

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  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА**  
**за першим (бакалаврським) рівнем вищої освіти**  
на тему  
**“Програмне забезпечення системи батьківського контролю  
доступу до мережі”**

Виконав здобувач вищої освіти  
IV курсу, групи КІ-21-2  
ОПП «Комп’ютерна інженерія»  
спеціальності 123 «Комп’ютерна інженерія»  
\_\_\_\_\_ Берестенко Д.О.  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.

Керівник проекту  
кандидат технічних наук, доцент  
\_\_\_\_\_ Марченко К.М.  
« \_\_\_\_ » \_\_\_\_\_ 2025 р.  
Рецензент \_\_\_\_\_  
\_\_\_\_\_

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

ЗАТВЕРДЖУЮ

Завідувач кафедри

д.т.н., проф.

Олексій СМІРНОВ

« 17 » січня 2025 року

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

*Берестенку Даниїлу Олексійовичу*

(прізвище, ім'я, по батькові)

1. Тема роботи *Програмне забезпечення системи батьківського контролю доступу до мережі*

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

затверджені наказом вищого навчального закладу № 47-02 від 17.01.2025 року

3. Строк подання студентом роботи до захисту *23.05.2025 р.*

4. Мета та завдання випускної кваліфікаційної роботи: *Метою роботи є розробка програмного забезпечення системи батьківського контролю доступу до мережі*

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

*1. Призначення та область використання.*

*2. Перегляд аналогічних існуючих систем.*

*3. Опис і обґрунтування проектних рішень.*

*4. Етапи програмування системи.*

*5. Впровадження системи в промислову експлуатацію.*

*6. Висновки*

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

*Структурна схема системи* *1 аркуш*

*Функціональна схема системи* *1 аркуш*

*Діаграма процесів* *1 аркуш*

*Блок-схема алгоритму роботи додатку* *2 аркуша*

7. Дата видачі завдання « 17 » січня 2025 р.

### КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Строк виконання етапів випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти	Примітка
1.	Аналіз існуючих систем	10.03.2025 р.	
2.	Постановка задачі, оформлення ТЗ	15.03.2025 р.	
3.	Розробка моделі компонента	20.03.2025 р.	
4.	Розробка структур даних	25.03.2025 р.	
5.	Розробка алгоритмів зв'язку та відображення	30.03.2025 р.	
6.	Програмування алгоритмів	10.04.2025 р.	
7.	Оформлення ПЗ	17.04.2025 р.	
8.	Попередній захист роботи	23.05.2025 р.	

Дата видачі завдання  
« 17 » січня 2025 р.

Підпис керівника

Марченко К.М.  
(прізвище та ініціали)

Завдання прийнято до виконання  
« 17 » січня 2025 р.

Підпис здобувача

Берестенко Д.О.  
(прізвище та ініціали)

## АНОТАЦІЯ

**Берестенко Д.О. Програмне забезпечення системи батьківського контролю доступу до мережі. 123 Комп'ютерна інженерія. Центральноукраїнський національний технічний університет. Кропивницький. 2025.**

В даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти розроблено програмне забезпечення, яке призначено для системи батьківського контролю доступу до мережі.

Метою розробки є програмне забезпечення системи батьківського контролю доступу до мережі.

Результат роботи – програмна реалізація системи батьківського контролю доступу до мережі.

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

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

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

Програму розроблено в середовищі PHP, Perl.

**Ключові слова:** комп'ютерна інженерія, батьківський контроль, доступ до мережі

## ABSTRACT

**Berestenko D.O. Software for the parental control system for network access. 123 Computer Engineering. Central Ukrainian National Technical University. Kropyvnytskyi. 2025.**

In this final qualification work for the first (bachelor's) level of higher education, software has been developed, which is intended for the parental control system for network access.

The purpose of the development is the software for the parental control system for network access.

The result of the work is the software implementation of the parental control system for network access.

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

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

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

The program was developed in the PHP, Perl environment.

**Keywords:** computer engineering, parental control, network access

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ .....	2
ВСТУП.....	3
1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ .....	5
1.1 Призначення системи.....	5
1.2 Область застосування.....	6
2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ .....	7
2.1 Огляд існуючих систем, технологій, архітектур та програмних рішень за профілем теми випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.....	7
2.2 Обґрунтування вибору засобів для побудови системи та мови програмування.....	19
2.3 Розгорнута постановка завдання .....	23
3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ .....	25
3.1 Опис функціонування системи .....	25
3.2 Розробка структурної схеми.....	30
3.3 Розробка функціональної схеми .....	34
3.4 Розробка діаграми процесів.....	36
4 РЕАЛІЗАЦІЯ РОБОТИ. РОЗРАХУНКИ І ЕКСПЕРИМЕНТАЛЬНІ ДАНІ, ЩО ПІДТВЕРДЖУЮТЬ ВІРНІСТЬ ПРОЕКТНИХ ТА ПРОГРАМНИХ РІШЕНЬ.....	38
4.1 Розробка блок-схем та опис алгоритмів функціонування системи.....	38
4.2 Захист розробленого програмного забезпечення.....	55
5 ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ .....	58
6 ОСНОВНІ ВИСНОВКИ.....	62
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	64

					ВКРБ-123.25.0025.00.00.ПЗ			
Вим	Арк.	№ докум.	Підп.	Дата				
Розроб.	Берестенко Д.О.				Програмне забезпечення системи батьківського контролю доступу до мережі	Літ.	Аркуш	Аркушів
Перев.	Марченко К.М.					Б	1	70
Н.контр.	Коваленко А.С.				ЦНТУ КІ-21-2			
Затв.	Смірнов О.А.							

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ І ТЕРМІНІВ

ЛОМ	–	локальна обчислювальна мережа
MME	–	міжмережні екрани
ATM	–	асинхронний режим передачі
BSD	–	адаптована для Internet реалізація операційної системи UNIX
ICMP	–	міжмережний протокол управляючих повідомлень
IP	–	Internet Protocol – міжмережний протокол
NFS	–	мережна файлова система
PPP	–	протокол передачі від точки до точки
RFC	–	опис набору протоколів Internet
RPC	–	віддалений виклик процедури
SLIP	–	міжмережний протокол для послідовного каналу
SMTP	–	Simple Mail Transfer Protocol – простий протокол передачі пошти
TCP	–	Transmission Control Protocol – протокол управління передачею
UDP	–	User Datagram Protocol – протокол користувальницьких датаграм
UNIX	–	багатозадачна операційна система
UTP	–	незахищена вита пара
URL	–	уніфікований покажчик інформаційного ресурсу

## ВСТУП

**Актуальність теми.** Неймовірно, але факт, у боротьбі з комп'ютерними вірусами розбираються 90% батьків, а про те, як установити батьківський контроль у комп'ютері, знають тільки 5%. А способів усього три – блокування, моніторинг і обмеження за часом гри або по запуску певних додатків.

Взагалі, відповідно до визначення Вікіпедії, батьківський контроль – це комплекс правил і мір по запобіганню негативного впливу мережі Інтернет і комп'ютера на опікувану людину (звичайно – дитини).

І різні способи цього контролю пропонують і більшість телекомунікаційних операторів, і творці антивірусного ПЗ, убудовані ця система й у налаштування Windows. Є й спеціальні програми батьківського контролю – «Інтернет Цензор», K9 Web Protection, NetKids, NetPolice, KidGid, Content Keeper Express, Гогуль, «Кібермама». Деякі компанії взаємодіють одна з одною.

Найпоширеніші інтернет-погрози для родини й дітей: спам, фішинг, крадіжка особистих даних, крадіжка паролів, шахрайство, шкідливі програми, віруси, порнографія, образливі повідомлення й пропозиції сексуального характеру.

Системи дозволяють забороняти скачивание й видалення програм і файлів, а також забезпечують обмеження доступу в Інтернет для дітей у певний час доби.

Можна контролювати доступ до онлайн-ігор, вибираючи вік і тип змісту, що не слід дозволяти.

**Мета й завдання дослідження.** Метою роботи є програмне забезпечення системи батьківського контролю доступу до мережі.

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

- Огляд існуючих систем батьківського контролю доступу до мережі.

					ВКРБ-123.25.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		3

- Дослідження системи батьківського контролю доступу до мережі.
- Програмна реалізація системи батьківського контролю доступу до мережі.

**Практична цінність отриманих результатів** полягає в тому, що розроблені алгоритми дозволяють успішно вирішувати задачі батьківського контролю доступу до мережі.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи батьківського контролю доступу до мережі, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

КБПЗ\_2025

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		4

# 1 ПРИЗНАЧЕННЯ ТА ОБЛАСТЬ ВИКОРИСТАННЯ

## 1.1 Призначення системи

У багатьох системах рівні захисту можна вибирати й налаштовувати під свої потреби. Так, мінімальний рівень захищає комп'ютер або планшет від порнографії й погроз безпеки, високий – від усього дорослого змісту, нелегальної діяльності й невідомих сайтів. Гірше всього системи (майже всі) захищають від контенту з категорій «екстремізм», «секти» і «суїцид».

Блокування за часом закриває доступ до будь-яких сайтів. Наприклад, щодня комп'ютер блокується з 21.00 до 08.00, а в понеділок – на весь день. Можна встановити різні години доступу для кожного дня тижня.

Якщо до кінця дозволеного часу дитина усе ще в комп'ютері, система завершить роботу автоматично.

Недоліки в систем батьківського контролю теж є. У деяких систем часто бувають пробої, і тоді програма дозволяє доступ до сайтів, які дітям були б не потрібні. В інших системах немає українськомовного інтерфейсу, і багато українських сайтів доводиться вручну вносити в базу. У певних системах при спробі відкрити сайт із каталогу заборонених дитина попадає на сторінку з рекомендованим списком ресурсів, що добре, але якщо поставити на Інтернет для дітей максимальний фільтр, то закритється доступ до багатьох корисних ресурсів, просто тому що їх немає в основному каталозі системи, що вже не занадто радує. «Кібермама» контролює тільки час у мережі. В Windows виявляються забороненими багато дитячих сайтів, освітні й інформаційні портали й пошукові системи, тобто знайти щось вийде із працею.

Ще один загальний мінус майже для всіх систем – при блокуванні доступу до сайтів дитина одержує оповіщення про це. Якщо він не знав, що ви встановили таку програму, він може засмутитися, тому що такий контроль побічно говорить

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		5

про вашу недовіру до нього. Погодитися, це й дорослому неприємно. Виходом може стати попереднє обговорення всіх плюсів і мінусів мережі з поясненням, чому така система контролю принесе користь.

## 1.2 Область застосування

Технологія налаштування докладно розписується у всіх компаній, звичайно з картинками, що полегшують всі дії, тому встановити систему не становить праці. Головне, щоб у вас із дитиною були окремі облікові записи для входу в комп'ютер, у вас – із правами адміністратора, у дитини – із правами користувача. У мобільних операторів батьківський контроль підключається як окремий тариф через USSD-команду: зірочка – номер – грати – кнопка виклику або за допомогою SMS.

Якщо дитина ще маленька, але ви все-таки дозволяєте йому грати з планшетом, то зробити гаджет безпечним дозволять так звані пісочниці або лаунчери (sandbox або launcher). Це додатки для налаштування робочого стола планшета або смартфона, дитина зможе грати в розвиваючі ігри, учитися й читати книжки, але при цьому буде закритий доступ до системних налаштувань, використанню Wi-Fi і деяких додатків. Крім усього іншого є ще й відстеження по GPS, віддалений контроль і синхронізація фотографій з вашим аккаунтом, щоб перевіряти, що знімають ваші діти. Майте на увазі, що обійти всі системи контролю діти, що вийшли з наївного віку, можуть із ангельською простотою – вони підуть до друга або вийдуть у мережу в громадських місцях. Тобто заборони потрібно підтримувати профілактикою й поясненнями ризиків в інтернеті, так, щоб рішення про власний захист дитина приймала сама.

Таким чином, виходячи з вищеперерахованого, програмне забезпечення системи батьківського контролю доступу до мережі, є актуальною задачею, яка потребує вирішення у даній випускній кваліфікаційній роботі за першим (бакалаврським) рівнем вищої освіти.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		6

## 2 ПЕРЕГЛЯД АНАЛОГІЧНИХ ІСНУЮЧИХ СИСТЕМ

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

Сфера застосування подібного ПЗ може бути розширена, включати практично будь-якого користувача, якому потрібно обмежити доступ до локальних або мережних функцій комп'ютера. Даний матеріал є доповненням огляду, у якому вже висвітлювалися деякі аспекти батьківського контролю.

#### **ChildWebGuardian Pro**

Після установки програми буде потрібно налаштувати первинні параметри: пароль, вказати email-адреса для відправлення звітів, вибрати профіль комп'ютера – домашній, шкільний або офісний. Далі, після уведення пароля, користувач попадає в секцію налаштувань програми, де може управляти групами користувачів з обмеженими правами й адміністраторів. Для зміни ролі досить перенести ім'я системного профілю в потрібну групу. Також нескладно створити додаткові профілі, які можуть знадобитися для більш-менш строгого контролю дій.

У розділі «Заборонені слова» ведеться список стоп-слів, по яких блокується зміст інтернет-сторінок, email-повідомлень і переписка в ICQ. Тут можна створювати власні групи слів, а також регулювати максимальну кількість заборонених слів, прийнятних у змісті тексту.

У розділі «Фільтр Веб» перебувають білі й чорні списки сайтів, куди можна додати інші адреси. Одна з найбільш актуальних опцій – блокування соціальних мереж, у числі яких є «ВКонтакте», «Однокласники» і Facebook. Ні опції для відключення всіх соцмереж одним кліком, також недоступні способи

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		7



адрес, які не будуть блокуватися при передачі. А от застарілий ICQ, видимо, розроблювачам варто доповнити іншими ІМ-клієнтами, у числі яких – значно більше популярний Skype із власним протоколом передачі. Тому дивно, що ICQ в ChildWebGuardian Pro приділяється занадто багато уваги. З іншого боку, як варіант, можна скористатися сусіднім розділом – «Фільтр по програмах», де можна заблокувати будь-який додаток.

Вертаючись до звітів. Якщо виникає необхідність віддаленого контролю й запису дій користувача в журнал, можна зайти в однойменний розділ і активувати опцію «Записувати події в журнал». У лог-файл фіксуються такі події, як відвідані сайти, заблокований зміст із заборонених списків. Також у даному розділі можна налаштувати відправлення звітів на email, зазначений відразу після установки програми. Мабуть, не вистачає більше розгорнутої статистики за часом, проведеному в Інтернеті/системі, запису в журнал уведених сполучень клавіш і подібної інформації – інструментів, які виявляються в інших програмах огляду.

Можна вільно ознайомитися з усіма функціями без обмежень і нав'язливих повідомлень. Крім того, є безкоштовна й менш функціональна редакція програми.

### **Резюме**

ChildWebGuardian – програма із гранично інтуїтивним інтерфейсом і організацією розділів. У ній присутні базові опції, які допоможуть батькам обмежити доступ дітей до інтернету й програм, іграм. Основний упор зроблений на мережну безпеку.

[+] Можливість заборони доступу до соціальних мереж.

[+] Українськомовний інтерфейс.

[-] Немає розширеної статистики по діях користувача.

[-] Немає заборони Skype і інших актуальних протоколів, крім ICQ.

### **Hidetools Parental Control**

В Hidetools Parental Control убачається чітке розмежування функцій на 3 модулі: звіти, обмеження контролю й налаштування. Доступ до кожного з

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		9

пунктів цих розділів доступний через меню в головному вікні, а також у вкладках.



Рисунок 2.2 – Hidetools Parental Control

У першому розділі, Reports, можна детально ознайомитися зі статистикою: запуск додатків, відвідані сайти, введення тексту із клавіатури (з поділом по відповідних програмах, де відбувався набір), скриншоти активних вікон. Одним словом, функція звітності реалізована досить повно. Все питання в тім, як використовувати можливості програми, не «перегинаючи ціпок», адже, по суті, багато програм, і ця в тому числі, нічим не відрізняються від кейлоггерів і інших «шпигунських» програм.

Другий розділ, Controls, не стільки контролює, скільки накладає всілякі обмеження на дії користувача. Втім, залишилося незрозумілим, як заборонені категорії у вкладці Website Blocking співвідносяться з веб-сайтами. Зважаючи на все, вони існують роздільно друг від друга. У цій же секції налаштувань можна

також створювати білі й чорні списки ресурсів, додавши адреси/ключові слова в потрібний список.

У відмінність, скажемо, від ChildWebGuardian, немає розкладу, по якому можна регулювати доступ до того або іншого додатка, сайту. З наявних засобів, доступно лише обмеження загального доступу до ПК (PC Usage Limits), цю опцію можна знайти в Time Controls > Time Limits.

У вкладці Application blocking можна заборонити запуск зазначених додатків (найчастіше, комп'ютерних ігор) або вибрати варіант зі списку популярних браузерів і месенджерів. В Windows Security доступне блокування Панелі керування, Диспетчера пристроїв, Редактору реєстру й інших системних компонентів, які відкривають «лазівку» користувачеві з обмеженими правами (будь те дитина або будь-яка стороння людина).

У модулі налаштувань слід зазначити кілька цікавих опцій. По-перше, для кожного акаунту у вкладці User Account можна активувати правила, які будуть застосовуватися під час сеансу. Сюди входять: моніторинг, фільтр веб-сайтів і додатків, контроль за часом і системною безпекою – тобто, уже згадані опції. Втім, у даному розділі для повної зручності не вистачає груп і механізму керування ними (хоча б такого спрощеного, як в ChildWebGuardian Pro).

По-друге, доступне детальне налаштування статистики в Monitoring Settings, у третій – відправлення звітів на email адміністратора, з певним інтервалом.

14 днів Hidetools Parental Control доступна в ознайомлювальному режимі.

### **Резюме**

Програма Hidetools Parental Control призначена для спостереження за активністю користувача (функція моніторингу) і обмеження доступу до системних налаштувань і Мережі. Якщо моніторингова частина досить функціональна, по обмеженню доступу хотілося б бачити більше опцій – інакше, грубо говорячи, програма має «шпигунську» спрямованість.

[+] Функціональний модуль моніторингу.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		11

[+] Зручне перемикання захисту в активний/неактивний режим.

[-] Слабко розвинене керування профілями.

[-] Немає заборони доступу за розкладом для окремих сайтів.

### **KinderGate Parental Control**

KinderGate Parental Control – додаток для обмеження доступу дітей до веб-ресурсів, розробка відомої в сфері інтернет-безпеки компанією Entensys. Містить у собі ряд інструментів для застосування обмежень за розкладом, фільтрації по категоріях сайтів, блокування небезпечних сайтів, а також моніторинг і статистику.

Під час інсталяції, KinderGate вимагає встановити пароль на вхід в інтерфейс адміністрування й вибрати рівень доступу, що визначає, наскільки строгою буде фільтрація інтернет-ресурсів.

