

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»**

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітня програма: «Розробка програмного забезпечення»

Група: 4РП-05

Дипломний проект

**здобувача освіти денної форми навчання
РП.05.24.000.ДП**

**УДОТА
ДМИТРА ВІТАЛІЙОВИЧА**

**м. Одеса
2022 р.**

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: **121 «Інженерія програмного забезпечення»**

Освітня програма: «**Розробка програмного забезпечення»**

Група: **4РП-05**

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту (роботи) на тему:

Розробка програмного забезпечення системи контролю і управління доступом на об'єкт

Проектний матеріал складається з пояснювальної записки на 71 сторінках та графічного (презентаційного) матеріалу на 15 аркушах (слайдах).

Дипломник _____ (Удог Д.В.)

Керівник _____ (Кіреєв І.А.)

Консультанти:

з економічної частини _____ (Копайгородська Т.Г.)

з охорони праці _____ (Чорновол Н.І.)

з дотримання вимог ЄСКД _____ (Петрашова В.І.)

старший консультант _____ (Скорнякова О.В.)

До захисту допущений

Голова циклової комісії _____ (Скорнякова О.В.)

Завідувач відділення _____ (Суліма Ю.Ю.)

Захист « » _____ 2022 р. Протокол ДКК № _____

Оцінка ДКК _____

Секретар ДКК _____

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та Ш
Спеціальність 121 «Інженерія програмного забезпечення»
Освітня програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР Беркань І.В.

“ _____ ” _____ 2022 р.

ЗАВДАННЯ

на дипломний проект (роботу)

Здобувачеві (здобувачці) освіти Удог Дмитру Віталійовичу
(прізвище, ім'я, по батькові)

1. Тема проекту (роботи) Розробка програмного забезпечення системи контролю і управління доступом на об'єкт

затверджена наказом по коледжу від “ _____ ” _____ 2022 р. № _____

2. Термін здачі закінченого проекту (роботи) _____

3. Вихідні данні до проекту (роботи) _____

1. Передбачити застосування автоматизованої прохідної з турнікетом та зчитувачем ID-карт;
2. Зчитувач ідентифікаційних карт під'єднувати до USB-порту для контролю пропуску персоналу;
3. Створити СУБД засобами візуального та об'єктно-орієнтованого програмування
4. Застосовувати таблиці Paradox та механізм BDE для створення бази даних

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)

Склад системи контролю і управлінням доступом на об'єкт; Проектування автоматизованої прохідної на об'єкт; Проектування мережі автоматизованих прохідних на об'єкт; Розробка бази даних автоматизованої прохідної; Розробка програми моніторингу та віддаленого доступу; Конструювання та опис інтерфейсу користувачів; Тестування проекту; Економічна частина; Охорона праці

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Загальна структура організації СКУД; Загальний вигляд автоматизованої прохідної; Ідентифікація співробітника карткою із магнітною стрічкою; Схема мережі автоматизованих прохідних; Логічна схема бази даних автоматизованої прохідної СКУД; Список команд керування контролером турнікету; Блок-схема роботи алгоритму у модулі зв'язку; Головна форма модулю "Monitor"; Інтерфейс модулю "Manager"; Форма інформації про співробітника; Форма переведення співробітника; Форма задавання інтервалів робочого часу; Форма завдання прав доступу

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Технологічний	Кіреєв І.А.		
Економічна частина	Копайгородська Т.Г.		
Охорона праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Скорнякова О.В.		

7. Дата видачі завдання _____

Керівник Кіреєв І.А. _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Вступ. Постановка мети та задач проектування	5.05.2022	
2	Аналіз предметної галузі	7.05.2022	
3	Огляд засобів розробки проекту	9.05.2022	
4	Вибір мови програмування, технології розробки	12.05.2022	
5	Розробка СКУД, проект автоматизованої прохідної	14.05.2022	
6	Проектування мережі автоматизованих прохідних	16.05.2022	
7	Аналіз мережевих ресурсів та обмеження доступу	18.05.2022	
8	Розробка бази даних для ПЗ СКУД	19.05.2022	
9	Розробка структури СУБД ПЗ СКУД	21.05.2022	
10	Розробка алгоритмів програми моніторингу та модулю зв'язку із зчитувачем ID-карт	23.05.2022	
11	Розробка інтерфейсу ПЗ СКУД	26.05.2022	
12	Програмна реалізація ПЗ СКУД у ICP RAD Studio	28.05.2022	
13	Аналіз результатів, підготовка слайдів презентації	30.05.2022	
14	Економічні розрахунки та питання з охорони праці	4.06.2022	
15	Підготовка графічної частини проекту	10.06.2022	
16	Підготовка проекту до захисту та тестування ПП	13.06.2022	

Дипломник _____
(підпис)

Керівник _____
(підпис)

ЗМІСТ

Вступ.....	7
1 Технологічний розділ.....	8
1.1 Склад системи контролю і управлінням доступом на об'єкт.....	8
1.2 Проектування автоматизованої прохідної на об'єкт.....	10
1.2.1 Апаратне забезпечення автоматизованої прохідної на об'єкт.....	10
1.2.2 Робота комплексу автоматизованої прохідної на об'єкт.....	12
1.3 Проектування мережі автоматизованих прохідних на об'єкт.....	12
1.3.1 Мережеві ресурси Windows Network.....	13
1.3.2 Обмеження доступу до мережевих ресурсів.....	14
1.3.3 Схема мережі автоматизованих прохідних на об'єкт.....	14
1.4 Розробка бази даних автоматизованої прохідної.....	15
1.4.1 Логічна схема локальної БД автоматизованої прохідної.....	16
1.4.2 Опис таблиць локальної бази даних автоматизованої прохідної..	17
1.4.3 Створення локальної бази даних автоматизованої прохідної.....	18
1.5 Розробка програми моніторингу.....	19
1.5.1 Модуль для входу в систему.....	20
1.5.2 Модуль даних.....	21
1.5.3 Модуль зв'язку з USB -портом.....	21
1.5.4 Головний модуль.....	23
1.6 Розробка програми віддаленого доступу.....	27
1.6.1 Модуль для входу в систему.....	30
1.6.2 Модуль для зв'язку з базою даних.....	30
1.6.3 Модуль для мережевих підключень.....	31
1.6.4 Модуль для генерації звітів і їх експорту.....	32
1.6.5 Модуль для роботи з базою даних.....	36
1.6.6 Головний модуль.....	40
1.7 Конструювання та опис інтерфейсу користувачів.....	42
1.7.1 Інтерфейс програми "Monitor".....	42

Зм	Арх	Не доступ	Гідник	Дата

РП 05. 24 000. 00 ДП ПЗ

Арх.

5

ВСТУП

Захист будь-якого об'єкта включає декілька рівнів, число яких залежить від рівня режимності об'єкта. При цьому у всіх випадках важливим рівнем буде система контролю і управління доступом на об'єкт (СКУД). Система контролю доступу представляє собою сукупність програмно-технічних засобів та організаційно-методичних заходів, за допомогою яких вирішуються задачі контролю та управління відвідуванням окремих приміщень, а також задачі оперативного контролю персоналу і часу його перебування на території об'єкта.

Рішення про створення програмно-апаратного комплексу системи контролю доступу та обліку роботи співробітників підприємства можуть прийматися при необхідності обслуговування великого потоку людей, що проходять через контрольно-пропускний пункт підприємства. Крім того, акцент часто ставиться також і на те, щоб на основі зібраних статистичних даних про переміщення співробітників через прохідку, була можливість отримання декількох видів звітів, наприклад таких, як звіт про проведений робочий час співробітником на території підприємства. Наступним кроком у розвитку СКУД є створення цілої мережі автоматизованих прохідок. У такому випадку треба розробити програмне забезпечення для сеансів віддаленої мережевої взаємодії з кожною прохідкою окремо з метою отримання звітів по зібраним статистичним даним та адмініструванню бази даних співробітників підприємства, розташованої на ній. Така програма повинна встановлюватися на головний комп'ютер, розташований, наприклад, в бухгалтерії або у відділі кадрів, з якого буде здійснюватися доступ до всіх прохідок підприємства.

Таким чином, для реалізації програмного забезпечення у даному дипломному проекті ставиться завдання створити програмне забезпечення:

- для моніторингу, яке встановлюється на автоматизовану прохідку, що контролює доступ на територію підприємства;
- для віддаленого доступу, яке встановлюється на головний комп'ютер і по'язується з комп'ютерами автоматизованих прохідних систем.

Зм	Дот	№ сторінки	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Ас
7

1 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

1.1 Склад системи контролю і управління лініям доступом на об'єкт

Система контролю і управління доступом на об'єкт (СКУД) зазвичай складається з серверів СКУД (в залежності від навантаження та розгалуженості контрольованої мережі в цій ролі може бути як застарілий ноутбук так і найсучасніший, найпотужніший кластер серверів) і підключених до них контролерів СКУД. Контролер (контрольна панель) – це спеціалізований високонадійний комп'ютер, у якому зберігається інформація про конфігурацію, режими роботи системи, список людей, які мають право доступу до ресурсу, а також їх привілеї доступу до цього ресурсу. У простих випадках мінімальної варіант контролера може бути вбудованим у зчитувач, турнікет, замок або інший виконавчий пристрій (рис. 1.1).

Блокуючі пристрої (БлП) – це пристрої, що забезпечують фізичну перешкоду доступу і обладнані виконавчими пристроями для керування їх станом (турнікети, прохідні кабінки, двері і ворота, обладнані виконавчими пристроями СКУД).

Зчитувальний пристрій, зчитувач – це пристрій, призначений для зчитування (введення) ідентифікаційних ознак. Цю інформацію він передає контролеру, який і приймає рішення про допуск людини до ресурсу. Можна налаштувати контролер так, що він буде запитувати підтвердження прийнятого рішення у комп'ютера. Для підвищення надійності ідентифікації крім зчитувачів до контролера може підключатися клавіатура для набору персонального ідентифікаційного номера (ПІН-коду).

Ще одним важливим поняттям СКУД є ідентифікатор користувача – унікальна ознака суб'єкта або об'єкта доступу. В якості ідентифікатора може використовуватися код, біометрична ознака, або речовинний код. Ідентифікатор, що використовує речовинний код – предмет, в який (на який) за допомогою спеціальної технології занесена ідентифікаційна ознака у вигляді кодової інформації (картки, електронні ключі, брелоки та ін. пристрої).

					РП 05.24.003.00 ДП ПЗ	Лист
						3
Зм	Дрз	№ версії	Підпис	Дата		

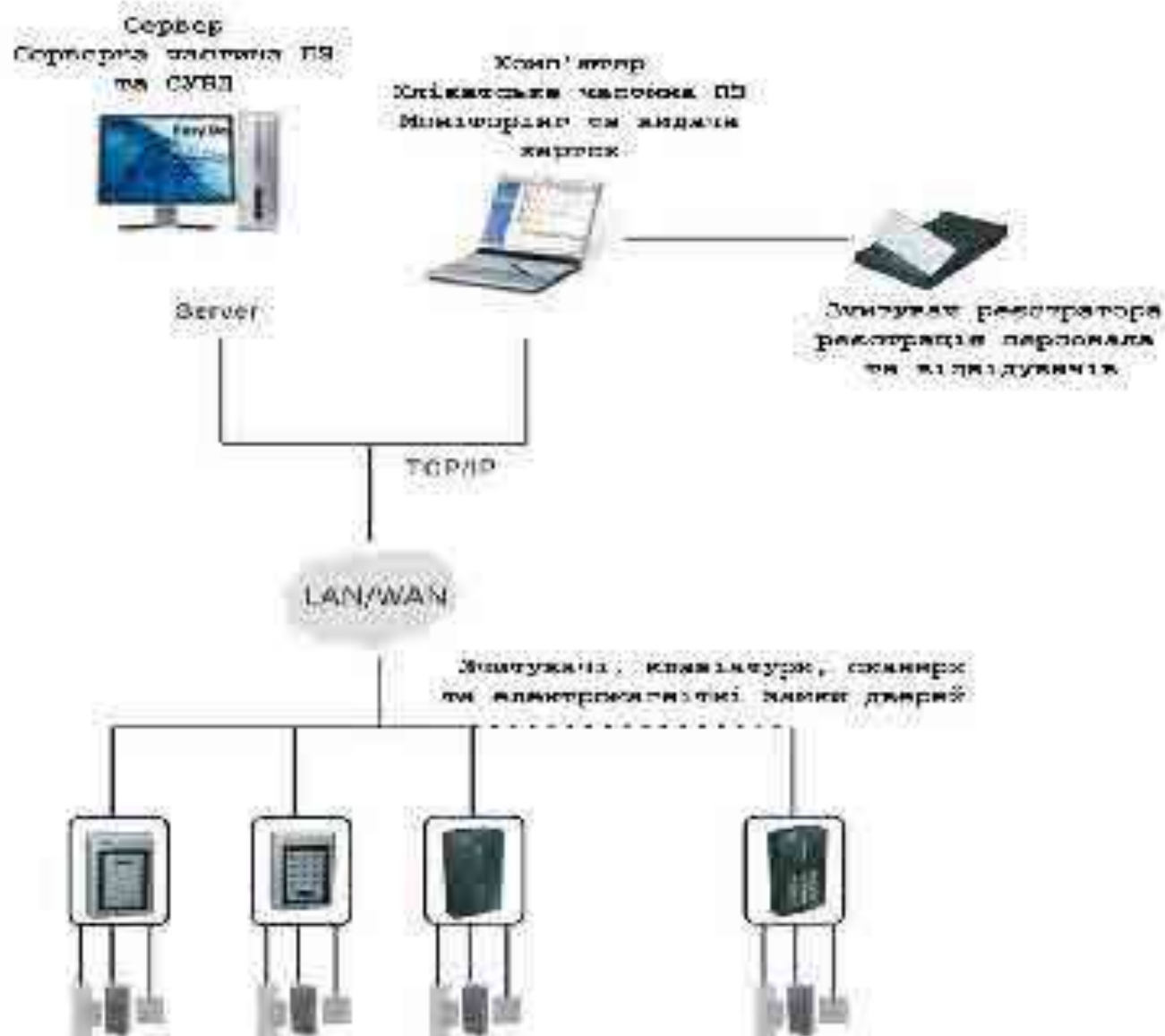


Рисунок 1.1. Склад СК УД на об'єкт

Інший тип пристроїв, які можна підключити до контролера – це охоронні панелі. Це також спеціалізований контролер, який відстежує стан охоронних датчиків (датчики на дверях, вікнах, об'ємні датчики та інші). Якщо стан будь-якого датчика змінюється, то інформація про це тут же надходить в основний контролер [3].

Виконавчі пристрої – це пристрої або механізми, що забезпечують приведення у відкритий або закритий стан БлП (електромеханічні, електромагнітні замки, електромагнітні засувки, механізми приводу шлюзів, воріт, турніжів і інші подібні пристрої). В них може бути набір реле, за допомогою яких вони здійснюють управління виконавчими пристроями.

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

електромеханічними замками, турнікетами, ліфтами, автоматичними воротами та іншими (рис. 1.2).



Рисунок 1.2. Різноманітність засобів обмеження доступу

1.2 Проектування автоматизованої прохідної на об'єкті

Перш ніж приступити до розробки програмного забезпечення системи потрібно спроектувати загальний вигляд комплексу СК УД на об'єкті, адже деякі параметри програмного забезпечення доведеться налаштувати відповідно до розташування апаратних частин системи.

Проектування складатиметься з двох етапів:

- проектування автоматизованої прохідної;
- побудова мережі автоматизованих прохідних.

1.2.1 Апаратне забезпечення автоматизованої прохідної на об'єкті

До апаратного забезпечення, яким буде обладнуватися комплекс автоматизованої прохідної, належать такі пристрої:

- спеціалізований комп'ютер;
- контролер віддалених пристроїв;

Зм	Арт	№ версії	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

- ручний пулт управління контролером;
- безконтактні зчитувачі пластикових карт;
- електромеханічний турнікет.

Спеціалізований desktop-комп'ютер (або ноутбук) служить для зберігання інформації та управління віддаленими пристроями через контролер, зв'язок з яким реалізований через стандартний USB-порт. Контролер віддалених пристроїв забезпечує зв'язок спеціалізованого комп'ютера з безконтактними зчитувачами, електромеханічним турнікетом і ручним пултом управління.

Ручний пулт управління контролером дозволяє управляти електромеханічним турнікетом через контролер у ручному режимі. Безконтактні зчитувачі пластикових карт зчитують унікальний PIN-код з карт за допомогою радіосигналу. Електромеханічний турнікет реалізує механізм пропуску або блокування проходження людей. Загальний вигляд автоматизованого комплексу, що встановлюється на контрольно-пропускній пункті підприємства, наведено на рис. 1.3:



Рисунок 1.3. Загальний вигляд автоматизованої прохідної на об'єкті

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

1.2.2 Робота комплексу автоматизованої прохідної на об'єкті

При піднесенні безконтактної картки (Proximity-картки) до считувача унікальний PIN-код картки за допомогою радіосигналу передається на контролер. Програмне забезпечення, встановлене на комп'ютері прохідної, через USB-порт періодично опитує поточний стан контролера і як тільки отримує у відповідь символ готовності для передачі зчитаного коду, надсилає запит на його отримання. Отриманий код картки треба шукати у локальній базі даних співробітників, а після знаходження визначити права доступу для знайденого співробітника і внести рішення про відкриття або блокування турнікета, для чого через USB-порт на контролер надсилається відповідна команда. Після цього в журнали прохідної записується подія, що відбулася. Журнали прохідної повинні мати такий вигляд, щоб по записам в них даних можна було генерувати чотири основні типи звітів. Вони є такими:

1. Звіт про прохідну – надає відомості про всі входи і виходи, а також невдалі спроби входу співробітників, які відзначалися на прохідній за певний період;
2. Звіт робочого часу співробітників – отримання сумарного робочого часу, проведеного співробітниками на робочому місці за певний період;
3. Звіт про відсутніх – список співробітників, які були відсутні в певний інтервал часу на робочому місці;
4. Звіт про тих, хто запізнився – список осіб, які мають запізнення на робоче місце за певний інтервал часу.

