це показує рішення або функцію перемикального типу з одним входом і двома або більше альтернативними виходами, з яких тільки один може бути обраний після обчислення умов, визначених всередині цього елемента. Вхід в елемент позначається лінією, що входить зазвичай у верхню вершину елемента. Якщо виходів два чи три то зазвичай кожен вихід позначається лінією, що виходить з решти вершин (бічних і нижній). Якщо виходів більше трьох, то їх слід показувати однією лінією, що виходить з вершини (частіше нижній) елемента, яка потім розгалужується. Відповідні результати обчислень можуть записуватися поруч з лініями, що відображають ці шляхи.

Зумовлений процес (підпрограма) це символ відображає виконання процесу, що складається з однієї або кількох операцій, що визначені в іншому місці програми (у підпрограмі, модулі). Всередині символу записується назва процесу і передані в нього дані.

Дані це перетворення у форму, придатну для обробки (введення) або відображення результатів обробки (виведення). Цей символ не визначає носія даних (для вказівки типу носія даних використовуються специфічні символи).

З'єднувач це символ відображає вихід в частину схеми і вхід з іншої частини цієї схеми. Використовується для обриву лінії та продовження її в іншому місці (приклад: поділ блок-схеми, що не поміщається на листі). Відповідні сполучні символи повинні мати одне (при тому унікальне) позначення.

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

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

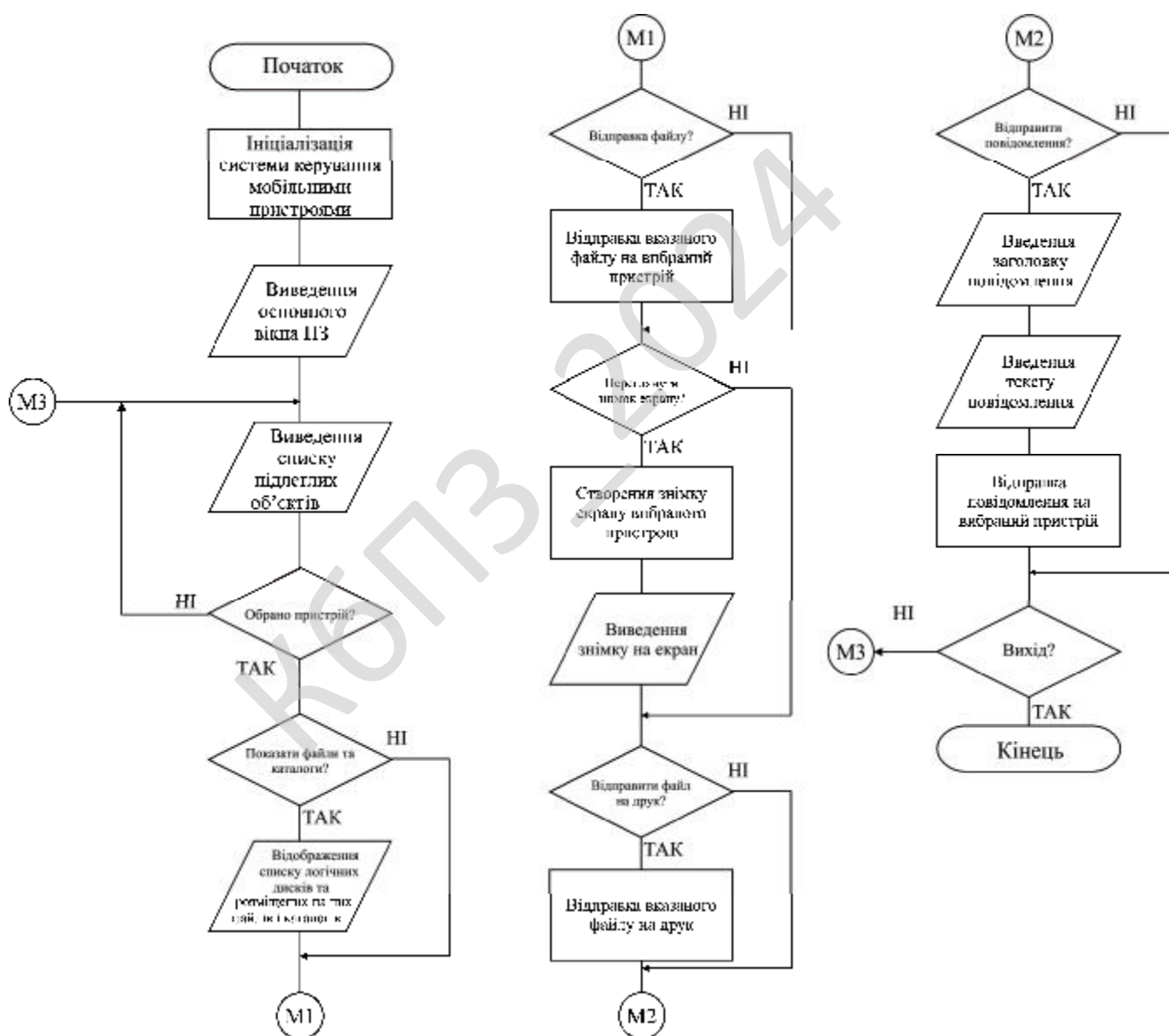


Рисунок 4.1 – Блок-схема основної програми

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю.

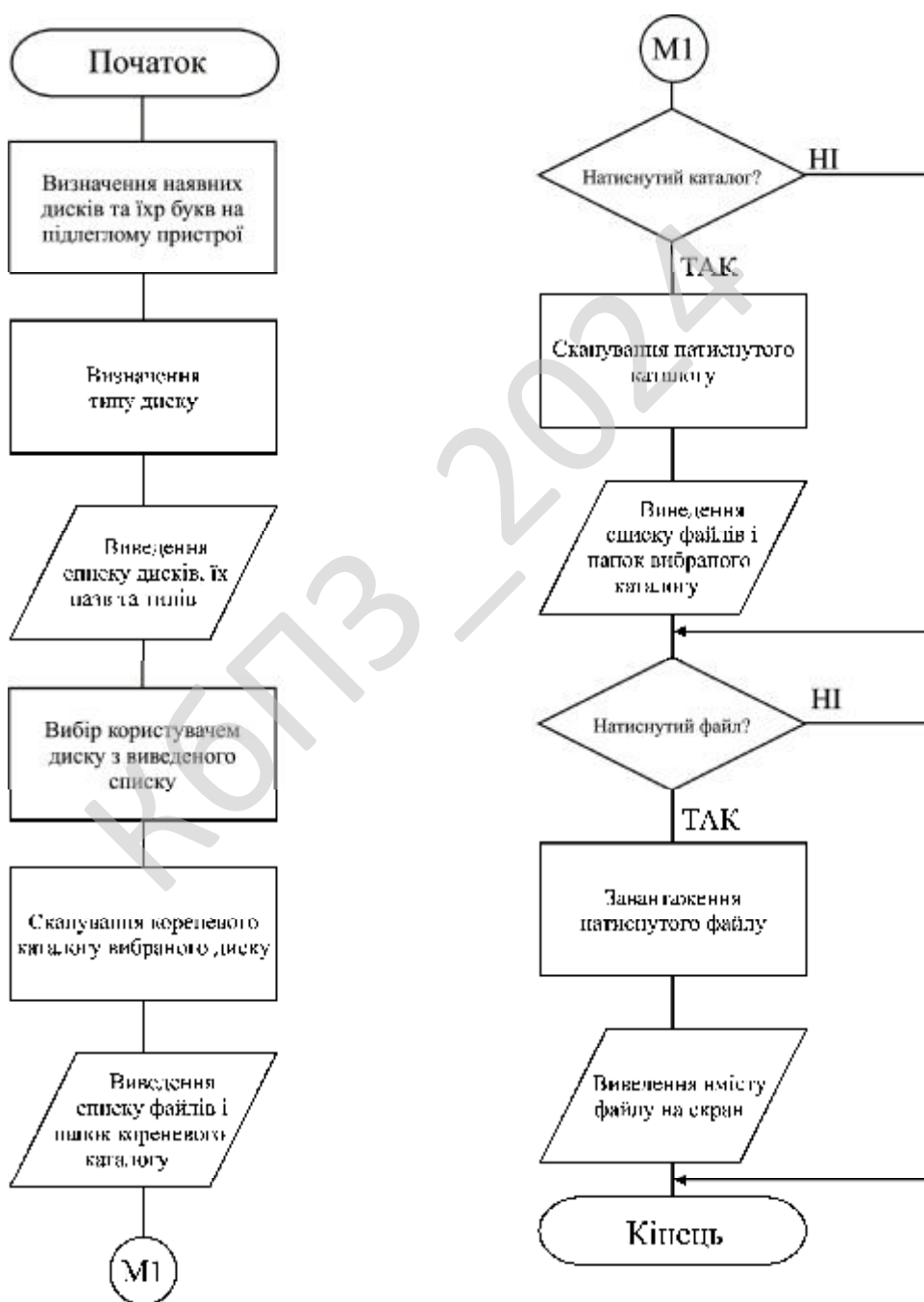


Рисунок 4.2 – Блок-схема роботи підпрограми

UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

UML може бути застосовано на всіх етапах життєвого циклу аналізу бізнес-систем і розробки прикладних програм. Різні види діаграм які підтримуються UML, і найбагатший набір можливостей представлення певних аспектів системи робить UML універсальним засобом опису як програмних, так і ділових систем.

Діаграми дають можливість представити систему (як ділову, так і програмну) у такому вигляді, щоб її можна було легко перевести в програмний код. Основною причиною використання мови UML є спілкування розробників між собою.

Крім того, UML спеціально створювалася для оптимізації процесу розробки програмних систем, що дозволяє збільшити ефективність їх реалізації у кілька разів і помітно поліпшити якість кінцевого продукту.

UML прекрасно зарекомендувала себе в багатьох успішних програмних проектах. Засоби автоматичної генерації кодів дозволяють перетворювати моделі мовою UML у вихідний код об'єктно-орієнтованих мов програмування, що ще більш прискорює процес розробки. Практично усі CASE-засоби (програми автоматизації процесу аналізу і проектування) мають підтримку UML. Моделі розроблені в UML, дозволяють значно спростити процес кодування і направити зусилля програмістів безпосередньо на реалізацію системи.

Діаграми підвищують супроводжуваність проекту і полегшують розробку документації.

UML необхідний:

– Керівникам проектів, які керують розподілом завдань і контролем за проектом.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		41

– Проектувальникам інформаційних систем які розробляють технічні завдання для програмістів.

– Бізнес-аналітикам, які досліджують реальну систему і здійснюють інжиніринг і реінжиніринг бізнесу компанії.