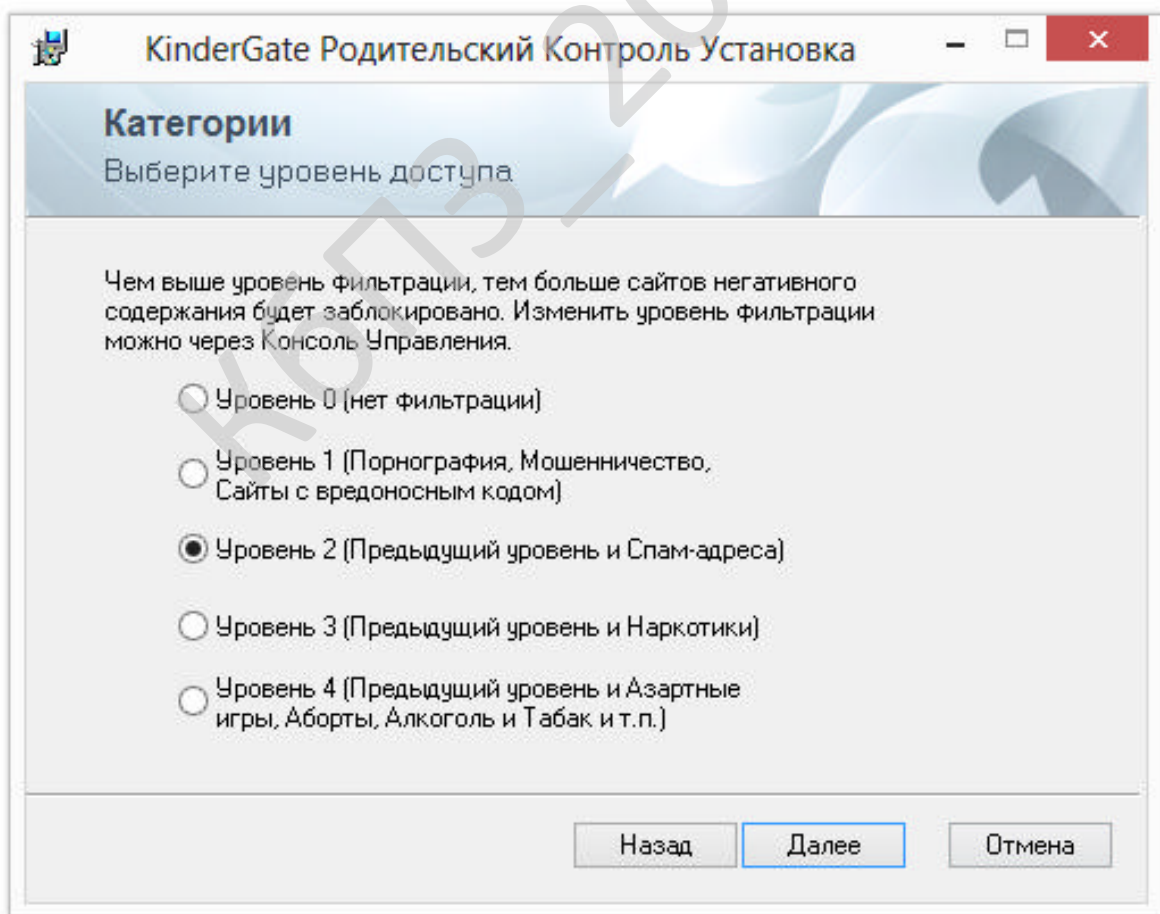


Рисунок 2.3 – Налаштування рівня фільтрації в KinderGate Parental Control



Налаштування розкладу доступне в розділі «Розклад роботи». Обмеження може бути настроєне за графіком або в довільному форматі. На жаль, не можна вказати ліміт за часом без прив'язки до графіка.

В Parental Control є інструмент за назвою «Розширені правила», що дозволяє сполучити кілька правил одночасно: для заборони доступу до протоколу, сайтам, із прив'язкою до розкладу й ін. На жаль, офлайн документація не приводить достатньої кількості повноцінних прикладів, які говорили б про практичну користь від застосування подібних правил. В онлайн довідці прикладів більше.

Невелика, але важлива ремарка. Установлювати й видаляти KinderGate PC потрібно дуже уважно. Робота додатка торкається мережних драйверів і системних сервісів, відключення яких спричинить неприємні наслідки, зокрема – обмеження доступу до Мережі. Це ж правило стосується всіх згаданих в огляді програм. Не слід установлювати їх більше однієї-двох одночасно щоб уникнути конфліктів.

### **Резюме**

KinderGate Parental Control – програма із передвстановленою системою фільтрів і можливістю налаштування доступу за розкладом. Постачена підказками, довідкою. Одне з найбільше з оптимальних рішень по позначеній проблемі.

- [–] Проблематичне видалення.
- [–] Обмежена функціональність налаштування доступу за розкладом.
- [+] База фільтрів, що обновляється.
- [+] Комбіновані правила.

### **Kids PC Time Administrator**

Функції Kids PC Time Administrator, відповідно до назви, сфальцьовані на обмеженні доступу за розкладом.

Споконвічно створюється враження, що програма має у своєму розпорядженні майстра налаштування. Однак кнопки переходу, як виявилось,

призначене тільки для навігації між розділами. Робити налаштування у встановленому порядку або переходити в кожний розділ за допомогою меню – принципової різниці немає.

Для кожного користувача можна вказати індивідуальний розклад, вибір профілю доступний у кожному розділі. Крім того, у деяких випадках простіше встановити однакові обмеження для всіх.

Налаштування розкладу здійснюється в розділі «Дозволені періоди часу», де за допомогою календаря можна дозволити або заборонити доступ із щохвилинною точністю. Якщо даний варіант обмежень адміністратор порухує незручним, можна перейти в сусідній розділ і встановити кількість годин у день, які будуть у розпорядженні в користувача.

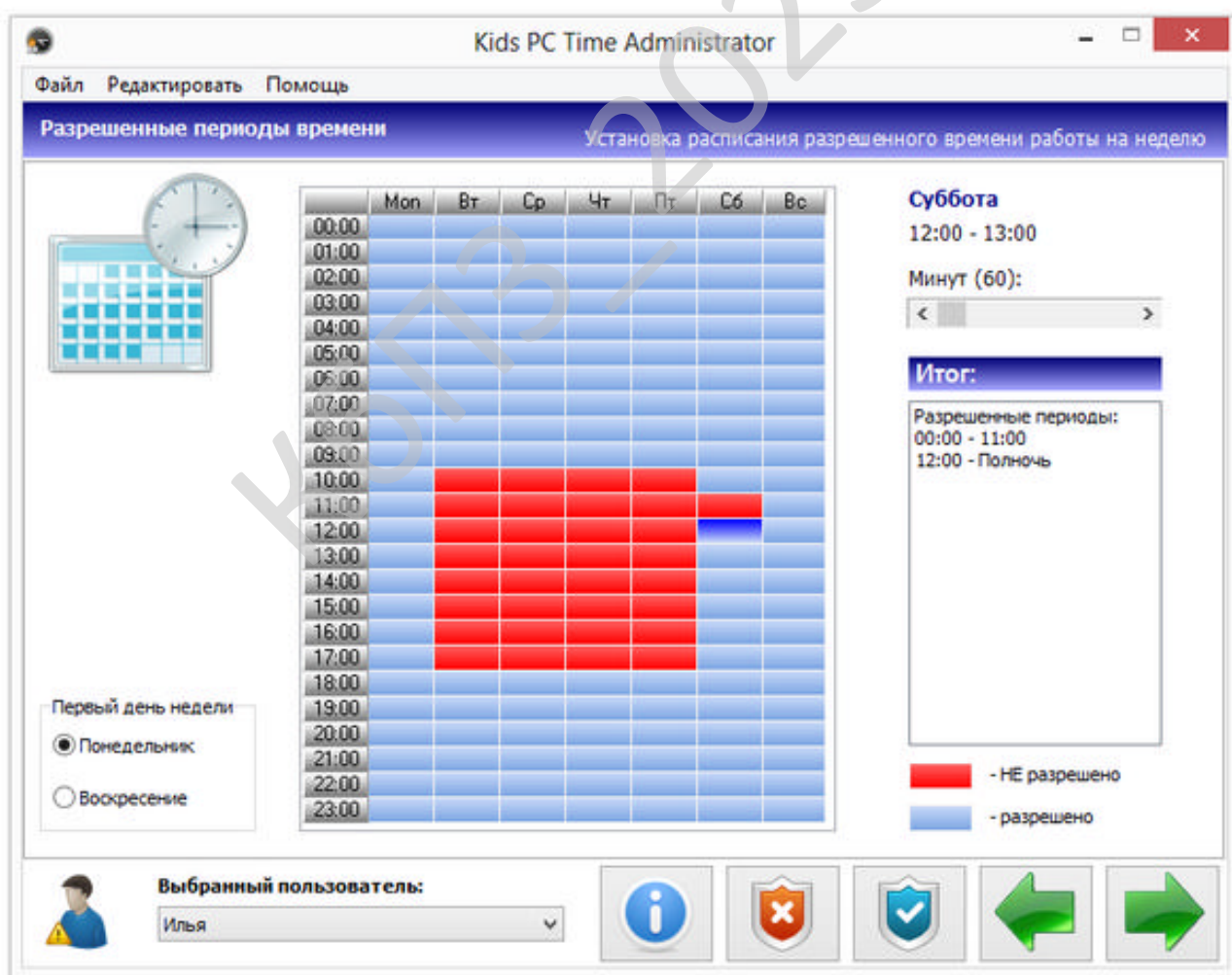


Рисунок 2.5 – Налаштування розкладу в Kids PC Time Administrator

Певна частина налаштувань Kids PC Time Administrator відноситься до звичних системних «твиків»: блокування Диспетчера пристроїв, видалення програм через панель керування, заборона на керування обліковими записами й т.п. Тут вибір опцій досить скромний. Ще один обмежуючий розділ, не пов'язаний з розкладом – «Захист папок». Засобами програми можна сховати обрані папки з важливими даними або захистити їх від зміни.

Деякі інструменти для ведення звітності в Time Administrator у наявності, але от відсутність яких-небудь налаштувань, віддаленого контролю й відправлення балок на email – все це говорить про слабку реалізацію інших сторін програми.

### **Резюме**

Kids PC Time Administrator можна задіяти в тому випадку, якщо дитина проводить занадто багато часу за комп'ютером і сеанс потрібно обмежити. Фільтрів, заборонених списків і інших додаткових інструментів тут не передбачено.

[+] Широкі можливості обмеження за часом.

[-] Відсутність фільтрів.

[-] Незручний інтерфейс календаря.

### **Child Control 2013**

На відміну від Kids PC Time Administrator, програма Child Control 2013 дозволяє обмежувати не тільки доступ до системних ресурсів, але також доступ до Інтернету за розкладом. Це й подібні завдання спрощує покроковий майстер. Коротенько, усього 4 секції з параметрами:

– Time Limits: погодинне обмеження роботи в системі/в Інтернеті.

– Blocked Times: часові інтервали, коли доступ до системи/Інтернету буде заблокований.

– Internet Control: інтернет-фільтр, що містить у списках дозволені й заборонені ресурси. Опціонально, можна дозволити доступ тільки до сайтів з

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		16



У сусідньому розділі «Programs» – практично ті ж опції, з тією лише різницею, що вони застосовні до програм. Тут також можна управляти забороненими й дозволеними списками, налаштовувати запуск за розкладом (група додатків Time-limited). Для Windows 8 передбачена відповідна вкладка, одним кліком блокуються всі Windows 8 Apps, а також Магазин Windows.

У підрозділі «Windows Security» роздягнула «Security» здійснюється блокування системних розділів, сторонній доступ до яких небажаний. Причому Child Control 2013 має у своєму розпорядженні досить широкий набір опцій: відключення безпечного завантаження, налаштувань часу, зміни користувальницьких акаунтів і т.п.

Розділ «Usage Log» надає детальну статистику по використанню системних ресурсів, а також відвідуванню веб-сайтів. Статистика містить у собі інформацію за день, тиждень і місяць. Для зручності її можна відкрити в переглядачі по натисканню кнопки «Report». Крім того, можливе відправлення статистики на зазначений email (див. «E-mail Report»).

В Child Control 2013 розроблювачі передбачили велику кількість додаткових налаштувань. Розділ «Settings» дозволяє налаштовувати повідомлення й інші повідомлення користувачеві, поведження програми при вичерпанні лімітів; можливе розширене налаштування фільтрів, керування групами.

Нарешті, варто звернути увагу на таку можливість, як віддалене керування програмою через браузер – для її активації призначений розділ «Remote Control».

### **Резюме**

Одна з найбільш функціональних програм даного огляду, у тому числі по кількості додаткових налаштувань. Одна із привабливих сторін Child Control 2013 – звітність у вигляді графіків.

[+] Детальність звітів, графіки.

[+] Універсальність.

[+] Віддалене керування.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		18

## 2.2 Обґрунтування вибору засобів для побудови системи та мови програмування

### PHP

Для написання програмного забезпечення була обрана мова програмування PHP.

PHP – це мова серверних скриптів (server scripting language), що вбудовується в HTML, який інтерпретується і виконується на сервері. PHP працює як частина Web-сервера, і цим самим схожий на ASP від Microsoft або Coldfusion від Allaire. Синтаксис PHP дуже схожий на синтаксис таких мов програмування C або Perl. Люди, що мають деякий досвід програмування, дуже швидко зможуть почати писати програми на PHP. У цій мові немає строгої типізації даних і немає необхідності в діях з виділення/звільнення пам'яті. Програми, написані на мові програмування PHP, читаються достатньо легко. На відміну від Perl-программ PHP-коду властива легка читабельність та зрозумілість.

PHP є препроцесором HTML.

До того, як сервер відправить файл браузеру, його проглядає препроцесор-інтерпретатор. Для того, щоб це відбувалося, файли, які піддаються обробці препроцесором, повинні мати визначене розширення (звичайно це .phtml або .php3, але ці значення можна поміняти) і містити (хоча це не обов'язкова вимога) код для препроцесора. Перед відправкою сторінки PHP-код програється на сервері і браузеру видається результат у вигляді знову таки HTML-сторінки, яка може сильно відрізнитися від тієї, що зберігається на сервері. Звичайні ж сторінки, що мають розширення .html/.htm Web-сервер буде відправляти браузеру без будь-якої обробки. Основна відмінність від CGI-скриптів, написаних на інших мовах, типу Perl або C – це те, що в CGI-програмах розробник самостійно пише HTML-код, що виводиться, а, використовуючи PHP – вбудовує свою програму-скрипт в готову HTML-сторінку, використовуючи відкриваючий і закриваючий теги (<?php та ?>). PHP називається мовою серверних скриптів на відміну від

					ВКРБ-123.25.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		19

Javascript/jscript/vbscript, які є мовами клієнтських скриптів. Це означає, що PHP-скрипт виконується на сервері, а клієнтові передається результат його роботи, тоді як в JavaScript, код повністю передається на клієнтську машину і лише там виконується браузером.

При написанні на інших мовах програмування, наприклад, на Perl або C – замість того, щоб створювати програму, яка займається формуванням HTML-коду і містить незліченну безліч призначених для цього команд, PHP-програміст має можливість створювати HTML-код з декількома упродовженими командами PHP. Код PHP відділяється спеціальними початковим і кінцевим тегами, які дозволяють процесору PHP визначати початок і кінець ділянки HTML-коду, що містить PHP-скрипт.

Значною відмінністю PHP від якого-небудь коду, що виконується на стороні клієнта, наприклад, Javascript, є те, що PHP-скрипти виконуються на сервері. PHP-скрипт розміщений на сервері, клієнт отримує тільки результат виконання скрипта, причому клієнт не має можливості з'ясувати, який саме код виконується. Існує також можливість конфігурувати власний сервер так, щоб HTML-файли оброблялися процесором PHP, так що клієнти навіть не в змозі дізнатися, чи отримують вони звичайний HTML-файл, чи це результат виконання скрипта.

Мова програмування PHP досить проста для освоєння, але разом з тим здатна задовольнити запити професійних програмістів. Хоча PHP, головним чином, призначений для роботи в середовищі web-серверів, область його застосування не обмежується тільки цим.

Можливості мови PHP дуже великі. Головним чином, область застосування PHP сфокусована на написання скриптів, що працюють на стороні сервера; таким чином, PHP здатний виконувати все те, що виконує будь-яка інша програма CGI, наприклад, обробляти дані форм, генерувати динамічні сторінки або посилати і приймати cookies. Окрім вищенаведеного PHP здатний виконувати і безліч інших завдань.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		20

Існують такі основні області, де використовується PHP.

– Створення скриптів для виконання на стороні сервера. Саме таким чином мова програмування PHP найширше використовується. Все, що може знадобитися програмісту – це парсер PHP (у вигляді програми CGI або серверного модуля), вебсервер і браузер. Для того, щоб існувала можливість проглядати результати виконання PHP-скриптів у браузері, потрібен працюючий веб-сервер і встановлений PHP.

– Створення скриптів для виконання в командному рядку. Існує можливість створити PHP-скрипт, здатний запускатися незалежно від веб-серверу та браузера. Все, що потрібно – це парсер PHP. Такий спосіб використання PHP ідеально підходить для скриптів, які повинні виконуватися регулярно, наприклад, за допомогою cron (на платформах Unix або Linux) або за допомогою планувальника завдань (Task Scheduler) на платформах Windows. Ці скрипти також можуть бути використані в завданнях простої обробки даних.

Існує величезна кількість документації і списків розсилки, до яких можна звернутися у разі виникнення яких-небудь питань.

### **Perl**

Perl – високорівнева, інтерпретована, динамічна мова програмування загального призначення. Perl запозичує можливості з багатьох інших мов програмування, як то C, shell scripting, AWK та sed. Мова надає потужні можливості для обробки тексту без довільних обмежень на довжину даних багатьох сучасних інструментів Unix, полегшуючи процес маніпуляції текстових файлів. Використовується для програмування графіки, системного адміністрування, у мережному програмуванні, у написанні програмного забезпечення, яке взаємодіє з базами даних, у програмуванні CGI для веб. Perl за свою гнучкість і потужність отримав прізвисько «швейцарського армійського ножа мов програмування».

Perl – мова програмування загального призначення, котра на початку розроблялась, як інструмент для обробки тексту, і тепер використовується для

вирішення дуже широкого кола завдань, включно із системним адмініструванням, веб-розробкою, розробкою мережного програмного забезпечення, та програмного забезпечення з графічним інтерфейсом користувача.

Мова віддає перевагу практичному (простота у використанні, ефективність, повнота) над гарним (крихітність, елегантність, мінімалістичність). Вона має багато можливостей, включно із підтримкою кількох парадигм програмування(процедурне програмування, об'єктно-орієнтоване програмування, а також функціональне програмування), управлінням пам'яттю, вбудованою підтримкою системи обробки текстів, і величезною колекцією сторонніх модулів.

Загальна структура програм на мові Perl була запозичена у C. Perl є типовою процедурною мовою, зі змінними, виразами, присвоюваннями, дужками-розділювачами, блоками, циклами та підпрограмами.

Perl також запозичив деякі особливості з shell-програмування. Назви усіх змінних розпочинаються зі спеціальних знаків, кожен з яких характеризує певний тип даних (наприклад, скаляри, масиви та хеш-таблиці). Важливо зазначити, що це дозволяє правильно інтерпретувати назви змінних безпосередньо у текстових рядках. Perl має багато вбудованих функцій, які забезпечують програміста інструментарієм, котрий часто використовується у shell-програмуванні (хоча, багато з цих інструментів реалізовується за допомогою зовнішніх інструментів), як то сортування або виклик зовнішніх програм.