1.3 Проектування мережі автоматизованих прохідних на об'єкті

При наявності двох і більше автоматизованих прохідних на одному підприємстві, з'являється проблема збору їх накопичених статистичних даних. Якщо підключити комп'ютери прохідних до комп'ютерної мережі підприємства, то за допомогою програми віддаленого доступу можна було б приєднуватися до локальних баз даних прохідних і переключувати інформацію з них на головний комп'ютер, де вже її обробляти і роздруковувати.

					РП 05.24.003.00 ДП ПЗ	Арт.
Зм	Дат	№ докум.	Підпис	Дата		12

Як випливає з [7], в даний час більшість комп'ютерних мереж підприємств будується на основі мережі Windows Network, в якій використовується кілька основних типів протоколів: TCP/IP, NetBEUI, IPX/SPX.

1.3.1 Мережеві ресурси Windows Network

В операційних системах сімейства Windows є можливість призначати доступ до ресурсів комп'ютера. Завдяки цій можливості можна через мережу Windows Network користуватися ресурсами інших комп'ютерів, як, наприклад, мати доступ до їх файлів на жорстких дисках. Для реалізації проекту нас буде цікавити, перш за все, доступ до файлів. Щоб отримати доступ до певних ресурсів комп'ютера необхідно оголосити ці ресурси доступними для користувачів мережі. У нашому випадку, щоб мати мережевий доступ до локальної бази даних автоматизованої прохідної, необхідно оголосити її доступним ресурсом в мережі. Для цього можна скористатися програмовою експлоатером, в якій вказати базу даних – каталог з файлами-таблицями Рагадок, і, натиснувши на цьому каталозі праву кнопку миші, вибрати пункт "доступ".

1.3.2 Обмеження доступу до мережевих ресурсів

Доступ до мережевих ресурсів необхідно чимось обмежувати, щоб уникнути втрати і навіть пошування інформації. Іншими словами, необхідно захистити оголошену як доступний мережевий ресурс базу даних від проникнення в неї сторонніх осіб. Це можна виконати двома типами обмеження доступу: на рівні ресурсів, задавши паролі на читання і на повний доступ та на рівні користувачів, перераховувачи їх імена і групи. На рівні ресурсів при спробі підключення до ресурсу у користувача запитується пароль і в разі, якщо пароль невірний, в доступі відмовляється. На рівні користувачів на кожен ресурс зберігається свій список привілейованих користувачів мережі Windows Network, яким дозволяється доступ до нього. Для більш детального ознайомлення з призначенням доступу до ресурсів мережі можна скористатися [8].

Зм	Дот	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

1.3.3 Схема мережі автоматизованих прохідних на об'єкті

Загальний вигляд мережі автоматизованих прохідних підприємства наведений на рис. 1.4.

Локальна база даних кожної автоматизованої прохідної є доступним ресурсом в мережі Windows Network для головного комп'ютера, на якому встановлена програма віддаленого доступу. За допомогою програми віддаленого доступу буде виконуватися під'єднання до кожної бази, а потім отримання її накопичених статистичних даних.

Також програма віддаленого доступу буде використовуватися для редагування інформації в локальних базах даних, наприклад додавання, зміни, видалення інформації про конкретну організацію, відділ, співробітника. Приєднаний до головного комп'ютера контролер разом з безконтактним зчитувачем дозволить отримувати код карти і записувати його у під'єднану базу даних для зазначеного співробітника.

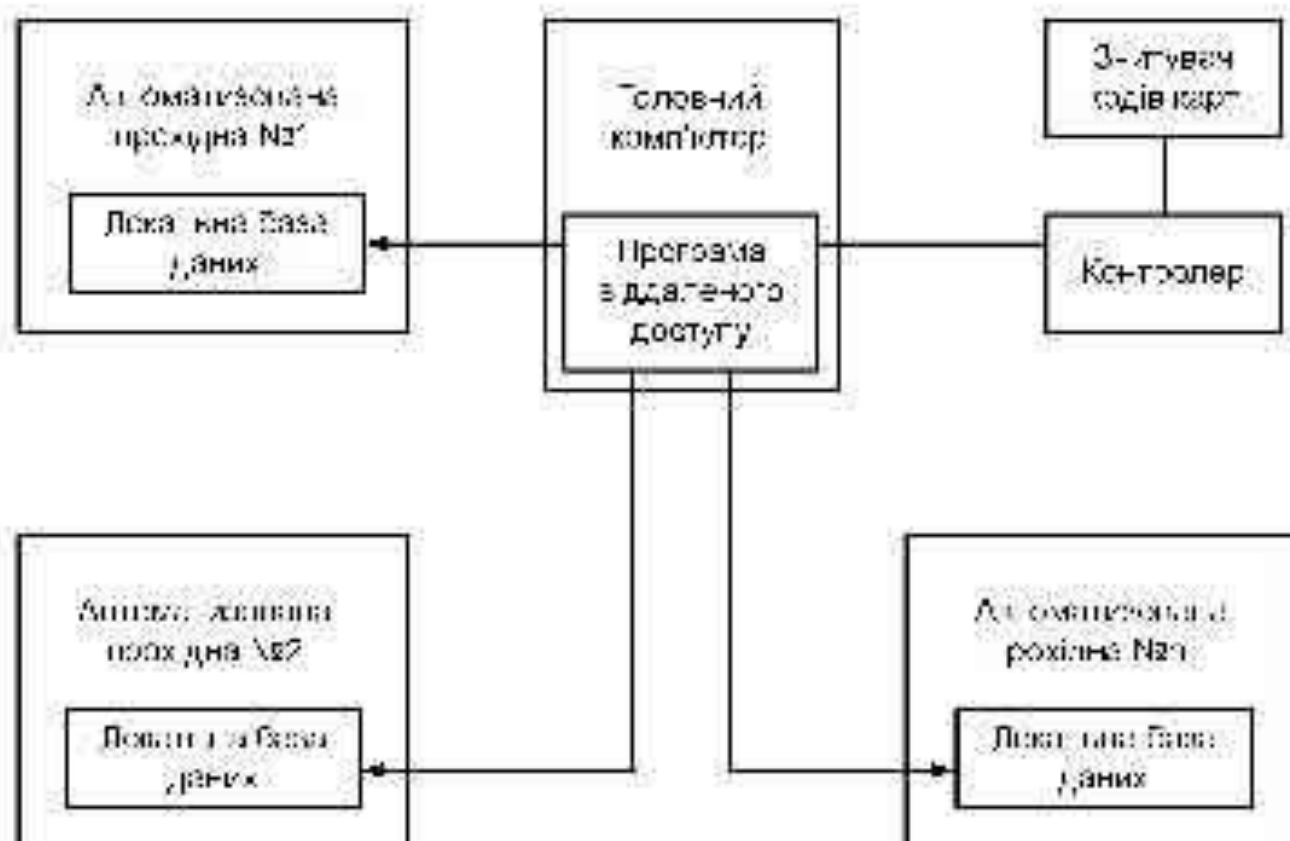


Рисунок 1.4. Схема мережі автоматизованих прохідних

Зм	Авт	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Для побудови мережі автоматизованих прохідних (контрольно-пропускних пунктів – КПП), на комп'ютер кожного КПП буде встановлена програма «Monitor» (що включає в себе спочатку порожню базу даних), крім того, комп'ютер повинен бути підключений до комп'ютерної мережі Windows Network цього підприємства. Це потрібно для того, щоб можна було мати віддалений доступ з головного комп'ютера до локально збереженої бази даних комп'ютера КПП.

Встановлення доступу до бази даних здійснюється після інсталяції програми «Monitor». Каталог з файлами бази даних "Base" буде розташовуватися в тому ж каталозі, де буде розміщена сама програма «Monitor».

Для встановлення доступу з головного комп'ютера до бази даних (до каталогу "Base"), що зберігається на комп'ютері кожного КПП, слід оголосити її загальнодоступним ресурсом мережі Windows Network. Після цього слід встановити пароль на цей ресурс, щоб запобігти небажаному підключенню до бази сторонніми особами.

Програмне забезпечення, яке необхідно встановити на комп'ютер кожної з автоматизованих прохідних, назвемо програмю моніторингу "Monitor". Моніторинг являє собою незалежну програму, це означає, що вона функціонує окремо від інших частин комплексу.

Програмне забезпечення, яке встановлюється на головний комп'ютер, назвемо програмю віддаленого доступу до автоматизованих прохідних "Manager". Віддалений доступ буде можливий тільки в тому випадку, якщо комп'ютер віддаленої автоматизованої прохідної включений і функціонує.

1.4 Розробка бази даних автоматизованої прохідної

Загальна база даних підприємства буде розподілена по локальних базах даних автоматизованих прохідних, це є необхідним для найбільш швидкого доступу до неї. У базі даних кожної автоматизованої прохідної будуть зберігатися відомості про тих співробітників, яких вона обслуговує. Крім того, для взаємодії програмного забезпечення з локальною базою даних на комп'ютері

					РП 05.24.003.00 ДП ПЗ	Акт
Зм	Дот	№ докум.	Підпис	Дата		15

Зв'язки між таблицями встановлені таким способом, щоб завжди по номеру потрібного нам співробітника можна було отримати певну інформацію, що стосується його.

1.4.2 Опис таблиць локальної бази даних автоматизованої прохідної

Перерахуємо назви всіх таблиць бази даних і опишемо, для чого кожна з них призначена:

- "Organization.db" - зберігає список організацій підприємства. Містить поля: org - номер (індекс) організації, org_desc - назва організації;
- "Departments.db" - зберігає список відділів організацій підприємства, для чого має зв'язок з таблицею "Organization.db". Містить поля: org - номер організації в таблиці "Departments.db", dep - номер (індекс) відділу, dep_desc - назва відділу;
- "Empbyers.db" - таблиця співробітників підприємства, для кожного з яких вказано відділ і організація, жими він належить. Поля таблиці: org, dep - номер організації, відділу для співробітника, empkey - номер співробітника в таблиці, tabname - займана посада, Fname, Lname, Mname - прізвище, ім'я, по батькові, keycode - код картки, photo - фотографія, startaccess, endaccess - дати початку і кінця доступу по картці, blocking - блокування доступу, addinfo - додаткова текстова інформація;
- "Rights.db" і "WorkRange.db" - необхідні відповідно для зберігання прав доступу співробітників на територію підприємства і інтервалів робочого часу, протягом яких співробітники зобов'язані перебувати на робочому місці. Поля: org, dep, empkey - номери організації, відділу, співробітника, для яких встановлюється часовий інтервал: t_from, t_to;
- "Simplog.db" - журнал для фіксування подій (входи / виходи) і їх можливих порушень. Поля: empkey - номер співробітника в таблиці "Empbyers.db", today, time - дата і час, event - подія, alert - порушення;

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Лист

17

- "EmpLog.db" - журнал для запису проведених інтервалів робочого часу. Поля empkey - номер співробітника в таблиці "Employers.db", today, time - дата і час, accesstime - інтервал проведеного робочого часу;
- "LateComers.db" - журнал для фіксування запізень співробітників, відно відно. Поля: empkey - номер співробітника в таблиці "Employers.db", today, time - дата і час, mustcome - час необхідного приходу співробітника, latimg - запізнення;
- "Systemt.db" - зберігає закодовані паролі для входу в систему. Поля: loginname - ім'я користувача, password - пароль.

1.4.3 Створення локальної бази даних автоматизованої прохідної

Створити описану базу даних можна за допомогою утиліти DataBase DeskTop (DBD), що входить в стандартну поставку Embarcadero RAD Studio Delphi. Опис для роботи з утилітою можна знайти в [5].

Після створення необхідно зареєструвати аліас бази даних в файлі конфігурації BDE "Idaricfg", що полегшить в подальшому розробку програми. Для цього скористаємося утилітою BDE Administrator. У розділі "Object" виберемо пункт "New", вкажемо у вікні тип драйверу "STANDARD" для роботи з базовою базою даних і в розділі визначення параметрів нового аліасу (розділ "Definition"), навпроти рядка "Path" вкажемо повний шлях до місця положення нашої бази даних на жорсткому диску, а потім назвемо створений аліас "Base".

1.4.4 Оголошення бази даних доступним мережевим ресурсом

Після створення локальну базу даних необхідно оголошити доступним ресурсом в мережі Windows Network і обмежити доступ до неї. Це необхідно для того, щоб за допомогою програми віддаленого доступу мати можливість приєднуватися до неї.

Скористаємося програмою explorer.exe, у якій вкажемо шлях до створеної бази даних (каталогу з файлами Paradox), натиснемо праву кнопку миші на ньому, виберемо пункт "доступ". У вікні встановимо опцію "Загальний ресурс",

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

в поле "Мережеве ім'я" призначимо ресурсу ім'я, під яким він буде видний в мережі (наприклад "Base"), далі визначимо тип доступу як "повний" і в полі "пароль для повного доступу" встановимо пароль.

Тепер для мережевого під'єднання до локальної бази даних необхідно буде знати пароль, який обмежить доступ до неї.

1.5 Розробка програми моніторингу

Програма моніторингу буде включати в себе наступні модулі:

- Модуль для входу в систему;
- Модуль даних для зв'язку з базою даних;
- Модуль для реалізації зв'язку з USB-портом;
- Головний модуль, до якого будуть підключатися інші.

Нижче перелічені модулі програми "Monitor" і список основних процедур в них:

1. USBThread.pas – робота з USB-портом:

constructor USBPortThread.Create – створює екземпляр класу USBPortThread для роботи з USB-портом;

function USBPortThread.getRights(empkey: integer; blocking: smallint): boolean – визначає права доступу співробітника з номером empkey в таблиці "Employers.db";

procedure USB.WrStat(mode: integer; emp: longint) – записує в журналі подій статистику;

function sendcommand(com: string): boolean – відсилає команду контролеру турнікета;

procedure USBPortThread.Open_dir(direction, mode: short) – посилає команду відкриття турнікета в потрібному напрямку;

procedure USBPortThread.Execute – створює новий процес, що запускає екземпляр класу USBPortThread на виконання;

procedure LRClose; LOpen(ROpen) – б'лює турнікет;

function LOpen: boolean – відкриває турнікет зліво;

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

function ROpen: boolean – відкриває турнікет вправо;

function WrongReadCardCRC(s: string): boolean – підраховує контрольну суму зчитаного коду карти;

2. Configurator.pas – налаштування програми:

procedure TConfigForm.FormShow(Sender: TObject) – відображає на формі налаштування програми;

procedure TConfigForm.FormClose(Sender: TObject; var Action: TCloseAction) – кнопка змінени налаштування в системний реєстр;

3. ConpMain.pas – головний модуль програми:

procedure TMainForm.FormCreate(Sender: TObject) – створює форму головного модуля, завантажує з реєстру налаштування програми;

procedure TMainForm.USBPort1WaitChar(CP: TObject; C: Char) – читання з USB-порту чергового символу відно віді контролера;

procedure TMainForm.Button1Click(Sender: TObject) – кнопка увімкнення / вимкнення процесу взаємодії з USB-портом;

procedure TMainForm.RxClock1Alarm(Sender: TObject) – подія автовидалення застарілих даних з журналі прохідної, користуючись програмними налаштуваннями;

4. Passwords.pas – вхід до системи:

procedure TPasswordDlg.OKButtonClick(Sender: TObject) – обробка введеного імені користувача та його паролю для входу до програми.

1.5.1 Модуль для входу в систему

Головна функція роботи цього модуля полягає в тому, щоб запобігти несанкціонованому доступу третіх осіб до роботи в програмі. На вході в систему буде запитуватися роль користувача в системі і пароль до неї. Визначено дві ролі: "Admin" і "Operator", вони відрізняються тим, що користувач з роллю "Admin" має право доступу і зміни програмних налаштувань, в той час, як "Operator" може лише спостерігати коректність роботи програми і вживати необхідні заходи у екстрених випадках (тобто виконує функції охоронця).

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Паролі будуть зберігатися в Paradox файлі-таблиці "SystemT.db" в закодованому вигляді, причому для кодування паролів використовується компонента TBlowFish. Нижче наведено фрагмент програми кодування паролів:

```
LoginName := 'ADMIN';  
BF1.Key := 'password';  
StrPCopy(Buffer, LoginName);  
BF1.EncryptBlock(buffer, length(LoginName));
```

Вигляд вікна із запитом ролі користувача і пароля наведений на рис. 1.7.

1.5.2 Модуль даних

Для зв'язку програми з базою даних через BDE в середовищі розробки додатків Embarcadero RAD Studio Delphi передбачений спеціальний модуль даних TDataModule, форму якого можна створити в меню File, пункт New DataModule.

Форма модуля даних спеціально призначена для розташування на ній компонент Embarcadero RAD Studio Delphi для роботи з базами даних [9]. Перерахуємо потрібні для програми компоненти:

RightsQuery:TQuery – служить для генерування запитів до бази даних з метою визначення прав доступу співробітника за кодом його картки;

AutoDelQuery:Tquery – використовується з метою автоматичного Автовидалення застарілих даних з журналів прохідної;

StatQuery:Tquery – для занесення записів подій в журнали прохідної.

Щоб прив'язати ці компоненти конкретно до нашої створеної локальної бази даних, необхідно в їх властивостях напроти напису "DataBaseName" написати ім'я зареєстрованого раніше апарату "Base".

1.5.3 Модуль зв'язку з USB-порт

Для керування програмов турнікетом, через контролер турнікета потрібно реалізувати зв'язок "комп'ютер-контролер" через стандартний USB-порт. Для цього можна скористатися компонентом RAD Studio TApdUSBPort [10].

Зм	Дрз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Арт

21

Розмістимо компоненту на головної формі проекту в RAD Studio Delphi. Компонента має перелік властивостей, які необхідно правильно налаштувати:

USBNumber – номер USB-порту комп'ютера, до якого буде підключатися контролер. Встановимо значення цієї властивості у "2". Інші налаштування можна залишити так, як вони встановилися за замовчуванням.

У самій програмі для встановлення зв'язку комп'ютера з контролером через USB-порт створимо новий процес, описуваний класом TThread, який за допомогою компоненти TArpUSBPort прив'язується до вказаного номеру USB-порту і посилає контролеру команди керування, одночасно читаючи відповіді, що надходять.