– Програмістам які реалізують модулі інформаційної системи.

При модифікації системи об'єктний підхід дозволяє легко включати в систему нові об'єкти і виключати застарілі без істотної зміни її життєздатності. Використання побудованої моделі при модифікаціях системи дає можливість усунути небажані наслідки змін, оскільки вони не ламають структури системи, а тільки змінюють поведінку об'єктів.

При розробці магістерської дипломної роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу); Діаграма класів; Діаграма компонент; Діаграма об'єктів; Діаграма розгортання.

Діаграма діяльності. Це візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Дія є фундаментальною одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

Одна із цих множин, або обидві водночас, можуть бути порожніми. Виконання дії відповідає виконанню окремої дії. Подібно до цього, виконання діяльності є виконанням окремої діяльності, буквально, включно із виконанням тих дій, що містяться в діяльності. Кожна дія в діяльності може виконуватись один, два, або більше разів під час одного виконання діяльності. Щонайменше, дії мають отримувати дані, перетворювати їх та тестувати, деякі дії можуть вимагати певної послідовності.

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

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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

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

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

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

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

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

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

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

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

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

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

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

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

Діаграма класів (class diagram) служить для представлення статичної структури моделі системи в термінології класів об'єктно-орієнтованого програмування. На цій діаграмі показують класи, інтерфейси, об'єкти й кооперації, а також їхні відносини.

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

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

з незафарбованим ромбом з боку «цілого». Графічно агрегація представляється порожнім ромбом на блоці класу, і лінією, яка від цього ромба до міститься класу.

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

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

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

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

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

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

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

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

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

Діаграма об'єктів в UML це діаграма, що відображає об'єкти та їх зв'язки в певний момент часу. Діаграма об'єктів може розглядатись як окремий випадок

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46

діаграми класів, на якій можуть бути представлені як класи, так і екземпляри (об'єкти) класів. Схожою за змістом є діаграма взаємодії (collaboration diagram).

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

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

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

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

Для реалізації системи віддаленого керування мобільними пристроями під керуванням ОС Android на Python використовується архітектура клієнт-сервер, де сервер є центральним елементом для керування пристроями, а клієнт – додаток на самому пристрої Android. Система містить кілька модулів, які виконують різні функції. Основні модулі включають:

– Серверний модуль обробляє запити користувача, керує з'єднаннями з мобільними пристроями та виконує команди.

– Клієнтський модуль встановлюється на Android-пристрої, приймає команди від сервера та виконує їх (наприклад, знімки екрана, керування файлами, моніторинг стану пристрою).

– Модуль безпеки забезпечує захищене з'єднання між сервером та клієнтами.

– Модуль аналітики зберігає дані про активність користувачів та пристроїв для подальшого аналізу.

Код серверного модуля.

```
import socket
import threading

class RemoteControlServer:
    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.clients = []

self.server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
self.server_socket.bind((self.host, self.port))
self.server_socket.listen(5)
self.is_running = True

def start(self):
    print("Server started, waiting for connections...")
    while self.is_running:
        client_socket, client_address = self.server_socket.accept()
        print(f"Connected to {client_address}")
        self.clients.append(client_socket)
        threading.Thread(target=self.handle_client, args=(client_socket,)).start()

def handle_client(self, client_socket):
    while True:
        try:
            data = client_socket.recv(1024).decode('utf-8')
            if not data:
                break
            self.process_command(client_socket, data)
```

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		48

```

        except Exception as e:
            print(f"Error: {e}")
            break
        client_socket.close()

    def process_command(self, client_socket, command):
        if command == "SCREENSHOT":
            # Request screenshot from client
            client_socket.send("GET_SCREENSHOT".encode('utf-8'))
        elif command == "SHUTDOWN":
            # Shutdown command execution
            client_socket.send("SHUTDOWN_DEVICE".encode('utf-8'))

    def stop(self):
        self.is_running = False
        self.server_socket.close()
        for client in self.clients:
            client.close()

```

Код клієнтського модуля для Android.

```

import socket
import os
import subprocess

class RemoteControlClient:
    def __init__(self, server_host, server_port):
        self.server_host = server_host
        self.server_port = server_port
        self.client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def connect(self):
        try:
            self.client_socket.connect((self.server_host, self.server_port))
            self.listen()
        except Exception as e:
            print(f"Connection failed: {e}")

    def listen(self):
        while True:
            command = self.client_socket.recv(1024).decode('utf-8')
            self.execute_command(command)

```

					БКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		49

```

def execute_command(self, command):
    if command == "GET_SCREENSHOT":
        self.get_screenshot()
    elif command == "SHUTDOWN_DEVICE":
        self.shutdown_device()

def get_screenshot(self):
    # Code to take a screenshot using ADB
    os.system("adb exec-out screencap -p > /sdcard/screenshot.png")
    print("Screenshot captured")

def shutdown_device(self):
    # Code to shutdown device using ADB
    subprocess.run(["adb", "shell", "reboot", "-p"])
    print("Device shutting down")

```

Основні дані для підтвердження проектних рішень:

1. Пропускна здатність мережі – оцінюється використання ресурсів під час передачі даних. Виходячи з тестових вимірів, кожна передача скріншоту займає до 2 МБ, що є допустимим для стандартних Wi-Fi-з'єднань.

2. Час відповіді – в середньому сервер обробляє команди за 100-200 мс, що є достатнім для реального часу.

3. Кількість активних пристроїв – підтримується до 50 пристроїв одночасно, що перевіряється шляхом тестування з використанням віртуальних пристроїв.

Ця система забезпечує гнучкість у керуванні Android-пристроями та може масштабуватись для використання в корпоративних мережах.

4.2 Захист розробленого програмного забезпечення

Захист розробленого програмного забезпечення буде відбуватися за допомогою алгоритму UMAC (код автентифікації повідомлення на основі універсального гешування) – один з видів коду автентичності повідомлень (MAC).

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50

Швидка «універсальна» функція використовується, для того, щоб гешувати вхідне повідомлення M у короткий рядок. До цього рядка потім застосовується функція XOR із псевдовипадковим значенням, у результаті чого ми одержуємо тег UMAC:

$$\text{Tag} = H_{K1}(M) \oplus F_{K2}(\text{Nonce})$$

де $K1$ і $K2$ – секретні випадкові ключі, які мають одержувач і відправник.

Звідси видно, що безпека UMAC залежить від того, яким випадковим способом відправник і одержувач вибрали таємну геш-функцію й псевдовипадкову послідовність. При цьому значення Nonce міняється кожний такт. Через використання Nonce, приймач і передавач повинні знати час відправлення повідомлення й принцип створення значення Nonce. Замість цього можна використовувати в якості Nonce будь-яке інше неповторюване значення, наприклад порядковий номер повідомлення. При цьому даний номер не зобов'язано бути секретним, головне щоб він не повторювався.

UMAC розрахований на використання 32-х, 64-х, 92-х, і 128-бітових тегів, залежно від необхідного рівня безпеки. UMAC звичайно використовується разом з алгоритмом шифрування AES.

Функція створення ключа й псевдовипадкової послідовності

Створення псевдовипадкових байтів необхідно для роботи UHASH і при створенні тегів

Вибір блокового шифру

Для своєї роботи UMAC використовує блоковий шифр, вибір якого визначають наступні константи:

- BLOCLLEN – довжина, у байтах, блоку з яким працює блоковий шифр.
- KEYLEN – довжина, у байтах, ключа блокового шифру.

При цьому використовується функція

– ENCRYPTER(K, P) – зашифрувати рядок P з BLOCLLEN байтів, використовуючи ключ K .

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		51

Приклад: якщо використовується AES з 16-байтним ключем, то BLOCLLEN буде рівним 16(тому що AES працює з 16-байтними блоками).

KDF – функція створення ключа

Ця функція генерує послідовність псевдовипадкових байтів, використовуваних для ключових геш-функцій.

Вхід:

- K – рядок довжиною KEYLEN байт. // Ключ блокового шифру.
- Index – ненегативне ціле число менше, чим 2^{64} .
- Numbytes – ненегативне ціле число менше, чим 2^{64} .

Вихід:

- Y – рядок довжини numbytes байт.

PDF: функція створення псевдовипадкового числа

Ця функція ухвалює ключ і даний час і повертає псевдовипадкове число для використання його в тегу покоління. За допомогою цієї функції можуть бути отримані числа довжиною 4, 8, 12 або 16 байт.

Вхід:

- K – рядок довжиною KEYLEN байт.
- Nonce – рядок довжиною від 1 до BLOCKLEN байт.
- Taglen – ціле число 4, 8, 12 або 16.

Вихід:

- Y – послідовність байтів довжини taglen.

Генерація UMAC-тегів

Генерація UMAC-тегів відбувається за допомогою UHASH функції при використанні Nonce значенні й отриманої до цього рядка. Їхня довжина може бути 4, 8, 12 або 16 байт.

Вхід:

- K – рядок довжиною KEYLEN байт.
- M – рядок довжиною менше 267 біт.
- Nonce – випадкове число від 1 до BLOCKLEN байт.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		52

– Taglen – ціле 4, 8, 12 або 16.

Вихід:

– Тег, послідовність байтів довжиною taglen.

Алгоритм обчислення тегів:

Hashedmessage = UHASH(K, M, Taglen)

Pad = PDF(K, nonce, Taglen)

Tag = Pad xor Hashedmessage

UMAC-32 UMAC-64 UMAC-96 UMAC-128

Дані позначення містять у своїй назві певне значення довжини тегу:

– UMAC-32 (K, M, Nonce) = UMAC (K, M, Nonce, 4).

– UMAC-64 (K, M, Nonce) = UMAC (K, M, Nonce, 8).

– UMAC-96 (K, M, Nonce) = UMAC (K, M, Nonce, 12).

– UMAC-128 (K, M, Nonce) = UMAC (K, M, Nonce, 16).

Універсальна функція гешування(UHASH)

UHASH – універсальна функція гешування, серцевина алгоритму UMAC.

UHASH – функція працює в три етапи. Спочатку до вхідного повідомлення застосовується L1-HASH, потім до цього результату застосовується L2-HASH і, нарешті, до результату застосовується L3-HASH . Якщо при цьому довжина вхідного повідомлення не більш 1024 біт, то L2-HASH не використовується. Тому що функція L3-hash повертає тільки слово довжини 4 байта, те якщо потрібно одержати геш довжини більше 4 байт, здійснюється кілька ітерацій даної трирівневої схеми.