Perl запозичив списки з мови Lisp, асоціативні масиви (хеші) з AWK, і регулярні вирази з sed. Це спростило процес парсингу і обробки великих обсягів текстової інформації.

У Perl 5, була додана підтримка комплексних структур даних, функції першого класу, і об'єктно-орієнтованої моделі програмування. Що включає у себе посилання, пакунки, та інше. Серед найбільших нововведень у Perl 5 слід відмітити наявність модулів, які дозволяють повторно використовувати один і той же код.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		22

Усі версії Perl здійснюють автоматичне типізування змінних і управління пам'яттю. Інтерпретатор знає про тип та способи зберігання усіх об'єктів даних у програмі.

Дизайн Perl можна описати, як відповідь на три основні тенденції в комп'ютерній індустрії: зниження вартості апаратного забезпечення, зростання вартості робочої сили, а також вдосконалення технології компіляторів. Багато ранніх мов програмування, як то Fortran та C, були розроблені таким чином, щоб якомога ефективніше використовувати на той час дороге апаратне забезпечення. У протилежність, Perl було розроблено для підвищення ефективності роботи дорогих в наш час програмістів.

Perl має багато можливостей, які збільшують ефективність програміста за рахунок інтенсивного використання мікропроцесора та великих обсягів оперативної пам'яті. Серед них: автоматичне управління пам'яттю; динамічна типізація; стрічки, списки, та хеші; регулярні вирази; самоаналіз; та функція eval().

Perl підтримує мовні конструкції, які є короткими і природними для людей при їхньому читанні і написанні, навіть якщо це ускладнює реалізацію інтерпретатора Perl.

Синтаксис Perl має багато спільного з синтаксисом мов Cі, Awk, Sed і Shell. Перший рядок початкового коду може починатися з «#!/Шлях/до/Perl [-ключі]» – що указує системі шлях до інтерпретатора Perl для виконання програми в \*NIX системах і виконання їх на Веб-сервері.

### 2.3 Розгорнута постановка завдання

Згідно з технічним завданням на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, реалізації підлягає програмне забезпечення, яке призначено для системи батьківського контролю доступу до мережі.

					ВКРБ-123.25.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		23

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

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

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

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

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

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

е) провести розрахунки по визначенню економічної ефективності розробленої системи;

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

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

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ док.ум.	Підпис	Дата		24

## 3 ОПИС І ОБҐРУНТУВАННЯ ПРОЕКТНИХ РІШЕНЬ

### 3.1 Опис функціонування системи

Найчастіше батьки не усвідомлюють всю проблему «дітей, що пропадають» у Мережі. А зштовхнувшись із цим, губляться в пошуках рішення. У розділі розглянуті погрози, які можуть перерости в більші неприємності при безконтрольному веб-серфінгу вашої дитини, а також запропоновано кілька рекомендацій, які допоможуть упоратися з виниклою проблемою без жертв.

#### Погрози при безконтрольному використанні інтернету

На сучасний момент вік інтернет-користувача знизився настільки, що часом п'ятирічні малята звертаються з комп'ютером і мобільними пристроями більш спритно, ніж дорослі.

Вся причина в стрімкому розвитку технологій і їхньої інтеграції в повсякденне життя людей.

Якщо розглядати доросле покоління, то можна із упевненістю сказати, що більша частина все-таки розуміє всі ризики й наслідки роботи з інтернетом.

А як же діти? Чи усвідомлюють вони погрози, які підстерігають їх на кожному кроці?

#### Шкідливі програми

До шкідливих програм ставляться погрози, здатні завдати значної шкоди комп'ютеру й перехопити персональні дані, у тому числі паролі від банківських систем.

От лише деякі з дій дитини в Мережі, які можуть привести до проблем:

- перехід на шкідливий сайт;
- завантаження файлів з файлообмінних ресурсів;
- завантаження піратських матеріалів (музика, відео);

					ВКРБ-123.25.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		25



Така інформація часто зустрічається в рекламних баннерах.

На сторінках соціальних мереж можна знайти цілі групи користувачів, які обговорюють питання суїциду й самокатування.

Одна 13-літня дівчинка пише про те, як вона утомилася від життя й мріє про те, щоб заснути назавжди. Інша, порізавши себе, виклала фото в Мережу з написом: «Тивсезрозумієш, алебудепізно».

Інформація про дитячу порнографію, сексуальні збочення найчастіше втримується на файлообмінних ресурсах, а також у торрентах у вигляді краплень, окремих сторінок.

І в таких випадках перегляд може виявитися неминучим.

У даний момент ведеться досить активна боротьба з подібними ресурсами. Але погроза усе ще залишається.

### **Домагання**

Подібні випадки звичайно відбуваються через онлайн-чати, соціальні мережі, мережні ігрові площадки та інші ресурси, що дозволяють користувачам вступати в контакт.

Так, відповідно до джерела, недавно жертвою 25-літнього педофіла стала 11-літня дівчинка. Познайомившись зі своєю жертвою в соціальній мережі й вступивши з нею в «довірчі відносини», він відправляв їй матеріали еротичного змісту й шантажем вимагав інтимні знімки.

Слід зазначити, що таке спілкування не обмежується онлайн-розмовами.

Як затверджують дослідники, діти досить часто зустрічаються зі своїми новими знайомими в реальності.

За результатами дослідження, близько 50% українських дітей призначають реальні зустрічі.

Однак замість подружки із чата на зустріч може прийти доросла людина з кепськими намірами, а інший раз такий «друг» може запропонувати спробувати наркотики. І все, що знають батьки – це вигаданий никнув нового знайомого. І проте багато батьків не вважають потрібним контролювати життя дітей у Мережі.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		27

За результатами дослідження, проведеного 2015 року компанією Eset, 51% батьків зовсім відмовилися від онлайн-спостереження за своєю дитиною в соціальних мережах.

### **Інтернет-залежність**

Ця залежність практично невиліковна для дитини. У цьому випадку фахівці радять тільки батьківську любов і увагу.

### **Можливі рішення проблеми**

Отже, які ж можливі рішення? Заборони завжди діяли тільки до певного віку, а для деяких дітей їх зовсім не існує.

Бували навіть випадки, коли діти підліткового віку знаходили способи обходу встановленої батьками програмного захисту заради того, щоб відвідати ресурс із чорного списку. Тому необхідні комплексні міри.

### **Бесіда з дитиною**

Фахівці рекомендують у першу чергу поговорити зі своєю дитиною й обговорити виниклі проблеми, причини їхньої появи й можливі наслідки.

Обговоріть думки фахівців, покажіть кілька прикладів неправильного поведіння в інтернеті й до яких результатів це все може привести.

Це дозволить нейтралізувати дух протиборства й відторгнення всього того, що ви скажете.

### **Технічні засоби**

Прийміть всі необхідні міри безпеки при роботі з паролями, доступом у Мережу.

Рекомендується стежити за відновленнями безпеки операційної системи й додатків. Ця міра дозволить мінімізувати ризик зараження комп'ютера, забезпечить технічний захист від можливих неприємностей.

Встановіть антивірусні продукти з функцією «Батьківського контролю» на комп'ютер і мобільні пристрої, це дозволить:

– обмежити час знаходження дитини в Мережі й користування комп'ютером;

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		28

- створити графік із припустимими годинниками роботи протягом дня;
- заблокувати сайти із забороненим контентом – створити чорні списки на основі баз даних антивірусного виробника по категоріях (наркотики, соціальні мережі й т.д.) і створити білі списки;
- установити обмеження на запуск додатків (наприклад, ігор) і установку нових програм;
- стежити за перепискою дитини в соцмережах і чатах (ICQ, Skype і т.п.);
- контролювати пересилання особистої інформації (в основному, це платні пакети послуг).

Для мобільних пристроїв на ринку представлені рішення, що дозволяють відслідковувати дзвінки, SMS-діалоги дитини, переписку в месенджерах, а також одержувати інформацію про його місцезнаходження.

Крім того, існують послуги «Батьківського контролю» від інтернет-провайдерів.

Так, наприклад, деякі компанії надають послугу, що дозволяє батькам вводити обмеження за часом перебування в інтернеті дитини в їхню відсутність, і надає звіти про переглянуті сайти.

### **Правила**

Обговоріть правила користування інтернетом і обмежте час присутності в Мережі. Погодьтеся, що ви не зможете увесь час контролювати своєї дитини й постійно перебувати поруч із ним. Але ви зможете встановити правила, при яких обидві сторони залишаться задоволені:

- Призначити час онлайн-присутності й установити норми на завантаження тих або інших файлів, фільмів і програм.
- Контролюйте дітей онлайн.
- Зареєструйтеся в соціальній мережі й установите контакт із вашою дитиною.
- Час від часу переглядайте сторінку на предмет появи там небезпечних повідомлень.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		29

На закінчення відзначимо, що захистити дитину від усього на світі неможливо. Однак можна навчити її, як правильно приймати рішення, якщо все-таки вона потрапила в негарну ситуацію.

### 3.2 Розробка структурної схеми

Система батьківського контролю доступу до мережі дозволяє розподілити обчислювальне навантаження на будь-яке число серверів. На рисунку 3.1 зображена структурна схема розробленої системи. На ньому показано конфігурацію, у якій сервіси FTP і HTTP обслуговуються декількома серверами.

Розміщення системи батьківського контролю доступу до мережі як граничного пристрою між внутрішньою мережею (інтранетом) і Інтернетом дозволяє контролювати весь вихідний і вхідний Інтернет-трафік і управляти його проходженням. Між внутрішньою й зовнішньою мережею створюється чіткий захисний рубіж. Однак деяким зовнішнім клієнтам може знадобитися доступ до внутрішніх ресурсів. Для цього можна передбачити демілітаризовану зону (DMZ).

У комп'ютерних мережах демілітаризованою зоною називається ділянка мережі, доступна як внутрішнім, так і зовнішнім користувачам. Вона більше захищена у порівнянні із зовнішньою мережею, але менш захищений у порівнянні із внутрішньою мережею. DMZ створюється за допомогою одного або декількох міжмережних екранів, побудованих за технологією NGFW, що розмежовують внутрішню мережу, DMZ і зовнішню мережу. В DMZ часто розміщуються веб-сервери, відкриті для доступу ззовні.

#### **Конфігурація з однією системою батьківського контролю доступу до мережі**

Одна систему батьківського контролю доступу до мережі ділить мережний простір на три зони: зовнішня мережа, внутрішня мережа й DMZ. Весь трафік надходить на систему батьківського контролю доступу до мережі із зовнішньої

мережі. Система батьківського контролю доступу до мережі повинна контролювати трафік і ухвалювати рішення щодо його пересиланні в DMZ або у внутрішню мережу, або про заборону пересилання.

### Конфігурація із двома системами батьківського контролю доступу до мережі

У конфігурації із двома системами батьківського контролю доступу до мережі передбачені внутрішні і зовнішні системи батьківського контролю доступу до мережі, між якими розташовується DMZ. Зовнішня система батьківського контролю доступу до мережі застосовує менш строгі обмеження й дозволяє доступ користувачів з Інтернету в DMZ, а також наскрізне проходження трафіка, запитаного внутрішніми користувачами. Внутрішня система батьківського контролю доступу до мережі застосовує більше строгі обмеження й захищає внутрішню мережу від несанкціонованого доступу.

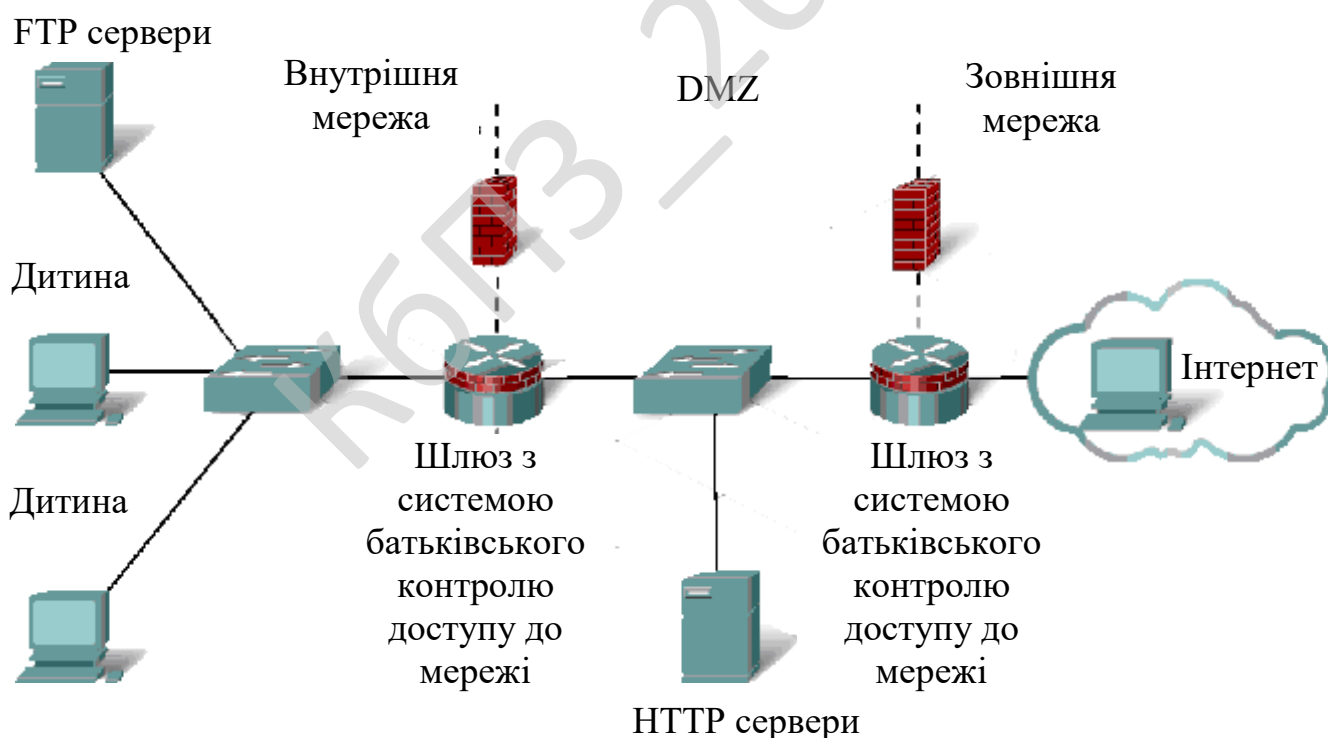


Рисунок 3.1 – Структурна схема системи

Для невеликих мереж з низьким трафіком підходить конфігурація на основі однієї системи батьківського контролю доступу до мережі, що, однак, є критичною точкою відмови й може виявитися перевантаженим. Конфігурація із двома системами батьківського контролю доступу до мережі доцільна для великих і складних мереж зі значно більшими об'ємами трафіка.

Всі НТТР сервери здатні дати клієнтові однакові сервіси (однак не всі НТТР сервери захищені мостом). У такий же спосіб, всі FTP сервери забезпечують клієнтів однаковим сервісом.

Системному адміністраторові важливо, щоб завантаження по обслуговуванню була розподілена між серверами. Клієнт не буде підозрювати про наявність декількох різних серверів. З погляду клієнта, є тільки один НТТР і один FTP сервер. Коли запит на обслуговування досягає системи батьківського контролю доступу до мережі, то він визначає який із серверів буде обслуговувати даний запит, ґрунтуючись на алгоритмі балансування завантаження, заданому системним адміністратором.

#### **Алгоритм розподілу навантаження**

Доступні наступні алгоритми розподілу:

1. По завантаженню сервера. Система батьківського контролю доступу до мережі запитує сервери щоб визначити, що з них найкраще здатний обслужити нове з'єднання.
2. За часом відповіді на ping (round trip). Система батьківського контролю доступу до мережі використовує ping для визначення часу проходження пакета між системою батьківського контролю доступу до мережі і кожним із серверів і вибирає сервер з найменшим часом відповіді.
3. По колу. Система батьківського контролю доступу до мережі просто призначає наступний сервер у списку.
4. Випадковий. Система батьківського контролю доступу до мережі призначає сервер у випадковому порядку.
5. По доменному імені. Система батьківського контролю доступу до

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		32



### 3.3 Розробка функціональної схеми

На рисунку 3.2 зображена функціональна схема системи. Нижче розглянемо її більш докладно.

Функції Батьківського контролю дозволяють обмежувати час і параметри доступу до комп'ютера, програмам, ОС Windows і мережі Інтернет.

Після того, як ви завершите процес установки, створіть пароль, збережіть його, до того ж вам не потрібно запускати програму постійно, щоб довідатися – вийшли чи відновлення ні, тому що повідомлення про відновлення будуть приходити на електронну пошту.

Також ви можете запустити майстер налаштувань, з його допомогою можна легко встановити необхідні параметри програми. Ви можете встановити проміжок часу, коли вашій дитині можна буде виходити в Інтернет. Коли час підійде до кінця, комп'ютер завершить роботу запущених додатків, а потім вимкнеться (цю опцію можна також встановити в параметрах програми). Також можна встановити час, коли доступ до інтернету буде заборонений для конкретного користувача.

Скориставшись Інтернет-фільтром, ви можете створити список дозволених для доступу сайтів, а також виставити Налаштування рівня безпеки (від низького до гранично високого).

Також можливо змінити параметри доступу до комп'ютера й мережі Інтернет (активувати опцію – Перерва), створити білий і чорний списки. У білий список можна включити дозволених, у чорний список – веб-сайти, які необхідно блокувати.

Крім того, за допомогою програмного забезпечення, що розроблено в даній роботі, можна заборонити доступ до командного рядка, редактору реєстру Windows, панелі керування, зміні дати й часу, диспетчеру завдань, кнопці «Пуск» і ін.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		34

Не тільки програми й веб-сайти можна додати в чорні списки, але й користувачів. Наприклад, певним користувачам можна виставити заборону на вихід в Інтернет, перегляд лог файлів, зміну параметрів (самостійно переналаштувати додаток, якщо ліміт часу на перегляд веб-сайтів вичерпаний, зробити синхронізацію часу й дати, використовуючи Інтернет).

**Основні функції програмного забезпечення, що розроблено в даній роботі:**

- опція додавання в білий список користувачів, яким дозволений доступ до ПК;
- можна задавати обмеження доступу (за часом);
- налаштувати час блокування доступу до ПК, до мережі Інтернет на щогодини кожного дня;
- в автоматичному режимі занести в чорний список «шкідливі» веб-сайти, сайти для дорослих, утримуючі задані ключові слова;
- строки роботи визначаються для кожного EXE-файлу;
- у звіт заносяться дані по кожному користувачі, відслідковується кожна дія;
- відключити основні служби операційної системи;
- стелс-режим – програма невидима для користувачів;
- білий список .EXE файлів – визначити, які програми дитина може запускати;
- білий список веб-сайтів – установити, які сайти можна відвідувати;
- оповіщення за допомогою e-mail: бачити, що відбувається на домашньому комп'ютері з віддаленого робочого місця;
- у кілька кліків можна виставити налаштування Імпорту й Експорту – синхронізувати з іншими ПК;
- підтримка доменів і робочих груп;
- налаштування власних спливаючих повідомлень і попереджень.

Розглянувши усі блоки функціональної схеми перейдемо до розгляду

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		35

діаграми взаємодії процесів, які відбуваються у системі.