Таблиця 1.1. Список команд керування контролером

Get_st2	Команда, після якої контролер посилає у відповідь статус-байт про свій поточний стан
Id1 (Id2)	Отримати зчитаний код із зчитувача N1 (або N2)
Llamp_on(Llamp_off)	Запалити (погасити) ліву стрілку на турнікеті
Rlamp_on(Rlamp_off)	Запалити (погасити) праву стрілку на турнікеті
Slamp_on(Slamp_off)	Запалити (погасити) хрестик на турнікеті
Lsol_on(Lsol_off)	Відкрити (закрити) турнікет для проходу вліво
Rsol_on(Rsol_off)	Відкрити (закрити) турнікет для проходу вправо

У таблиці 1.1 наведено список команд керування, що розуміються контролером. Для зв'язку з контролером турнікету створимо клас USBPortThread, що має своїм предком клас TThread, в який були включені наступні процедури і функції:

constructor Create (Query:TQuery) – створення екземпляра класу з передачею йому у якості параметру компонента TQuery для реалізації необхідних запитів до бази даних;

процедура Execute; override; – для запуску і синхронізації з основним потоком;

процедура MainCycle; – тіло нового процесу, в якому були написані виконуваний дії;

function GetRights (EmpKey: integer):boolean; – функція визначення прав доступу для співробітника з ключем EmpKey, що повертає у якості результату роботи "true" – якщо доступ дозволений і "false" – якщо у доступі відмовлено;

procedure Open_tun(direction:short); – відкриття турнікета в ту чи іншу сторону, в залежності від параметру "direction";

procedure WriteLog(mode:integer; emp:longint); – запис в журнал реєстрації події, що сталася.

Блок-схема роботи алгоритму у модулі може бути представлена у вигляді, показаному на рис. 1.6.

Детальніше зупинимося на кроці алгоритму «Запис події в журнал». У таблицю "Simpllog.db" записується дата, час, дані про співробітника і тип події: "увійшов" або "вийшов", щоб надалі по цій таблиці можна було отримати перший тип звіту ("Прохідна").

У таблицю "EmpLog.db" в разі "входу" співробітника буде занесено запис у вигляді: дата, час, співробітник. У разі "виходу" спочатку знайдеться найостанній запис "входу" для цього співробітника, потім відбудеться підрахунок різниці часу "виходу" і "входу" з урахуванням накладення маски інтервалів робочого часу (наприклад, обідній час не береться до уваги) з таблиці "WorkRange.db" і остаточно оновиться знайдений запис, прийнявши вигляд: дата, час, співробітник, проведений робочий час.

При записі у таблицю "LateComers.db" в разі "виходу" співробітника спочатку визначиться, чи входить він вже сьогодні. При негативній відповіді час "виходу" співробітника порівнюється з часом приходу на роботу з таблиці "WorkRange.db" і при необхідності записується запізнення у вигляді: дата, час, співробітник, має прийти, запізнення.

1.5.4 Головний модуль

Головний модуль буде включати в себе початковий завантажувач програми, основні елементи інтерфейсу для зв'язку з користувачем і зв'язок з іншими модулями.

					РП 05.24.003.00 ДП ПЗ	Акт
Зм	Дот	№ докум.	Підпис	Дата		23

При запуску програми спочатку буде запускатися на виконання модуль входу в систему (в разі трьох невдалих спроб входу програма завершує свою роботу), який можна прив'язати, наприклад, на подвійне створення і форми:

```
procedure TMainForm.FormCreate (Sender: TObject);  
begin  
if not (PasswordDialog.ShowModal=mrOK) then MainForm.Close;
```

Далі за допомогою компоненти Regl:TLMDIniCtrl [Registry] з системного реєстру зчитуються програмні налаштування:

```
// Номер USB-порту  
USB_PORT:= Regl.ReadInteger (IDS_ROOT,'COM_PORT',0);  
// Швидкість обміну по швидкості  
COM_SPEED :=Regl.ReadInteger (IDS_ROOT,'COM_SPEED',0);  
// Автовидалення застарілих записів (узімк./взімк.)  
AUTODEL:=Regl.ReadInteger (IDS_ROOT,'AUTODEL',0);  
// Період старіння записів за місяцями  
AUTODELPERIOD:=Regl.ReadInteger (IDS_ROOT,'AUTODELPERIOD',0);
```

Потім перевіряється, чи підключений контролер до потрібного USB-порту, в разі позитивного результату створюється і запускається на виконання новий процес класу TArpUSBPort, що працює паралельно з основною програмою, інакше видається попередження про неможливість роботи з USB-портом.

```
try  
if USB1 = nil then  
begin  
USB1 := USBPortThread.Create; // Створення екземпляра класу  
USB1.Resume;  
end;  
except  
ShowMessage ('USB-порт для турнікета зайнятий іншим пристроєм!');  
end;
```

					РП 05.24.003.00 ДП ПЗ	Акт
Зм	Дат	№ докум.	Підпис	Дата		25

У разі вдалого розпізнавання введеного імені користувача і пароля з'являється форма головного модуля, як показано на рис. 1.8.

Після появи форми робота головного модуля полягає в обробці чотирьох типів подій:

1) Вибір типу керування турнікетом: автоматичне або ручне. Подія настає при натисканні на кнопку з написом «Увімкнути» (або «Вимкнути»). У разі активізації ручного керування, відкриття або блокування механізму турнікета здійснюється кнопками ручного пульта.

```
procedure TMainForm.Button1Click(Sender: TObject);
begin
  if (Button1.Caption = 'Увімкнути') then
  begin
    USB1.Resume; // Управління ведеться з процесу USB1
    Button1.Caption := 'Вимкнути';
  end
  else
  begin
    USB1.Suspend; // Управління бере на себе контролер
    Button1.Caption := 'Увімкнути';
  end;
end;
```

2) Поміняти програмні налаштування. За допомогою компонента TLMD InCtrl записуємо у реєстр нові налаштування

```
// Номер USB-порту
Reg1.WriteInteger(IDS_ROOT, USB_PORT, USB_PORT);
// Швидкість обміну по порту
Reg1.WriteInteger(IDS_ROOT, 'USB_SPEED1', USB_SPEED);
// Автовидалення застарілих записів (увімк. / вимк.)
Reg1.WriteInteger(IDS_ROOT, 'AUTODEL', AUTODEL);
```

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

```
// Період старіння записів за місцями
```

```
Reg1.WriteInteger (IDS_ROOT, 'AUTODELPERIOD', AUTODELPERIOD);
```

3) Автовидалення застарілих записів в журналах прохідної. Подія спрацює кожний день у визначений час (компонента TRxClock [RxClock]) і, керуючись програмними налаштуваннями (Автовидалення) здійснює очищення журналів від старих записів через SQL-запит за допомогою компонента TAutoDelQuery:

```
procedure TMainForm.RxClock1Alarm (Sender: TObject);
begin
  SQL.Clear;
  SQL.Add ('delete from simplog, EmpLog, LateComers where
  TODAY<: TODAY');
  Parambyname ('TODAY').Asdatetime :=Date-AUTODELPERIOD*30;
try
  rgerate;
  Exec SQL;
except
  MessageDlg ('Помилка підключення до БД!', mtError, [mbOk], 0);
  exit;
end;
```

1.6 Розробка програми віддаленого доступу

Програма віддаленого доступу до локальних баз даних, розташованих на комп'ютерах автоматизованих прохідних, в загальному вигляді вимагає наявності наступних ресурсів комп'ютера:

- Встановлений BDE для взаємодії з віддаленою базою даних;
- Підключення до мережі Windows Network підприємства для віддаленого доступу до даних;

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Арт

27

- Підключення до USB-порту локальний контролер для призначення кодів карт співробітникам.

Перерахуємо також модулі, з яких буде складатися програма:

- Модуль авторизації;
- Модуль для зв'язку з базою даних;
- Модуль для мережних підключень;
- Модуль для генерації звітів і їх експорту;
- Модуль для роботи з базою даних;
- Головний модуль, до якого будуть підключатися інші.

Нижче перелічені модулі програми "Manager" і список основних процедур

в них:

1. EmpDialog.pas - заповнення інформації про співробітника в БД:

procedure TEmpEditDlg.FormActivate(Sender: TObject) - активує форму модуля, відкриваючи таблицю "Employers.db" на запис;

procedure TEmpEditDlg.GetFotoBtnClick(Sender: TObject) - дозволяє вибрати фотографію співробітника і внести її у локальний кеш БД;

procedure TEmpEditDlg.BitBtn1Click(Sender: TObject) - вносить іншу інформацію про співробітника у локальний кеш БД;

procedure TEmpEditDlg.FormCloseQuery(Sender: TObject;
var CanClose: Boolean) - закриває форму, вносячи всі зміни з локального кешу БД в саму БД;

2. RightsDlg.pas, WorkRangeDialog.pas - модулі для встановлення прав доступу та інтервалів робочого часу співробітникам підприємства:

procedure AssertIntervalBtnClick(Sender: TObject)

вносить введений інтервал до БД, перевіряючи коректність введення;

procedure FormShow(Sender: TObject) - читає з БД вже задані інтервали часу і відображає їх;

3. KppReport.pas, WorkReport.pas, EpsReport.pas, LateComersReport.pas - описують форми для відображення звітів, містять в собі елементи для їх подання.

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Арт

28

4. `CompanyMain.pas` – головний модуль програми;
- процедура `TMainForm.buildtree` – будує ієрархічну структуру підприємства, читаючи її з БД і відображає її на формі;
- процедура `TMainForm.EditPopUpMenuPopUp(Sender: TObject)` – викликає меню для модифікування інформації в БД;
- процедура `TMainForm.AddClick(Sender: TObject)` – додає в БД організацію, відділ або співробітника;
- процедура `TMainForm.DelClick(Sender: TObject)` – видаляє з БД організацію, відділ або співробітника;
- процедура `TMainForm.EditClick(Sender: TObject)` – редагує в БД організацію, відділ або співробітника;
- процедура `TMainForm.FormCreate(Sender: TObject)` – створює головну форму програми, читаючи налаштування з реєстру;
- процедура `TMainForm.GetReportBtnClick(Sender: TObject)` – генерує один з чотирьох типів звітів;
- процедура `TMainForm.ViewReportBtnClick(Sender: TObject)` – передає отримані звітні дані у відповідну форму для відображення звіту;
- процедура `TMainForm.FormShow(Sender: TObject)` – читає список прохідних і відображає його на екрані;
- процедура `TMainForm.ListView1Db1Click(Sender: TObject)` – виконує підключення до обраної зі списку прохідної;
- процедура `TMainForm.RightsMenuItemClick(Sender: TObject)` – викликає форму для визначення прав доступу;
- процедура `TMainForm.WorkRangeClick(Sender: TObject)` – викликає форму для визначення інтервалів часу роботи співробітників;
- процедура `TMainForm.PrintReportBtnClick(Sender: TObject)` – виконує експорт отриманих звітних даних в EXCEL.

1.6.1 Модуль для входу в систему

Модуль авторизації аналогічний модулю для входу в систему програми моніторингу та виконує таку ж функцію захисту від несанкціонованого доступу до функцій програми.

1.6.2 Модуль для зв'язку з базою даних

Скористаємося все тим же модулем даних TDataModule для створення зв'язку з локальними базами даних автоматизованих прохідних. Розмістимо на ньому компоненти:

- DataBase1:TDatabase – для підключення бази даних;
- EmpTable:TTable – для доступу до записів в таблиці співробітників;
- ReportQuery:TQuery – для отримання зібраних статистичних даних з віддаленої бази даних;
- PasswordQuery1:TQuery – для доступу до таблиці з паролями для авторизації в програмі віддаленого доступу;
- EmpQuery:TQuery – для редагування інформації про співробітників у віддаленій базі даних;
- Query1:TQuery – для тимчасових потреб;

Для компоненти DataBase1 необхідно зареєструвати порожній аліас з ім'ям BossClientBase, параметри якого програмно будуть заповнюватися кожен раз інформацією про ту базу даних, до якої програма буде виконувати підключення. Потім встановимо властивості:

AliasName = BossClientBase; // Назва порожнього аліасу

DatabaseName = ClientBase; // Назва локальної бази даних

Для всіх інших компонент необхідно встановити властивість:

DatabaseName = ClientBase, щоб вказати, що ці компоненти відносяться безпосередньо до роботи з тією базою даних, параметри якої задані у властивостях компоненти DataBase1.

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

1.6.3 Модуль для мережеских підключень

Щоб приєднатися до бази даних, скористаємося компонентом Database1.TDatabase, розміщеним на формі модуля даних. У цього компонента є властивість Params, яка містить список параметрів, переданих BDE при встановленні зв'язку з якою-небудь базою даних. При підключенні задамо наступний параметр:

Path = \\<IP_address>\<Net_Resource_Name>\<DB_Path >, де Path – назва переданого параметра (в даному випадку параметр, що містить шлях до БД), IP_address – чотирьохбайтова адреса комп'ютера у мережі, Net_Resource_Name – назва доступного (Shared) мережевого ресурсу, а DB_Name – шлях до бази даних на доступному мережевому ресурсі, наприклад:

```
"Path=\\198.162.10.40 \D_DISK\DATABASES\DB1"
```

Весь цей ланцюжок носить назву UNC-ім'я і є значенням параметру Path. Таким чином виконувач підключення буде використовувати протокол TCP/IP для встановлення зв'язку і передачі даних.

У разі, якщо потрібно підключитися до локальної бази даних, параметр буде мати спрощений вигляд: Path = <DB_Path>, де DB_Path містить повний шлях до бази даних на локальному або підключеному мережевому диску, наприклад, "Path=C:\DATABASES\DB1".

Програмно всі описані дії можна реалізувати так:

```
Database1.Connected=false; \ Від'єднуємо БД
```

```
Database1.Params.Clear; \ Очищуємо список параметрів
```

```
Database1.Params.Add (PATH ='+DBName); \ Додаємо
```

```
де DBName – рядок, що містить UNC-ім'я.
```

Але, треба не забувати, що для доступу до бази даних, що підключається, як до мережного ресурсу необхідно з'явиться чинком передавати пароль при підключенні, інакше в доступі буде відмовлено (тому що автоматично пароль запитуватися не буде). Для цього скористаємося засобами WinAPI [1]. Спеціально призначена функція WNetAddConnection3() дозволить програмістам

Зм	Дрз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Асв

31

шляхом встановлювати з'єднання з ресурсами мережі, передаючи у якості одного з параметрів пароль до мережного ресурсу.

Дії для під'єднання мережевого ресурсу, такі:

```
NR.dwType:=RESOURCE_TYPE_DISK; // Тип ресурсу - диск
```

```
NR.lpLocalName:=nil; // Ім'я локального диску
```

```
NR.lpRemoteName:=PChar(DBName); // UNC-ім'я
```

```
NR.lpProvider:=nil; // Провайдер за замовчуванням
```

```
WNetAddConnection3(0, NR.Password, UserName,
```

```
CONNECT_UPDATE_PROFILE);
```

де NR – структура типу TNetResourceA, DBName – UNC-ім'я ресурсу, що під'єднується, Password – рядок з паролем, UserName – рядок з ім'ям користувача (у разі визначення доступу до ресурсів на рівні користувачів), а CONNECT_UPDATE_PROFILE – константа, оновлює профіль поточного користувача в системі шляхом додавання UNC-імені в список мережних ресурсів, що підключаються. Далі необхідно встановити властивість Database1.Connected=true для виконання під'єднання до віддаленої бази даних.

1.6.4 Модуль для генерації звітів і їх експорту

Після того, як віддалена база даних буде успішно приєднана, можна почати роботу зі збору накопичених статистичних даних і генерації звітів по ним.

Скористаємося компонентом ReportQuery:TQuery, що дозволяє за допомогою SQL-запитів одержувати записи з бази даних, що задовольняють параметрам цього запиту. Компонента ReportQuery має властивість SQL, яка у текстовому вигляді містить SQL-запит. Для генерації чотирьох типів звітів запити будуть мати вигляд:

1. Звіт про прохідну. Потрібно вибрати всіх співробітників, які відзначалися на прохідній за певний період часу:

```
ReportQuery.SQL.Clear; // очистимо запит
```

```
ReportQuery.SQL.Add ('select today, entertime, event, lname, org_desc,
```

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Арт

32

```
dep_desc from simplog.db, employers.db, organizations.db, departments.db  
where simplog.today>=datestart and simplog.today<=:dateend');
```

```
ReportQuery.Open; // Зробити запит
```

де datestart і dateend визначають період звіту і описуються типом Tdate Time.

2. Звіт про робочий час. Необхідно підраховати сумарний робочий час для конкретного співробітника, проведений на робочому місці за певний період часу, або для всіх співробітників відділу, організації або підприємства в цілому. Одним SQL-запитом цього зробити не можна, тому звіт будемо отримувати в кілька етапів.

Спочатку отримаємо список співробітників для звіту:

```
ReportQuery.SQL.Clear;
```

```
case Type of
```

```
// Отримати список всіх співробітників
```

```
-1: SQL.Add ('select *from Employers.db');
```

```
// Отримати всіх співробітників обраної організації
```

```
0: SQL.Add ('select *from Employers.db and Org =' + Org_desc);
```

```
// Отримати всіх співробітників обраного відділу
```

```
1: SQL.Add ('select *from Employers.db where Dep =' + Dep_desc);
```

```
// Отримати одного співробітника
```

```
2: SQL.Add ('select *from Employers.db where Lname =' + LName);
```

```
end;
```

```
ReportQuery.open;
```

де змінна Type – підтип звіту, Org_desc, Dep_desc – рядки, що містять назву організації або відділу, LName – прізвище конкретного співробітника.