Універсальна функція

Нехай функція гешування вибирається із класу геш-функцій H, які відображають повідомлення в D, набір усіляких образів повідомлення. Цей клас називається універсальним, якщо для яких-небудь окремих пар повідомлень, існує на безлічі H/D функцій, функція, яка відображає їх в елемент D. Зміст цієї функції в тому, що якщо третя сторона прагне замінити одне повідомлення

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		53

іншим, але при цьому вважає, що геш-функція була обрана абсолютно випадково, те ймовірність не виявлення підміни стороною, що ухвалює, прагне до $1/D$.

L1-hash – перший етап

L1-hash розбиває повідомлення на шматки з 1024 байт і до кожного шматка застосовує алгоритм гешування називаний NH. Вихідний результат алгоритму NH в 128 раз менше вхідного.

L2-hash – другий етап

L2-hash працює з виходом L1-hash, використовує поліноміальний алгоритм POLY. Другий етап гешування використовується, тільки якщо довжина вхідного повідомлення більше 16 мегабайт. Використання алгоритму POLY потрібно для того, щоб уникнути тимчасову атаку. На виході з алгоритму POLY виходить 16 байтне число.

L3-hash – третій етап

Цей етап потрібно для того щоб з вихідних 16 байтів алгоритму L2-hash одержати 4-байтне значення.

КБПЗ-2024

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання магістерської дипломної роботи. Розроблене програмне забезпечення віддалене керування мобільними пристроями під керівництвом ОС Android складається з наступних функціональних блоків: Навігаційне меню (Мережа, Параметри підключення, Налаштування, Довідка); Функції представлені у графічному вигляді (іконки); Розділу відображення підключеного пристрою; Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші; Функціональних кнопок ПЗ (Вікно пристроїв, Редагувати додані пристрої, Налаштування миші, Файловий менеджер, Вікно команд, Налаштування, Фільтр дій).



Рисунок 5.1 – Головне вікно розробленого ПЗ

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		55

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

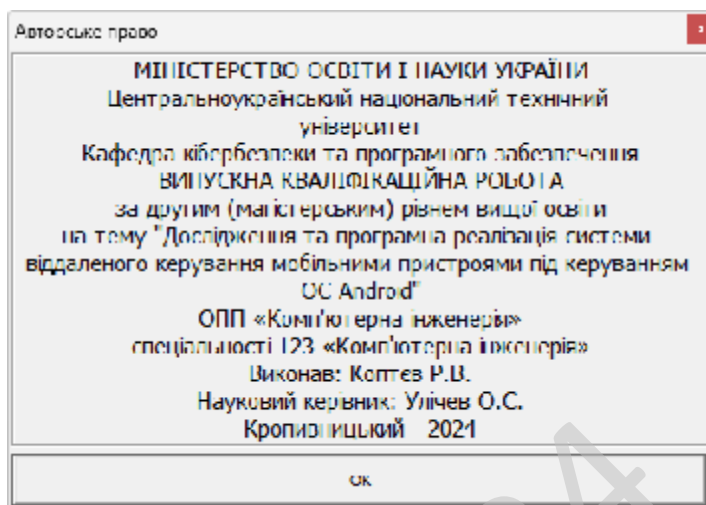


Рисунок 5.2 – Вікно розробника ПЗ

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

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

Проводилась оцінка:

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

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

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		56

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

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

У цьому випадку формуються тестові варіанти, в яких:

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

Недоліки тестування "білої скриньки":

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

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

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

– Деякі результати в програмі залежать не від вихідних даних, а від внутрішніх станів програми.

Обрано умови розповсюдження – proprietary software.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		57

Програмне забезпечення, на яке зберігаються як немайнові, так і майнові авторські права. Отримавши або придбавши таке програмне забезпечення, користувач отримує обмежені права користування ним: може бути заборонено або закрито доступ до коду (вивчення), внесення змін, тиражування, розповсюдження та перепродаж. Програмне забезпечення вважається власницьким, якщо наявне хоча б одне з перелічених обмежень.

Найчастіше основним методом захисту майнових прав на власницьке ПЗ, поза ліцензійною угодою, власник обирає закриття сирцевого коду, захищаючи свій продукт від модифікації і вбудовуючи системи обмеження користування через авторизацію. Таке програмне забезпечення називається закритим. Проте, код власницького продукту може бути і відкритим, але власник може обмежити права користувача умовами користувацької ліцензії.

Власницьке програмне забезпечення та комерційне програмне забезпечення не є синонімами – власницьким може бути і безплатне (тобто, некомерційне) програмне забезпечення.

На противагу власницькому ПЗ існує вільне програмне забезпечення, автори і власники якого дозволяють вивчати, модифікувати і поширювати свій продукт. Саме визначення власницького програмного забезпечення виникло в результаті діяльності громадського руху вільного програмного забезпечення (представленого Фондом вільного програмного забезпечення та іншими організаціями) і осмислення умов свободи користування програмами. Визначенням власницького програмного забезпечення є не невідповідність хоча б одній з базових умов вільного програмного забезпечення. Сама назва власницьке ПЗ підкреслює визначальне значення власника у способі використання і можливостях розвитку цього програмного забезпечення.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		58

6 НАУКОВА НОВИЗНА

У випускній кваліфікаційній роботі за другим (магістерським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи віддаленого керування мобільними пристроями під керуванням ОС Android.

Метою розробки є дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android.

Об'єктом дослідження є процес віддаленого керування мобільними пристроями під керуванням ОС Android.

Предметом дослідження є методи віддаленого керування мобільними пристроями під керуванням ОС Android.

Методи дослідження базуються на методах теорії комп'ютерних мереж, методах математичної статистики, методах розробки програмного забезпечення.

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

– Удосконалено метод віддаленого керування мобільними пристроями під керуванням ОС Android.

– Розроблено вітчизняний продукт віддаленого керування мобільними пристроями під керуванням ОС Android, який має більш широкі можливості, на відміну від існуючих аналогів.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

7 МАРКЕТИНГОВЕ ТА ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ ІТ-ПРОЄКТУ

7.1 Визначення цільової аудиторії кінцевого готового продукту

Результати дослідження та програмної реалізації системи віддаленого керування мобільними пристроями під керуванням ОС Android можуть зацікавити досить широке коло (рисунок 7.1).

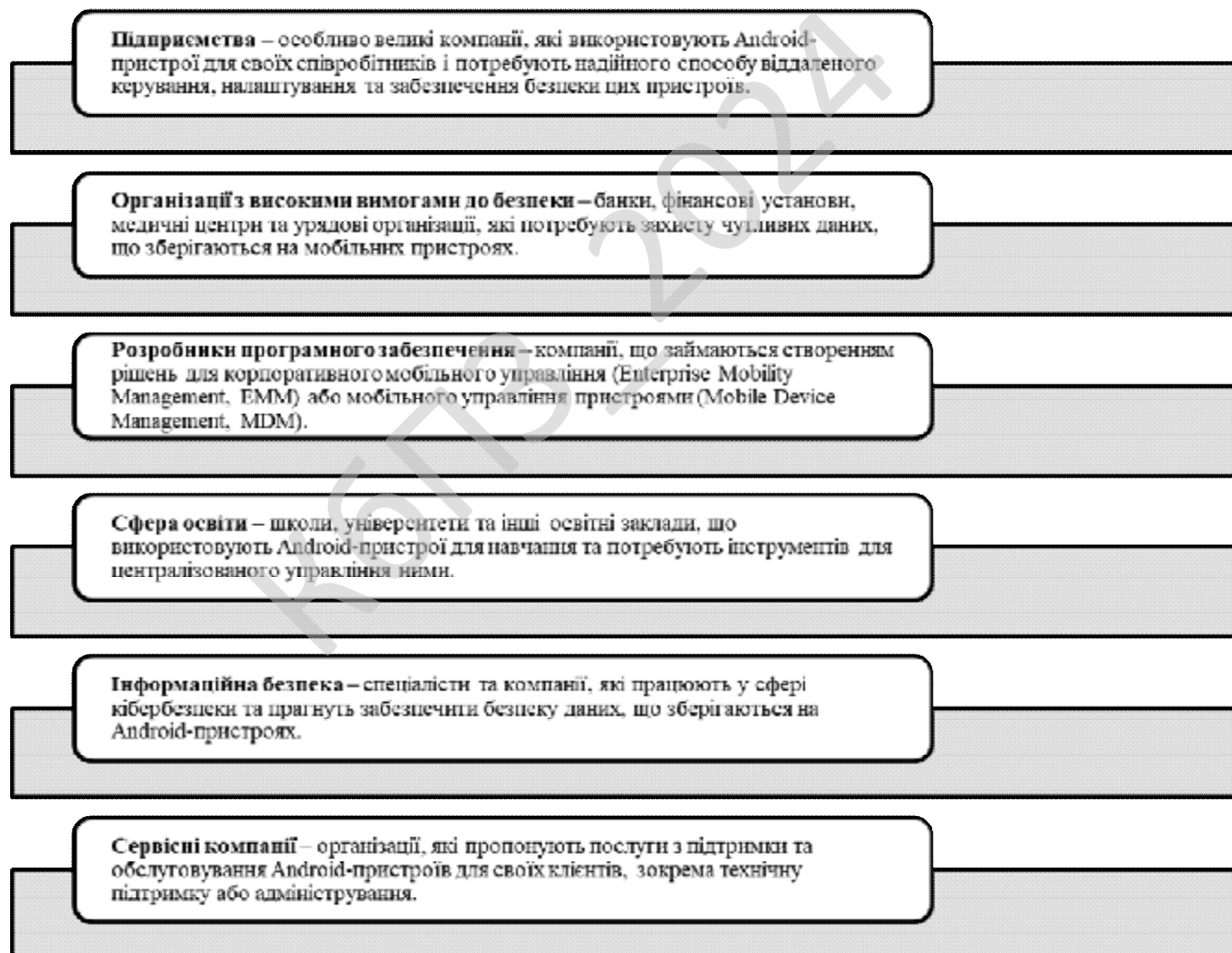


Рисунок 7.1 – Цільова аудиторія

Розробка таких систем може допомогти їм у впровадженні рішень, що полегшать дистанційне керування, підвищать ефективність роботи та забезпечать високий рівень безпеки.

7.2 Оцінка привабливості шляхом застосування методів експертних оцінок

Оцінка привабливості програмної реалізації системи віддаленого керування мобільними пристроями під ОС Android за допомогою методів експертних оцінок може бути проведена через метод бальної оцінки, де експерти оцінюють проект за різними критеріями.

Визначимо ключові критерії, за якими буде оцінюватися привабливість системи (рисунок 7.2).