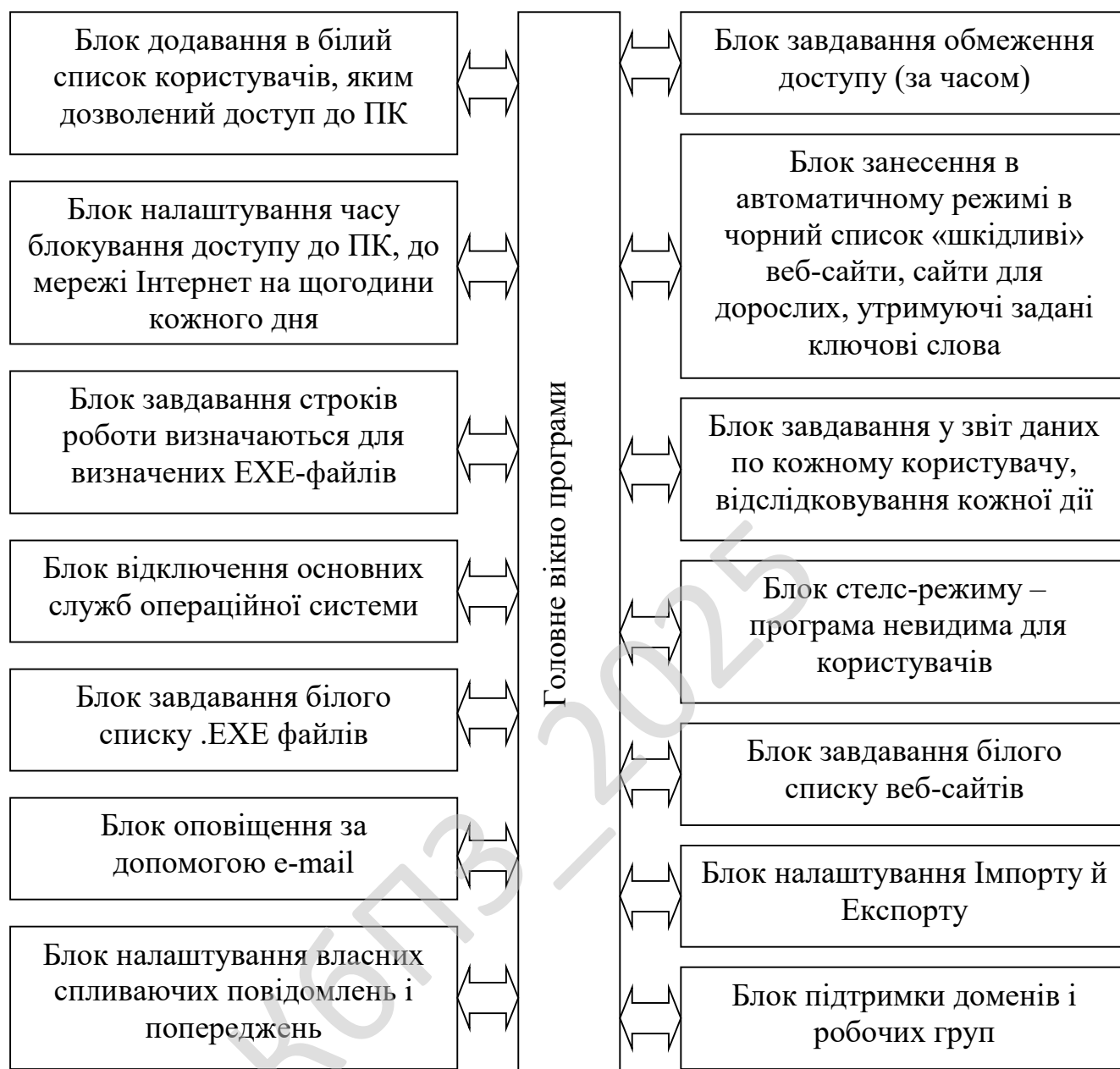


Рисунок 3.2 – Функціональна схема системи

### 3.4 Розробка діаграми процесів

Діаграма процесів розробленої системи зображена на рисунку 3.3. При детальному її розгляді можна побачити як саме проходить взаємодія у розробленій



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

### 4.1 Блок-схеми та опис алгоритмів функціонування системи

Первинною стадією без якої не відбувається розробка програмного забезпечення це звичайно розробка блок-схем.

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

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.

При виборі початкової точки відліку при побудові схем було враховано, що виходячи з вибору мови програмування і інших технічних засобів, програма буде об'єктно-орієнтована що вимагає оптимізації програми високого рівня, також те, що при розробці програми слід надати особливу увагу модулю системи батьківського контролю доступу до мережі.

При складанні блок-схем програмного забезпечення і напрацювання алгоритмів я зіткнувся з масою проблем, які вимагали напрацювання процедур і функцій над основною проблематикою. Для чого були створені додаткові класи, типи даних і константи, що забезпечило вирішення проблем.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		<b>38</b>

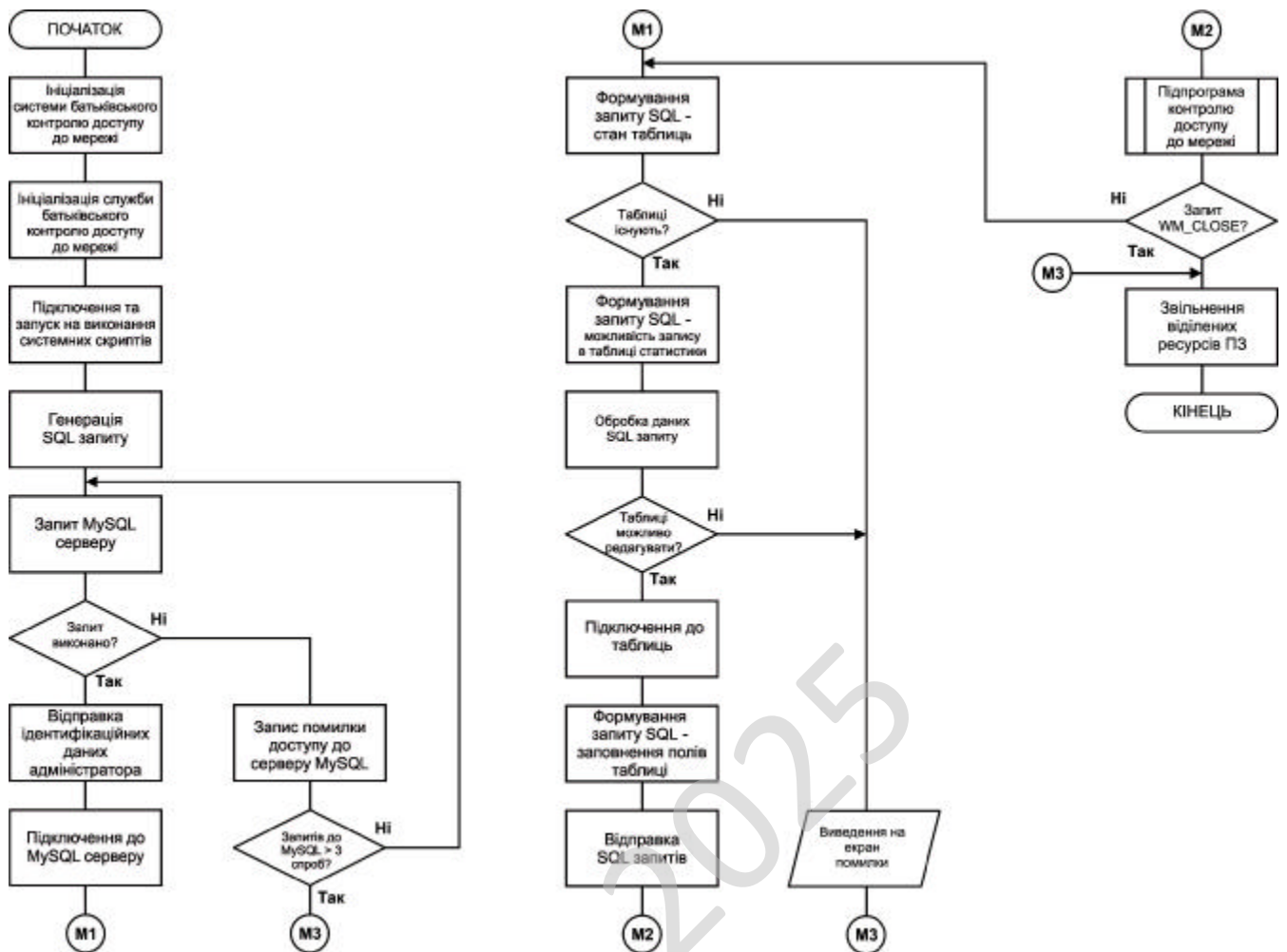


Рисунок 4.1 – Блок-схема основної програми

Під час роботи над бакалаврською 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.

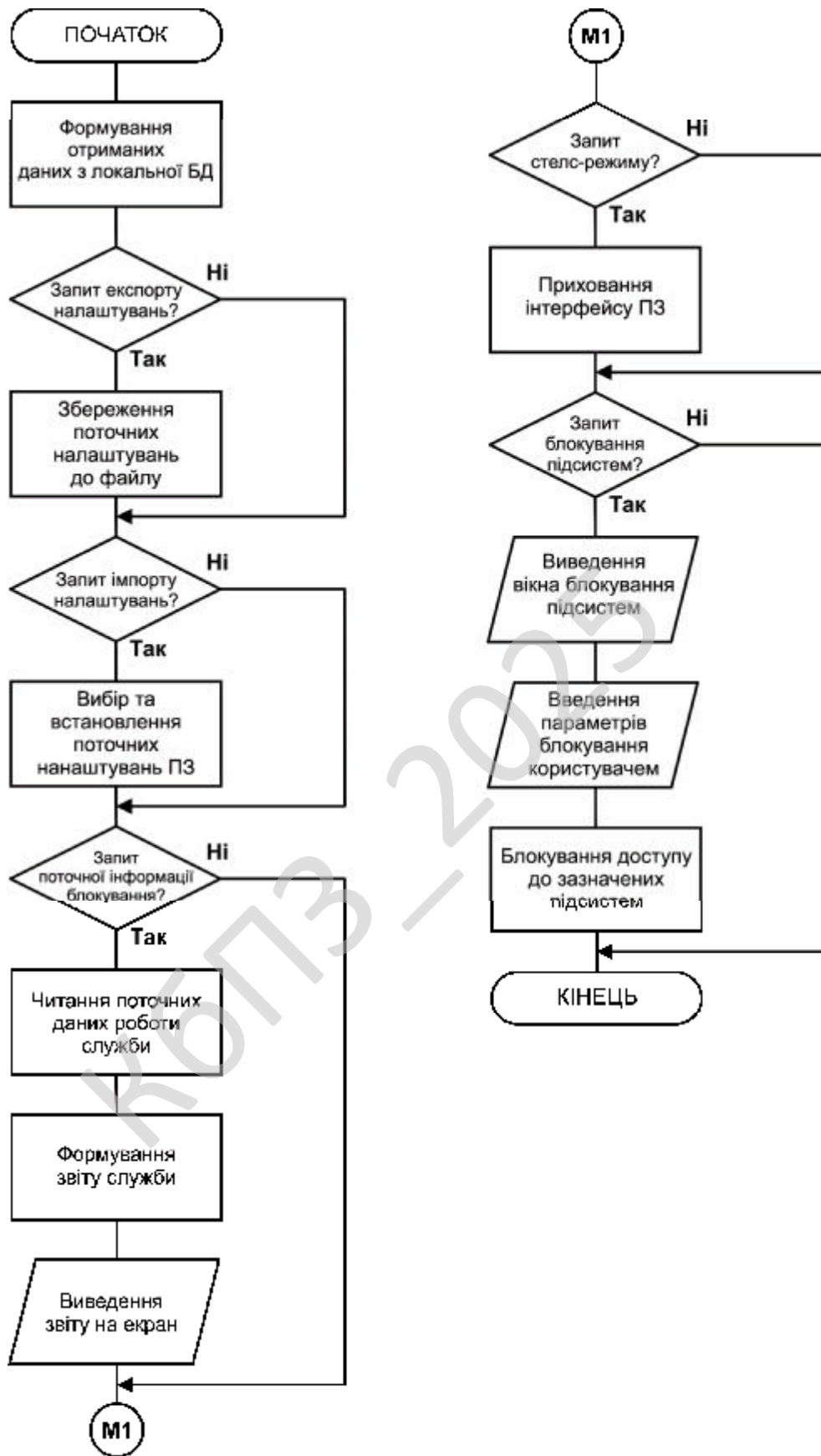


Рисунок 4.2 – Блок-схема роботи підпрограми



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.

Було використано підходи з використанням UML, це уніфікована мова моделювання, використовується у парадигмі об'єктно-орієнтованого програмування. Є невід'ємною частиною уніфікованого процесу розробки програмного забезпечення. UML є мовою широкого профілю, це відкритий стандарт, що використовує графічні позначення для створення абстрактної моделі системи, називаної UML-моделлю. UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але в засобах виконання UML-моделей як інтерпретованого коду можлива кодогенерація.

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.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		42

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.

– 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.

Також при розробці бакалаврської дипломної роботи було використано наступні підходи UML: діаграма діяльності (діаграми поведінки типу); діаграма прецедентів (діаграми поведінки типу).

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

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		43

одиницею визначення поведінки в специфікації. Дія отримує множину вхідних сигналів, та перетворює їх на множину вихідних сигналів.

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

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

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

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

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

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

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		44

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

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

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

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

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

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

При роботі із системою виникла проблема взаємодії з XML. Інформація на сервері зберігається саме в цьому форматі. Тому був розроблений перетворювач XML в HTML на мові програмування PHP. Як неважко припустити по його назві, мова HTML (Hypertext Markup Language) ставиться до числа так званих мов розмітки тексту (markup languages).

Під терміном "розмітка" розуміється загальна службова інформація, яка не виводиться разом з документом, але визначає; як повинні виглядати ті або інші фрагменти документа. Наприклад можна зажадати, щоб яке-небудь слово виводилося жирним або курсивним шрифтом, вивести окремий абзац особливим

шрифтом або оформляти заголовки збільшеним шрифтом. Текстовий редактор теж використовує особливу форму розмітки для вистави тих атрибутів форматування, які я вибираю.

Таким чином, у ньому теж використовується особливий різновид мови розмітки. У наші дні існує безліч різних мов розмітки. Наприклад, у комунікаційних програмах особлива форма розмітки визначає зміст кожного пакета з нулів і одиниць, що пересилається в Інтернеті. Коли ми підкреслюємо слова в книзі, це теж можна вважати свого роду розміткою. Втім, будь-яка мова розмітки повинен вирішувати два важливі завдання:

1. Мова визначає синтаксис розмітки. Наприклад, у відповідності зі специфікацією HTML конструкція `<b>text</b>` визначає синтаксично правильну розмітку тексту, а конструкція `<xr5t>text</x4rt>` вважається неправильної через розбіжність відкриваючого й закриваючого тегів.

2. Мова визначає зміст розмітки. Звичайно, ви знаєте, що команда `<b>text</b>` виводить слово `text` жирним шрифтом. У цьому випадку визначається зміст, пов'язаний з оголошенням деякого компонента документа.

Стрімкий розвиток Web за останні кілька років наочно показує, що самою популярною мовою розмітки тексту є HTML.

### Перетворення XML в HTML.

```
<?
Class XMLHTML {
VAR $xmlparser;
VAR $tagcolor = "#800000";
VAR $datacolor = "#0000ff";
function XMLHTML() {
$this->xmlparser = xml_parser_create();
xml_set_object($this->xmlparser, $this);
xml_set_element_handler($this->xmlparser, "starttag", "endtag");
xml_set_character_data_handler($this->xmlparser, "characterdata");
}
// Функція відповідає за обробку всіх відкриваючих тегів.
function starttag($parser, $tagname, $attributes)
{
GLOBAL $tagcolor;
```

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		46







```

close(INFO);
foreach $string(@data) {
@item=split(/&/,$string);
foreach (@item) {
if (($item[0] eq $login) && ($item[1] eq $password)) {
# Якщо все нормально, переходимо в користувачький каталог.
print "Content-type:text/html\n\n";
print <<HTML;
<html><body bgcolor="e6e8fa">
# Вітаємо користувача.
<p><h2 align=center><font color="ff0000">Hello,$login!</font></h2>
<p><center>Welcome to your home directory!
<p>Your URL is <a href="$basedir$login">$basedir$login.</a></center>
HTML
# Список директорії
$userdir=$dir.$login;
chdir ("$userdir");
# Відкриваємо каталог і читаємо список файлів у масив.
opendir(DIR,"$userdir") || die "Cannot open $userdir!";
while (@files=readdir(DIR)) {
# Якщо каталог містить підкаталоги, виводимо їх
# окремо, а також не показуємо
# каталоги "." і ".." Друкуємо шапку таблиці.
print <<HTML;
<p><center>
<table bgcolor="\bfbfbf\" width=600 border cellspacing=0
cellpadding=0 nowrap>
<tr><td colspan=5 align=center nowrap>
<b><font color="ff0000">Directories</font></b></td></tr>
<tr><td>.</td><td align=center><b>List</b>
</td><td><b>Size</b>
<td><td><b>Last accessed</b></td><td><b>Last modified</b></td>
#HTML
foreach $file(@files) {
# Статистика файлів - розмір, час останнього звернення й модифікації.
$size=(stat("$userdir/$file"))[7];
$time=localtime((stat("$userdir/$file"))[8]);
$mtime=localtime((stat("$userdir/$file"))[9]);
# друкуємо список підкаталогів.
if ( -d "$userdir/$file" && "$file" ne "." && "$file" ne "..") {
print "<tr><td width=30><img src=\"$url/image/folder.gif\">
</td><td width=100 align=left>$file</td>\n";

```

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		50







```

foreach (@fields=split(/&/,$query)) {
if (/login=(.*)/) { $login=&urldecode ($1); }
if (/pass=(.*)/) { $password=&urldecode ($1); }
if (/email=(.*)/) { $email=&urldecode ($1); }
}
# Перевіряємо, не чи існує дане ім'я в системі.
open(INFO,"login.txt") ||die;
# читаємо рядок в масив.
@data=<INFO>;
close(INFO);
foreach $string(@data) {
@item=split(/&/,$string);
# Розбиваємо рядок на частини
foreach (@item) {
if ($item[0] eq $login) {
# Порівнюємо отримане ім'я з першим полем файлу
# для кожного рядка і якщо таке знайдене видаємо
# помилку.
print <<HTML;
<h2 align=center><font color="ff0000">Error!</font></h2>
<p><center><b>The name <font color="ff0000">$login</font> already
exists in the system. <p>Please,go back and choose another name.</b>
<p><form><input type="button" value="Back" onclick=
"history.back()"></form>
</center>
HTML
exit;
}
}
}
# Якщо ім'я не знайдене,відкриваємо базу даних і додаємо інформацію.
if ($item[0] ne $login) {
open(DATA,">>login.txt");
$string=join('&', $login, $password, $email, scalar localtime, $ENV
{'REMOTE_ADDR'});
print DATA "$string\n";
close(DATA);
# Створюємо домашній каталог користувача й переходимо в нього.
mkdir("$userdir/$login",0700);
chdir("$userdir/login");
opendir(USER,"$userdir/$login");
# Поміщаємо файл index.html у каталог користувача.