Далі, перебираючи отриманих співробітників, будемо підраховувати для кожного з них інтервали робочого часу з таблиці "Emp Log.db".

```
ReportQuery.First; // Встановимо вказівник на перший знайдений запис
```

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Лист

33

```

while not ReportQuery.EOF do //перебираємо співробітників
begin
  TempQuery.SQL.Clear;
  TempQuery.SQL.Add (' select AccessTime from Emplog.db where
  Empkey= Empkey and today>= Date Start and today<= :Dateend and
  accesstime is not null');
  TempQuery.ParamByName ('EmpKey').AsInteger :=ReportQuery.FieldName
  ('EmpKey'). AsInteger;
  TempQuery.Open;
  // якщо знайшли інтервали робочого часу
  if TempQuery.Recordcount>0 then
  begin
    TempQuery.First; // Беремо перший
    While not TempQuery.EOF do // Підсумовуємо з іншими
      begin
        de codetime (fieldname ('accesstime'). asdatetime ,hour,min, sec ,msec);
        hours:=hours+hour;
        mins:=mins+min;
        if mins>=60 then
          begin hours:=hours+(mins div 60); mins:=mins mod 60; end;
        TempQuery.Next;
      end;
    end;
  end;
  ReportQuery.Next; // Беремо наступного співробітника
end;

```

Таким чином можна підраховати повний робочий час за певний період.

3. Звіт про відсутніх. По суті, цей звіт вимагає перевірки, чи був зареєстрований співробітник в журналі проходної в певний період. SQL-запит буде таким:

					РП 05. 24 003. 00 ДП ПЗ	Акт
Зм	Дот	№ докум.	Підпис	Дата		34

```

ReportQuery.Clear;
ReportQuery.SQL.Add('select org_desc, dep_desc, lname from
Organization.db, Departments.db where empkey not in (select distinct empkey
from Simpleg.db where today>=:DateStart and today<=:Dateend);
ReportQuery.Open; // Зробити запит

```

4. Звіт про тих, хто запізнився. Отримання списку співробітників, що мають запізнення за певний період. Може бути сформований таким запитом:

```

ReportQuery.Clear;
ReportQuery.SQL.Add('select org_desc, dep_desc, lname, mustcome, lating
from Organization.db, Departments.db, Employers.db, LateComers.db where
Employers.empkey = LateComers.empkey and today>=:DateStart and
today<=:Dateend');
ReportQuery.Open; // Зробити запит

```

Після отримання відповідей з бази даних на запити звіт можна переглянути на екрані, скориставшись в Delphi палітрою компонент QReport. З усієї палітри знадобляться наступні компоненти:

QuickReport: TQuickRep – макет сторінки формату А4 для розташування на ньому елементів звіту. Володіє можливостями перегляду сторінки під різним збільшенням, запису і завантаження сторінки у формат ".QRP", а також виведення сторінки на друк;

Band1, Band2: TQRBand – для створення "шапки" і "тіла" таблиці;

Org, Dep, Name: TQRLabel – написи назв в "шапці" таблиці;

SysData: TQRSysData – виведення на макет службової інформації (дата, час, номер сторінки звіту);

Org_desc, Dep_desc, Fname: TQRDBText – для виведення списку отриманих записів з відно відних назв полів бази даних;

Alert: TQRDBExpr – для створення тригерів на виведення записів з бази даних (умовні оператори, логічні вирази);

У якості джерела даних у властивості "DataSet" всіх компонент потрібно вказувати назву компоненту "ReportQuery", яка буде містити список отриманих записів з бази даних.

Якщо потрібно експортувати отримані звітні дані для їх подальшого перерахунку, наприклад, за допомогою спеціально призначеної для цього програми Microsoft EXCEL, то найбільш простим варіантом буде перевести їх у зрозумілий EXCEL формат ".CSV".

Як можна дізнатися з [12], формат ".CSV" влаштований таким чином: це текстовий файл, у якому дані, що розміщуються за різними стовпцями таблиці EXCEL, розділяються між собою комами (Абревіатура "CSV" розшифровується як Comma Separated Values, що в перекладі з англійської означає значення, розділені комами), а рядки таблиці EXCEL – це рядки текстового файлу.

1.6.5 Модуль для роботи з базою даних

Модуль для роботи з базою даних призначений для редагування і адміністрування інформації, що зберігається в ній. Наприклад, з його допомогою можна буде додавати, видаляти, редагувати інформацію про співробітників; заводити, видаляти, перейменовувати відділи та організації в структурі підприємства; призначати права доступу співробітникам і визначати рамки їх робочого часу. Крім того, можна буде експортувати дані за чотирма типами звітів в Microsoft Excel і видаляти застарілі записи з журналів проходної.

Для наочного уявлення структури підприємства скористаємося стандартною компонентою Delphi TTreeView [13]. З її допомогою можна буде візуалізувати деревоподібну ієрархію підрозділів на головному формі програми. Дерево структури внутрішнього устрою всього підприємства можна представити трьома ярусами: організація – відділ – співробітник.

Спочатку дані про структуру підприємства будуть зчитуватися послідовно з трьох таблиць бази даних, що під'єднується "Organization.db", "Departments.db", "Employers.db" за допомогою найпростіших SQL-запитів на

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

отримання списку всіх наявних записів в конкретній таблиці і візуалізуватися у
видимій області компоненти Emp Tree View: T Tree View, викликаючи методи:

```
Emp Tree View.Items.Add(organization);  
// додавання самого верхнього рівня в дерево;  
Emp Tree View.Items.AddChild(Node, Name); // додавання підрізня
```

де organization – рядок з назвою організації, Node – структура типу
TTreeNode, яка вказує рівень попередника у дереві, Name – назва відділу або
прізвище співробітника.

Далі, створивши компоненту EmpFormMenu, описану класом
TFormMenu, і перерахувавши в її властивості "Items" необхідні пункти для
роботи з базою даних "Додати", "Видалити", "Змінити", вкажемо її в якості
меню, що з'являється, у властивості "FormMenu" нашої компоненти
Emp Tree View, що активується при натисканні мишкою на видимому області
компоненти Emp Tree View. Потім додамо обробники перерахованих пунктів
меню:

```
// Пункт меню "Додати"  
Emp Query.SQL.Clear;  
Case TreeLevel of  
0: // додати нову організацію номер org, з назвою org_desc  
Emp Query.SQL.Add ('insert into organization values (org, org_desc);  
1: // додати новий відділ організації з номером org під номером dep  
// з назвою dep_desc  
Emp Query.SQL.Add ('insert into departments values (org, dep, dep_desc);  
2: // додати нового співробітника у відділ номер dep організації номер  
// org під номером EmpKey, з особистими даними  
Emp Query.SQL.Add ('insert into Emp byers values (org, dep, empkey,  
tablename, keycode ...)');  
end;  
Emp Query.Prepare; // Підготувати запит
```

Зм	Дот	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Лист

37

```

Emp Query Ехес SQL; // Реалізувати з апнт
// Пункт меню "Видалити"
Emp Query .SQL .Clear;
Case TreeLevel of
0: // видалити організацію з назвою org_desc з усіма її відділами та
    // співробітниками
Emp Query .SQL .Add ('delete from organization, departments, employers where
org_desc=:org_desc and organization.dep=departments.dep and
organization.org=employers.org ');
1: // видалити відділ з назвою dep_desc з усіма його співробітниками
Emp Query .SQL .Add ('delete from departments, employers where dep_desc=:
dep_desc and departments.dep=employers.dep ');
2: // видалити співробітника під номером Emp Key
Emp Query .SQL .Add ('delete from Employers where empkey=:empkey ');
end;
Emp Query Prepare;
Emp Query Ехес SQL;
// Пункт меню "Змінити"
Emp Query .SQL .Clear; Case TreeLevel of
0: // перейменувати організацію з назвою org_desc в new_org_desc
Emp Query .SQL .Add ('update organization set org_desc =:new_org_desc ');
1: // перейменувати відділ з назвою dep_desc в new_dep_desc
Emp Query .SQL .Add ('update departments set dep_desc =:new_dep_desc ');
end;
Emp Query Prepare;
Emp Query Ехес SQL;

```

Для того, щоб змінити особисті дані співробітника, необхідно створити і додати в проект нову форму, всередині якої буде відображатися особиста

					РП 05.24.003.00 ДП ПЗ	Лист
Зм	Дат	№ докум.	Підпис	Дата		38

інформація про співробітника, що береться з бази даних для можливості її подальшої модифікації. На формі будуть розміщені такі компоненти:

- DVEditRole, DVEditFName, DVEditLName, DVEditMName, CodeEdit: TDVEdit – для зміни строкових полів в базі даних (посада, прізвище, ім'я, по батькові, код перепустки);
- EmpImage: TDVImage – для роботи з зображеннями формату "BMP" (фотографія);
- Blocking: TDBRadioGroup – для вибору пунктів, що перераховуються (пропуск: блокований / не блокований);
- EmpMemo: TDVMemo – для відображення великих текстових полів (додаткова інформація);
- StartEdit, EndEdit: TDVDateEdit – для роботи з датами (період дії пропуску).

У властивості "DataSource" всіх цих компонент слід вказати джерело отримання даних з бази: DataSource: TDataSource – зв'язок таблиці EmpTable бази даних з компонентами. У самій таблиці EmpTable необхідно встановити властивість CashedUpdates = true, щоб всі зміни, виконувані за допомогою описаних компонент, накопичувалися в локальному кеші, а по завершенню редагування відсилалися в базу за допомогою методу EmpTable.Post.

Тепер зупинимось на адмініструванні бази даних і розглянемо питання про призначення прав доступу і встановлення рамок робочого часу для працівників. Таблиця "Rights.db" містить інформацію про права доступу, які можуть призначатися як безпосередньо для конкретного співробітника, так і для всього відділу співробітників або всієї організації в цілому.

Для цього в полях таблиці org, dep, empkey відповідно встановлюються номер організації, відділу або співробітника, для яких діють тимчасові обмеження T_From, T_To – інтервал часу доступу на територію підприємства. Механізм успадкування прав такий, що пошук прав доступу буде здійснюватися

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Арт

39

спочатку для співробітника, потім для відділу, якому належить співробітник, і вже потім для організації, до якої входить співробітник.

Аналог іншим чином влаштована таблиця "WorkRange.db", що містить інтервали робочого часу для працівників.

1.6.6 Головний модуль

Головний модуль об'єднує в собі всі інші модулі і надає інтерфейс з вибору локальної бази даних, що підключається, віддаленої автоматизованої прохідної.

Зовнішній вигляд форми вікна головного модуля наведено на рис. 1.10.

Інтерфейс вибору бази даних, що підключається, надає можливість вносити до списку автоматизованих прохідних нові прохідні, вказувачи у якості реєстраційних даних назву, шлях (UNC-м'я), по якому буде здійснюватися підключення, і будь-яку додаткову інформацію у якості коментаря.

Позиціонування у мережі Windows Network автоматизованої прохідної (її повне UNC-м'я) можна реалізувати наступним чином через функції WinAPI [11]:

```
// Функція визначення місця розташування ресурсу в мережі
function GetSelectedDir (Handle : THandle): string;
var
    s: TBrowseInfoA;
    IDList: PItemIDList; // Список ресурсів
function GetPathFromIDList: string;
var
    Buf: array [0..MAX_PATH-1] of Char;
    Res: Boolean;
begin
    Result := "";
    if IDList <> nil then
        begin
            Res := SHGetPathFromIDList (IDList, Buf); // вибрати шлях до БД
```

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

```

    if Res then Result: = Buf else Exit;
    end;
end;
begin
    // Заповнюємо структуру TBrowseInfoA
    s.hwndOwner: = Handle; // хендл вікна
    s.pidlRoot: = nil;
    s.pszDisplayName: = nil;
    s.pszTitle: = PChar ('Огляд мережевих ресурсів'); s.ulFlags: = 0;
    s.pfn: = nil;
    s.Param: = 0;
    s.Image: = 0;
    IDList: = SHBrowseForFolder (s); // виклик функції вибору шляху до БД
    Result: = GetPathFromIDList; // Повернення результату: UNC-ім'я до БД
end;

```

Виклик функції GetSelectedDir (0) з передачею їй у якості параметру значення "0" (хендл поточного вікна), відобразить на екрані форму для навігації по всім ресурсам мережі. Після вибору необхідного мережевого шляху у якості результату роботи функція поверне повне UNC-ім'я для доступу до ресурсу, яке можна буде додати у список.

Алгоритм роботи наступний: зі списку зареєстрованих автоматизованих прохідних, що зберігається у локальному файлі "KrrpData.ini", вибирається одна з них, до якої здійснюється мережеве підключення.

Далі користувач за допомогою модуля для отримання звітів може зібрати накопичену інформацію з журналіве прохідної за певний період з можливістю її подальшого експорту в Microsoft Excel, або модифікувати інформацію в базі даних співробітниць підприємства за допомогою модуля для роботи з базою даних.

					РП 05.24.003.00 ДП ПЗ	Акт
Зм	Дат	№ докум.	Підпис	Дата		41

1.7 Конструювання та опис інтерфейсу користувачів

Система контролю доступу та обліку робочого часу розроблена для можливості побудови мережі КПП підприємства з метою ведення обліку робочого часу та контролю доступу співробітників. Програмне забезпечення складається з двох програм: "Monitor" і "Manager". Програма "Monitor" встановлюється на комп'ютер контрольно-пропускного пункту і здійснює автоматизований контроль прохідної. Програма керує процесом проходження співробітників через електромагнітні турнікети, керувачись правами доступу для співробітників, закладеними в локальній базі даних. Програма "Manager" встановлюється на так званій головній комп'ютер підприємства, який має можливість віддаленого доступу до локальних баз даних комп'ютерів КПП, включених в комп'ютерну мережу, з метою їх адміністрування і отримання звітних відомостей по прохідним системам за певний період часу.

1.7.1 Інтерфейс програми "Monitor"

1. Реєстрація в системі.

GUI-інтерфейс вікна реєстрації в системі має вигляд, показаний на рис. 1.7.



Рисунок 1.7. Реєстрація у системі при вході до програми "Monitor"

Напроти напису «Користувач» слід набрати на клавіатурі ім'я користувача. Ім'я може бути «Operator» або «Admin». Напроти напису «Пароль» слід ввести пароль для відповідного імені користувача і після цього натиснути клавішу «ОК». Відмінності між «Admin» і "Operator" полягає в тому, що вхід в систему під іменем "Admin" дозволяє вносити зміни в налаштування програми. Вхід під іменем «Operator» дає тільки право контролювати коректність роботи

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

системи і стежити за ситуацією на прохідній, отримувачи на екрані монітора інформацію про тих, хто увійшов або вийшов.

2. Функціональні можливості програми.

Після натиснення на кнопку «ОК» з'являється головна форма вікна програми контролю-пропускнуої системи на екрані монітора. Її вигляд наведено на рис.

1.8.

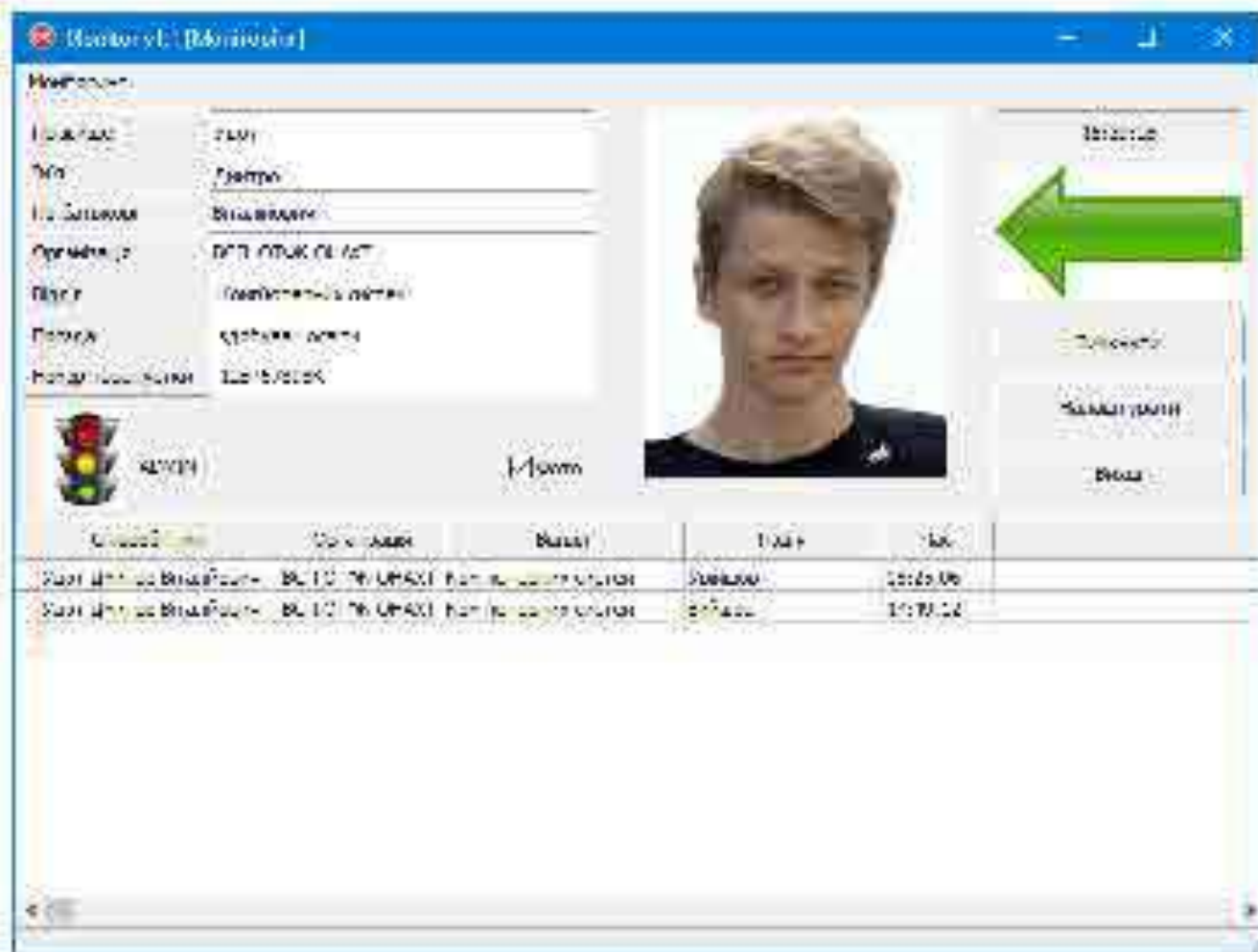


Рисунок 1.8. Головна форма програми "Monitor"

У вікні передбачені такі елементи:

1. Відомості про співробітника (П.І.Б., організація, відділ, посада, номер пропуски, фотографія). Ці поля заповнюються кожен раз відомостями про працівника, який останнім проїшов прохідку в ту чи іншу сторону;

2. Системна кнопка контролю-пропускнуої пункту. Після її натиснення з'являється меню з пунктами: «Увійти в систему», «Змінити пароль», «Про систему...», які дозволять, відповідно, увійти в систему під іншим ім'ям, змінити

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

поточний пароль на інший, дізнатися відомості про версію системи;

3. Значок поточного стану турнікета. Стрілка показує напрямок переміщення людини через турнікет (якщо стрілка вліво – людина увійшла, якщо вправо – вийшла). Хрестик означає, що турнікет знаходиться в даний момент в режимі очікування проходження через нього.