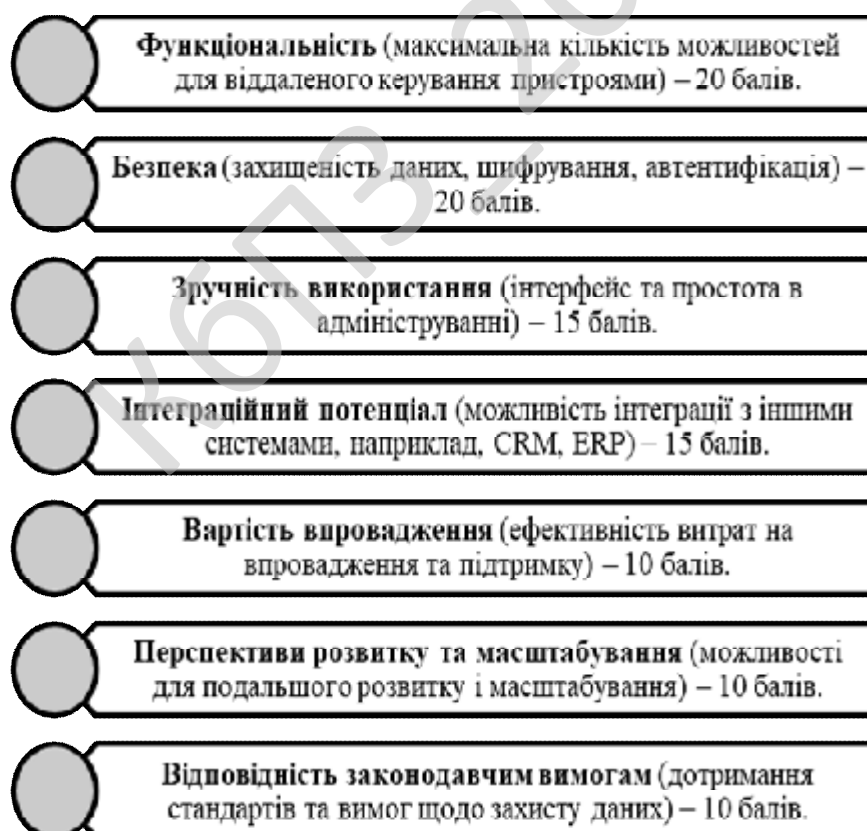


Рисунок 7.2 – Критерії оцінювання проекту

Формуємо групу експертів. Група експертів може складатися з фахівців у галузі мобільного управління пристроями, інформаційної безпеки, ІТ-менеджерів і представників бізнесу, які мають досвід у впровадженні та експлуатації подібних систем.

Кожен експерт оцінює кожен критерій за 10-бальною шкалою, де 10 – максимальний бал (критерій повністю задовольняє вимоги), а 1 – мінімальний. Надалі зводимо середнє значення по критеріям: функціональність: 18 із 20, безпека: 17 із 20, зручність використання: 12 із 15, інтеграційний потенціал: 13 із 15, вартість впровадження: 8 із 10, перспективи розвитку та масштабування: 9 із 10, відповідність законодавчим вимогам: 9 із 10.

Підсумуємо отримані значення: Загальна оцінка = $18 + 17 + 12 + 13 + 8 + 9 + 9 = 86$ із 100 можливих. Отримані 86 балів із 100 свідчать про високу привабливість проєкту для його цільової аудиторії. Виходячи з цього, можна зробити висновок, що програмна реалізація системи віддаленого керування мобільними пристроями під ОС Android є перспективною для впровадження.

7.3 Вибір методу оцінки вартості ПЗ

Для оцінки вартості програмної реалізації системи віддаленого керування мобільними пристроями під ОС Android можна застосувати кілька методів, залежно від специфіки проєкту, ресурсів і вимог. Нижче наведені оптимальні (рисунок 7.3).

Для проєкту програмної реалізації системи віддаленого керування мобільними пристроями під ОС Android найкраще підійде метод оцінки на основі трудовитрат, якщо проєкт динамічний і потребує адаптації на різних етапах. Якщо ж вимоги чітко окреслені, метод функціональних точок також буде ефективним, оскільки він дозволяє оцінити вартість залежно від запланованого функціоналу. Обидва методи можуть доповнюватися експертним оцінюванням для підвищення точності, особливо на початкових етапах планування.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		62

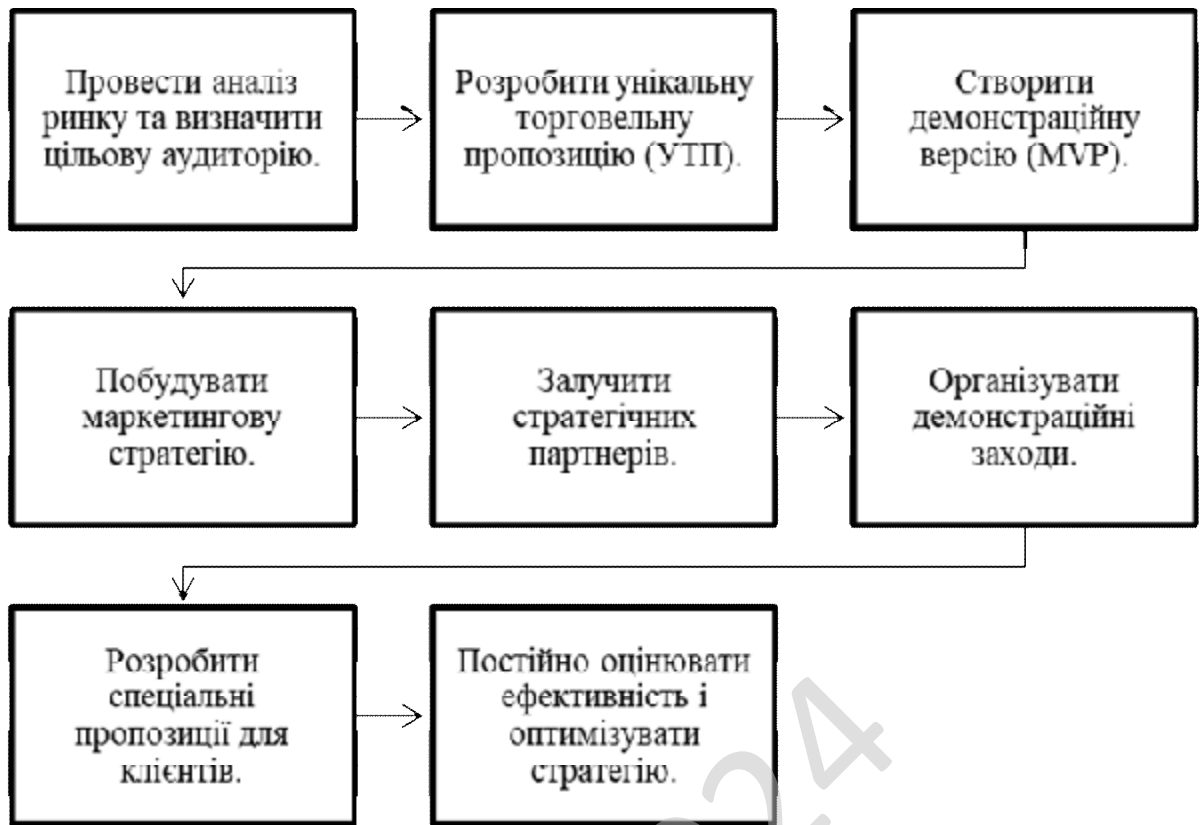


Рисунок 7.5 – Етапи просування проекту

Цей алгоритм допоможе поступово просувати проект, адаптуючи підходи відповідно до відгуків клієнтів і реакції ринку.

7.6 Оптимізація каналів збуту та шляхів реалізації ПЗ

Для оптимізації каналів збуту та шляхів реалізації проекту програмної реалізації системи віддаленого керування мобільними пристроями під ОС Android можна застосувати такі стратегії (рисунок 7.6).

Ці кроки допоможуть ефективно оптимізувати канали збуту та способи реалізації, що забезпечить успішне розгортання та стійке зростання проекту на ринку.

1. Оптимізація каналів збуту

- Прямий продаж великим корпораціям: Зосередьтеся на B2B-продажах, оскільки великі організації, що активно використовують мобільні пристрої, можуть бути зацікавлені у централізованому віддаленому управлінні ними. Для таких клієнтів можна пропонувати індивідуальні умови.
- Партнерські програми з IT-компаніями та інтеграторами: Запропонуйте партнерські умови для IT-компаній, які впроваджують мобільні рішення. Це дозволить розширити охоплення, оскільки партнери зможуть інтегрувати систему у свої рішення.
- Сервіс-провайдери керування мобільними послуг (MDM): Співпраця з провайдерами, які надають послуги з керування мобільними пристроями (MDM), дозволить швидше досягти потенційних клієнтів і пропонувати систему як додаткову послугу або компонент MDM.
- Маркетплейси та каталоги ПЗ: Розміщення на платформах, де компанії шукають спеціалізовані рішення для управління мобільними пристроями, наприклад, Google Play (для бізнес-програм), AWS Marketplace, або спеціалізованих платформах для корпоративного ПЗ.

2. Шляхи реалізації та адаптація продукту

- Гнучка цінова політика: Запровадьте різні тарифні пакети (наприклад, для малих, середніх і великих підприємств) з відповідними функціями. Це дозволить охопити ширшу аудиторію, де кожен клієнт зможе обрати оптимальний пакет.
- Пробо-версія безкоштовного тестового періоду: Надання безкоштовного тестового періоду або демонстраційної версії допоможе компаніям спробувати систему перед покупкою, що збільшить ймовірність підписання довгострокових контрактів.
- Гібридна модель розгортання: Додайте можливість як хмарного, так і локального розгортання системи. Хмарне розгортання зменшить витрати для малих і середніх компаній, тоді як локальне може підійти великим корпораціям із сувереними вимогами до безпеки.
- Інтеграція з існуючими платформами: Зробіть систему сумісною з популярними платформами управління, такими як Microsoft Intune, Google Workspace, або CRM-системами. Інтеграція з існуючими бізнес-інструментами значно підвищить її привабливість для компаній.

3. Маркетингові стратегії для просування

- Фокус на безпеку та ефективність: Поділіться системою як ефективне рішення з високим рівнем захисту даних. Це буде актуально для клієнтів, які шукають безпечні методи для управління своїми пристроями, особливо у галузях фінансів та охорони здоров'я.
- Вебінари та тренінги: Проводьте вебінари для потенційних клієнтів і демонструйте ключові переваги системи. Тренінги можуть бути орієнтовані на IT-фахівців компаній, що потенційно сприятиме впровадженню системи в організації.
- Сертифікація для партнерів: Запровадьте програму сертифікації для партнерів, які хочуть продавати та підтримувати систему. Це допоможе підвищити довіру до продукту й забезпечить якість обслуговування.