```

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		54

```

open(IN,">$userdir/$login/index.html");
print IN "This is the test!\n";
close(IN);
closedir(USER);
# Зміст файлу може бути будь-яким,це тільки для прикладу.
# Генеруємо відповідь користувачеві.
print <<HTML;
<p><h1 align=center><font color="ff0000">Congratulations!</font></h1>
<p><b>Your registration was successful and your data were added to
our database.Thank you for your time.</b>
<p><center><b><font color="ff0000">
You entered:</font>(print this page and keep it in safe place)</b>
<p><table>
<tr><td><b>Your login name:</b><td><font color="0000ff">$login</font>
<tr><td><b>Your password:</b><td><font color="0000ff">$password</font>
<tr><td><b>Your e-mail address:</b>
<td><font color="0000ff">$email</font>
</table></center>
HTML
}
}

```

Скрипт видає відповідь у вигляді html сторінки, що містить всю інформацію, введену користувачем.

## 4.2 Захист розробленого програмного забезпечення

Для захисту розробленого програмного забезпечення запропоновано використовувати алгоритм LOKI\_91. Механізм алгоритму LOKI\_91 подібний DES (рисунок 4.4). Блок даних розщеплюється на ліву й праву половини й проходить 16 раундів, що досить нагадує DES. У кожному раунді права половина спочатку піддається операції XOR із частиною ключа, а потім розширювальній перестановці (таблиця 4.1).

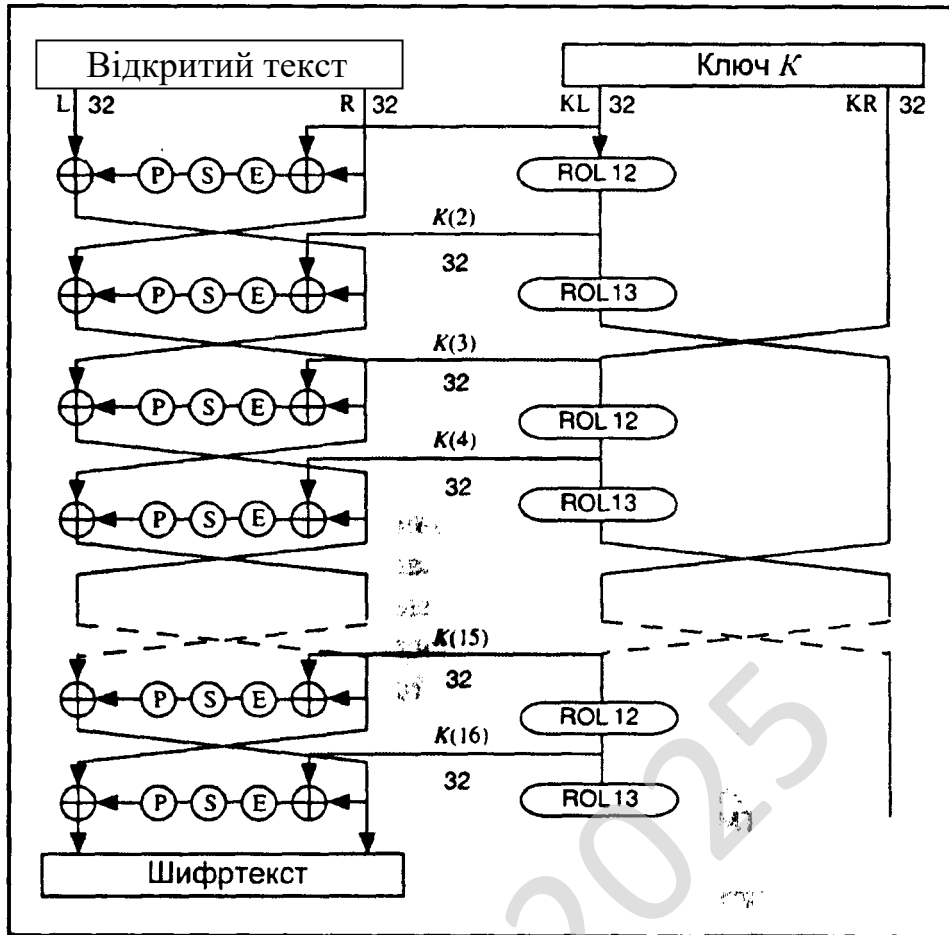


Рисунок 4.4 – Алгоритм LOKI91

Таблиця 4.1 – Перестановка з розширенням

4	3	2	1	32	31	30	29	28	27	26	25
28	27	26	25	24	23	22	21	20	19	18	17
20	19	18	17	16	15	14	13	12	11	10	9
12	11	10	9	8	7	6	5	4	3	2	1

48-бітовий вихід розділяється на чотири 12-бітових блоки. У кожному блоці виконується така підстановка з використанням S-блоку: береться кожний 12-бітовий вхід, 2 старших і 2 молодших біти використовуються для утворення номера  $r$ , а вісім внутрішніх біт утворюють номер  $s$ . Вихід S-блоку,  $O$ , має наступне значення:  $O(r,c) = (c + ((r*17) \oplus 0xff) \& 0xff)^{31} \bmod P_r$ .



## 5 МЕТОДИКА ВПРОВАДЖЕННЯ СИСТЕМИ В ПРОМИСЛОВУ ЕКСПЛУАТАЦІЮ

На рисунку 5.1 зображено інтерфейс програмного забезпечення, розробленого у результаті виконання бакалаврської дипломної роботи.

Розроблене програмне забезпечення системи батьківського контролю доступу до мережі складається з наступних функціональних блоків:

- Навігаційне меню: Файл; Налаштування; Довідка.
- Обрання інтерфейсу роботи – мережної карти.
- Введення даних автентифікації.
- Вікно виведення налаштування системи.
- Навігаційного меню яке викликається натисканням правої клавіші маніпулятора миші.
- Функціональних кнопок ПЗ: Підключитись/Відключитись; Статистика доступу; Журнал роботи; Налаштування.

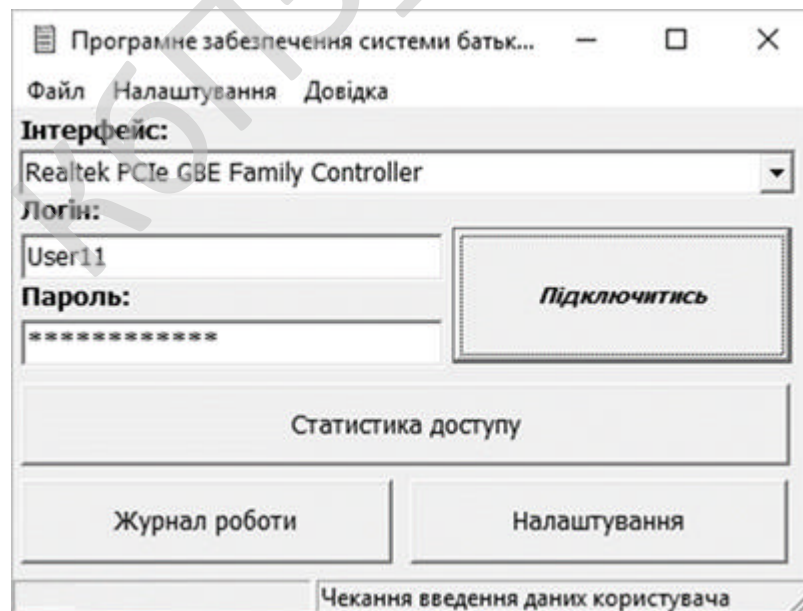


Рисунок 5.1 – Головне вікно розробленого ПЗ

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.

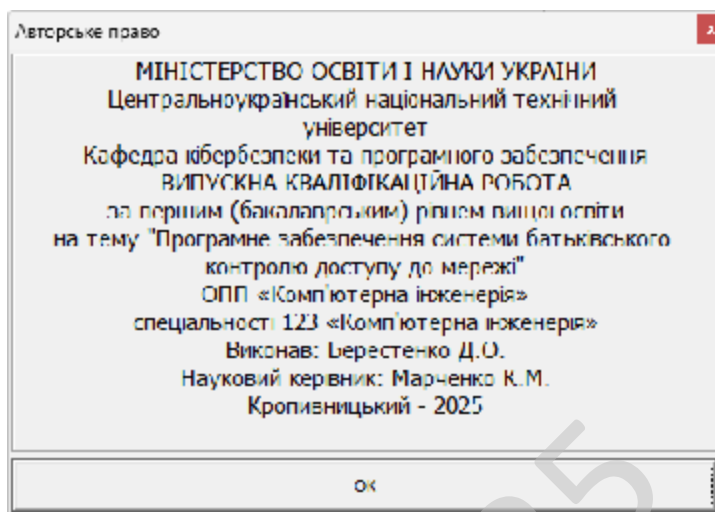


Рисунок 5.2 – Вікно розробника ПЗ

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.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		59

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.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		60

Обрано умови розповсюдження – proprietary software.

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 протипагу 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.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		61



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

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

Даються необхідні рекомендації з установки розробленого програмного забезпечення.

Для підвищення рівня безпеки запропоновано застосовувати алгоритм LOKI\_91.

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.

КБПЗ\_2025

					VKPB-123.25.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		63

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Loren Kohnfelder. Designing Secure Software. No Starch Press. 2022. 332 p.
2. Samir Kumar Rakshit. Ethical Hacker's Penetration Testing Guide. BPB Online. 2022. 509 p.
3. Corey J. Ball. Hacking APIs. No Starch Press. 2022. 353 p.
4. Kevin Beaver. Hacking for Dummies. John Wiley & Sons. 2022. 419 p.
5. Mark S. Merkow. Practical Security for Agile and DevOps. CRC Press. 2022. 236 p.
6. Derek Fisher. Application Security Program Handbook. Manning Publications. 2021. 155 p.
7. Cameron Wyatt PH.D. Kali Linux Tutorial. Independently published. 2021. 60 p.
8. Alex Matrosov, Eugene Rodionov, Sergey Bratus. Rootkits and Bootkits. No Starch Press. 2019. 450 p.
9. Lakhno, V., Malyukov, V., Smirnov, O., Bebeshko, B., Chubaievskiy, V., Zhumadilova, M., Malyukova, I., Smirnov, S. «Multifactorial Model for Targeted Attacks Counteracting Within the Framework of a Multi-Step Quality Game with Fuzzy Information». *8th International Symposium on Intelligent Informatics, ISI 2023, 2025*. vol 389. pp 377-389. Springer, Singapore.
10. Kuznetsov, O., Frontoni, E., Kryvinska, N., Chevardin, V., Smirnov, O. «Wireless Network Encryption Stream Ciphers, Computational Modeling, and Security Analysis». *Computational Modeling and Simulation of Advanced Wireless Communication Systems, 2024*, pp. 379–402.
11. Kuznetsov, O., Frontoni, E., Kryvinska, N., Smirnov, O., Imoize, G.L. «Computational Modeling of Enhanced Spread Spectrum Codes for Asynchronous Wireless Communication». *Computational Modeling and Simulation of Advanced Wireless Communication Systems, 2024*, pp. 403–447.

					ВКРБ-123.25.0025.00.00.ПЗ	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		64

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. Akhalaia, G., Iavich, M., Iashvili, G., Prysiazhnyy, D., Smirnova, T. «Secure Encrypted Connection on Georgian Website». *CEUR Workshop Proceedings*, 2023, 3550, pp. 313-320.

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. Kuznetsov, O., Kandiy, S., Frontoni, E., Smirnov, O. «Trade-offs in Post-Quantum Cryptography: A Comparative Assessment of BIKE, HQC, and Classic McEliece». *CEUR Workshop Proceedings*, Volume 3504, 2023, pp. 1-11.

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

18. Smirnov, O., Lakhno, V., Akhmetov, B., Chubaievskyi, V., Khorolska, K., Bebesko, B. «Selection of a Rational Composition of Information Protection Means Using a Genetic Algorithm». In: Rajakumar, G., Du, KL., Vuppapapati, C., Beligiannis, G.N. (eds) *Intelligent Communication Technologies and Virtual Mobile Networks. Lecture Notes on Data Engineering and Communications Technologies*, vol 131. 2023. Springer, Singapore. pp. 21-34.

19. Smirnov O.A., Al-Oraiqat A.M., Ulichev O.S., Meleshko Ye.V., Al-Rawashdeh H.S., Polishchuk L.I. «Modeling strategies for information influence dissemination in social networks». *Journal of Ambient Intelligence and Humanized Computing* Volume 13, Issue 5. Springer, Cham. 2022, pp. 2463-2477.

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

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

22. Smirnov O., Kuznetsov A., Girzheva O., Kiian A., Nakisko O., Kuznetsova T. «Advanced Code-Based Electronic Digital Signature Scheme». *2020 IEEE International Conference on Problems of Infocommunications Science and Technology, PIC S and T 2020*, Kharkiv, 6 October 2020-9 October 2020, P. 358-362.

23. Smirnov O., Kuznetsov A., Kiian A., Kuznetsova K. «Data hiding scheme based on spread sequence addressing». *CEUR Workshop Proceedings Volume 2805*, 2020, Pages 44-58.

24. 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.

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

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

27. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings* Volume 2654, 2020, Pages 122-131.

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

29. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84.

30. Smirnov O., Kuznetsov A., Onikiychuk A., Makushenko T., Anisimova O., Arischenko A. «Adaptive pseudo-random sequence generation for spread spectrum image steganography». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 161-165.

31. Smirnov O., Kuznetsov A., Kiian A., Babenko V., Perevozova I., Chepurko I. «New Approach to the Implementation of Post-Quantum Digital Signature Scheme». *2020 IEEE 11th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, Ukraine, Kyiv, May 14-18. 2020. P. 166-171.

32. 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.

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

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

35. 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.

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

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

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

39. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019.

40. 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.

41. Smirnov, O., Kuznetsov, A., Kiian, A., Gorbenko, Y., Cherep, O., Bexhter L. «Code-based Pseudorandom Generator for the Post-Quantum Period», *2019 IEEE International Conference on Advanced Trends in Information Theory (IEEE ATIT 2019)*. 18.12.19-20.12.19 Kyiv Ukraine. P. 204 – 209.

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		68

42. Smirnov, O., Kuznetsov, A., Nariezhnii, O., Stelnyk, S., Kokhanovska, T., Kuznetsova T., «Side Channel Attack on a Quantum Random Number Generator», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18 - 21 September 2019. P.713-718.

43. 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.

44. 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.

45. Smirnov, O., Hu, Z., Vasiliu, Y., Sydorenko, V., Polishchuk, Y., «Abstract Model of Eavesdropper and Overview on Attacks in Quantum Cryptography Systems», *10th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS 2019*; Metz; France; 18-21 September 2019. P.399-405.

46. 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.

47. Smirnov, O., Kuznetsov, A., Kiian, A., Babenko, B., Zhosan, H., Prokopovych-Tkachenko, D., «Soft Decoding Method for Turbo-Productive Codes», *2019 3rd International Conference on Advanced Information and Communications Technologies, AICT 2019, Lviv, Ukraine, 2-6 July, 2019*, P. 129-134.

48. Smirnov, O., Kuznetsov, A., Kiian, A., Zamula, A., Rudenko, S., Hryhorenko, V., «Variance Analysis of Networks Traffic for Intrusion Detection in

Smart Grids», *2019 IEEE 6th International Conference On Energy Smart Systems (2019 IEEE ESS)*, Kyiv, Ukraine April 17-19, 2019 P. 353-358.

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

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

51. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», *CEUR Workshop Proceedings Volume 2353, CEUR Workshop Proceedings 2019*, Pages 873-884.

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

					<b>ВКРБ-123.25.0025.00.00.ПЗ</b>	Арк.
Вим.	Арк.	№ докум.	Підпис	Дата		70

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 Перелік документів, що розробляються.....	5
8 Етапи розробки.....	6
9 Порядок контролю та приймання.....	6

					<b>ВКРБ-123.25.0025.00.00.ТЗ</b>			
<i>Вим.</i>	<i>Арк.</i>	<i>№ документа</i>	<i>Підпис</i>	<i>Дата</i>				
<i>Розробив</i>	<i>Берестенко Д.О.</i>				<i>Програмне забезпечення системи батьківського контролю доступу до мережі</i>	<i>Літ.</i>	<i>Аркуш</i>	<i>Аркушів</i>
<i>Перевірів</i>	<i>Марченко К.М.</i>					<i>Б</i>	<i>1</i>	<i>6</i>
<i>Н. Контр.</i>	<i>Коваленко А.С.</i>					<i>ЦНТУ КІ-21-2</i>		
<i>Затв.</i>	<i>Смірнов О.А.</i>							

## 1 Найменування та область застосування

Це технічне завдання розповсюджується на розробку системи батьківського контролю доступу до мережі.

## 2 Підстава для розробки

Підставою для розробки служить завдання на випускню кваліфікаційну роботу за першим (бакалаврським) рівнем вищої освіти, видане на кафедрі кібербезпеки та програмного забезпечення (нак. № 47-02 від 17.01.2025 року).

## 3 Мета та призначення розробки

Метою випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти є розробка програмного забезпечення системи батьківського контролю доступу до мережі.

## 4 Джерела розробки

Джерелом цієї випускної кваліфікаційної роботи за першим (бакалаврським) 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;

					ВКРБ-123.25.0025.00.00.ТЗ	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		2

- розробка програмної частин системи, а також розробка взаємодії системи з ОС та з користувачем;
- розробка програми, що реалізує спроектовані алгоритми роботи системи.

## 5.2 Показники призначення

Система повинна забезпечувати:

- системи батьківського контролю доступу до мережі;
- 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.

					<b>ВКРБ-123.25.0025.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		3

## 5.6 Умови експлуатації

Робочі місця користувачів ПЗ повинні задовольняти наступним умовам експлуатації:

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

## 5.7 Вимоги до складу та параметрів технічних засобів

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

## 5.8 Вимоги до інформаційної і програмної сумісності

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

### 5.8.1 Обладнання

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

### 5.8.2 Мова програмування

Середовище PHP, Perl.

					<b>ВКРБ-123.25.0025.00.00.ТЗ</b>	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 Перелік документів, що розробляються

- Структурна схема системи – 1 аркуш.
- Функціональна схема системи – 1 аркуш.
- Діаграма процесів – 1 аркуш.
- Блок-схема алгоритму роботи програми – 2 аркуша.
- Пояснювальна записка – 70 аркушів.

## 8 Етапи розробки

8.1 Збір і обробка інформації по темі випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти. Постановка задачі на виконання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти (складання ТЗ).

					<b>ВКРБ-123.25.0025.00.00.ТЗ</b>	Арк.
Вим.	Арк.	№ документа	Підпис	Дата		5

8.2 Проведення досліджень або експериментальних робіт для уточнення основних положень випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти.

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

8.4 Побудова схем взаємодії даних.

8.5 Створення прототипу ПЗ.

8.6 Віднаходження ПЗ, аналіз отриманих результатів.

8.7 Оформлення пояснювальної записки і виконання робіт по графічній частині.

## 9 Порядок контролю та приймання

9.1 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на попередній захист 23.05.2025 р.

9.2 Подання випускної кваліфікаційної роботи за першим (бакалаврським) рівнем вищої освіти на захист 3.06.2025 р.

					<b>ВКРБ-123.25.0025.00.00.ТЗ</b>	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  
першим (бакалаврським) рівнем вищої освіти

\_\_\_\_\_ Марченко К.М.