Крім того, можливі поля системних повідомлень замість малюнків, що сигналізують про різні порушення режиму входу або виходу (наприклад «Доступ заборонений», «Пропуск при вході» і т.д.). Під цим значком розташовані кнопки: «Увімкнути» («Вимкнути») – для переведення турнікета з автоматичного режиму в ручний і навпаки. При автоматичному режимі турнікетом керує програма, при ручному – турнікетами може керувати оператор, користуючись для цього ручним пултом управління. «Налаштувати» – дозволяє змінювати налаштування програми (тільки користувачі зі "Admin"). При натисканні на цю кнопку з'являється вікно, показане на рис. 1.9.

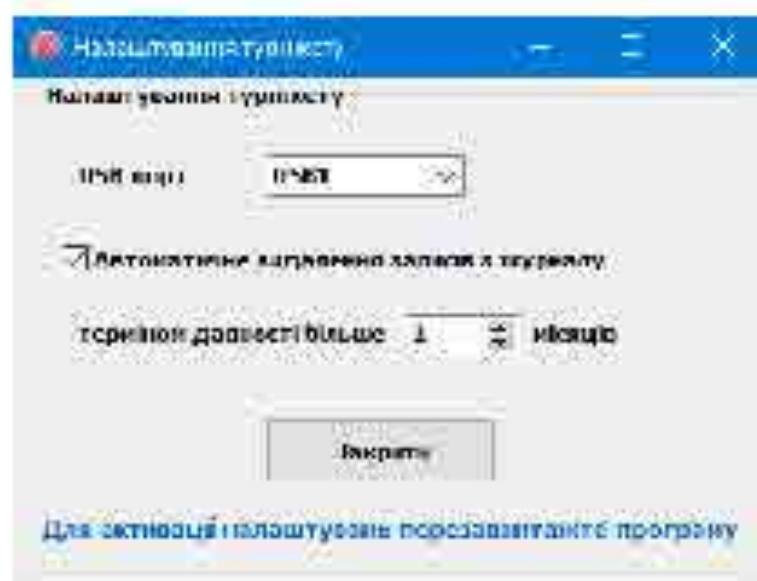


Рисунок 1.9. Налаштування програми "Monitor"

У ньому можна встановити номер USB-порту для зв'язку комп'ютера з контролером турнікета, швидкість обміну даними по USB-порту, встановити режим автоматичного видалення застарілих записів з журналу реєстрації проходної, термін давності яких задається в інтервалі від 1 до 12 місяців.

4. Журнал проходної (нескінченна форма). У ньому фіксуються записи

					РП 05.24.003.00 ДП ПЗ	Акт
Зм	Доз	% датум.	Підпис	Дата		44

про входи і виходи співробітників через прохідну. Також позначаються порушення прохідного режиму (коли співробітник намагається пройти, а програма визначає, що його не можна пропускати з будь-якої причини).

1.7.2 Інтерфейс програми "Manager"

Процес реєстрації в системі для програми «Manager» аналогічний процесу реєстрації в програмі «Monitor». Користувач "Admin" наділений повними правами для управління і адміністрування контрольно-пропускними постами. Зовнішній вигляд інтерфейсу програми «Manager» представлений на рис. 1.10.

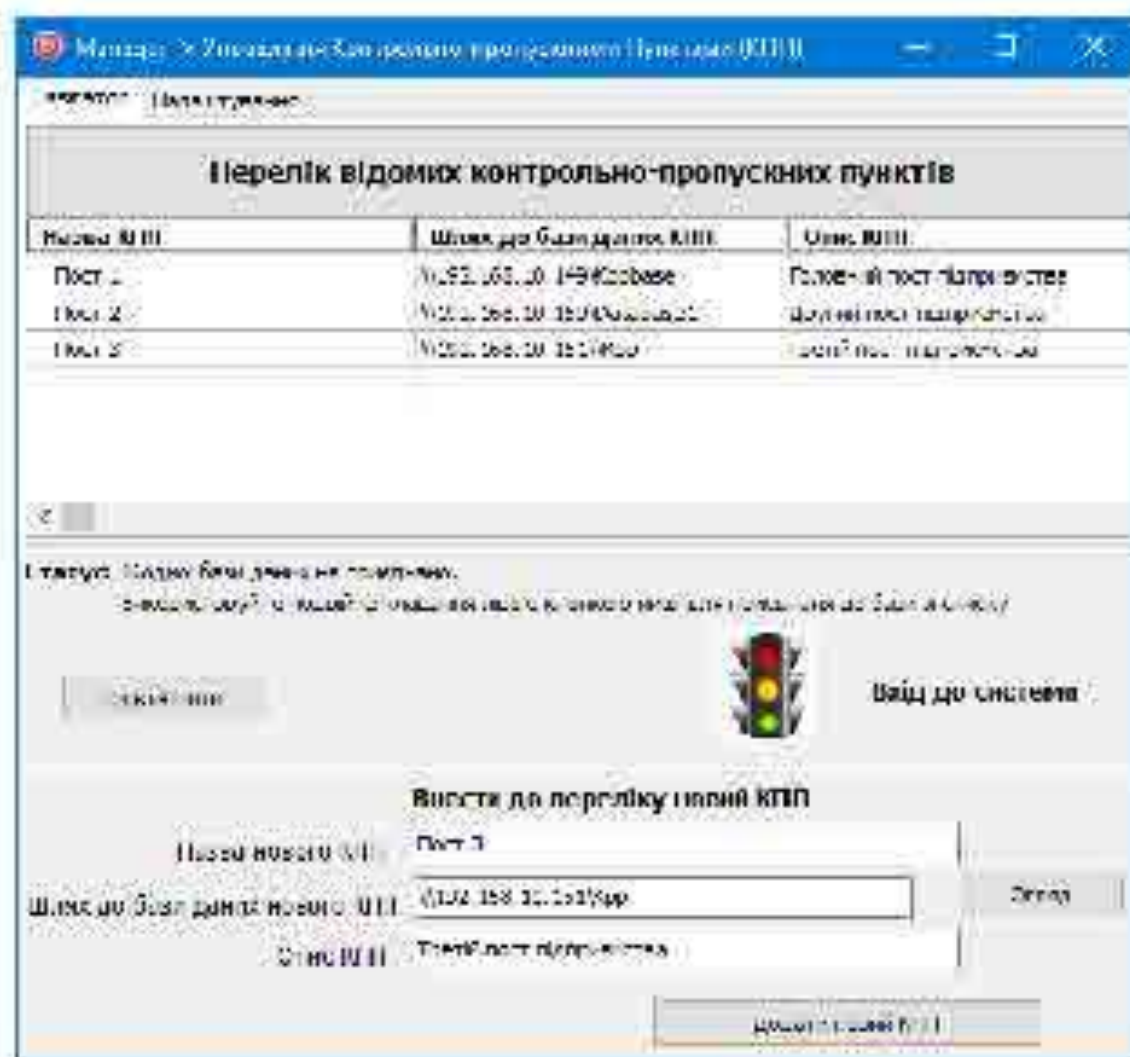


Рисунок 1.10. Зовнішній вигляд програми "Manager"

На формі розташовані дві закладки: «Навігатор» і «Налаштування», кожна з яких, при натисканні на ній, відкриває окрему сторінку.

На сторінці «Навігатор» розташовані:

- список відомих контрольно-пропускних пунктів (із рядком статусу під ним). У списку містяться: символічна назва КПП, адреса КПП у вигляді UNC-імені і опис цього КПП;
- поля «Назва нового КПП», «Шлях до бази даних нового КПП», «Опис КПП» разом із кнопками «Огляд» і «Додати»;
- значок «світлофора» – кнопка, натискаючи на яку, відображується меню: «увійти в систему», «змінити пароль», «про систему...».

Сторінка «Навігатор» надає механізм навігації по базах даних контрольно-пропускних пунктів:

1. Список відомих КПП. Список відомих КПП потрібен для того, щоб містити в собі необхідну інформацію для доступу до кожного КПП. У нього повинні вклучатися всі контрольно-пропускні пункти підприємства. Список автоматично запам'ятовується програмою "Manager" і при наступних сеансах роботи завантажується з файлу;

2. Додавання КПП в список. Для цього необхідно заповнити обов'язкові поля: «Назва нового КПП» і «Шлях до бази даних нового КПП». Поле «Опис КПП» є додатковим і може не заповнюватися. В полі «Шлях до бази даних нового КПП» слід ввести повний шлях до бази даних комп'ютера, на якому встановлена програма «Монітор» або можна, скориставшись кнопкою «Огляд», через «Мережеве оточення» знайти потрібний комп'ютер і вибрати його мережевий ресурс з базою даних. Формат шляху в полі «Шлях до бази даних нового КПП» задається наступними способами:

- Локальний диск з повним шляхом до бази даних на цьому диску, наприклад: "C:\Program Files\Monitor\Base\";
- Підключений мережевий диск з повним шляхом до бази даних на ньому, наприклад: "S:\Monitor\Base\" ("S:\" – підключений мережевий диск віддаленого комп'ютера КПП);
- В вигляді UNC-імені, що має формат: "\<ім'я комп'ютера в мережі>\<ім'я загального ресурсу>\<шлях до бази даних>". Наприклад: \Computer1\CommonResource\Base, або \192.168.12.150\CommonResource\Base, де

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

"CommonResource" – оголошений загальний ресурс.

Після заповнення полів треба натиснути кнопку «Додати» і новий КПП з'явиться у списку.

3. Видалення КПП із списку. Видалити КПП із списку можна двома способами:

- вибрати із списку необхідний для видалення КПП і натиснути на клавіатурі клавішу «DEL»;
- вибрати із списку необхідний для видалення КПП і натиснути на ньому правою кнопкою мишки, обрати пункт «Видалити».

4. Підключення до КПП. Необхідно обрати курсором мишки потрібний КПП зі списку відомостей і натиснути кнопку «Підключитися» або двічі клацнути мишкою на обраному пункті. У процесі з'єднання з віддаленою базою даних КПП з'явиться вікно з проханням ввести ім'я користувача і пароль, необхідні для визначення доступу до мережного ресурсу даного КПП, як показано на рис. 1.11. Необхідно вказати ім'я користувача в мережі Windows Network і пароль на загальний ресурс.

При невдачі приєднання до віддаленої бази даних з'явиться повідомлення, в якому буде вказана причина помилки або її код, який можна інтерпретувати за допомогою програми "NET.EXE", що входить в поставку Microsoft Windows. Для цього треба запустити програму "NET.EXE", передавши їй у якості параметру рядок "HELPMSG [Код помилки]". У разі успішного під'єднання на формі з'являться ще дві закладки «Сервіс» та «Експорт», а також кнопка «Відключитися».



Рисунок 1.11. Доступ до мережного ресурсу

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

5. Відключення КПП. Після вдалого підключення до бази даних віддаленого КПП з'являється кнопка «Відключити». Натискання на ній призведе до відключення від приєднаної бази даних (зникнуть закладки «Сервіс» та «Експорт»). Відключати з'єднані базу даних необхідно для того, щоб не блокувати доступ до неї іншим користувачам програми "Manager", якщо такі є. При приєднанні до чергової бази даних поточна під'єднана база даних автоматично відключається. Те ж саме відбувається при виході з програми "Manager".

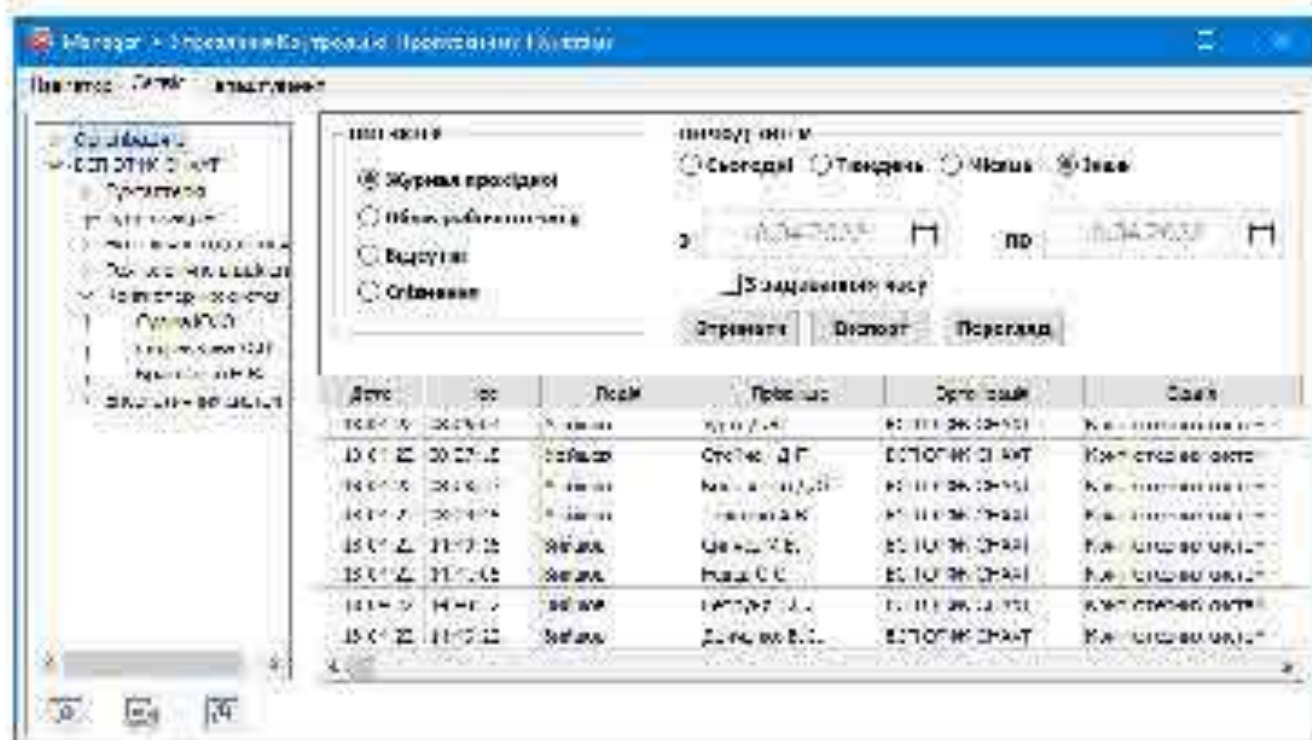


Рисунок 1.12. Загальний вигляд сторінки «Сервіс»

Сторінка «Сервіс» (рис. 1.12) з'являється після успішного підключення до бази даних контрольно-пропускного пункту і призначена для роботи з під'єднаною базою даних:

1. Дерево організації, відділів, співробітників (зліва на формі). Дерево – це структура, яка відображає ієрархію підприємства. На дереві видно розташування організації, що включають в себе відділи, а також відділи, що включають в себе співробітників. Таким чином всі співробітники належать жомусь відділу і організації. Під деревом розташовані кілька іконок:

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Арт

48




 – знайти співробітника у дереві по його прізвищу. Натискаючи на цю кнопку, відкривається вікно пошуку (Рис. 1.13), у якому можна, розгорнувши список всіх існуючих співробітників, знайти необхідного.



Рисунок 1.13. Вікно пошуку співробітника

 – оновити дерево. Оновлювати дерево організації, відділів, співробітників необхідно тоді, коли в ньому можуть з'явитися нові записи, внесені кимось іншим під час роботи (наприклад, інший користувач програми "Manager" підключився до цієї ж бази даних і зробив деякі зміни: додав/видавлив співробітника, організацію, відділ, перевів співробітника у інший відділ або організацію).

 – розгорнути все дерево цілком, щоб було видно всі його організації, відділи і всі співробітники у всіх відділах.

2. Отримання звітів (права частина форми). В даній версії програми передбачені чотири форми звітів, які мають задоволяти запити користувачів:

1) Звіт журналу проходної. Надає відомості про всі входи і виходи співробітників, які проходили турнікет. Дані видаються у вигляді списку в форматі: Дата, Час, Подія, П.І.Б., Організація, Відділ, Порушення;

2) Звіт робочого часу співробітників. У ньому є можливість отримати сумарний час, проведений співробітником на роботі за певний період. Формат звіту: Організація, Відділ, П.І.Б., Сумарний робочий час;

3) Звіт про відсутніх. Результат буде представлений у вигляді списку співробітників, які були відсутні в певний інтервал часу на робочому місці. Формат звіту: Організація, Відділ, П.І.Б.;

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

4) Звіт про тех, хто запізнився. Видається список осіб, що мають запізнення на робоче місце за певний інтервал часу. Формат звіту наступний: Організація, Відділ, П.І.Б., Дата, Час приходу, Запізнення.

Для отримання звітів у списку «Тип звіту» мишкою обрати потрібний тип, у списку «Період звіту» вказати інтервал часу: «Сьогодні», «Тиждень», «Місяць», «Інше» і натиснути на кнопку «Отримати».

Під пунктом «Інше» мається на увазі довільне вказання початкової і кінцевої дати для отримання звіту, які задаються в розташованих нижче полях «з» і «по». Також, якщо необхідно визначити не тільки дату початку і кінця терміну звіту, а ще й час доби для цих дат, то натискаючи мишкою на напис «З задаванням часу» з'являться ще два поля «з» і «по» – які повинні бути заповнені необхідним часом доби.