4. Аналіз результатів та коригування

- Моніторинг зворотного зв'язку: Проводьте регулярні опитування клієнтів і партнерів для збору зворотного зв'язку, що дозволить поліпшувати систему відповідно до вимог ринку.
- Аналіз каналів збуту: Визначайте найбільш результативні канали продажів та фокусуйте ресурси на тих, які демонструють найвищу рентабельність інвестицій (ROI).

Рисунок 7.6 – Оптимізація каналів збуту та шляхів реалізації

7.7 Визначення ключових факторів успіху конкретного проєкту

Для успішної реалізації проєкту віддаленого керування мобільними пристроями на базі ОС Android важливо звернути увагу на кілька ключових факторів (рисунок 7.7).

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		66

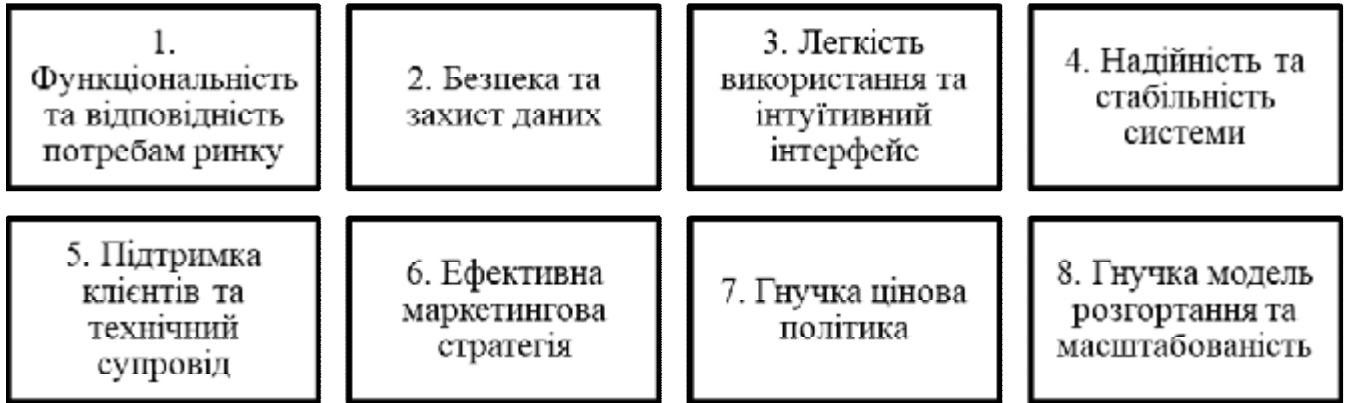


Рисунок 7.7 – Ключові фактори успіху проекту

Ключові фактори успіху проекту системи віддаленого керування Android-пристроями лежать в основі якісного продукту, орієнтованого на безпеку, зручність, гнучкість та підтримку, підкріплених сильним маркетингом та адаптивною ціновою політикою.

8 ЗАХОДИ З ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

8.1 Вступ

Законом України “Про охорону праці” [3] 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 [2].

Загальні вимоги пожежної безпеки під час експлуатації комп’ютерної техніки визначають «Правила пожежної безпеки в Україні» (затверджені наказом МВС від 30.12.2014 № 1417) [13], комп’ютерних класів – пункт 3 розділу VIII «Правил пожежної безпеки для навчальних закладів та установ системи освіти України» (затверджені наказом МОН від 15.08.2016 № 974). [12] та інші державні стандарти, що регламентують експлуатування комп’ютерної техніки як радіоелектронної апаратури.

Охорона здоров'я робітників, забезпечення безпеки умов праці, ліквідація та профілактика професійних захворювань і виробничого травматизму складає одну з головних турбот людського суспільства.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

8.2 Пожежна безпека

Вимог пожежної безпеки на підприємстві неухильно повинен дотримуватися кожен співробітник, а організаційна складова при цьому покладається на посадових осіб за відповідним рішенням керівництва і прописується в посадових інструкціях і положеннях по структурним підрозділам.

Зокрема, вказуються конкретні території, ділянки, зони, об'єкти, цілі будівлі і їх частини, поверхи, на яких відповідального співробітника повинне проводити такі організаційні роботи.

Відповідальні особи зобов'язуються розробити, впровадити та підтримувати в певному інструкцією і положенням на ввірених їм об'єктах протипожежний режим і інструкції відповідно до вимог, викладених в нормативних актах.

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

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

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		69

приміщеннях. Звуковбирне облицювання стін, стель приміщень треба виконувати з негорючих та важко горючих матеріалів.

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

З точки зору забезпечення пожежної безпеки до цих заходів можна віднести наявність схеми евакуації з приміщення вузла, у випадку пожежі, повішену на вхідні двері.

8.3 Характеристика умов праці програміста

В приміщенні, в якому проводиться розробка і дослідження програмного продукту, відсутні умови, які можуть створювати підвищену або особливо підвищену небезпеку, тому воно відноситься до класу звичайних приміщень згідно Правил улаштування електроустановок (ПУЕ). Джерелом живлення є трифазна мережа напруги 380/220 В з глухо заземленою нейтралі, з частотою 50 Гц згідно За вибухопожежонебезпеку приміщення відноситься до класу В. В таблиці 8.2 наведена загальна характеристика приміщення щодо вибухопожежонебезпеки та важкістю робіт.

Температура повітря в приміщенні визначається температурою зовнішнього повітря і тепловою енергією, що виділяється всередині приміщення. Джерелами теплоти в даному приміщенні є люди, електроустаткування, а також освітлювальні прилади в темний час доби. Зовнішнім джерелом надлишкового тепла є сонячна радіація у світлий час доби. Робота, виконувана в даному приміщенні, відноситься до категорії I-а. Людиною в цьому випадку виділяється до 120 ккал теплової енергії в годину. Вологість повітря в приміщенні визначається вологістю атмосферного і видихуваного людьми повітря, а також випарами з поверхні шкіри.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

Таблиця 8.1 – Загальна характеристика приміщення щодо вибухопожежонебезпеки та важкістю робіт

Характеристика приміщень за вибухопожежною категорією та класом зони	Загальна характеристика приміщення	Категорія за важкістю робіт згідно ГН 3.3.5-8.6.6.1 -2002
В – пожежонебезпечне клас П – П	Звичайне без ознак хімічного забруднення та нормальної вологості і за санітарними нормами	1а до 139 Вт/м ² 1б до 140-174 Вт/м ² Клас умов праці – оптимальний

У таблиці 8.1 приведені оптимальні значення параметрів мікроклімату для категорії робіт I-а, а також фактичні значення цих параметрів у розглянутому приміщенні. У приміщеннях з використанням обчислювальної техніки рекомендується застосування тільки оптимальних значень показників мікроклімату, тобто таких, при яких людина відчуває себе комфортно.

У приміщенні немає виділення шкідливих газів. Тому що в ньому не проводиться монтажних робіт, пайки чи інших робіт, при яких виділяються шкідливі гази.

Для нормалізації параметрів повітряного середовища також періодично здійснюється провітрювання приміщення і вологе прибирання. У всьому будинку діє встановлена загально обмінна витяжна вентиляція.

Раціональне освітлення приміщення сприяє кращому виконанню виробничого завдання і забезпеченню комфорту при роботі. Для забезпечення нормального освітлення застосовуються природне, однобічне, бічне і штучне освітлення, а також сполучене, нормуються згідно ДБН В.2.5-28-2006 Природне і штучне освітлення [1]. Дані по нормах освітлення наведені в таблиці 8.4

За результатами виміру освітленості величина освітленості від системи загального штучного освітлення дорівнює 310 лк, що відповідає вимогам, які пред'являються до даного приміщення.

Основними джерелами шуму на робочих місцях, обладнаних відео дисплейними терміналами, є принтер, сканер факс і обладнання для кондиціонування повітря, в самих відео дисплейних терміналах – вентилятори систем охолодження і трансформатори.

Таблиця 8.2 – Значення параметрів небезпечних факторів праці

Найменування параметра	Значення параметра		Нормативний документ	
	Фактичне	Нормоване		
Освітленість штучна, лк	300	300	ДБН.В 2.5-28:2018 [1]	
Значення КПО,%	1,0	1,1	ДБН.В 2.5-28:2018 [1]	
Повітрообмін, м /год	взимку	76	80	ДСН 3.3.6.042-99
	влітку	36	80	ДСН 3.3.6.042-99
Температура повітря. °С	взимку	22	21-25	ДСанПіН 3.3.2-007-98
	влітку	24	27-28	ДСанПіН 3.3.2-007-98 [2]
Відносна вологість,%	взимку	60	<75	ДСанПіН 3.3.2-007-98
	влітку	55	<60	ДСанПіН 3.3.2-007-98 [2]
Швидкість переміщення повітря, м/с	взимку	0,16	<0,2	ДСанПіН 3.3.2-007-98 [2]
	влітку	0,10	<0,2	ДСанПіН 3.3.2-007-98 [2]

Крім рекомендацій щодо конкретного приміщення, де було проведено дослідження умов праці, існують загальні вимоги, які зарекомендовані відповідними нормативними документами.

Правильна організація робочих місць запобігає передчасній втомлюваності користувача і сприяє збереженню здоров'я. Організація робочого місця передбачає:

- правильне розміщення робочого місця у виробничому приміщенні;
- вибір ергономічного обґрунтованого робочого положення, виробничих меблів з урахуванням характеристик людини;
- раціональне компонування обладнання на робочих місцях;
- урахування характеру й особливостей трудової діяльності.

Стосовно робочих місць користувача ВДТ, то організація робочого має забезпечуватися відповідно до ДСанПіН 3.3.2-007-98. Для запобігання перевтомленню необхідно виконувати вправи для очей та дотримуватись розпорядку роботи та відпочинку. На робочому місці реалізовувався режим відпочинку: кожні дві години – перерва для виконання фізичних вправ для м'язів очей.

Додаткові 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'язковою наявністю масок-клапанів, або іншого спорядження для штучного дихання.

8.5 Розрахункова частина

Для захисного штучного заземлення застосовуються вертикальні електроди: металевий куток 63х63х6 мм, (згідно з ДСТУ 2251-93 «Кутики сталеві гарячекатані рівнополічні. Сортамент») довжиною L=1,5 м, та горизонтальний електрод – металева полоса з перетином 60х5 мм. Напряга – 220/380 В.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		74

Розрахункова схема розташування заземлюючих електродів – по контуру прямокутником (рис. 8.1).

Розрахунок проведемо за допустимим опором розтіканню струму заземлювача.