*Програмне забезпечення системи батьківського контролю доступу до  
мережі*

Лістинг програми

Код документу 12

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

Загальна кількість аркушів: 33

Літера: РП

Кропивницький – 2025 року

## ВИХІДНИЙ КОД МОДУЛЯ АДМІНІСТРУВАННЯ

```

<?php
//Берестенко Даниїл Олексійович
// ЦНТУ, кафедра КБПЗ, КІ-21-2, 2025

    $path=getenv("DOCUMENT_ROOT");
    $path."/MONITORING /"; // шлях до папки зі скриптами
    include( "$path"."common.php" ); // підключення модулю common.php
    include( "$path"."global.php" ); // підключення модулю global.php
    Top("./", "сторінка адміністратора");
    ConnectToBase($conf{'host'}, $conf{'username'}, $conf{'password'},
    $conf{'database'});
    $CurrentDateDB = GetCurrentDateDB();
    if ($action=="search") // виклик пошуку?
    {
    search();
    }
    elseif ($action=="show") // виклик відображення інтерфейсу
    {
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="incbalance") // виклик відображення балансу
    {
    IncBalance ();
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="update") // виклик оновлення
    {
    UpdateUser ();
    ShowUserAdmin($ip,$month);
    }
    elseif ($action=="delete") // виклик видалення
    {
    if ($is_js_confirmed)
    {
    DeleteUser ();
    }
    ShowSearchForm(GetMonths());
    }
    elseif ($action=="adduserform")
    {
    AddUserForm ( );
    }
    elseif ($action=="adduser") // виклик додавання користувача
    {
    AddUser( );
    ShowSearchForm(GetMonths());
    }
    elseif ($action=="addgroup")
    {
    AddGroup ();
    EditGroup ();
    }elseif ($action=="editgroup")
    {
    EditGroup ();
    }
    elseif ($action=="delgroup")
    {
    DelGroup ();
    EditGroup ();
    }
    elseif ($action=="updategroup")
    {
    UpdateGroup ();
    EditGroup ();
    }
    elseif ($action=="stat")

```

```
{
  ShowStat ();
}
elseif ($action=="deleteold")
{
  DeleteOld ();
} else {
ShowSearchForm(GetMonths ());
}
Footer();
?>
```

К6П3\_2025

## ВИХІДНИЙ КОД WWW-ІНТЕРФЕЙСУ

```
<?php
//Берестенко Даниїл Олексійович
// ЦНТУ, кафедра КБПЗ, КІ-21-2, 2025
include( "./common.php" ); // підключення модулю common.php
include( "./global.php" ); // підключення модулю global.php
// початок формування HTML коду
Тор("./", "сторінка користувача.");
// підключення та робота з БД, формування HTML коду
ConnectToBase($conf{'host'}, $conf{'username'}, $conf{'password'},
              $conf{'database'});
// Дані користувачів, формування HTML коду
ShowUser(ResolveUser());
// кінець формування HTML коду
Footer();
?>
```

КБПЗ\_2025

**ФАЙЛ MONITORING.PL - ВИХІДНИЙ КОД МОДУЛЯ ЗЧИТУВАННЯ ТРАФІКА**

```
#!/usr/bin/perl
# Diplom
# Берестенко Даниїл Олексійович
# ЦНТУ, кафедра КБПЗ, КІ-21-2, 2025
use DBI;
use Common;
LoadConfig();
GetDate();
ConnectToDB();
TestNextMonth();
PutTrafToDBOutput();
PutTrafToDBForward();
SubtractAbon();
TurnOffAccess();
TurnOnAccess();
KillUserOutput();
KillUserForward();
Disconnect();
```

КБПЗ\_2025

## ФАЙЛ COMMON.PM - ПАКЕТНИЙ ФАЙЛ ПРОЕКТУ

```

package Common;
BEGIN {
//Берестенко Даниїл Олексійович
// ЦНТУ, кафедра КБПЗ, КІ-21-2, 2025
    use Exporter ();
    @ISA = 'Exporter';
    @EXPORT = qw(&SubtractAbon &ConvertMoneyToDB &TurnOnAccess &TurnOffAccess
&SubtractMoney &GetDate &PutTrafToDBOutput &PutTrafToDBForward &LoadConfig
&ConnectToDB &TestNextMonth &KillUserOutput &KillUserForward &Disconnect);
}
// завантаження конфігурації
sub LoadConfig
{
for $file("./Monitoring.conf")
    {
        // повернення коду читання файлу
        unless ($return = do $file) {
            warn "couldn't parse $file: $@" if $@;
            warn "couldn't do $file: $!" unless defined $return;
            warn "couldn't run $file" unless $return;
        }
    }
}

// читання таблиць БД
sub TurnOffAccess
{
    my $query_counter = ("SELECT `ip` FROM `counter` WHERE
        `moneymonth`='n' AND `yearmonth`='$date'");
    my $sth_counter = $dbh -> prepare($query_counter);
    $sth_counter -> execute;
    $sth_counter -> bind_columns(undef, \($ip));
    while ($sth_counter -> fetch)
    {
        my $query = ("UPDATE `user` SET access = 'n' WHERE `ip` = '$ip' AND
            `access`='y'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        $sth -> finish;
    }
    $sth_counter -> finish;
my $query = ("UPDATE `user` SET access = 'n' WHERE `balance`<=0 AND
    `access`='y'");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    $sth -> finish;
}
sub TurnOnAccess
{
    my $query_counter = ("SELECT `ip` FROM counter WHERE
        `moneymonth`='y' AND `yearmonth`='$date'");
    my $sth_counter = $dbh -> prepare($query_counter);
    $sth_counter -> execute;
    my ($ip);
    $sth_counter -> bind_columns(undef, \($ip));
    while ($sth_counter -> fetch)
    {
        my $query = ("UPDATE `user` SET access = 'y' WHERE `balance` > 0 AND
`ip` = '$ip' AND `access`='n'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        $sth -> finish;
    }
    $sth_counter -> finish;
}

// Запис до таблиць БД

```

```

sub ConvertMoneyToDB
{
    $money = $_[0];
    $newmoney = $money*1000000000;
return $newmoney; // повернення коду затрачених коштів
}
sub SubtractAbon
{
$ip = $_[0];
    # $balancelimit = ConvertMoneyToDB(100);
    my $query_user = ("SELECT `ip`,`group`,`balance` FROM user");
    #print "$query_user\n";
    my $sth_user = $dbh -> prepare($query_user);
    $sth_user -> execute;
    $sth_user -> bind_columns(undef,\($ip,$group,$balance));
    while( $sth_user -> fetch )
    {
        my $query_counter = ("SELECT `moneymonth`,`counter` FROM `counter`
                                WHERE `yearmonth`='$date' AND `ip`='$ip'");
        my $sth_counter = $dbh -> prepare($query_counter);
        $sth_counter -> execute;
        my ($moneymonth,$counter);
        $sth_counter -> bind_columns(undef,\($moneymonth,$counter));
        $sth_counter -> fetch;
        $sth_counter -> finish;
        if ($moneymonth eq 'n')
        {
            my $query_group = ("SELECT `trafcost`,`abon`,`abontraf` FROM
                                `group` WHERE `group`='$group'");
            my $sth_group = $dbh -> prepare($query_group);
            $sth_group -> execute;
            my ($trafcost,$abon,$abontraf);
            $sth_group -> bind_columns(undef,\($trafcost,$abon,$abontraf));
            $sth_group -> fetch;
            $sth_group -> finish;
            if ($balance >= ConvertMoneyToDB($abon))
            {
                my $MoneyForAbon = ConvertMoneyToDB($abon)-1;
                my $query_abon = ("UPDATE `user` SET balance = balance -
                                    $MoneyForAbon WHERE `ip`='$ip'");
                my $sth_abon = $dbh -> prepare($query_abon);
                $sth_abon -> execute;
                $sth_abon -> finish;
            }
        }
    }
    $sth_user -> finish;
}
sub SubtractMoney
{
    $ip = $_[0];
    $traffic = $_[1];
    my $query_user = ("SELECT `group` FROM user WHERE `ip`='$ip'");
    #print "$query_user\n";
    my $sth_user = $dbh -> prepare($query_user);
    $sth_user -> execute;
    my ($group);
    $sth_user -> bind_columns(undef,\($group));
    $sth_user -> fetch;
    $sth_user -> finish;
    my $query_group = ("SELECT `trafcost`,`abontraf` FROM `group`
                        WHERE `group`='$group'");
    my $sth_group = $dbh -> prepare($query_group);
    $sth_group -> execute;
}

```

```

my ($trafcost, $abontraf);
$sth_group -> bind_columns(undef, \($trafcost, $abontraf));
$sth_group -> fetch;
$sth_group -> finish;
    my $query_counter = ("SELECT `moneymonth`,`counter` FROM `counter`
                          WHERE `yearmonth`='$date' AND `ip`='$ip'");
    my $sth_counter = $dbh -> prepare($query_counter);
$sth_counter -> execute;
my ($moneymonth, $counter);
$sth_counter -> bind_columns(undef, \($moneymonth, $counter));
$sth_counter -> fetch;
$sth_counter -> finish;
if ($counter > $abontraf*1048576)
{
    my $ByteCost = $trafcost/(100*1024*1024);
    my $MoneyToSubtract = ConvertMoneyToDB($ByteCost*$traffic);
my $query_balance = ("UPDATE `user` SET balance = balance - $MoneyToSubtract
                     WHERE `ip`='$ip'");
    my $sth_balance = $dbh -> prepare($query_balance);
    $sth_balance -> execute;
    $sth_balance -> finish;
}
}
sub KillUserForward
{
    my $query = ("SELECT ip,access FROM user");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    my ($ip, $access);
    $sth -> bind_columns(undef, \($ip, $access));
    while($sth -> fetch)
    {
        #print "$ip $access\n";
        if ($access eq "n")
        {
            my $AllreadyOff='n';
            open (KILLHFORWARD, "iptables -L FORWARD -n |");
            while (my $s=<KILLHFORWARD>)
            {
                if (($.>2))
                {
                    @ar = split /\s+/, $s;
                    #print "$ar[0] $ar[3]\n";
                    if (($ar[0] eq "REJECT") && ($ar[3] eq "$conf{'network'}"."$ip"))
                    {
                        $AllreadyOff = 'y';
                    }
                }
            }
            close(KILLHFORWARD);
            if ($AllreadyOff eq 'n')
            {
                $string = "iptables -I FORWARD -s ".$conf{'network'}.$ip." -j REJECT";
                system($string);
            }
            else {
                my $AllreadyOff='n';
                open (KILLHFORWARD, "iptables -L FORWARD -n |");
                while (my $s=<KILLHFORWARD>)
                {
                    if (($.>2))
                    {
                        @ar = split /\s+/, $s;
                        #print "$ar[0] $ar[3]\n";
                        if (($ar[0] eq "REJECT") && ($ar[3] eq
"$conf{'network'}"."$ip"))
                        {
                            $AllreadyOff = 'y';
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
    }
    close(KILLHFORWARD);
    if ($AllreadyOff eq 'y')
    {
$string = "iptables -D FORWARD -s ".$conf{'network'}. $ip." -j REJECT";
        system($string);
    }
}
}
$sth -> finish;
}
sub KillUserOutput
{
    my $query = ("SELECT ip,access FROM user");
    my $sth = $dbh -> prepare($query);
    $sth -> execute;
    my ($ip,$access);
    $sth -> bind_columns(undef,\($ip,$access));
    while($sth -> fetch)
    {
        print "$ip $access\n";
        if ($access eq "n")
        {
            my $AllreadyOff='n';
            open (KILLHOUTPUT, "iptables -L OUTPUT -n |");
            while (my $s=<KILLHOUTPUT>)
            {
                if (($.>2))
                {
                    {
                        @ar = split /\s+/, $s;
                        #print "$ar[0] $ar[3]\n";
                    if (($ar[0] eq "REJECT") && ($ar[4] eq "$conf{'network'}. "$ip"))
                    {
                        $AllreadyOff = 'y';
                    }
                }
            }
            close(KILLHOUTPUT);
            if ($AllreadyOff eq 'n')
            {
my $string = "iptables -I OUTPUT -d ".$conf{'network'}. $ip." -p tcp --sport 3128
                -j REJECT";
                system($string);
            }
            } else {
                my $AllreadyOff='n';
                open (KILLHOUTPUT, "iptables -L OUTPUT -n |");
                while (my $s=<KILLHOUTPUT>)
                {
                    if (($.>2))
                    {
                        {
                            @ar = split /\s+/, $s;
                            #print "$ar[0] $ar[3]\n";
                        if (($ar[0] eq "REJECT") && ($ar[4] eq "$conf{'network'}. "$ip"))
                        {
                            $AllreadyOff = 'y';
                        }
                    }
                }
            }
            close(KILLHOUTPUT);
            if ($AllreadyOff eq 'y')
            {
                my $string = "iptables -D OUTPUT -d ".$conf{'network'}. $ip." -
                p tcp --sport 3128 -j REJECT";
                system($string);
            }
        }
    }
}
}

```

```

        $sth -> finish;
    }

    sub ConnectToDB
    {
        $datasource = "DBI:mysql:". $conf{ 'DBName' }. ":". $conf{ 'host' };
        $dbh = DBI -> connect($datasource, $conf{ 'username' }, $conf{ 'password' });
    }

    sub GetDate
    {
        open (DATE_H, "date +%y%m |");
        $date = <DATE_H>;
    }

    sub TestNextMonth
    {
        my $query = ("SELECT ip FROM counter WHERE yearmonth='$date'");
        my $sth = $dbh -> prepare($query);
        $sth -> execute;
        my $numrows = $sth -> rows;
        $sth -> finish;
        if ($numrows == 0)
        {
            for (my =1;$i<=254;$i++)
            {
                my $query = ("INSERT INTO counter (ip,yearmonth) VALUES
($i,'$date')");
                my $sth = $dbh -> prepare($query);
                $sth -> execute;
                $sth -> finish;
            }
        }
    }

    sub PutTrafToDBOutput
    {
        $count=0;
        open (FH, "iptables -L -Z OUTPUT -nvx |");
        print "$date\n";
        while(my $s=<FH>)
        {
            if (($.>2))
            {
                # $s =~ s/\s+/_/g;
                @ar = split /\s+/, $s;
                if (($ar[9] =~ /\$conf{'network'}\d+/)&&($ar[3] ne "REJECT"))
                {
                    @ip = split /\./, $ar[9];
                    if ($ar[2] ne "0")
                    {
                        if (($ip[3] >= 2) && ($ip[3] <= 254))
                        {
                            my $query = ("UPDATE counter SET counter = counter + $ar[2] WHERE
ip='$ip[3]' AND yearmonth='$date'");
                            my $sth = $dbh -> prepare($query);
                            $sth -> execute;
                            $sth -> finish;
                            SubtractMoney("$ip[3]","$ar[2]");
                        }
                        $count+=$ar[2];
                    }
                }
            }
        }
        print "\nOverall OUTPUT: $count\n";
        close(FH);
    }

    sub PutTrafToDBForward
    {
        $count=0;
        open (FH, "iptables -L -Z FORWARD -nvx |");
    }

```

```

print "$date\n";
while(my $s=<FH>)
{
    if (($.>2))
    {
        # $s =~ s/\s+/_/g;
        @ar = split /\s+/, $s;
        if (($ar[9] =~ /\$conf{'network'}\d+/)&&($ar[3] ne "REJECT"))
        {
            @ip = split /\./, $ar[9];
            if ($ar[2] ne "0")
            {
                if (($ip[3] >= 2) && ($ip[3] <= 254))
                {
                    my $query = ("UPDATE counter SET counter = counter + $ar[2] WHERE
ip='$ip[3]' AND yearmonth='$date'");
                    my $sth = $dbh -> prepare($query);
                    $sth -> execute;
                    $sth -> finish;
                    SubtractMoney("$ip[3]","$ar[2]");
                }
                $count+=$ar[2];
            }
        }
    }
}
print "\nOverall FORWARD: $count\n";
close(FH);
}
sub Disconnect
{
    $dbh -> disconnect;
}
return 1; // повернення коду
END { }

```

## ВИХІДНИЙ КОД ФАЙЛА COMMON.PHP

```

<?php
//Берестенко Даниїл Олексійович
// ЦНТУ, кафедра КБПЗ, КІ-21-2, 2025
// функція підключення та роботи з БД
function ConnectToBase($host,$username,$password,$database)
{
    global $link;
    if (!($link = mysql_pconnect( $host, $username, $password)))
        {
            echo ("<br>Соединение с базой не установлено<br>");
            exit();
        }
// вибор БД
    if (!mysql_select_db($database, $link))
        {
            echo ( "Помилка при виклику БД. ".mysql_errno($link)."
                ".mysql_error($link) );
            exit();
        }
}

// Функція відображення форми пошуку користувачів
function ShowSearchForm($month)
{
    ?>
<table cellpadding="0" cellspacing="0" border="0">
<tr><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <form method="GET" action=?echo ("$_PHP_SELF");?>
    <tr class="even"><td height=30>
        Login:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="login">
    </td>
    </tr>
    <tr class="even"><td height=30>
        IP:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="ip">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Им'я:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="name">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Адреса:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="address">
    </td>
    </tr>
    <tr class="even"><td height=30>
        Доступ:
    </td>
    <td bgcolor="#ffffff">
        <input type="radio" name="access" value="y">Есть&nbsp;
        <input type="radio" name="access" value="n">Нет&nbsp;
        <input type="radio" name="access" value="-1" checked>-
    </td>

```

```

</tr>
<tr class="even"><td height=30>
    Місяць:
</td>
<td bgcolor="#ffffff">
    <select name="month">
<?
        $i=1;
        while ($month[$i])
        {
            if (GetCurrentDateDB()==$month[$i])
            {
                echo "<OPTION VALUE=\"\$month[$i]\"
SELECTED>".ConvertDateDB($month[$i])."</option>";
            } else {
                echo "<OPTION
VALUE=\"\$month[$i]\">".ConvertDateDB($month[$i])."</option>";
            }
            $i++;
        }
?>
    </select>
</td>
</tr>
<tr class="odd"><td></td><td height=30 align="left">
    <input type="submit" value="Искать">
</td></tr>
    <input type="hidden" name="action" value="search" >
</form>
</table>
</td></tr>
</table>
</td>
<td valign="top">
<table cellpadding="0" cellspacing="0" border="0" width="300">
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=adduserform");?>">Додати нового
користувача</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=editgroup");?>">Редагування груп</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=stat");?>">Статистика</a>
</td></tr>
<tr class="odd"><td height=40 align="right">
<a href="<? echo ("\$PHP_SELF"."?action=deleteold");?>">Видалення інформації</a>
</td></tr>
</table>
</td></tr>
</table>
<?
}

// Функція видалення
function DeleteOld()
{
    global $delete, $link, $month;
    $month_ar = GetMonths();
?>
<table cellpadding="0" cellspacing="2" border="0">
<tr class="odd"><td>
<a href="<? echo ("\$PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd"><td>
Видалити інформацію о трафіке та платежах за:
</td></tr>