Після натиснення на кнопку «Перегляд» з'явиться повноекранне вікно, у якому можна буде переглянути вид отриманого звіту для його подальшого друку на принтері. Угорітього вікна є ряд кнопок (рис. 1.14):




Рисунок 1.14. Панель управління переглядом

 – масштабує документ для зручності перегляду;

 – здійснюють переміщення по сторінках звіту;

 – видає відомості про конфігурацію принтеру;

 – подає на друк список сторінок звіту;

 – дозволять записати звіт у файл з розширенням .QRP і завантажити звіт з файлу з таким же розширенням;

 – закриває перегляд звіту.

3. Експорт даних звіту. Може виникнути необхідність зберегти або перерахувати отримані звітні дані програмов "Manager" за допомогою інших програм, спеціально призначених для цього, як, наприклад, Microsoft EXCEL. Для цього запроваджено механізм експорту даних у формат даних Microsoft EXCEL. Експорт даних з журналів прохідної здійснюється у таблиці EXCEL.

Для експорту звітних даних потрібно:

- отримати необхідний звіт;
- натиснути кнопку «Експорт»;
- вказати у діалоговому вікні, під яким ім'ям зберегти експортовані дані для EXCEL (автоматично зрозширенням .CSV).

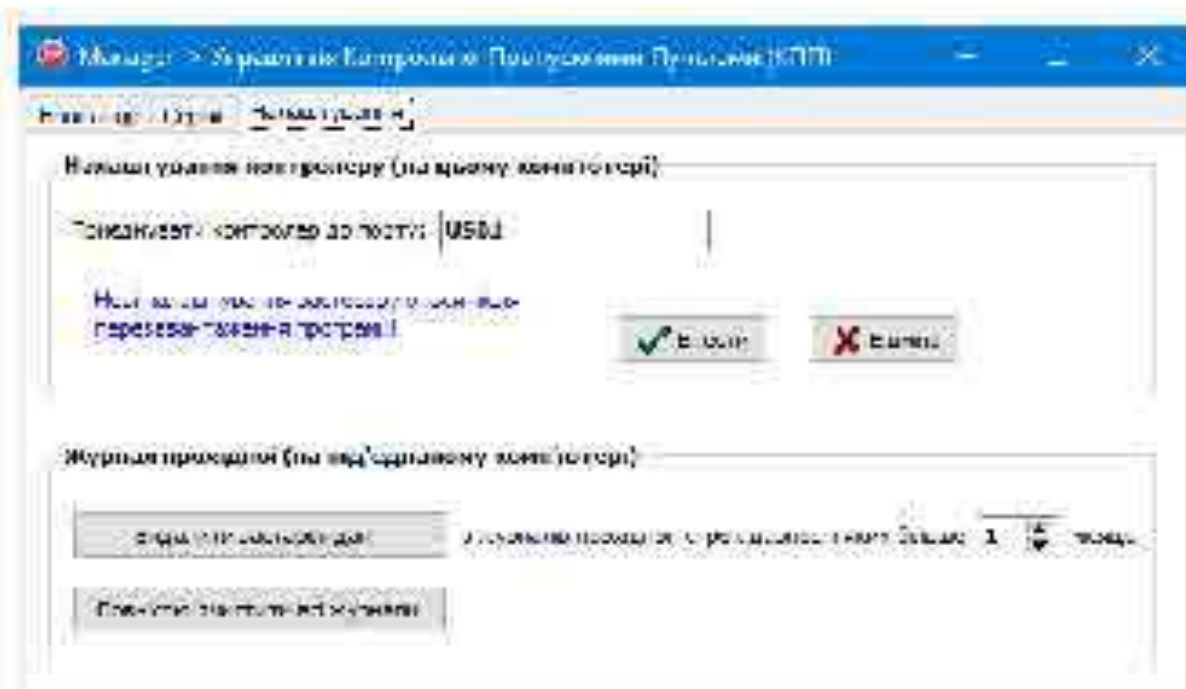


Рисунок 1.15. Сторінка «Налаштування»

Сторінка «Налаштування» (рис. 1.15) надає такі можливості:

На панелі «Налаштування контролера» встановлюються параметри зв'язку головного комп'ютера з підключеним до нього через USB-порт контролером. Контролер підключається до головного комп'ютера з метою мати можливість заносити у під'єдану базу даних віддаленого КПП зчитані коди ідентифікаційних карт за допомогою безконтактних зчитувачів, підключених до контролера.

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Щоб активізувати нові налаштування, необхідно натиснути кнопку «Внести», а потім перезапустити програму. На панелі «Журнали прохідної» є можливість видалити застарілі записи з бази даних приєднаного комп'ютера КПП. Кнопка «Видалити застарілі дані» зітре всі записи з усіх журналів прохідної, термін давності яких перевищує встановлене значення в полі навпроти кнопки. Кнопка «Повністю очистити всі журнали» – виконає очистку всіх журналів від всіх наявних в них записів. Після натискання на будь-яку з цих кнопок треба підтвердити видалення.

Адміністрування приєднаного КПП виконується тільки користувачем «Admin», наділений правом адмініструвати базу даних, користуючись інтерфейсом, наданим йому за допомогою програми «Manager». Спочатку, після встановлення на віддаленому комп'ютері КПП програми «Monitor», його база даних буде порожньою. Після з'єднання з нею за допомогою програми «Manager» її можна заповнити:

1. Заповнення дерева підприємства. Відразу після підключення програмою «Manager» до порожньої бази даних КПП на сторінці «Сервіс» відображується порожнє дерево організації, відділів, співробітників. Після наведення на нього мишки і натиснення її правої кнопки з'являється пункт меню «Додати організацію», який треба обрати. Треба ввести назву організації та натиснути «ОК». У дереві з'являється пункт з назвою щойно введеної організації. Після наведення на нього мишки і натиснення її правої кнопки у меню з'являється пункт:

«Додати відділ» – для додавання відділу цієї організації. При виборі цього пункту з'являється діалогове вікно, схоже на попереднє, у якому треба написати назву відділу;

«Видалити організацію» – організація буде видалена;

«Змінити» – поміняти назву організації;

«Час доступу» – задати інтервали часового доступу для співробітників, що належать даній організації;

Зм	Дот	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Ас

52

«Робочий час» – задати інтервали робочого часу для працівників організації.

Якщо тепер додати відділ і аналогічно натиснути на ньому праву кнопку мишки, то далі можете додати в цей відділ співробітника, вибравши відповідний пункт меню. Діючи таким чином, можна заповнити самостійно дерево підприємства, включаючи в нього нові організації, відділи та співробітників.

Вибір інформації про співробітника

Ім'я:

Прізвище:

Дата народження:

Стать:

Дата прийняття на роботу:

Код підприємства: 123456789

Відділ:

Посада:

Додати фото

Додати

Додаткова інформація

Розшифровка: 123456789

OK

Скасувати

Рисунок 1.16. Форма інформації про співробітника

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

2. Заповнення інформації про співробітника. Якщо обрати пункт «Додати співробітника» для якогось відділу або вибрати у дереві будь-якого співробітника, натиснути на ньому правою кнопкою мишки, після чого у меню обрати пункт «Змінити», то з'явиться вікно вкву, показано на рис. 1.16. В ньому можна ввести прізвище, ім'я, по батькові, посаду співробітника, визначити час дії перепустки (вибравши дату початку і закінчення), ввести код перепустки (або видалити його), заблокувати пропуск («Блокування перепустки»), вставити або прибрати фотографію співробітника, вказати додаткову інформацію про співробітника, перевести співробітника у іншу організацію і/або відділ (кнопка «Перевести»).

Для переведення співробітника треба натиснути на кнопку «Переведення». З'явиться діалогове вікно, зображене на рис. 1.17, де буде запропоновано вибрати із списків, що розкриваються, назву нової організації для співробітника та назву відділу цієї організації. Після натиснення кнопки «Переведення» обраний співробітник буде переведений на нове місце роботи, що буде показано у дереві організацій, відділів, співробітників.

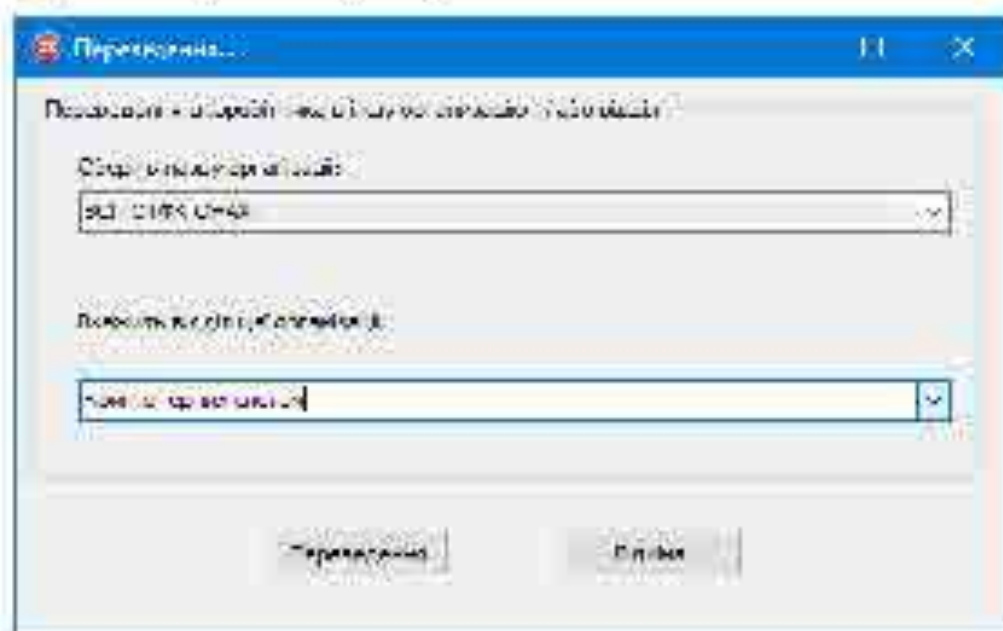


Рисунок 1.17. Форма переведення співробітника

3. Отримання коду перепустки. Для цього необхідно вибрати співробітника в дереві, натиснути правою кнопкою мишки, в меню вибрати «Отримати перепустку».

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Екранне вікно із запрошенням. Тепер треба піднести ідентифікаційну карту доступу, видану співробітнику, до безконтактного зчитувача контролера, приєднаного до комп'ютера. У разі успіху з'явиться напис «Картка успішно закріплена!», а в іншому випадку з'явиться напис, який говорить про те, за ким уже закріплена дана карта. При успішному «закріпленні коду перепустки» в особистій картці співробітника навпроти напису «Код перепустки» буде стояти код картки. Також код перепустки можна записати вручну в особисту картку співробітника в поле «Код перепустки». Таким чином, наприклад, можна видалити код перепустки.

4. Час доступу. Під цим терміном розуміються тимчасові інтервали, встановлені для організації, або для відділу, або безпосередньо для співробітника, під час яких співробітник може допускатися на своє робоче місце (тобто проходити турнікет). Щоб встановити інтервали доступу треба вибрати мишкою потрібний об'єкт у дереві (підприємство, відділ або співробітника), натиснути на цьому об'єкті праву кнопку мишки і з меню натиснути на пункт «Час доступу», після чого з'явиться вікно (рис. 1.18).

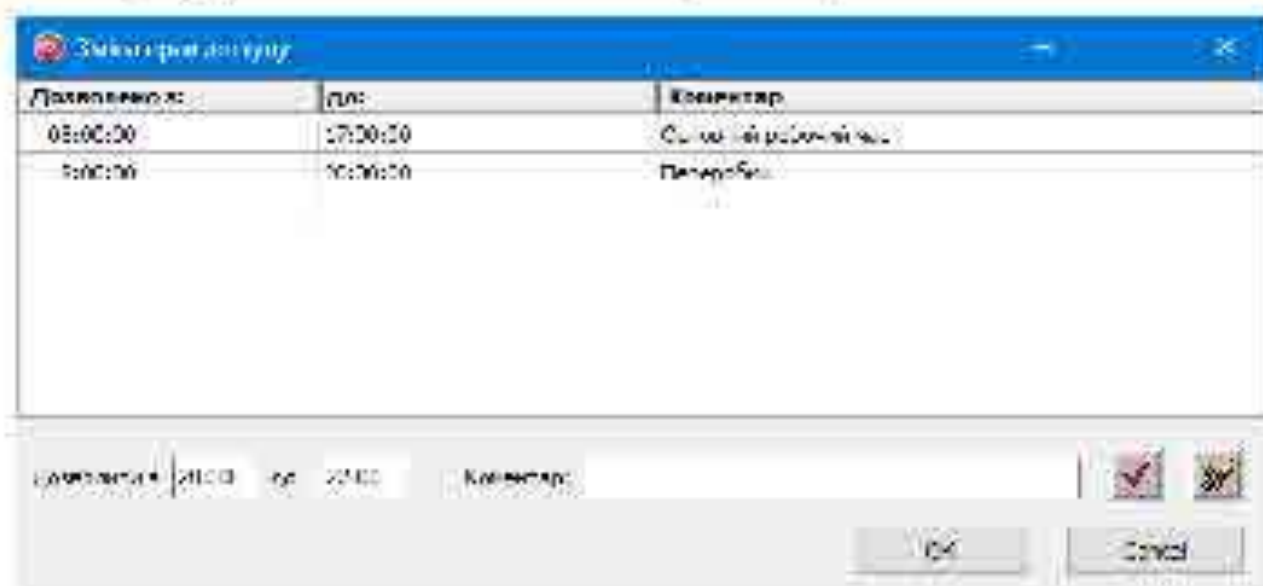



Рисунок 1.18. Форма задання прав доступу

У вікні змінення прав доступу можна задавати інтервали часу доступу у вигляді списку. Тут передбачені наступні елементи керування:

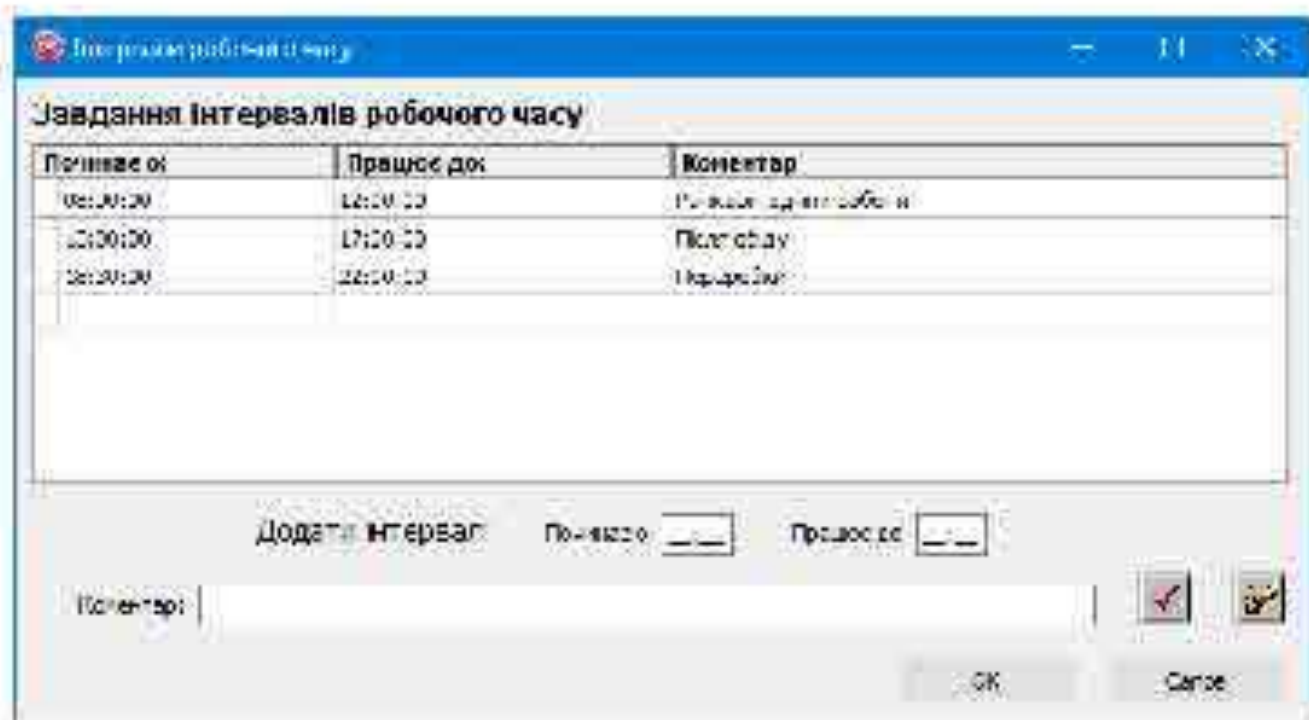
 – кнопка, яка вносить введені інтервали у список;

 – кнопка, яка повністю очищує список інтервалів часу доступу.

Для видалення жодного одного інтервалу часу потрібно навести курсор миші на початок цього інтервалу (стовпчик «Дозволяти з») і натиснути клавішу «DEL» на клавіатурі.

Кнопка «OK» заносить список інтервалів часу доступу до бази даних, а «Cancel» – скасує внесені зміни. Тепер, якщо співробітник прийде поза часом доступу, програма видасть повідомлення оператору про те, що доступ для нього заборонений і не відкриє турнікет.

5. Робочий час. Аналогічно механізму встановлення часу доступу встановлюються інтервали робочого часу – потрібно тільки вибрати пункт «Робочий час» (замість «Час доступу»). З'явиться форма, схожа на попередню, зі списком інтервалів (рис. 1.19).



Починає о:	Працює до:	Коментар
08:00:00	16:00:00	Робочий день
17:00:00	17:00:00	Після обіду
22:00:00	22:00:00	Відпочинок

Додати інтервал Починає о: _____ Працює до: _____

Коментар: _____

OK Cancel

Рисунок 1.19. Форма задавання інтервалів робочого часу

6. Підразунок робочого часу.