Початкові дані для розрахунку захисного заземлення: тип верхнього шару ґрунту – чорнозем, нижнього шару ґрунту – глина (питомий опір $\rho_2 = 40 \text{ Ом}\cdot\text{м}$). Умовна товщина верхнього шару ґрунту: $H=0,45 \text{ м}$. Відстань між вертикальними заземлювачами (електродами) $A=3 \text{ м}$. Глибина закладення горизонтального контура заземлення $t=0,65 \text{ м}$. Опір заземлювача, який нормується: $R_{3Н} = 4 \text{ Ом}$. Необхідно визначити необхідну кількість вертикальних заземлювачів та довжину полоси (горизонтального заземлювача).

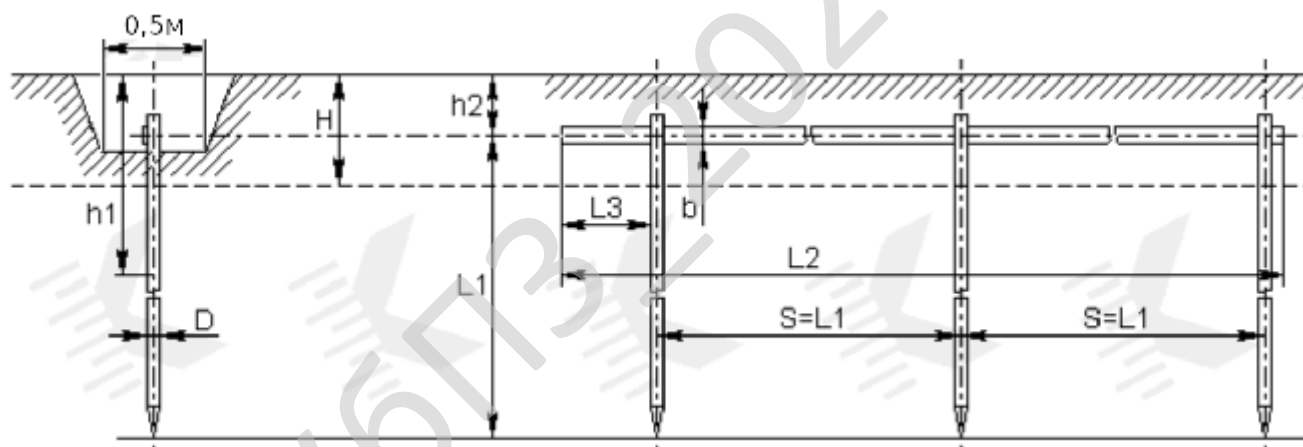


Рисунок 8.1 – Схема штучного заземлення

Виконуємо розрахунок.

Відстань від центра вертикального заземлювача до поверхні землі:

$$T = t + L/2 = 0,65 + 1,5/2 = 1,35 \text{ м.}$$

Розрахунковий питомий опір ґрунту (з врахуванням того, що фактично вся конструкція заземлювача розташовується у нижньому шарі ґрунту):

$$\rho = \psi \cdot \rho = 1,36 \cdot 40 = 54,5 \text{ Ом}\cdot\text{м.}$$

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		75

де $\eta_{II} = 0,6$ – табличне значення коефіцієнта екранування з'єднуючої полоси [11].

Умова $R \leq R_{3H}$ виконується ($3,28 \leq 4$).

Оскільки при 10 вертикальних електродах R суттєво більше R_{3H} , зменшимо кількість вертикальних електродів N до 8 і виконаємо перерахунок. У результаті остаточно отримали: $R=3,95$ Ом. при кількості вертикальних електродів $N=8$.

Висновки до розділу

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

Виконано розрахунок захисного штучного заземлення, як одного з ключових факторів безпеки програміста.

КБПЗ-2024

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		77

9 ОСНОВНІ ВИСНОВКИ

Програмне забезпечення, створене в результаті виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти, призначено для системи віддаленого керування мобільними пристроями під керуванням ОС Android.

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 віддаленого керування мобільними пристроями під керуванням ОС Android.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем віддаленого керування мобільними пристроями під керуванням ОС Android.
- Досліджена система віддаленого керування мобільними пристроями під керуванням ОС Android.
- На основі отриманих результатів досліджень створена програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання віддаленого керування мобільними пристроями під керуванням ОС Android.

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-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		78

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

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

Програма реалізована на мові високого рівня Python. Дана мова програмування дозволяє найбільш ефективно обробляти дані. 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.

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм UMAC.

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.

Проведено маркетингове та економічне обґрунтування IT-проєкту, що дозволило визначити ключові фактори успіху даного проєкту.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		79

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Коптєв Р.В. Дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android // Збірник праць молодих науковців ЦНТУ. – Вип. 14. – Кропивницький: ЦНТУ, 2024.

2. Dawn Griffiths, David Griffiths. Head First Android Development. O'Reilly Media, Inc. 2021. 1414 p.

3. Nathan Metzler. Kotlin Programming for Beginners. Independently published. 2021. 158 p.

4. Aaron Torres. Go Programming Cookbook Second Edition. Packt Publishing Ltd. 2019. 427 p.

5. Мелешко Є.В., Якименко М.С., Поліщук Л.І. Алгоритми та структури даних: Навчальний посібник для студентів технічних спеціальностей денної та заочної форми навчання. – Кропивницький: Видавець – Лисенко В.Ф., 2019. – 156 с.

6. Knuth D. The Art of Computer Programming, Vol. 1: Fundamental Algorithms, 3rd Edition 3rd Edition. – Addison-Wesley Professional, 2019. – 672 p.

7. Knuth D. Art of Computer Programming, Vol. 2: Seminumerical Algorithms 3rd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 672 p.

8. Knuth D. The Art of Computer Programming: Vol. 3: Sorting and Searching 2nd Edition, Kindle Edition. – Addison-Wesley Professional, 2019. – 800 p.

9. Cormen T.H., Leiserson C.E., Rivest R.L., Stein C. Introduction to Algorithms, 3rd Edition (The MIT Press) 3rd Edition – The MIT Press, 2019. – 1292 p.

10. Kuznetsov O., Ilchenko O., Kryvinska N., Buravchenko K., Smirnov O., Savchenko Iu. «An Empirical Assessment of Leading Blockchain Financial Services». 2023 IEEE 1st Ukrainian Distributed Ledger Technology Forum (UADLTF), Kyiv, Ukraine, 2023, pp. 1-6,

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		80

11. Smirnov O., Fedorov E., Neskorođieva A., Neskorođieva T. «Intellectual Classification method of Gymnastic Elements Based on Combinations of Descriptive and Generative Approache». *CEUR Workshop Proceedings* Volume 3664, 2024, Pages 11-23.

12. Kuznetsov, O., Kryvinska, N., Ilchenko, O., Smirnova, T., Ulianovska, Y. «Comparative Analysis of Cryptocurrency Trading Platforms Using the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, 2023, 3628, pp. 106-115.

13. Malyukov V., Bebishko B., Lakhno V., Smirnov O., Malyukova I., Mohylnyi H. «Managing the Purchase-Sale Process of Digital Currencies Under Fuzzy Conditions». *Lecture Notes in Networks and Systems*, 2023, 729 LNNS, pp. 104–112.

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

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

16.

17. 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.

18. Smirnov, O., Karapetyan, A., Fedorov, E., «Creating Neural Network and Single Solution Human-Based Metaheuristic Methods of Solving the Traveling Salesman Problem». *CEUR Workshop Proceedings*, Volume 3312, 2022, pp. 47-58.

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

20. Smirnov O., Kovalenko O., Kovalenko A., Kavun S. «Quantitative Risk Assessment Method Development in the Context of the SDLC-model». 2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T), 2021, pp. 203-208, doi: 10.1109/PICST54195.2021.9772143

21. Smirnova T., Gnatyuk S., Berdibayev R., Avkurova Zh., Iavich M. «Cloud-Based Cyber Incidents Response System and Software Tools». Communications in Computer and Information Science, 2021, vol 1486. Springer, Cham. pp 169-184.

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

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

24. Smirnov O., 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.

25. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». International Journal of Computer Network and Information Security (IJCNIS). Vol. 12, No. 3, 2020. PP.33-43.

26. 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.

27. 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.

28. 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.

29. 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.

30. 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.

31. Kuznetsova, T., «Code-Based Schemes for Post-Quantum Digital Signatures», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P. 707-712.

32. Smirnov, O., Kuznetsov, A., Stefanovych, O., Gorbenko, Y., Krasnobaev, V., Kuznetsova K. «Information Hiding Using 3D-Printing Technology», 10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019; Metz; France; 18-21 September 2019. P.701-706.

33. 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.

34. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

35. Вінтенко Б.Ю., Смірнов О.А., Коваленко А.С., Смірнов С.А., Буравченко К.О. «Дослідження вимог міжнародних стандартів ІЕС60880 та ІЕС62138 з розробки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 3(73), С. 155-166.

36. Вінтенко, Б., Миронець, І., Смірнов, О., Кравчук, О., Козірова, Н., Савеленко, Г., Коваленко, А. «Дослідження вимог та аналіз кібербезпеки програмного забезпечення інформаційно-керуючих систем АЕС, важливих для безпеки». *Кібербезпека: освіта, наука, техніка*. 2024. №3(23), С. 111-131.

37. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А., Коваленко А.С. «Дослідження нормативних документів та галузевих стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». *Системи управління, навігації та зв'язку*, 2023, вип. 2(72), С. 170-178.

38. 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.

39. Вінтенко Б.Ю., Смірнов О.А., Коваленко О.В., Смірнов С.А. «Дослідження нормативної документації та стандартів розробки програмного забезпечення комп'ютерних систем управління АЕС, важливих для безпеки». VI міжнародна науково-практична конференція «Інформаційна безпека та комп'ютерні технології», м. Кропивницький. 20-21 квітня 2023 р. – Кропивницький: ЦНТУ. – 2023. – С. 35-36.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		84

40. Смірнов, О.А., Усік П.С., Полігенко О.О., Одарченко Р.С., Терещенко Л.Ю. «Інформаційна технологія та програмне забезпечення для підвищення ефективності планування підсистеми базових станцій стільникового зв'язку». Проблеми телекомунікацій. № 1(26). С. 83-96. 2020.

41. 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.

42. 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.

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

44. 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).

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

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. O. Smirnov, O. Kovalenko, A. Kovalenko, S. Smirnov, V. Vialkova. The mathematical model of the testing technology for DOM XSS vulnerabilities.

Scientific & practical cyber security journal (SPCSJ) Vol 2 Issue 1, 22-28 pp.
[Электронный Журнал]. Georgia. Tbilisi: SCSA – 2018.