```

```

<tr class="odd"><td>
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
    <select name="month">
<?
        $i=1;
        while ($month_ar[$i])
        {
            if (GetCurrentDateDB()!=$month_ar[$i])
            {
                echo "<OPTION
VALUE=\"$_month_ar[$i]\">".ConvertDateDB($month_ar[$i])."</option>";
            }
            $i++;
        }
?>
    </select>
</td></tr>
<tr><td>
<br>
        <input type="hidden" name="action" value="deleteold">
        <input type="hidden" name="delete" value="1">
        <input type="submit" value="Удалить">
</form>
</td></tr>
</table>
<?
    if (($delete)&&($month))
    {
        $query1="DELETE FROM `counter` WHERE `yearmonth`='$_month.'";
        $query2="DELETE FROM `money` WHERE `date` LIKE '$_month.%'";
        echo $query1."<br>".$query2;
        if (!( $result1 = mysql_query($query1, $link ) ))
        {
            echo ( "Помилка при виконанні $query1
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        if (!( $result2 = mysql_query($query2, $link ) ))
        {
            echo ( "Помилка при виконанні $query2
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
    }
}

// Функція пошуку
function Search()
{
    global $name, $login, $ip, $address, $access, $month, $link, $sortBy;
    if ($ip != "")
    {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE `ip`='$ip'";
    }
    elseif ($login != "")
    {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE `login` like '%$login%'";
    } else {
        $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
`balance`, `group` FROM user WHERE 1=1";
        if ($name != "") {$query_user.=" AND `name` like '%$name%'"; }
        if ($address != "") {$query_user.=" AND `address` like
'%$address%'";}
        if ($access !="-1") {$query_user.=" AND `access` like
'$access'";}
    }
    if ($sortBy=="ip_asc")

```

```

{
$query_user.=" ORDER BY ip ASC";
} elseif ($sortby=="ip_desc")
{
$query_user.=" ORDER BY ip DESC";
} elseif ($sortby=="login_desc")
{
$query_user.=" ORDER BY login DESC";
} elseif ($sortby=="login_asc")
{
$query_user.=" ORDER BY login ASC";
} elseif ($sortby=="group_desc")
{
$query_user.=" ORDER BY `group` DESC";
} elseif ($sortby=="group_asc")
{
$query_user.=" ORDER BY `group` ASC";
} elseif ($sortby=="access_desc")
{
$query_user.=" ORDER BY `access` DESC";
} elseif ($sortby=="access_asc")
{
$query_user.=" ORDER BY `access` ASC";
} elseif ($sortby=="balance_desc")
{
$query_user.=" ORDER BY balance DESC";
} elseif ($sortby=="balance_asc")
{
$query_user.=" ORDER BY balance ASC";
} elseif ($sortby=="address_desc")
{
$query_user.=" ORDER BY address DESC";
} elseif ($sortby=="address_asc")
{
$query_user.=" ORDER BY address ASC";
} elseif ($sortby=="name_desc")
{
$query_user.=" ORDER BY name DESC";
} elseif ($sortby=="name_asc")
{
$query_user.=" ORDER BY name ASC";
} else {
$query_user.=" ORDER BY ip ASC";
}
if ($sortby == "traffic_asc")
{
$query_counter = "SELECT `ip`, `counter`, `moneymonth` FROM `counter`
WHERE `yearmonth`='".$$.month.'" AND `counter` > 0 ORDER BY 'counter' ASC";
if (!( $result_counter = mysql_query( $query_counter, $link ) ))
{
echo ( "Помилка при виконанні $query_counter ".mysql_errno($link)."
.mysql_error($link) );
exit();
}
} elseif ($sortby == "traffic_desc")
{
$query_counter = "SELECT `ip`, `counter`, `moneymonth` FROM `counter`
WHERE `yearmonth`='".$$.month.'" AND `counter` > 0 ORDER BY 'counter' DESC";
if (!( $result_counter = mysql_query( $query_counter, $link ) ))
{
echo ( "Помилка при виконанні $query_counter
.mysql_errno($link)." ".mysql_error($link) );
exit();
}
}
}
if (!( $result_user = mysql_query( $query_user, $link ) ))
{
echo ( "Помилка при виконанні $query_user
.mysql_errno($link)." ".mysql_error($link) );
}

```

```

        exit ();
    }
    $backst = getenv("QUERY_STRING");
    $row_user = mysql_fetch_object( $result_user );
    if ( mysql_num_rows($result_user)==1 )
    {
        ShowUserAdmin($row_user -> ip,$month);
    } elseif (mysql_num_rows($result_user) > 1)
    {
        ?>
        <table cellpadding="0" cellspacing="0" border="0">
        <tr class="odd"><td>
        <a href="<? echo ("$_PHP_SELF"."?action=");?>">Вернуться</a>
        </td></tr>
        <tr class="edge"><td>
        <table cellpadding="2" cellspacing="1" border="0">
        <tr class="top" align="center">
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="ip_asc") { echo "sortby=ip_desc"; }
        else { echo "sortby=ip_asc"; }?>>IP</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="login_asc") { echo
        "sortby=login_desc"; } else { echo "sortby=login_asc"; }?>>Login</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="name_asc") { echo "sortby=name_desc";
        } else { echo "sortby=name_asc"; }?>>Имя</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="group_asc") { echo
        "sortby=group_desc"; } else { echo "sortby=group_asc"; }?>>Группа</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="address_asc") { echo
        "sortby=address_desc"; } else { echo "sortby=address_asc"; }?>>Адрес</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="access_desc") { echo
        "sortby=access_asc"; } else { echo "sortby=access_desc"; }?>>Доступ</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="traffic_desc") { echo
        "sortby=traffic_asc"; } else { echo "sortby=traffic_desc"; }?>>Трафик</a></td>
        <td><a href=<?echo
        ("$_PHP_SELF"."?action=search&name=$name&login=$login&ip=$ip&address=$address&acc
        ess=$access&month=$month&"); if ($sortby=="balance_asc") { echo
        "sortby=balance_desc"; } else { echo "sortby=balance_asc";
        }?>>Баланс (грн.)</a></td>
        <td>user.balance</td>
        <td>Абонентка</td>
        <td></td></tr>
        <?
        if (($sortby=="traffic_asc") || ($sortby=="traffic_desc"))
        {
            $i=1;
            mysql_data_seek( $result_user, 0);
            while (( $row_counter = mysql_fetch_object( $result_counter ))
            {
                $query_user = "SELECT `ip`, `login`, `access`, `name`, `address`,
                `balance`, `group` FROM `user` WHERE `ip`='".$row_counter -> ip.'"";
            if (!( $result_user = mysql_query($query_user, $link ) ))
            {
                echo ( "Помилка при виконанні $query_user ".mysql_errno($link)."
                ".mysql_error($link) );
                exit ();
            }
            $row_user = mysql_fetch_object( $result_user );

```

```

        if ($i%2) {$class="odd";} else {$class="even";}
        $i++;
        printf("<tr class=\"\$class\"><td>%s</td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td
align=\"center\">%s</td><td>%s</td><td align=\"center\">%s</td><td
align=\"right\">%s</td><td align=\"right\">%s</td><td align=\"right\">%s</td><td
align=\"center\">%s</td><td><a href=\"\$PHP_SELF?action=delete&ip=%s\"
onclick=\"return confirmLink(this, 'видалити користувача
`s`s`)\">>Видалити</a></td></tr>\n",

$row_user -> ip, $row_user -> ip, $month, $row_user -> login, $row_user -> ip,
$month, $row_user -> name, $row_user -> group, $row_user -> address, $row_user -
> access, number_format($row_counter -> counter,0, chr(44), "."),
ConvertMoneyFromDB($row_user -> balance), $row_user -> balance, $row_counter ->
moneymonth, $row_user -> ip, $row_user -> login);
    }
} else {
    $i=1;
    mysql_data_seek( $result_user, 0);
    while (( $row_user = mysql_fetch_object( $result_user )))
    {
        $query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`='.$row_user -> ip.'" AND `yearmonth`='.$month;
        if (!( $result_counter =
mysql_query($query_counter, $link ) ))
        {
            echo ( "Помилка при виконанні
$query_counter ".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        $row_counter = mysql_fetch_object(
$result_counter );
        if ($i%2) {$class="odd";} else {$class="even";}
        $i++;
        printf("<tr class=\"\$class\"><td>%s</td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td><a
href=\"\$PHP_SELF?action=show&ip=%s&month=%s\">%s</a></td><td
align=\"center\">%s</td><td>%s</td><td align=\"center\">%s</td><td
align=\"right\">%s</td><td align=\"right\">%s</td><td align=\"right\">%s</td><td
align=\"center\">%s</td><td><a href=\"\$PHP_SELF?action=delete&ip=%s\"
onclick=\"return confirmLink(this, 'видалити користувача
`s`s`)\">>Видалити</a></td></tr>\n",
        $row_user -> ip, $row_user -> ip, $month, $row_user ->
login, $row_user -> ip, $month, $row_user -> name, $row_user -> group, $row_user
-> address, $row_user -> access, number_format($row_counter -> counter,0,
chr(44), "."), ConvertMoneyFromDB($row_user -> balance), $row_user -> balance,
$row_counter -> moneymonth, $row_user -> ip, $row_user -> login);
    }
}
?>
</table>
</td></tr>
</table>
<?
} else
{
?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("\$PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd"><td>
Нема користувачів, відповідних пошукові
</td></tr>
</table>
<?>
mysql_free_result( $result_user );
    if ($result_counter)

```

```

        {
            mysql_free_result( $result_counter );
        }
echo $query_user;
echo "<br>".$query_counter;
}

// Функція конвертування коштів
function ConvertMoneyFromDB($money)
{
    $NewMoney = number_format($money/100000000, 2, chr(44), ".");
    return $NewMoney; // повернення коду коштів
}

// Функція конвертування коштів
function ConvertMoneyToDB($money)
{
    $NewMoney = $money*100000000;
    return $NewMoney; // повернення коду коштів
}

// Функція конвертування дати у БД
function ConvertDateDBMoney($dateDB)
{
    $date_ar = explode(" ", $dateDB);
    $time = $date_ar[1];
    $year = substr($date_ar[0], 0, 2);
    $month = substr($date_ar[0], 2, 2);
    $date = substr($date_ar[0], -2);
    switch ( $month ) {
        case "1":
            $monthstring = "січня";
            break;
        case "2":
            $monthstring = "лютого";
            break;
        case "3":
            $monthstring = "березня";
            break;
        case "4":
            $monthstring = "квітня";
            break;
        case "5":
            $monthstring = "травня";
            break;
        case "6":
            $monthstring = "червня";
            break;
        case "7":
            $monthstring = "липня";
            break;
        case "8":
            $monthstring = "серпня";
            break;
        case "9":
            $monthstring = "вересня";
            break;
        case "10":
            $monthstring = "жовтня";
            break;
        case "11":
            $monthstring = "листопаду";
            break;
        case "12":
            $monthstring = "грудня";
            break;
        default:
            $monthstring = "-1";
    }
}

```

```

$newdate = $date." ".$monthstring." 20".$year." ".$time;
return $newdate; // повернення коду дати
}

```

```

// Функція конвертування дати
function ConvertDateDB($DateDB)
{

```

```

$year = substr($DateDB,0,2);
$month = substr($DateDB,-2);
    switch ( $month ) {
        case "1":
            $monthstring = "січень";
            break;
        case "2":
            $monthstring = "лютий";
            break;
        case "3":
            $monthstring = "березень";
            break;
        case "4":
            $monthstring = "квітень";
            break;
        case "5":
            $monthstring = "травень";
            break;
        case "6":
            $monthstring = "червень";
            break;
        case "7":
            $monthstring = "липень";
            break;
        case "8":
            $monthstring = "серпень";
            break;
        case "9":
            $monthstring = "вересень";
            break;
        case "10":
            $monthstring = "жовтень";
            break;
        case "11":
            $monthstring = "листопад";
            break;
        case "12":
            $monthstring = "грудень";
            break;
        default:
            $monthstring = "-1";
    }

```

```

$newdate = $monthstring." 20".$year;
return $newdate; // повернення коду дати
}

```

```

// Функція отримання поточного місяця
function GetMonths()
{

```

```

    global $link;
    $query = "SELECT yearmonth FROM counter WHERE `ip`=1";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
//друкування результату
    $i = 1;

```

```

while (( $row = mysql_fetch_object( $result ))
{
    $month[$i] = $row -> yearmonth;
    $i++;
}
mysql_free_result( $result );
return $month; // повернення коду місяця
}

// Функція лічильника
function ShowCounter($ip, $dateDB)
{
    global $link;
    $query = "SELECT ip,counter, yearmonth FROM counter WHERE `ip`=$ip AND
`yearmonth`=$dateDB";
    if (!( $result = mysql_query($query, $link ) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    //друкування результату
    echo("<table cellpadding=0 cellspacing=5><tr><td colspan=2
class=left>");
    echo("</td></tr>");
    while (( $row = mysql_fetch_object( $result ))
    {
        printf("<tr><td>%s</td><td>%s</td><td>%s</td></tr>\n",
$row -> yearmonth, $row -> ip, $row -> counter);
    }
    echo("</table>");
    mysql_free_result( $result );
}

// Функція роботи з трафіком
function TrafRemainder($balance, $group, $counter, $moneymonth)
{
    global $link;
    $query_group = "SELECT `trafcost`,`abontraf` FROM `group` WHERE
`group`='$.$group.'";
    if (!( $result_group = mysql_query($query_group, $link ) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $row_group = mysql_fetch_object( $result_group );
    if ($row_group -> trafcost > 0)
    {
        if ($moneymonth=="y")
        {
            if ($counter < ($row_group -> abontraf*1048576))
            {
                $TrafRemainder = number_format(($row_group -> abontraf*1048576-
$counter)/1048576+ConvertMoneyFromDB(($balance)/(($row_group -> trafcost)/100),
2, chr(44), "."));
            } else {
                $TrafRemainder = ConvertMoneyFromDB(($balance)/(($row_group ->
trafcost)/100));
            }
        } else { $TrafRemainder = 0; }
    } else {
        $TrafRemainder = "-";
    }
}
return $TrafRemainder; // повернення коду трафіку
}

// Функція відображення сумарної статистики
function ShowStat()

```

```

{
    global $link, $monthtoshow;
$query1 = "SELECT `ip` FROM `user` WHERE `access`='y'";
$query2 = "SELECT `ip` FROM `user`;
if ($monthtoshow)
{
$query3 = "SELECT `money` FROM `money` WHERE `date` LIKE '". $monthtoshow. "%'";
$query4 = "SELECT `counter` FROM `counter` WHERE `yearmonth` LIKE
'". $monthtoshow. "'";
} else {
$query3 = "SELECT `money` FROM `money` WHERE `date` LIKE
'".GetCurrentDateDB()."%'";
$query4 = "SELECT `counter` FROM `counter` WHERE `yearmonth` LIKE
'".GetCurrentDateDB()."';
}

    if (!( $result1 = mysql_query($query1, $link ) ))
    {
        echo ( "Помилка при виконанні $query1
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result2 = mysql_query($query2, $link ) ))
    {
        echo ( "Помилка при виконанні $query2
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result3 = mysql_query($query3, $link ) ))
    {
        echo ( "Помилка при виконанні $query3
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if (!( $result4 = mysql_query($query4, $link ) ))
    {
        echo ( "Помилка при виконанні $query4
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $activeusers=mysql_num_rows($result1);
    $allusers=mysql_num_rows($result2);
    while ( $row_money = mysql_fetch_object( $result3 ) )
    {
        $allmoney+=$row_money -> money;
    }
    while ( $row_counter = mysql_fetch_object( $result4 ) )
    {
        $alltraffic+=$row_counter -> counter;
    }
    while ( $row_user = mysql_fetch_object( $result2 ) )
    {
        if ($monthtoshow)
        {
            $query5 = "SELECT `counter` FROM `counter` WHERE `ip`='". $row_user -> ip. "' AND
`yearmonth` LIKE '". $monthtoshow. "'";
        } else {
            $query5 = "SELECT `counter` FROM `counter` WHERE `ip`='". $row_user -> ip. "'
AND `yearmonth` LIKE '".GetCurrentDateDB()."';
        }
        if (!( $result5 = mysql_query($query5, $link ) ))
        {
            echo ( "Помилка при виконанні $query5
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        $row_counterusers = mysql_fetch_object( $result5 );
        $alltrafficusers+=$row_counterusers -> counter;
    }
?>

```

```

<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF"."?action=");?>">Вернуться</a>
</td></tr>
<tr class="odd"><td valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <tr class="odd">
    <td height=30>Кількість активних користувачів</td><td width="30"
align="center"><? echo $activeusers;?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальна кількість користувачів</td><td width="30"
align="center"><? echo $allusers;?></td>
    </tr>
    <tr class="odd">
    <td height=30>Кількість зібраних грошей (грн.)</td><td width="30"
align="center"><? echo number_format($allmoney,0, chr(44), ".");?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальний трафік:</td><td width="30" align="center"><? echo
number_format($alltraffic,0, chr(44), ".");?></td>
    </tr>
    <tr class="odd">
    <td height=30>Загальний трафік (використовуючи таблицю user):</td><td
width="30" align="center"><? echo number_format($alltrafficusers,0, chr(44),
".");?></td>
    </tr>
</table>
</td>
</tr>
<tr class="odd"><td>
    <form method="GET" action="<? echo ("$_PHP_SELF");?>">
    <select name="monthtoshow">
<?
    $month = GetMonths();
    $i=1;
    while ($month[$i])
    {
    if (GetCurrentDateDB()==$month[$i])
    {
    echo "<OPTION VALUE=\"\$month[$i]\"
SELECTED>".ConvertDateDB($month[$i])."</option>";
    } else {
    echo "<OPTION
VALUE=\"\$month[$i]\">".ConvertDateDB($month[$i])."</option>";
    }
    $i++;
    }
?>
    </select>
    <input type="hidden" name="action" value="stat">
    <br>
    <input type="submit" value="Показать">
    </form>
</td></tr>
</table>
<?
}

// Функція оновлення групи
function UpdateGroup()
{
    global $link, $group, $abon, $abontraf, $trafcost, $message;
    $query = "UPDATE `group` SET `abon`='". $abon.'"',
    `abontraf`='". $abontraf.'"', `trafcost`='". $trafcost.'" WHERE
`group`='". $group.'"';
    if (!( $result = mysql_query($query, $link ) ))
    {

```

```

        echo ( "Помилка при виконанні $query ".mysql_errno($link)." ".
            mysql_error($link) );
            exit();
        } else {
            $message="Updated";
        }
    }

// Функція видалення групи
function DelGroup()
{
    global $link, $group, $message;
    $query = "DELETE FROM `group` WHERE `group`='".$group."'";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    } else {
        $message="Група ".$group." удалена";
    }
}

// Функція додавання групи
function AddGroup()
{
    global $link, $groupname;
    if ($groupname)
    {
        $query_add = "INSERT INTO `group` ( `group` ) VALUES ( '$groupname' )";
        if (!( $result = mysql_query($query_add, $link) ))
        {
            echo ( "Помилка при виконанні $query_add
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
    }
}

// Функція редагування групи
function EditGroup()
{
    global $link, $group, $message;
    ?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF.">action=");?>">Вернуться</a>
</td></tr>
<tr class="odd"><td valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="3" cellspacing="1" border="0">
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
<tr class="even"><td height=30>
    Група:
</td>
<td bgcolor="#ffffff" width="110">
<select name="group">
<?
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    while ( $row_group = mysql_fetch_object( $result_group ) )
    {
        if ($row_group -> group == $group)