Після задавання інтервалів робочого часу для організації, відділу або конкретного співробітника, звіт «Робочий час» буде працювати за такими

Зм	Доз	% варт.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

правилами:

1. Робочий час підраховується для співробітника тільки в той період, коли він був присутній на робочому місці під час інтервалів робочого часу;

2. Робочий час не рахується для співробітника, якщо він був присутній на робочому місці поза всіма заданими інтервалами робочого часу.

Наприклад, були задані інтервали робочого часу: з 8:00 до 12:00 – ранкові години роботи, з 13:00 до 17:00 – післяобідні години роботи. Нехай співробітник прийшов на роботу о 8:45, був на роботі до 12:30, пішов на обід, повернувся о 13:15 і пішов з роботи о 18:00. У цьому випадку програмою будуть виконані наступні дії: записується запізнення на 45 хвилин (має прийти о 8:00, запізнився на 45 хвилин); ранкові години роботи підрахує, як 3 години 15 хвилин (з 8:45 до 12:00); записується запізнення на 15 хвилин (має прийти о 13:00, запізнився на 15 хвилин); післяобідні години підрахує, як 3 години 45 хвилин (з 13:15 до 17:00). Співробітник пішов з роботи о 18:00, але робочий день закінчився о 17:00, тому зайва година переробок не враховується. Таким чином, отримуємо у звіті «Робочий час» за день для співробітника – 7 годин, а у звіті «Спізнились» – запізнився на 45 хвилин після 8:00 і запізнився на 15 хвилин після 13:00.

Якщо профіль організації такий, що важливо тільки те, скільки часу взагалі співробітник провів на роботі (з огляду на всі його «переробки» на робочому місці після закінчення робочого дня), то можна задати один інтервал робочого часу (для описаного вище прикладу буде: з 8:00 до 23:59) і програма підрахує: запізнення для співробітника (тобто прихід після 8:00), робочий час – скільки всього часу за день співробітник був присутній на роботі.

Програма, при пошуку в базі даних інтервалів часу («Час доступу» або «Робочий час») для конкретного співробітника, шукає спочатку інтервали безпосередньо для конкретного співробітника. У разі, якщо співробітнику нічого не приписано, пошук продовжується для відділу, де працює цей співробітник. Якщо не знайдено, то пошук ведеться для організації, де працює співробітник.

Зм	Дат	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Арт

57

Якщо взагалі не знайдені інтервали часу доступу, то співробітник не зможе пройти турнікет. Якщо взагалі не знайдені інтервали робочого часу, то робочий час буде підрахований повністю.

1.8 Тестування проекту

Працездатність програмного забезпечення СКУД персоналу на підприємстві перевірялася на двох комп'ютерах, на один з яких встановлювалася програма моніторингу з підключенням всього необхідного апаратного забезпечення, що грало роль спеціалізованого комп'ютера на автоматизованій прохідній. На інший комп'ютер була встановлена програма віддаленого доступу. Обидва комп'ютери були підключені до мережі Windows Network.

1.8.1 Тестування програми моніторингу

Тестування програми моніторингу проводилося за такими критеріями:

- швидкість прийняття рішення програмою для блокування або пропуску співробітника по зчитаному коду карти;
- довжина з'єднувального кабелю між комп'ютером і контролером турнікету;
- можливість роботи програми у фоновому режимі.

Була заведена база даних на 1000 осіб, встановлена довжина кабелю 8 метрів.

Результати тестування такі. Час пошуку співробітника склав не більше 0,05 секунди (при введенні фотографії співробітника розміром 10Кб на екран – не більше 0,1 секунди), що є достатнім для обслуговування безперервної черги людей на прохідній будь-якого підприємства. Довжина з'єднувального кабелю в цьому випадку цілком підходить для віддалення комп'ютера від турнікета на потрібну відстань.

Зм	Доз	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

1.8.2 Тестування програми віддаленого до ступу

Основним критерієм при тестуванні програми віддаленого до ступу був час отримання чотирьох типів звітів. Підключення до мережі здійснювалося з іншого сегмента мережі Windows Network, відстань між комп'ютерами була не менше 100 метрів. За допомогою програми "ring.exe" був визначений середній час отримання пакетів (по 32 байти), який склав 2 мс. База даних містила 2000 записів у всіх журналах прохідної. Всього було проведено 10 дослідів, результати тестування наведені у таблиці 1.2.

Таблиця 1.2. Результати часу отримання звітів

Тест	Звіт N1	Звіт N2	Звіт N3	Звіт N4
час	19 сек.	18 сек.	4 сек.	20 сек.

"Звіт N1" – звіт про прохідну; "Звіт N2" – звіт про робочий час; "Звіт N3" – звіт про відсутність; "Звіт N4" – звіт про тих, хто заперився. На підставі отриманих результатів можна розраховувати приблизний час отримання звітів за між'яць по накопиченим статистичними даними.

Нехай на підприємстві працює 1000 співробітників, в кожен з 22 робочих днів місяця в журнал прохідної додається в середньому по 10000 записів (два на прихід і відхід з роботи, два на прихід і відхід на обід і пізність – на прихід і відхід за випадковими обставинами), тоді за місяць буде накопичено 220000 записів, що у 100 разів більше, ніж було при тестуванні. Звідси випливає, що час отримання самого трудомісткого звіту за між'яць складе 2000 секунд (або близько 34 хвилин) у такому випадку.

Зм	Дот	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

2 ЕКОНОМІЧНА ЧАСТИНА

2.1 Резюме

В даному дипломному проекті розроблене програмне забезпечення системи контролю і управління доступом на об'єкт. З його допомогою можна значною мірою полегшити і оптимізувати роботу з охорони та обліку робочого часу співробітників на підприємстві і скоротити витрати на утримання робочого персоналу. Використання створеного програмного забезпечення дозволить перейти на новий етап впровадження високонадійних систем безпеки, в тому числі і для режимних об'єктів.

Ефективність кожного програмного продукту визначається його якістю та ефективністю процесу розробки. Якість ПП визначається наступними складовими: з точки зору користувача; з позиції використання ресурсів; виконання вимог до програмного забезпечення. Оцінка якості програмного продукту з точки зору користувача визначається необхідним на стадії функціонування розміром оперативної пам'яті ЕОТ, витратами машинного часу, пропускнув спроможністю каналів передачі даних. Оцінка якості програмного продукту включає визначення трудомісткості і вартості його створення.

2.2 Визначення трудомісткості розробки програмного забезпечення

Тривалість розробки програмного продукту залежить від його обсягу, трудомісткості розробки, кваліфікації виконавців, а також планових термінів, визначених умовами ринку. Методом структурної аналогії по відповідних каталогах аналогів програмного забезпечення визначається обсяг програмних засобів, у тисячах умовних машинних команд програми аналога

Зм	Арх	№ версії	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Таблиця 2.1 Каталог аналогів

Найменування ПП	Обсяг функції ПП – V_o , усл. машинних команд
1. ПП автоматизованих розрахунків	1300 – 8600
2. Комп'ютерні системи ведення БД	950 – 7430
3. ПП введення інформації	1060 – 5750

У таблиці 2.1 представлені аналоги програмного забезпечення, функції яких, у більшому або меншому ступені, виконує розроблений програмний продукт. Для цього варіанта виділено сірим кольором.

Вибраний аналог ПП, що містить V_o в умовних машинних командах, трудоміткою є визначати на основі табл. 2.2

Таблиця 2.2

Обсяг ПП, тис. умов машинних команд	Норма часу, люд.днів
1.00	229
2.00	244
3.00	262

На підставі отриманого значення, по до відношу, визначається укрупнена норма часу на розробку аналога програмного забезпечення (коректується поправочним коефіцієнтом враховуючої умови розробки ПП, тобто в умовах комп'ютера, $K_k=0,7+0,8$): $T^* = 229 \times 0,8 = 183,2$ (люд.днів).

Трудоміткість програмного продукту визначається по кожному етапу розробки окремо на підставі трудоміткості аналога з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_n = T^* \cdot p \times L_i \times K_n \quad (2.1)$$

$$T_m = T^* \cdot p \times L_i \times K_n \quad (2.2)$$

$$T_m = T^* \cdot p \times L_i \times K_n \times K_r \quad (2.3)$$

Для розрахунку необхідні наступні коефіцієнти:

L_i – питома вага i -го етапу розробки (див. табл. 2.2);

K_n – поправочний коефіцієнт, що враховує ступінь новизни (див. табл. 2.3);

K_r – поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм (див. табл. 2.4.).

Зм	Лист	№ сторінки	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Лист

61

Таблиця 2.2 Значення питомих коефіцієнтів трудомісткості стадії в загальній трудомісткості розробки ПП

Код стадії	Ступінь новизни		
	A	B	B
TЗ (L ₁)	0,15	0,12	0,12
ТП (L ₂)	0,16	0,15	0,11
РП (L ₃)	0,55	0,58	0,61

Для нашого варіанта виділено сірим кольором.

Таблиця 2.3 Значення поправочного коефіцієнта, що враховує ступінь новизни

Код ступеня новизни	Ступінь новизни	Значення K _n
A	Принципово нові ПП	1,75 – 1,2
B	ПП – розвиток визначеного параметричного ряду	1,0 – 0,8
B	ПП мавчий аналог	0,7

Для нашого варіанта виділено сірим кольором.

Таблиця 2.4 Значення коефіцієнта ступеня використання в розробці типових програм

Ступінь охоплення реалізованих функцій розроблявального ПП типовими програмами, %	Значення K _r
60 і вище	0,6
40-60	0,7
20-40	0,8
До 20	0,9

Для нашого варіанта виділено сірим кольором.

Тепер розраховуємо трудомісткість по кожному етапу окремо:

Трудомісткість технічного завдання

$$T_{тз} = T^* * L_1 * K_n = 183,2 * 0,12 * 0,7 = 15,39 \text{ (люд.доданок)} \quad (2.1)$$

Трудомісткість розробки технічного проекту

$$T_{тп} = T^* * L_2 * K_n = 183,2 * 0,11 * 0,7 = 17,42 \text{ (люд.доданок)} \quad (2.2)$$

Трудомісткість розробки робочого проекту

$$T_{рр} = T^* * L_3 * K_n * K_r = 183,2 * 0,61 * 0,7 * 0,7 = 54,76 \text{ (люд.доданок)} \quad (2.3)$$

Для подальших розрахунків визначили кількість паперу, витраченого на кожен етап: технічне завдання N_{тз} = 2 (стор), розробка ТП N_{тп} = 18 (стор), розробка

Зм	Дот	% витрат	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

робочого проекту $N_{\text{пр}}=25$ (стор), поживальна записка відно $N_{\text{ор}}= 25$
(стор) Розрахунок зведений у таблицю 2.5

Таблиця 2.5 Розрахунок трудомісткості ПП

Найменування етапів	Розрахунок, години.		
1	2	3	4
1.ТЗ	$T_{\text{пр}}=15,39$	$T_{\text{за}}=0,7 \cdot N_{\text{пр}}=0,7 \cdot 2=1,4$	$T_{\text{за}}=0,15 \cdot N_{\text{пр}}=0,15 \cdot 2=0,30$
2.Розробка ТП	$T_{\text{пр}}=14,12$	$T_{\text{за}}=0,7 \cdot N_{\text{пр}}=0,7 \cdot 18=12,6$	$T_{\text{за}}=0,15 \cdot N_{\text{пр}}=0,15 \cdot 18=2,7$
3.Розробка РП	$T_{\text{пр}}=54,76$	$T_{\text{за}}=0,7 \cdot N_{\text{пр}}=0,7 \cdot 25=17,5$	$T_{\text{за}}=0,15 \cdot N_{\text{пр}}=0,15 \cdot 25=3,8$
4.Розробка ПЗ	$T_{\text{пр}}=1,5$ $\cdot N_{\text{пр}}=1,5 \cdot 25=37,5$	$T_{\text{за}}=0,7 \cdot N_{\text{пр}}=0,7 \cdot 25=17,57$	$T_{\text{за}}=0,15 \cdot N_{\text{пр}}=0,15 \cdot 25=3,8$
Усього, в т.ч.:	$\Sigma T=181,4$		
- на розробку	$\Sigma T_{\text{р}}=121,8$		
- контроль керівника		$\Sigma T_{\text{за}}=49$	
- нормоконтроль			$\Sigma T_{\text{н}}=10,6$

2.3 Розрахунок ціни програного продукту

У цьому розділі для визначення ціни розраховуємо основну заробітну плату виконавців, матеріальні витрати, вартість машинно – години і витрати на розробку ПО. Розрахунок основної заробітної плати виконавців приведений у таблиці 4.6. Відповідно до статті 8 «Закоу про Державний бюджет України на 2022» встановлено мінімальну заробітну плату у місячному розмірі з 1 січня 2022 року - 6500 гривень; мінімальну погодинну тарифну ставку – 39,26 грн.

Таблиця 2.6 Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість робіт, години	Погодинна тарифна ставка, грн.	Розрахунок, грн.
1.Розробка ПП	121,8	39,26	4781,87
2.Контроль керівника	49	50,00	2450
3.Нормоконтроль	10,6	50,00	530
Усього	-	-	$\Sigma Z_{\text{о}}= 7761,87$

Зм	Дот	% витрат.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Зробимо розрахунок матеріальних витрат на розробку ПП. Розрахунок зведемо в таблицю 2.7

Таблиця 2.7 Розрахунок матеріальних витрат на розробку ПО

Найменування матеріальних витрат	Тип, модель	Кількість	Ціна одиниці, грн.	Вартість, грн.
Папір	Лист А4	80	2,00	160,00
				$B_{м.1} = 160,00$
Транспортно-заготівельні витрати (10%)				$B_{м.2} = 0,1 \times B_{м.1} = 1,60$
Усього				$B_{м.} = B_{м.1} + B_{м.2} = 161,60$

На підставі отриманих даних по окремих статтях витрат складена калькуляція планової собівартості в цілому ПП за формою, приведену в таблиці 2.8.

Таблиця 2.8. Розрахунок статей витрат планової собівартості

Стаття витрат	Значення, грн.	Формула розрахунку
1. Матеріали	176,00	$B_{м.}$ (див. табл. 4.7)
2. Основна заробітна плата	7761,87	Z_0 (див. табл. 4.6)
3. Додаткова заробітна плата	1164,28	$Z_0 = 0,15 \times Z_0 = 7761,87$
4. Відрахування до єдиного фонду соціального внеску	8926,15	$B_{с.с.в.} = 0,22 \times (Z_0 + Z_0) = 0,22 \times (7761,87 + 1164,28)$
5. Накладні витрати	2328,56	$B_{н.к.} = 0,3 \times Z_0 = 0,3 \times 7761,87$
6. Повна собівартість	20356,86	$C_{ном.} = B_{м.} + Z_0 + Z_0 + B_{с.с.в.} + B_{н.к.}$

Розмір прибутку, що включається в ціну, визначаємо по наступній формулі:

$$П = (C_{ном.} * P) / 100 = 20356,86 * 10 / 100 = 2035,68 \quad (2.4)$$

Де P – плановий рівень рентабельності (10-15%).

Оптимальна ціна (кошторисна вартість) визначається по формулі:

$$Ц_0 = C_{ном.} + П = 20356,86 + 2035,68 = 22392,54 \quad (2.5)$$

Податок на додану вартість визначаємо по наступній формулі:

$$ПДВ = 0,2 * Ц_0 = 0,2 * 22392,54 = 4478,51 \quad (2.6)$$

Виходячи з отриманих даних, ціна реалізації розробленого програмного продукту на основі наступної формули, становитиме:

$$Ц_0 = Ц_0 + ПДВ = 22392,54 + 4478,51 = 26871,05 \quad (2.7)$$

Зм	Дор	% вартості	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Арт.

64

3 ОХОРОНА ПРАЦІ

Вирішення завдань охорони праці базується на досягненнях ергономіки, наукової організації праці, технічної естетики, гігієни та фізіології праці, психофізіології. Крім того, успіх охорони праці визначається темпами впровадження передової техніки, підвищення рівня механізації і автоматизації виробничих процесів, удосконаленням технології та організації виробництва.

Умови праці впливають на здоров'я, працездатність і всебічний розвиток особи трудящого.

Виробниче середовище в умовах сучасних методів господарювання характеризується посиленням негативним впливом шкідливих та небезпечних чинників на гігієнічні показники й санітарний стан умов праці, а відтак і на організм людини. Подолання цього явища веде до зниження рівня професійних захворювань, до зміцнення здоров'я працівників, що забезпечується системово-соціально-економічними заходами.

Безпека праці, як галузь практичної діяльності, спрямована на створення небезпечних і нешкідливих умов праці. На сучасному етапі розвитку виробництва вона набуває все більше важливого значення.

Розглянемо умови праці і виконання основних видів робіт при використанні персональних комп'ютерів. Аналіз робіт по виконанню яких, що проектується показує, що в процесі праці на працівника можуть мати дію потенційно небезпечні і шкідливі чинники.

В дипломному розділі дипломного проекту розглядається питання охорони праці програміста на стадії вирішення ним питань розробки програмного забезпечення системи контролю і управління доступом на об'єкт.

3.1 Аналіз небезпечних та шкідливих чинників, що впливають на працівника

На операторів ПК і програмістів можуть мати вплив такі фізичні небезпечні і шкідливі виробничі фактори, як підвищений рівень шуму, підвищена температура зовнішнього середовища, недостатня освітленість

					РП 05.24.003.00 ДП ПЗ	Акт
Зм	Дот	№ докум.	Підпис	Дата		65

робочої зони, електричний струм та інші. Тому на робочому місці програміста повинні бути створені умови для високопродуктивної праці.

3.2 Розробка заходів з охорони праці

До засобів захисту відносять: вентиляцію, штучне освітлення, звукоізоляцію. Також забезпечують сприятливі умови праці естетичні фактори: оформлення виробничого інтер'єра, устаткування, застосування функціональної музики та інше, які впливають на організм людини.