48. Oleksii Smirnov, Oleksandr Kovalenko, Jamil Al-Azzeh, Anna Kovalenko, Serhii Smirnov. Qualitative risk analysis of software development. Asian Journal of Information Technology. – Volume 17(3). – Medwell Journals. – 2018. – P. 218-230.

49. Смірнов О.А., Коваленко О.В., Коваленко А.С., Смірнов С.А. Розробка методу передтестової компіляції й розподілу доступу. Збірник наукових праць III міжнародної науково-практичної конференції “Інформаційна безпека та комп’ютерні технології”, м. Кропивницький. 19-20 квітня 2018р. – Кропивницький: ЦНТУ. – 2018. – С. 214-215

50. Smirnov Oleksii, Kovalenko Oleksandr, Kovalenko Anna, Smirnov Serhii. Method of testing the DOM XSS vulnerability. International Conference «Information technologies, systems and networks ITSН-2017». Chisinau, Republic of Moldova. 17 – 18 October 2017. – Chisinau: Academy of Sciences of Moldova, Military Academy of Armed Forces “Alexandru cel Bun”. 2017. P7.

51. Смірнов О.А., Смірнов С.А., Коваленко О.В., Коваленко А.С. Технологія тестування DOM XSS уразливості. Науково-практичний журнал кібер безпеки (SPCSJ) № 1. [Електронний журнал]. Грузія. Тбілісі: SCSA – 2017.

52. Смірнов О.А., Лисенко І.А. Інформаційна технологія проектування тестових наборів з урахуванням вимог до програмного забезпечення. Системи управління, навігації та зв'язку. – Випуск 4 (44). – Полтава: ПолтНТУ. – 2017. – С. 112-115.

					ВКРМ-123.24.0017.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		86

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-123.24.0017.00.00.ТЗ		
Вим.	Арк.	№ документа	Підпис	Дата			
Розробив	Коптєв Р.В.				Літ.	Аркуш	Аркушів
Перевірів	Улічев О.С.						
Н. Контр.	Коваленко А.С.				ЦНТУ КІ-23М		
Затв.	Смірнов О.А.						

1 Найменування та область застосування

Це технічне завдання розповсюджується на дослідження та програмну реалізацію системи віддаленого керування мобільними пристроями під керуванням ОС Android.

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. № 19-13 від 07.08.2024 року).

3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти є дослідження та програмна реалізація системи віддаленого керування мобільними пристроями під керуванням ОС Android.

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-123.24.0017.00.00.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 віддаленого керування мобільними пристроями під керуванням ОС Android;
- 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-123.24.0017.00.00.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 під керуванням ОС Android 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.

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 Python.

					ВКРМ-123.24.0017.00.00.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 маркетингове та економічне обґрунтування ІТ-проєкту з урахуванням цін на 3 вересня 2024 року.

8 Вимоги щодо охорони праці

В частині охорони праці випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти повинна бути розглянута характеристика умов праці програміста.

					ВКРМ-123.24.0017.00.00.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 – 86 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 Маркетингове та економічне обґрунтування ІТ-проєкту.

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 02.12.2024 р.

11.2 Подання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти на захист 18.12.2024 р.

					ВКРМ-123.24.0017.00.00.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
_____ Улічев О.С.

*Дослідження та програмна реалізація
системи віддаленого керування мобільними пристроями під керуванням ОС
Android*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 19

Літера: РП

Кропивницький – 2024 року

Основна програма

```

import socket
import threading
import os
import subprocess
import time
import json

class RemoteControlServer:
    def __init__(self, host, port):
        self.host = host
        self.port = port
        self.clients = {}
        self.server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.server_socket.bind((self.host, self.port))
        self.server_socket.listen(5)
        self.is_running = True

    def start(self):
        print("Server started, waiting for connections...")
        while self.is_running:
            client_socket, client_address = self.server_socket.accept()
            print(f"Connected to {client_address}")
            self.clients[client_address] = client_socket
            threading.Thread(target=self.handle_client, args=(client_socket,
client_address)).start()

    def handle_client(self, client_socket, client_address):
        while True:
            try:
                data = client_socket.recv(1024).decode('utf-8')
                if not data:
                    break
                self.process_command(client_socket, client_address, data)
            except Exception as e:
                print(f"Error with {client_address}: {e}")
                break
            client_socket.close()
            del self.clients[client_address]
            print(f"Client {client_address} disconnected")

    def process_command(self, client_socket, client_address, command):
        command_data = json.loads(command)
        if command_data['action'] == "SCREENSHOT":
            # Request screenshot from client
            client_socket.send(json.dumps({"action":
"GET_SCREENSHOT"}).encode('utf-8'))
        elif command_data['action'] == "SHUTDOWN":
            # Shutdown command execution
            client_socket.send(json.dumps({"action":
"SHUTDOWN_DEVICE"}).encode('utf-8'))
        elif command_data['action'] == "RESTART":
            # Restart command execution
            client_socket.send(json.dumps({"action":
"RESTART_DEVICE"}).encode('utf-8'))
        elif command_data['action'] == "FILE_TRANSFER":
            # File transfer command
            client_socket.send(json.dumps({"action": "TRANSFER_FILE",
"file_path": command_data['file_path']}).encode('utf-8'))

    def stop(self):
        self.is_running = False
        self.server_socket.close()
        for client in self.clients.values():
            client.close()
        print("Server stopped")

```

```

class RemoteControlClient:
    def __init__(self, server_host, server_port):
        self.server_host = server_host
        self.server_port = server_port
        self.client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

    def connect(self):
        try:
            self.client_socket.connect((self.server_host, self.server_port))
            self.listen()
        except Exception as e:
            print(f"Connection failed: {e}")

    def listen(self):
        while True:
            try:
                command = self.client_socket.recv(1024).decode('utf-8')
                command_data = json.loads(command)
                self.execute_command(command_data)
            except Exception as e:
                print(f"Error: {e}")
                break

    def execute_command(self, command_data):
        if command_data['action'] == "GET_SCREENSHOT":
            self.get_screenshot()
        elif command_data['action'] == "SHUTDOWN_DEVICE":
            self.shutdown_device()
        elif command_data['action'] == "RESTART_DEVICE":
            self.restart_device()
        elif command_data['action'] == "TRANSFER_FILE":
            self.receive_file(command_data['file_path'])

    def get_screenshot(self):
        os.system("adb exec-out screencap -p > /sdcard/screenshot.png")
        print("Screenshot captured")

    def shutdown_device(self):
        subprocess.run(["adb", "shell", "reboot", "-p"])
        print("Device shutting down")

    def restart_device(self):
        subprocess.run(["adb", "shell", "reboot"])
        print("Device restarting")

    def receive_file(self, file_path):
        file_name = os.path.basename(file_path)
        try:
            os.system(f"adb pull {file_path} ./received_files/{file_name}")
            print(f"File {file_name} received successfully")
        except Exception as e:
            print(f"Failed to receive file: {e}")

class DeviceMonitoring:
    def __init__(self, client_socket):
        self.client_socket = client_socket

    def send_status_update(self):
        while True:
            battery_level = self.get_battery_level()
            status_data = {
                "action": "STATUS_UPDATE",
                "battery": battery_level,
                "timestamp": time.time()
            }
            self.client_socket.send(json.dumps(status_data).encode('utf-8'))
            time.sleep(60)

```

```

def get_battery_level(self):
    result = subprocess.run(["adb", "shell", "dumpsys", "battery"],
capture_output=True, text=True)
    for line in result.stdout.splitlines():
        if "level:" in line:
            return int(line.split(":")[1].strip())
    return -1

class FileManagement:
    def __init__(self):
        pass

    def upload_file(self, client_socket, local_path, remote_path):
        try:
            os.system(f"adb push {local_path} {remote_path}")
            client_socket.send(json.dumps({"action": "FILE_UPLOADED", "status":
"success"}).encode('utf-8'))
            print("File uploaded successfully")
        except Exception as e:
            client_socket.send(json.dumps({"action": "FILE_UPLOADED", "status":
"failure"}).encode('utf-8'))
            print(f"File upload failed: {e}")

    def download_file(self, client_socket, remote_path, local_path):
        try:
            os.system(f"adb pull {remote_path} {local_path}")
            client_socket.send(json.dumps({"action": "FILE_DOWNLOADED",
"status": "success"}).encode('utf-8'))
            print("File downloaded successfully")
        except Exception as e:
            client_socket.send(json.dumps({"action": "FILE_DOWNLOADED",
"status": "failure"}).encode('utf-8'))
            print(f"File download failed: {e}")

class SecurityModule:
    def __init__(self):
        pass

    def encrypt_data(self, data):
        # Encrypt data using basic algorithm (for demonstration)
        return ''.join(chr(ord(char) + 2) for char in data)

    def decrypt_data(self, data):
        # Decrypt data using basic algorithm (for demonstration)
        return ''.join(chr(ord(char) - 2) for char in data)

# Main function to start the server
if __name__ == "__main__":
    server = RemoteControlServer("127.0.0.1", 9999)
    try:
        server.start()
    except KeyboardInterrupt:
        server.stop()

```

Authentication and Authorization (auth.py)

```
import hashlib

class AuthModule:
    def __init__(self):
        self.users = {
            "admin": self.hash_password("admin123")
        }

    def hash_password(self, password):
        hashed = hashlib.sha256(password.encode('utf-8')).hexdigest()
        return hashed

    def authenticate(self, username, password):
        hashed_password = self.hash_password(password)
        if username in self.users and self.users[username] == hashed_password:
            return True
        else:
            return False

    def add_user(self, username, password):
        if username not in self.users:
            self.users[username] = self.hash_password(password)
            return True
        else:
            return False

    def remove_user(self, username):
        if username in self.users:
            del self.users[username]
            return True
        else:
            return False

# Example usage
if __name__ == "__main__":
    auth = AuthModule()
    print(auth.authenticate("admin", "admin123")) # Should return True
    auth.add_user("user1", "password1")
    print(auth.authenticate("user1", "password1")) # Should return True
```

Encrypting Traffic (encryption.py)

```
from cryptography.fernet import Fernet

class EncryptionModule:
    def __init__(self, key=None):
        if key:
            self.key = key
        else:
            self.key = Fernet.generate_key()
            self.cipher_suite = Fernet(self.key)

    def encrypt_data(self, data):
        encrypted_data = self.cipher_suite.encrypt(data.encode('utf-8'))
        return encrypted_data

    def decrypt_data(self, encrypted_data):
        decrypted_data = self.cipher_suite.decrypt(encrypted_data).decode('utf-8')
        return decrypted_data

# Example usage
if __name__ == "__main__":
    encryption = EncryptionModule()
    secret_message = "This is a secret"
    encrypted = encryption.encrypt_data(secret_message)
    print(f"Encrypted: {encrypted}")
    decrypted = encryption.decrypt_data(encrypted)
    print(f"Decrypted: {decrypted}")
```

```
import logging

class LoggingModule:
    def __init__(self, log_file='server.log'):
        logging.basicConfig(filename=log_file, level=logging.INFO,
                            format='%(asctime)s - %(levelname)s - %(message)s')

    def log_info(self, message):
        logging.info(message)

    def log_warning(self, message):
        logging.warning(message)

    def log_error(self, message):
        logging.error(message)

# Example usage
if __name__ == "__main__":
    logger = LoggingModule()
    logger.log_info("Server started")
    logger.log_warning("Low memory warning")
    logger.log_error("Connection lost")
```