```

```

        {
        ?>
        <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
        <?
        } else {
        ?>
        <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
        <?
        }
        }
?>

</select>
</td></tr>
<tr class="odd"><td></td><td height=30 align="left">
<input type="submit" value="Показати">
<input type="hidden" name="action" value="editgroup" >
</form>
</td></tr>
<tr class="even"><td colspan="2">
Додати групу:
</td></tr>
<tr class="odd"><td>
<form method="POST" action="<? echo ("$_PHP_SELF");?>">
<input type="text" size=8 name="groupname">
</td><td>
<input type="hidden" name="action" value="addgroup" >
<input type="submit" value="Додати">
</form>
</td></tr>
</table>
</td></tr>
<? if ($group)
{?>
<tr class="odd"><td>
<a href="<?echo ("$_PHP_SELF". "?action=delgroup&group=$group")?>"
onclick="return confirmLink(this, 'видалити групу `<? echo
$group;?>`')">Видалити групу <?echo $group;?></a>
</td></tr>
<?}?>
</table>
</td>

<td width="500" align="right" valign="top">
<table cellpadding="0" cellspacing="0" border="0">
<tr><td>
<?
    if ($group != "")
    {
        $query_group = "SELECT `trafcost`, `abon`, `abontraf` FROM `group` WHERE
`group`='<?echo $group;?>'";
        if (!( $result_group = mysql_query($query_group, $link ) ) )
        {
            echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        $row_group = mysql_fetch_object( $result_group );
    }
?>

<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<form name="updategroup" method="POST" action="<?echo
("$_PHP_SELF");?>>
<table cellpadding="2" cellspacing="1" border="0">
<tr class="odd"><td>Вартість трафіка (грн. за 100
Мбайт):</td><td align="right"><input type="text" name="trafcost" value="<? echo
$row_group -> trafcost; ?>" size="2"></td></tr>

```

```

                <tr class="odd"><td>Абонентська плата (грн. за
місяць):</td><td align="right"><input type="text" name="abon" value="<? echo
$row_group -> abon; ?>" size="2"></td></tr>
                <tr class="odd"><td>Кількість трафіка, вхідного в
абонентську плату (Мбайт):</td><td align="right"><input type="text"
name="abontraf" value="<? echo $row_group -> abontraf; ?>" size="2"></td></tr>
                <tr class="odd"><td colspan="2" align="right"><input
type="submit" value="Відіслати"></td></tr>
            </table>
        </td></tr>
        <input type="hidden" name="group" value="<? echo $group; ?>">
        <input type="hidden" name="action" value="updategroup">
    </form>
</table>
<?
}
?>
</td></tr>
</table>
</td>
<td valign="top" align="right" width="70"><font color="red"><? echo
$message;?></td>
</tr>
</table>
<?
}

// Функція додавання користувача
function AddUser()
{
    global $link, $login, $ip, $name, $address, $moneyinet, $moneynetwork,
$balance, $group, $trafficebegin, $SubtractAb;
    $balanceDB = ConvertMoneyToDB($balance);
    $query_user = "INSERT INTO `user` ( `login`, `ip`, `name`, `address`,
`moneyinet`, `moneynetwork`, `balance`, `group`) VALUES ( '$login', '$ip',
'$name', '$address', '$moneyinet', '$moneynetwork', '$balanceDB', '$group' )";
    if (!( $result = mysql_query($query_user, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_user
.mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    if ($trafficebegin)
    {
        $query_counter = "UPDATE `counter` SET `counter`='$trafficebegin',
`moneymonth`='$SubtractAb' WHERE `ip`='$ip' AND
`yearmonth`='".GetCurrentDateDB()."''";
    } else {
        $query_counter = "UPDATE `counter` SET `counter`='0',
`moneymonth`='$SubtractAb' WHERE `ip`='$ip' AND
`yearmonth`='".GetCurrentDateDB()."''";
    }
    if (!( $result_counter = mysql_query($query_counter, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_counter
.mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $date = GetCurrentDateDBMoney();
    $query_money = "INSERT INTO `money` ( `ip`, `money`, `date`) VALUES (
'$ip', '$balance', '$date' )";
    if (!( $result = mysql_query($query_money, $link ) ) )
    {
        echo ( "Помилка при виконанні $query_money
.mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
}
}

```

```

// Функція форми користувача
function AddUserForm()
{
    global $link;
?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo ("$_PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
    <table cellpadding="3" cellspacing="1" border="0">
    <form method="GET" action=<?echo ("$_PHP_SELF");?>>
    <tr class="even"><td height=30>
        Login:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="login" size="12">
    </td>
    <td bgcolor="#ffffff">
        маленькими літерами
    </td>
    </tr>
<tr class="even"><td height=30>
        IP:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="ip" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться тільки остання частина ip-адреса
    </td>
    </tr>
<tr class="even"><td height=30>
        Имя:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="name" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться у вигляді: Фамилія Ім'я
    </td>
    </tr>
<tr class="even"><td height=30>
        Адреса:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="address" size="12">
    </td>
    <td bgcolor="#ffffff">
        уводиться в вигляді: 15/1 7-2-3
    </td>
    </tr>
<tr class="even"><td height=30>
        Деньги за сеть:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="moneynetwork" size="12">
    </td>
    <td bgcolor="#ffffff"></td>
    </tr>
<tr class="even"><td height=30>
        Деньги за инет:
    </td>
    <td bgcolor="#ffffff">
        <input type="text" name="moneyinet" size="12">
    </td>
    <td bgcolor="#ffffff"></td>

```

```

</tr>
<tr class="even"><td height=30>
    Баланс:
</td>
<td bgcolor="#ffffff">
    <input type="text" name="balance" size="12">
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Вилучити абонентську плату?
</td>
<td bgcolor="#ffffff">
    <input type="radio" name="SubtractAb" value="n" checked>Да&nbsp;
    <input type="radio" name="SubtractAb" value="y">Нєт&nbsp;
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Початковий трафік (байт):
</td>
<td bgcolor="#ffffff">
    <input type="text" name="trafficbegin" value="0" size="12">
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="even"><td height=30>
    Група:
</td>
<td bgcolor="#ffffff">
    <select name="group">
<?
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    while ( $row_group = mysql_fetch_object( $result_group ) )
    {
        if ($row_group -> group=="comp")
        {
            ?>
            <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
            <?
            } else {
            ?>
            <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
            <?
            }
        }
    }
?>
    </select>
</td>
<td bgcolor="#ffffff"></td>
</tr>
<tr class="odd"><td></td><td height=30 align="left">
    <input type="submit" value="Додати">
</td>
<td bgcolor="#ffffff"></td>
</tr>
    <input type="hidden" name="action" value="adduser" >
</form>
</table>
</td></tr>
</table>

```

```

</td>
</tr>
</table>
<?
}

// Функція видалення користувача
function DeleteUser ()
{
    global $link, $ip;
    $query = "DELETE FROM `user` WHERE `ip`='". $ip. "'";
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query ".mysql_errno($link)."
            ".mysql_error($link) );
        exit();
    }
    $query_counter = "UPDATE `counter` SET `counter`='0', `moneymonth`='n'
WHERE `ip`='$ip' AND `yearmonth`='".GetCurrentDateDB()." ";
    if (!( $result_counter = mysql_query($query_counter, $link) ))
    {
        echo ( "Помилка при виконанні $query_counter
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
}

// Функція оновлення даних користувача
function UpdateUser ()
{
    global $link, $message, $ip, $login, $group, $address, $moneynetwork,
$moneynet;
    $query = "UPDATE `user` SET `login`='". $login. "', `group`='". $group. "',
`address`='". $address. "', `moneynetwork`='". $moneynetwork. "',
`moneynet`='". $moneynet. "' WHERE `ip`='". $ip;
    if (!( $result = mysql_query($query, $link) ))
    {
        echo ( "Помилка при виконанні $query
".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    } else {
        $message = "Updated";
    }
}

// Функція додавання до балансу
function IncBalance ()
{
    global $link, $message, $ip, $incbalance ;
    if ($incbalance)
    {
        $query = "UPDATE `user` SET balance = balance +
".ConvertMoneyToDB($incbalance)." WHERE `ip`='". $ip;
        if (!( $result = mysql_query($query, $link) ))
        {
            echo ( "Помилка при виконанні $query ".mysql_errno($link)."
            ".mysql_error($link) );
            exit();
        }
        $date = GetCurrentDateDBMoney();
        $query_money = "INSERT INTO `money` ( `ip`, `money`, `date`) VALUES (
            '$ip', '$incbalance', '$date' )";
        if (!( $result = mysql_query($query_money, $link) ))
        {
            echo ( "Помилка при виконанні $query_money
            ".mysql_errno($link)." ".mysql_error($link) );
            exit();
        }
        } else {
            $message = "Updated";
        }
    }
}

```

```

    }
}

// Функція відображення даних адміністратора
function ShowUserAdmin($ip, $month)
{
    global $link, $message, $showpayment;
    $query_user = "SELECT `name`, `login`, `access`, `balance`, `group`, `address`,
        `moneynetwork`, `moneyinet` FROM user WHERE `ip`=$ip";
    if (!( $result_user = mysql_query($query_user, $link) ))
    {
        echo ( "Помилка при виконанні $query_user
            ".mysql_errno($link)." ".mysql_error($link) );
        exit();
    }
    $row_user = mysql_fetch_object( $result_user );
    $query_group = "SELECT `group` FROM `group`";
    if (!( $result_group = mysql_query($query_group, $link) ))
    {
        echo ( "Помилка при виконанні $query_group ".mysql_errno($link)."
            ".mysql_error($link) );
        exit();
    }
}

?>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="odd"><td>
<a href="<? echo (" $PHP_SELF"."?action=");?>">Повернутися</a>
</td></tr>
<tr class="odd" valign="top"><td>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<form name="update" method="POST" action=<?echo (" $PHP_SELF");?>>
<table cellpadding="2" cellspacing="1" border="0">
<?
                $query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`=". $ip ." AND `yearmonth`=". $month;
                if (!( $result_counter = mysql_query($query_counter, $link) ))
                {
                    echo ( "Помилка при виконанні $query_counter
                        ".mysql_errno($link)." ".mysql_error($link) );
                    exit();
                }
                $row_counter = mysql_fetch_object( $result_counter );
?>
<tr class="odd"><td>Дані за:</td><td align="right"><? echo
ConvertDateDB($month);?></td></tr>
<tr class="even"><td>IP:</td><td align="right"><? echo $ip;?></td></tr>
<tr class="odd"><td>Login:</td><td align="right"><input type="text" name="login"
value="<? echo $row_user -> login; ?>" size="8"></td></tr>
<tr class="even"><td>Група:</td><td align="right">

                <select name="group">
?>
                while ( $row_group = mysql_fetch_object( $result_group ) )
                {
                    if ($row_group -> group==$row_user -> group)
                    {
                        ?>
                        <OPTION VALUE="<? echo $row_group -> group; ?>" SELECTED><? echo
$row_group -> group; ?></option>";
                        <?
                    } else {
                        ?>
                        <OPTION VALUE="<? echo $row_group -> group; ?>"><? echo
$row_group -> group; ?></option>";
                        <?
                    }
                }

```

```

    }
  }
?>
    </select>
        </td></tr>
        <tr class="odd"><td>Имя:</td><td align="right"><? echo
$row_user -> name; ?></td></tr>
        <tr class="even"><td>Адрес:</td><td align="right"><input
type="text" name="address" value="<? echo $row_user -> address; ?>"
size="8"></td></tr>
        <tr class="odd"><td>Гроші за мережу:</td><td
align="right"><input type="text" name="moneynetwork" value="<? echo $row_user ->
moneynetwork; ?>" size="2"></td></tr>
        <tr class="even"><td>Гроші за инет:</td><td
align="right"><input type="text" name="moneyinet" value="<? echo $row_user ->
moneyinet; ?>" size="2"></td></tr>
        <tr class="odd"><td>Трафік (байт):</td><td
align="right"><? echo number_format($row_counter -> counter,0, chr(44), ".");
?></td></tr>
        <tr class="even"><td>Баланс (грн.):</td><td
align="right"><font color="red"><? echo ConvertMoneyFromDB($row_user ->
balance); ?></font></td></tr>
        <tr class="odd"><td>balance:</td><td align="right"><?
echo $row_user -> balance; ?></td></tr>
        <tr class="even"><td>Залишилося (МБ):</td><td
align="right"><? echo TrafRemainder($row_user -> balance, $row_user -> group,
$row_counter -> counter, $row_counter -> moneymonth); ?></td></tr>
        <tr class="odd"><td>Доступ:</td><td align="right"><?
echo $row_user -> access; ?></td></tr>
        <tr class="even"><td>Абон. плата:</td><td
align="right"><? echo $row_counter -> moneymonth; ?></td></tr>
        <tr class="odd"><td colspan=2 align="center"><input
type="submit" value="Відправити"></td></tr>
        <input type="hidden" name="ip" value="<? echo $ip;?>" >
        <input type="hidden" name="month" value="<? echo $month;?>" >
        <input type="hidden" name="action" value="update" >
        </form>
    </table>
</td></tr>
</table>
</td>
    <td></td>
    <td valign="top" align="right" width="300">
    <table cellpadding="0" cellspacing="0" border="0">
    <tr class="edge"><td>
        <table cellpadding="2" cellspacing="1" border="0">
        <form name="incbalance" method="POST" action=<?echo ("$_PHP_SELF");?>>
        <tr class="odd">
            <td>Баланс (грн.):</td>
            <td><font color="red"><? echo ConvertMoneyFromDB($row_user -> balance);
?></font></td>
        </tr>
        <tr class="odd">
            <td>Додати (грн.):</td>
            <td><input type="text" name="incbalance" size="8"></td>
        </tr>
        <tr><td bgcolor="#ffffff" height=40 colspan=2 align="center">
            <input type="hidden" name="ip" value="<? echo $ip;?>" >
            <input type="hidden" name="month" value="<? echo $month;?>" >
            <input type="hidden" name="action" value="incbalance" >
            <input type="submit" value="Відправити">
        </td></tr>
        </form>
    </table>
</td></tr>
</table>

<table>
    <tr class="odd" colspan="2">

```

```

<td colspan="2" align="right">
<?
if ($showpayment)
{?>
<a href="<? echo ("$_PHP_SELF"."?action=show&ip=$ip&month=$month");?>">Вилучити
платежі</a>
</td></tr>
</table>
<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="2" cellspacing="1" border="0">
<tr class="odd"><td>Дата</td><td>руб.</td></tr>
<?
$query_money = "SELECT money, date FROM money WHERE `ip`=".$ip." AND `date` like
'".$month."%";
if (!( $result_money = mysql_query($query_money, $link ) ))
{
echo ("Помилка при виконанні $query_money ".mysql_errno($link)."
.mysql_error($link) );
exit();
}
while ($row_money = mysql_fetch_object( $result_money ))
{
echo "<tr class=\"odd\"><td>".ConvertDateDBMoney($row_money ->
date)."</td><td>".$row_money -> money."</td></tr>";
$allmoney+=$row_money -> money;
}
echo "<tr class=\"odd\"><td>Итого:</td><td>".$allmoney."</td></tr>";
echo "</table>";
} else {?>
<a href="<? echo
("$_PHP_SELF"."?action=show&ip=$ip&month=$month&showpayment=1");?>">
Дивитися платежі</a>
<?>?>
</td></tr>
</table>
</td>
<td valign="top"><font color="red"><? echo $message;?></td>
</tr>
</table>
<?
}

// Функція відображення користувача
function ShowUser($ip)
{
global $link;
$query_user = "SELECT `name`, `login`, `access`, `balance`, `group` FROM
user WHERE `ip`=$ip";
if (!( $result_user = mysql_query($query_user, $link ) ))
{
echo ( "Помилка при виконанні $query ".mysql_errno($link)."
.mysql_error($link) );
exit();
}
$row_user = mysql_fetch_object( $result_user );
//друкування результату
?>

<table cellpadding="0" cellspacing="0" border="0">
<tr class="edge"><td>
<table cellpadding="2" cellspacing="1" border="0">
<?
$query_counter = "SELECT counter, moneymonth FROM
counter WHERE `ip`=".$ip." AND `yearmonth`=" .GetCurrentDateDB();
if (!( $result_counter =
mysql_query($query_counter, $link ) ))
{

```

```

echo ( "Помилка при виконанні
$query_counter ".mysql_errno($link)." ".mysql_error($link) );
exit();
}
$row_counter = mysql_fetch_object(
$result_counter );
// виведення даних на екран
printf("
<tr class=\"odd\"><td>Дані за:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Login:</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Ім'я:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Трафік (байт):</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Баланс (грн.):</td><td align=\"right\"><font
color=\"red\">%s</font></td></tr>
<tr class=\"even\"><td>Залишилося (Мб):</td><td align=\"right\">%s</td></tr>
<tr class=\"odd\"><td>Доступ:</td><td align=\"right\">%s</td></tr>
<tr class=\"even\"><td>Абон. плата:</td><td align=\"right\">%s</td></tr>
\n",
ConvertDateDB(GetCurrentDateDB()), $row_user -> login, $row_user -> name,
number_format($row_counter -> counter,0, chr(44), "."),
ConvertMoneyFromDB($row_user -> balance), TrafRemainder($row_user -> balance,
$row_user -> group, $row_counter -> counter, $row_counter -> moneymonth),
$row_user -> access, $row_counter -> moneymonth);
?>
</table>
</td></tr>
</table>
<?
}

// Функція попередження
function ResolveUser()
{
global $network, $conf;
$host = getenv('REMOTE_ADDR');
$ip=explode(".", $host);
$network = $ip[0].".$ip[1].".$ip[2].".";
if ($network != $conf{'network'})
{
echo ("Зареєструйтеся в VPN.");
exit();
}
return $ip[3]; // повернення коду IP
}

// Функція роботи з датою
function GetCurrentDateDB()
{
$current_date=time();
$dateDB=date("ym", $current_date);
return $dateDB; // повернення коду дати у БД
}

function GetCurrentDateDBMoney()
{
$current_date=time();
$date=date("ymd H:i:s", $current_date);
return $date; // повернення коду дати внесення коштів
}

function Stuff()
{
$current_date=time();
echo $current_date."<br>";
$date=date("ymd H:i:s", $current_date);
$dateDB=date("ym", $current_date);
echo $date."<br>".$dateDB."<br>";
echo ConvertDateDBMoney($date);
}

```

```
// функція формування початку HTML сторінки
function Top($path,$pagename)
{
?>
<html>
<head>
<title>Моніторинг трафіку -> <? echo $pagename ?></title>
<META HTTP-EQUIV="Expires" CONTENT="Fri, Jan 01 1900 00:00:00 GMT">
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1251">
<link rel="stylesheet" type="text/css" href="<? echo $path; ?>Monitoring.css">

<script type="text/javascript" language="javascript">
<!--
// коди та текст помилок
var errorMsg0   = 'Потрібно значення для форми!';
var errorMsg1   = 'Це не число!';
var errorMsg2   = 'Неприпустима кількість рядків!';
var noDropDbMsg = 'Команда "Видалити БД" відключена.';
var confirmMsg  = 'Ви дійсно бажаєте ';
//-->
</script>
<script src="<? echo $path;?>functions.js" type="text/javascript"
language="javascript"></script>
</head>
<body>
<?
}

// формування HTML коду кінця сторінки
function Footer()
{
?>
</body>
</html>
<?
}
?>
```