3.2.1 Виробничі приміщення

Приміщення для роботи з ВДТ повинні мати природне та штучне освітлення, відповідно до ДЕН В.2.5-28-2006. У приміщеннях, призначених для роботи з відео терміналами, доцільно, щоб вікна були орієнтовані на північ або північний захід. На вікнах повинні бути штора або жалюзі, що регулюють рівень освітленості і захищають від прямого влучення сонячних променів на робоче місце.

При кольоровому оформленні виробничих і допоміжних приміщень необхідно враховувати орієнтацію їхніх вікон стосовно частин світу і використовувати гармонійне сполучення кольорів.

Для стін і робочих поверхонь використовують мало насичені (основні) кольори, для невеликих помещень або ділянок, що рідко потрапляють у поле зору працівників, а також для створення контрастності – кольори середньої насиченості (допоміжні), для маленьких по площі поверхонь – насичені (акценти) – як функціональне фарбування. Стелі у всіх приміщеннях повинні бути білими. Поверхні устаткування в приміщеннях повинні бути матовими або напівматовими, для виключення випадку відблисків світла в очі працівного, а стіни бути пофарбованими фарбами пастельних тонів.

3.2.2 Мікроклімат робочої зони працівників, вентиляція

У виробничих приміщеннях температура, відносна вологість і швидкість руху повітря на робочих місцях повинні відповідати санітарним нормам мікроклімату виробничих приміщень – ДСанПіН 3.3.2-007-98. Оптимальні параметри мікроклімату у виробничому приміщенні повинні становити:

					РП 05.24.003.00 ДП ПЗ	А см
Зм	Арт	№ докум.	Підпис	Дата		66

- ◆ температура повітря – 22-25⁰С;
- ◆ відносна вологість – 40-60 %;
- ◆ швидкість руху повітряних мас – 0,1-0,2 м/сек.

Для підтримки необхідних параметрів мікроклімату робоче приміщення оснащено системами опалення й кондиціювання.

У приміщеннях, де відбувається робота програміста вимоги до параметрів мікроклімату в цілому виконані.

Для підтримки в приміщеннях нормального, що відповідає гігієнічним вимогам складу повітря, видалення з нього шкідливих газів, пилу використовують вентиляцію. Переважно це припливно-втяжнню. При природній вентиляції (за допомогою вікон) повітря надходить у приміщення і видаляється з нього внаслідок різниці температур і тиску.

3.2.3 Освітлення робочого місця, шум, вібрація

Для штучного освітлення у приміщенні використовуються люмінесцентні лампи типу ЛБ, які в порівнянні з лампами розжарювання мають ряд істотних переваг: за спектральним складом світла вони близькі до природного світла, мають підвищену світлову віддачу (у 2-5 разів вищу, ніж у ламп розжарювання), мають триваліший термін служби – до 10 тис годин. Допускається застосування ламп розжарювання у світлоприймачах місцевого освітлення.

Оптимальні показники рівня шумів у робочих приміщеннях визначаються за ГОСТ 12.1.003-83. Припустимий рівень шуму при розумовій праці, що вимагає зосередженості – 50 дБ.

3.2.4 Електробезпека

Значення сили струму, що проходить через організм людини, залежать від напруги, під якою перебуває людина й від опору ділянки тіла, до якої прикладена ця напруга. Джерелом живлячої напруги є мережа змінного струму з напругою 229В, на яку поширюється ГОСТ 25861-83.

Основними причинами електротравматизму є:

- випадковий контакт до струмоведучих частин, у результаті ведення робіт поблизу або на цих частинах;

					РП 05.24.003.00 ДП ПЗ	Акт
Зм	Дат	№ докум.	Підпис	Дата		67

- Несправність захисних засобів, якими потерпілий доторкався до струмоведучих частин;
- Помилкове прийняття устаткування, що перебуває під напругою, як відключеного;
- Несподіване виникнення напруги через ушкодження ізоляції там, де в нормальних умовах його бути не повинно;
- Контакт струмопровідного устаткування із проводом, що перебуває під напругою.

3.2.5 Організація робочого місця користувача ПК

Конструкція робочого місця користувача персонального комп'ютера має забезпечити підтримання оптимальної робочої позиції працівника. Конструкція робочого столу має відповідати сучасним вимогам ергономіки і забезпечувати оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів.

Висота робочої поверхні робочого столу має регулюватися в межах 680-800 мм, а ширина і глибина – забезпечувати можливість виконання операцій у зоні досяжності моторного поля (рекомендовані розміри: 600-1400мм, глибина – 800-1000мм). Робочий стіл повинен мати простір для ніг заввишки не менше ніж 600мм, завширшки не менше ніж 500мм, заглибини (на рівні колін) не менше ніж 450мм, на рівні простягнутої ноги не менше ніж 650мм. Робочий стілець має бути підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння має бути плоскою, передній край – заокругленим. Регулювання за кожним із параметрів має здійснюватися незалежно, легко і надійно фіксуватися.

3.2.6 Безпека праці під час роботи з персональними комп'ютерами

Щодня перед початком роботи необхідно очищати монітор від пилу та інших забруднень. Після закінчення роботи персональний комп'ютер і периферійні пристрої повинні бути відключені від електричної мережі. У разі

виникнення аварійної ситуації необхідно негайно відключити персональний комп'ютер і периферійні пристрої від електричної мережі. Не допускається:

- виконувати обслуговування, ремонт та налагодження персонального комп'ютеру та периферійних пристроїв безпосередньо на робочому місці оператора;
- зберігати біля персонального комп'ютеру та периферійних пристроїв папір, будь-які носії інформації (диски, флешки тощо), запасні блоки, деталі тощо, якщо вони не використовуються для поточної роботи;
- відключати важливі пристрої, самостійно проводити зміни у конструкції та складі персонального комп'ютеру та периферійних пристроїв або їх технічне налагодження;
- працювати з персональним комп'ютером, у яких під час роботи з'являються нехарактерні сигнали, нестабільне зображення на моніторі тощо;

3.3 Пожежна безпека

Під пожежною безпекою розуміють систему державних і суспільних заходів, спрямованих на охорону від вогню людей і власності. Організаційно-технічними заходами.

Всі приміщення повинні бути забезпечені первинними засобами пожегасіння: пожежним водопостачанням (пожежні крани ПК), пожежні щити з набором пожежного інструменту, Пожежна безпека приміщень, що мають електричні мережі, регламентується ГОСТ 12.1.033-81, ГОСТ 12.1.004-85. Робота оператора ЕОМ повинна вестися в приміщенні, що відповідає категорії Д пожежної безпеки (негорючі речовини й матеріали в холодному стані).

Для гасіння пожеж на початкових стадіях широко застосовується вогнегасники. У виробничих приміщеннях це головним чином бутлекислотні вогнегасники, достоїнством яких є висока ефективність гасіння пожежі, збереження електричного устаткування. Розташовують вогнегасники на видних місцях, на висоті не більше ж 1,5 м від полу.

					РП 05.24.003.00 ДП ПЗ	Лист
						69
Зм	Дот	№ докум.	Підпис	Дата		

ВИСНОВКИ

У результаті виконання дипломного проекту було спроектовано і реалізовано програмне забезпечення системи контролю і управління доступом на об'єкт. З його допомогою можна значно міров полегшити і оптимізувати роботу з охорони та обліку робочого часу співробітників на підприємстві і скоротити витрати на утримання робочого персоналу.

З впровадженням розроблених програмно-апаратних засобів набгато підвищиться дисципліна робочого персоналу, адже працівники не зможуть спізнюватися на роботу або піти з неї раніше, залишаючись непоміченими. Ця обставина сприяє підвищенню продуктивності роботи підприємства, однак при відповідній конфігурації системи можна знизити такі жорсткі рамки, встановивши інший розпорядок робочого часу, наприклад ввести відпрацьовані пропущених годин.

Розроблена система контролю і управління доступом на об'єкт має подальші перспективи розвитку, зокрема – додавання автономних контролерів, здатних зберігати у своїй пам'яті коди карт, що мають права доступу до приміщення і накопичувати статистику про переміщення співробітників. Вартість таких пристроїв буде в кілька разів менше, ніж вартість спеціалізованих комп'ютерів, що встановлюються на автоматизованих прохідних, а для обслуговування всієї мережі потрібен тільки один головний комп'ютер, за допомогою якого будуть виконуватися приєднання до автономних контролерів з метою їх адміністрування (додавання, видалення кодів карт) і збору накопичених статистичних даних для складання звітів. Крім того, нові сучасні пристрої будуть споживати значно менше електроенергії і мати автономні елементи живлення, що робить їх надійніше у разі відключення електроживлення.

Використання створеного програмного забезпечення дозволить перейти на новий етап впровадження високонадійних систем безпеки, в тому числі і для режимних об'єктів.

					РП 05.24.003.00 ДП ПЗ	Акт
						70
Зм	Лист	№ докум.	Підпис	Дата		

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Практическое руководство по программированию / Пер. с англ. Б. Мькс, П. Хит, Н. Рашби и др.; под ред. Б. Мькс, П. Хит, Н. Рашби. – М.: Радио и связь, 2004. – 168 с., ил.
2. Зелкович М., Шоу А., Гэнон Дж. Принципы разработки программного обеспечения / Пер. с англ. – М.: Мир, 2003. – 386 с., ил.
3. Вычислительная техника, алгоритмы и системы управления. – М.: ИНЭУМ, 2015. – 275 с.
4. Четвериков, В.Н. Автоматизированные системы управления предприятиями / В.Н. Четвериков. – М.: Высшая школа, 2015. – 303 с.
5. Ворона В.А., Тихонов В.А. Системы контроля и управления доступом. – М.: Горячая линия – Телеком, 2010. – 272 с.: ил.
6. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли. – М.: Билльямс И.Д., 2017. – 1440 с.
7. Пирогов, В. Информационные системы и базы данных: организация и проектирование: Учебное пособие / В. Пирогов. – СПб: ВИНУ, 2009. – 528 с.
8. Гайдаржи В., Изварин І. Базы данных в информационных системах / В. Гайдаржи – канд. Университет "Україна", 2018. – 418 с.
9. Олифер Н.А. Использование сетевых технологий. – Материалы сайта: http://www.cifonin.ru/operating_systems/sos/glava_4.shtml#_1_4_3.
10. Проектировка, разработка и продажи систем контроля доступа. – Материалы сайта: <http://www.perc.ru>.
11. Материалы сайта: <http://www.lmd.de/download/edownload.html>.
12. Материалы сайта: <http://www.deiphimaster.ru>.
13. Материалы сайта: <http://deiphimworld.narod.ru/>

Зм	Дет	№ докум.	Підпис	Дата

РП 05.24.003.00 ДП ПЗ

Лист

21

ДОДАТОК А. Фрагмент коду головного модулю Monitor

```
function GetProgramPath: String;  
begin  
  GetProgramPath:=ExtractFilePath(ParamStr(0));  
end;  
procedure TForm3.Form.Create(Sender: TObject);  
var  
  pPos, cPos: Integer;  
  s: string;  
begin  
  AssignFile(NoteFile, GetProgramPath+'data'+#13#10#13#10#13#10);  
  if not FileExists(GetProgramPath+'data'+#13#10#13#10#13#10) then  
    begin  
      ForceDirectories(GetProgramPath+'data');  
      Rewrite(NoteFile);  
      _Pos:=0;  
    end  
  else  
    begin  
      CopyFile(pchar(GetProgramPath+'data'+#13#10#13#10#13#10),  
        pchar(GetProgramPath+'data'+#13#10#13#10#13#10), false);  
      Reset(NoteFile);  
      _Pos:=0;  
      if not Eof(NoteFile) then  
        begin  
          Seek(NoteFile, _Pos);  
          Read(NoteFile, NoteData);  
          ShowRecord;  
          if _Pos=0 then Previous.Enabled:=false;  
          Delete.Enabled:=true;  
          if FileSize(NoteFile) >=2  
            then Next.Enabled:=true;  
          end;  
        end;  
      cPos:=0;  
      Seek(NoteFile, cPos);  
      Read(NoteFile, NoteData);  
      s:=NoteData.reg;  
      for i:=0 to FileSize(NoteFile)-1 do  
        begin  
          if s='' then  
            begin  
              ComboBox2.Items.Add(i);  
              break;  
            end;  
          inc(cPos);  
          Seek(NoteFile, cPos);  
          Read(NoteFile, NoteData);  
          s:=NoteData.reg;  
        end;  
      for cPos:=0 to FileSize(NoteFile)-1 do  
        begin  
          Seek(NoteFile, cPos);  
          Read(NoteFile, NoteData);  
          s:=NoteData.reg;  
          if s='' then  
            begin  
              for i:=0 to ComboBox2.Items.Count-1 do  
                if pos(s, ComboBox2.Items.Strings[i])=0
```

```

then j:=2
else
begin
j:=1;
Break;
end;
end;
if j<2 then ComboBox2.Items.Add(j);
j:=0;
end;
ComboBox2.Sorted:=true;
cPos:=0;
Seek(NoteFile,cPos);
Read(NoteFile,NoteData);
s:=NoteData.vlob;
for i:=0 to FileSize(NoteFile)-1 do
begin
if s<>' then
begin
ComboBox3.Items.Add(j);
break;
end;
inc(cPos);
Seek(NoteFile,cPos);
Read(NoteFile,NoteData);
s:=NoteData.vlob;
end;
for cPos:=0 to FileSize(NoteFile)-1 do
begin
Seek(NoteFile,cPos);
Read(NoteFile,NoteData);
s:=NoteData.vlob;
if s<>' then
begin
for i:=0 to ComboBox3.Items.Count-1 do
if pos(s,ComboBox3.Items.Strings[i])=0
then n:=2
else
begin
j:=1;
Break;
end;
end;
if j<2 then ComboBox3.Items.Add(j);
j:=0;
end;
ComboBox3.Sorted:=true;
cPos:=0;
Seek(NoteFile,cPos);
Read(NoteFile,NoteData);
s:=NoteData.osnproc;
for i:=0 to FileSize(NoteFile)-1 do
begin
if s<>' then
begin
ComboBox4.Items.Add(j);
break;
end;
inc(cPos);
Seek(NoteFile,cPos);
Read(NoteFile,NoteData);
s:=NoteData.osnproc;

```

```

end;
for cPos:=0 to FileSize(NoteFile)-1 do
begin
  Seek(NoteFile,cPos);
  Read(NoteFile,NoteData);
  s:=NoteData.p.srprod;
  if s<>' ' then
  begin
    for i:=0 to ComboBox4.Items.Count-1 do
    if pos(s,ComboBox4.Items.Strings[i])=0
    then nj:=2;
    else
    begin
      i:=i;
      Break;
    end;
  end;
  if j=2 then ComboBox4.Items.Add(s);
  j:=0;
end;
ComboBox4.Sorted:=true;
cPos:=0;
Seek(NoteFile,cPos);
Read(NoteFile,NoteData);
s:=NoteData.m.meng;
for i:=0 to FileSize(NoteFile)-1 do
begin
  if s<>' ' then
  begin
    ComboBox5.Items.Add(s);
    break;
  end;
  inc(cPos);
  Seek(NoteFile,cPos);
  Read(NoteFile,NoteData);
  s:=NoteData.m.meng;
end;
for cPos:=0 to FileSize(NoteFile)-1 do
begin
  Seek(NoteFile,cPos);
  Read(NoteFile,NoteData);
  s:=NoteData.m.meng;
  if s<>' ' then
  begin
    for i:=0 to ComboBox5.Items.Count-1 do
    if pos(s,ComboBox5.Items.Strings[i])=0
    then nj:=2;
    else
    begin
      i:=i;
      Break;
    end;
  end;
  if j=2 then ComboBox5.Items.Add(s);
  j:=0;
end;
ComboBox5.Sorted:=true;
pPos:=FileSize(NoteFile); end;
Label4.Caption:=Services.Ne'+IntToStr(L_pos)+' us'+IntToStr(pPos);
end;
procedure TForm3.NewGibA(Sender: TObject);
var

```

```

pPos: Integer;
begin
  ClearData;
  Edit1.SetFocus;
  _Pos:=FileSize(NoteFile);
  Seek(NoteFile,_Pos);
  Button2.Enabled:=false;
  Previous.Enabled:=false;
  Next.Enabled:=false;
  Delete.Enabled:=false;
  Save.Enabled:=true;
  pPos:=FileSize(NoteFile); end;
Label4.Caption:='Strucla Nr '+IntToStr(_Pos)+' '+us'+IntToStr(pPos)+'';
end;
procedure TForm3.SaveClick(Sender: TObject);
begin
  NoteData.org:=Edit1.Text;
  NoteData.adr:=RichEdit1.Text;
  NoteData.tel:=Edit2.Text;
  NoteData.Aadr1:=Edit3.Text;
  NoteData.adr2:=Edit8.Text;
  NoteData.reg:=ComboBox2.Text;
  NoteData.vib1:=ComboBox8.Text;
  NoteData.asnprod:=ComboBox4.Text;
  NoteData.mereg:=ComboBox6.Text;
  Write(NoteFile,NoteData);
  Button2.Enabled:=true;
  Next.Enabled:=false;
  Save.Enabled:=false;
  Delete.Enabled:=true;
  if FileSize(NoteFile) >=2 then
    Previous.Enabled:=true;
end;
procedure TForm3.PreviousClick(Sender: TObject);
var
  pPos: Integer;
begin
  Next.Enabled:=true;
  if _Pos<=0 then dec(_Pos);
  Seek(NoteFile,_Pos);
  Read(NoteFile,NoteData);
  ShowRecord;
  if _Pos=0 then Previous.Enabled:=false;
  pPos:=FileSize(NoteFile); end;
Label4.Caption:='Strucla Nr '+IntToStr(_Pos)+' '+us'+IntToStr(pPos)+'';
end;
procedure TForm3.NextClick(Sender: TObject);
var
  pPos: Integer;
begin
  Previous.Enabled:=true;
  if _Pos=FileSize(NoteFile)-2
  then Next.Enabled:=false;
  inc(_Pos);
  Seek(NoteFile,_Pos);
  Read(NoteFile,NoteData);
  ShowRecord;
  pPos:=FileSize(NoteFile); end;
Label4.Caption:='Strucla Nr '+IntToStr(_Pos+1)+' '+us'+IntToStr(pPos)+'';
end;

```