K6П3_2024

Remote Installation/Removal of Apps and Camera Control (device_management.py)

```
import subprocess

class DeviceManagement:
    def install_app(self, apk_path):
        try:
            subprocess.run(["adb", "install", apk_path], check=True)
            print("App installed successfully")
        except subprocess.CalledProcessError:
            print("App installation failed")

    def uninstall_app(self, package_name):
        try:
            subprocess.run(["adb", "uninstall", package_name], check=True)
            print("App uninstalled successfully")
        except subprocess.CalledProcessError:
            print("App uninstallation failed")

    def capture_photo(self, save_path):
        try:
            subprocess.run(["adb", "shell", "input", "keyevent",
                "KEYCODE_CAMERA"], check=True)
            subprocess.run(["adb", "pull",
                "/sdcard/DCIM/Camera/captured_photo.jpg", save_path], check=True)
            print("Photo captured and saved")
        except subprocess.CalledProcessError:
            print("Photo capture failed")

# Example usage
if __name__ == "__main__":
    device = DeviceManagement()
    device.install_app("/path/to/app.apk")
    device.capture_photo("/local/save/path/captured_photo.jpg")
```

Remote Sound Control (sound_control.py)

```
import subprocess

class SoundControl:
    def mute_volume(self):
        subprocess.run(["adb", "shell", "media", "volume", "--stream", "music",
"--set", "0"])

    def set_volume(self, level):
        subprocess.run(["adb", "shell", "media", "volume", "--stream", "music",
"--set", str(level)])

    def play_sound(self, file_path):
        subprocess.run(["adb", "push", file_path, "/sdcard/"])
        subprocess.run(["adb", "shell", "am", "start", "-a",
"android.intent.action.VIEW", "-d", "file:///sdcard/" + file_path])

# Example usage
if __name__ == "__main__":
    sound = SoundControl()
    sound.set_volume(5)
    sound.mute_volume()
    sound.play_sound("sound.mp3")
```

K6П3_2024

Real-Time Location Monitoring (location_monitoring.py)

```
import subprocess
import json

class LocationMonitoring:
    def get_location(self):
        result = subprocess.run(["adb", "shell", "dumpsys", "location"],
                                capture_output=True, text=True)
        location_data = self.parse_location_data(result.stdout)
        return location_data

    def parse_location_data(self, data):
        # Example of parsing raw location data
        location = {
            "latitude": "unknown",
            "longitude": "unknown"
        }
        for line in data.splitlines():
            if "mLatitude=" in line:
                location["latitude"] = line.split("=")[1].strip()
            elif "mLongitude=" in line:
                location["longitude"] = line.split("=")[1].strip()
        return location

# Example usage
if __name__ == "__main__":
    loc = LocationMonitoring()
    location = loc.get_location()
    print(f"Latitude: {location['latitude']}, Longitude:
    {location['longitude']}")
```

Remote File Access (file_access.py)

```
import os
import subprocess

class FileAccess:
    def list_files(self, directory):
        result = subprocess.run(["adb", "shell", "ls", directory],
            capture_output=True, text=True)
        return result.stdout.splitlines()

    def download_file(self, remote_path, local_path):
        os.system(f"adb pull {remote_path} {local_path}")

    def upload_file(self, local_path, remote_path):
        os.system(f"adb push {local_path} {remote_path}")

# Example usage
if __name__ == "__main__":
    file_access = FileAccess()
    files = file_access.list_files("/sdcard/")
    print("Files:", files)
    file_access.download_file("/sdcard/test.txt", "./test.txt")
```

K6П3_2024

Remote Software Update (software_update.py)

```
import subprocess

class SoftwareUpdate:
    def update_firmware(self, update_file):
        try:
            subprocess.run(["adb", "push", update_file, "/sdcard/update.zip"],
                check=True)
            subprocess.run(["adb", "shell", "reboot", "recovery"], check=True)
            print("Firmware update initiated")
        except subprocess.CalledProcessError:
            print("Firmware update failed")

# Example usage
if __name__ == "__main__":
    update = SoftwareUpdate()
    update.update_firmware("/path/to/update.zip")
```

K6П3_2024

Push Notifications (push_notifications.py)

```
import requests

class PushNotifications:
    def send_notification(self, device_token, title, message):
        headers = {
            'Authorization': 'Bearer YOUR_API_KEY',
            'Content-Type': 'application/json'
        }
        data = {
            "to": device_token,
            "notification": {
                "title": title,
                "body": message
            }
        }
        response = requests.post("https://fcm.googleapis.com/fcm/send",
headers=headers, json=data)
        if response.status_code == 200:
            print("Notification sent successfully")
        else:
            print(f"Failed to send notification: {response.status_code}")

# Example usage
if __name__ == "__main__":
    push = PushNotifications()
    push.send_notification("DEVICE_TOKEN", "Test Notification", "This is a
test")
```

Integration with Video Surveillance Systems (video_integration.py)

```
import cv2

class VideoSurveillance:
    def __init__(self, camera_url):
        self.camera_url = camera_url
        self.cap = None

    def start_stream(self):
        self.cap = cv2.VideoCapture(self.camera_url)
        if not self.cap.isOpened():
            print("Error: Could not open video stream")
            return

        while True:
            ret, frame = self.cap.read()
            if not ret:
                print("Failed to retrieve frame")
                break

            cv2.imshow("Video Stream", frame)

            if cv2.waitKey(1) & 0xFF == ord('q'):
                break

        self.cap.release()
        cv2.destroyAllWindows()

    def capture_frame(self, save_path):
        if self.cap:
            ret, frame = self.cap.read()
            if ret:
                cv2.imwrite(save_path, frame)
                print(f"Frame captured and saved to {save_path}")
            else:
                print("Failed to capture frame")

# Example usage
if __name__ == "__main__":
    surveillance = VideoSurveillance("http://camera-ip/video")
    surveillance.start_stream()
```

Device Usage Statistics Collection (device_statistics.py)

```
import psutil
import time
import json

class DeviceStatistics:
    def collect_statistics(self):
        stats = {
            "cpu_usage": psutil.cpu_percent(interval=1),
            "memory_usage": psutil.virtual_memory().percent,
            "battery": psutil.sensors_battery().percent if
psutil.sensors_battery() else "N/A"
        }
        return stats

    def save_statistics(self, filename):
        stats = self.collect_statistics()
        with open(filename, 'w') as file:
            json.dump(stats, file)
        print(f"Statistics saved to {filename}")

# Example usage
if __name__ == "__main__":
    stats = DeviceStatistics()
    stats.save_statistics("device_stats.json")
```

K6П3_2024

Group Device Management (group_management.py)

```
class GroupManagement:
    def __init__(self):
        self.groups = {}

    def create_group(self, group_name):
        if group_name not in self.groups:
            self.groups[group_name] = []
            print(f"Group '{group_name}' created")
        else:
            print(f"Group '{group_name}' already exists")

    def add_device_to_group(self, group_name, device_id):
        if group_name in self.groups:
            self.groups[group_name].append(device_id)
            print(f"Device '{device_id}' added to group '{group_name}'")
        else:
            print(f"Group '{group_name}' not found")

    def remove_device_from_group(self, group_name, device_id):
        if group_name in self.groups and device_id in self.groups[group_name]:
            self.groups[group_name].remove(device_id)
            print(f"Device '{device_id}' removed from group '{group_name}'")
        else:
            print(f"Group '{group_name}' or device '{device_id}' not found")

# Example usage
if __name__ == "__main__":
    gm = GroupManagement()
    gm.create_group("OfficeDevices")
    gm.add_device_to_group("OfficeDevices", "device_001")
    gm.remove_device_from_group("OfficeDevices", "device_001")
```

Database Integration for Command History (db_integration.py)

```
import sqlite3

class DatabaseIntegration:
    def __init__(self, db_name="commands.db"):
        self.conn = sqlite3.connect(db_name)
        self.create_table()

    def create_table(self):
        cursor = self.conn.cursor()
        cursor.execute("""
CREATE TABLE IF NOT EXISTS command_history (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    command TEXT,
    timestamp DATETIME DEFAULT CURRENT_TIMESTAMP
)
""")
        self.conn.commit()

    def save_command(self, command):
        cursor = self.conn.cursor()
        cursor.execute("INSERT INTO command_history (command) VALUES (?)",
(command,))
        self.conn.commit()
        print(f"Command '{command}' saved to history")

    def fetch_command_history(self):
        cursor = self.conn.cursor()
        cursor.execute("SELECT * FROM command_history")
        return cursor.fetchall()

# Example usage
if __name__ == "__main__":
    db = DatabaseIntegration()
    db.save_command("SCREENSHOT")
    print(db.fetch_command_history())
```

Multilingual Interface Support (multilingual.py)

```
import gettext

class MultilingualSupport:
    def __init__(self, locale_dir="locales", lang="en"):
        self.translations = gettext.translation('messages',
        locale_dir=locale_dir, languages=[lang])
        self.translations.install()
        self._ = self.translations.gettext

    def greet(self):
        print(self._("Hello, welcome to the system!"))

# Example usage
if __name__ == "__main__":
    multilingual = MultilingualSupport(lang="es")
    multilingual.greet()
```

K6П3_2024

Backup and Restore (backup_restore.py)

```
import os
import shutil

class BackupRestore:
    def backup_data(self, source_dir, backup_dir):
        try:
            shutil.copytree(source_dir, backup_dir)
            print(f"Backup created at {backup_dir}")
        except Exception as e:
            print(f"Backup failed: {e}")

    def restore_data(self, backup_dir, target_dir):
        try:
            shutil.copytree(backup_dir, target_dir, dirs_exist_ok=True)
            print(f"Data restored from {backup_dir} to {target_dir}")
        except Exception as e:
            print(f"Restore failed: {e}")

# Example usage
if __name__ == "__main__":
    br = BackupRestore()
    br.backup_data("/path/to/source", "/path/to/backup")
    br.restore_data("/path/to/backup", "/path/to/restore")
```