АНОТАЦІЯ

Тема магістерської роботи «Розробка спеціалізованої підсистеми охоронної безпеки об’єктів».

Актуальність магістерської роботи полягає в необхідності розробки програмного забезпечення для спеціалізованої підсистеми охоронної безпеки об'єктів, яка б виконувала функції віддаленого пульта спостереження та виконувала функцію дублювання передачі тривог на локальний пост охорони цього територіально розподіленого об'єкта.

Об'єкт дослідження – процеси інтеграції та розробки спеціалізованої підсистеми охоронної безпеки об’єктів, що забезпечує швидке реагування локального пульта охорони на тривоги з структурних підрозділів територіально розподілених об'єктів.

Мета роботи – проектування та програмна реалізація спеціалізованої підсистеми охоронної безпеки територіально розподілених об’єктів, що реалізує функції віддаленого робочого місця чергового локального пульта охорони, розташованого безпосередньо на об’єкті та інтеграція підсистеми з програмним забезпеченням системи «Дунай».

Для реалізації поставленої мети були вирішені наступні питання: проведено дослідження та аналіз існуючих систем для пультів централізованого спостереження за безпекою об’єктів; визначені вимоги до інтеграції спеціалізованої підсистеми охоронної безпеки об’єктів з існуючим програмним забезпеченням систем пультів централізованого спостереження; визначені вимоги до розробки спеціалізованої підсистеми охоронної безпеки об’єктів; визначені протоколи взаємодії та формати передачі даних між підсистемою та системою «Дунай», що забезпечило виконання механізму інтеграції; проведено проектування та виконана програмна реалізація спеціалізованої підсистеми охоронної безпеки об’єктів. Практична цінність магістерської роботи полягає в тому, що розроблена підсистема охоронної безпеки об’єктів може бути використана для здійснення нагляду за безпекою територіально розподілених об’єктів безпосередньо на пульті розташованому на цьому об’єкті.

Ключові слова: спеціалізована підсистема, об’єкти охорони, системи охоронної безпеки, програмне забезпечення.

Магістерська робота містить 67 сторінок, 8 таблиць, 21 рисунок, 14 посилань та 13 листів додатку.

SUMMARY

The theme of the master's work is “Development of a Specialized Subsystem for Guarding Security of Facilities”.

The urgency of the master's work is the need to develop software for specialized security subsystem of objects, which would carry out the functions of remote control surveillance and perform function of duplicating transmission alarms to a local security post of that territorially distributed object.

The object of research is processes of integration and development of specialized security subsystem of objects, which provides a rapid response of local remote guard alarm with the structural subdivisions of territorially distributed objects.

The aim of the work is designing and implementation of specialized security subsystem of territorially distributed objects that implements functions of remote workplace of regular local security receiver located directly on object and integration of subsystems with the software system "Danube".

For implementation of the set aim the following issues have been resolved: carried out research and analysis of existing systems for remote centralized security monitoring facilities; defined the requirements for integration of specialized security subsystem of objects with existing software systems of remote centralized surveillance; identified requirements to develop specialized security subsystem of objects; defined interworking protocols and data transfer formats between the subsystem and the system "Danube", which ensured execution of the mechanism of integration; conducted designing and implemented software implementation of specialized security subsystem of the objects. The practical value of the master's work is that the developed security subsystem of the objects may be used to oversee security of territorially distributed objects directly on the panel located at that object.

Keywords: specialized subsystem, protection sites, security systems, software.

The master's work contains 67 pages, 8 tables, 21 pictures, 14 references and 13 letters of appendix.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ 7
ВСТУП 8
1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПУЛЬТІВ ЦЕНТРАЛІЗОВАНОЇ ОХОРОНИ ОБ'ЄКТІВ 11
1.1 Призначення та основні функції систем охоронної сигналізації об'єктів 11
1.2 Принципи функціонування пультів централізованого нагляду за безпекою об'єктів 14
1.3 Огляд та аналіз пультовських охоронних систем централізованого спостереження за безпекою об'єктів 18
1.4 Постановка завдання та вимоги до розробки підсистеми 23
2 ПРИНЦИПИ ФУНКЦІОНУВАННЯ ТА ОСНОВНІ СКЛАДОВІ СИСТЕМИ ПЦН «ДУНАЙ» 26
2.1 Апаратне забезпечення та технічні складові комплексної інтегрованої системи «Дунай» 26
2.2 Функціональні можливості та обладнання ПЦН «Дунай» 28
2.3 Програмне забезпечення ПЦН «Дунай» та його функції інтеграції 33
3 ПРОЕКТУВАННЯ СПЕЦІАЛІЗОВАНОЇ ПІДСИСТЕМИ ОХОРОННОЇ БЕЗПЕКИ ОБ'ЄКТІВ 37
3.1 Визначення функціональних можливостей підсистеми 37
3.2 Вибір та обґрунтування архітектури підсистеми 39
3.3 Середовище розробки інтерфейсу підсистеми 43
3.4 Проектування логічної моделі бази даних підсистеми охоронної безпеки об'єктів 44
3.5 Проектування фізичної моделі бази даних підсистеми 48
3.6 Протокол передачі даних між підсистемою та ПЗ системи «Дунай» ... 53
4 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОХОРОННОЇ БЕЗПЕКИ ОБ'ЄКТІВ 56
4.1 Реалізація інформаційних потреб користувачів системи 56
4.2 Реалізація інтерфейсу користувача-оператора підсистеми 57
4.3 Реалізація спільної роботи спеціалізованої підсистеми з ПЗ «Дунай» . 61
4.4 Реалізація функцій підсистеми 62
4.5 Реалізація обліку подій і тривог у підсистемі 65
ВИСНОВКИ 70
ПЕРЕЛІК ПОСИЛАНЬ 71
ДОДАТОК А ОСНОВНІ КОДИ ПРОГРАМНИХ МОДУЛІВ 72

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ І ТЕРМІНІВ

Скорочення:

АРМ – автоматизоване робоче місце

АТС – автоматична телефонна станція

КІСЦО – комплексна інтегрована система централізованої охорони

ОПС – охоронно-пожежна сигналізація

ПЗ КІСЦО «Дунай» – програмне забезпечення комплексної інтегрованої системи централізованої охорони «Дунай»

ПКП – пультові крайові пристрої

ППК – прибор приймально-контролюючий

ПЦН – пульт централізованого нагляду ПЦС – пульт централізованого спостереження

СПІ – система передачі сповіщень

API – Application Programming Interface – Прикладний програмний інтерфейс

IDE – Integrated Development Environment – інтегроване середовище розробки програм

LCL – Lazarus Component Library – бібліотека візуальних компонентів

ODBC – Open Database Connectivity – програмний інтерфейс, що забезпечує доступ до бази даних

RAD –Rapid Application Development – середовище швидкої розробки додатків

ВСТУП

Охоронна сигналізація має більш ніж вікову історію. Прототипи сучасних систем з'явилися чи не відразу після винаходу електричного дзвінка в першій половині IX століття. Скільки століть існує людство, стільки часу і присутня проблема крадіжки. У різні часи вона вирішувалася по-різному. Але з настанням ери технічного прогресу майже повсюдно стали використовувати технічні засоби захисту. І в першу чергу – охоронні сигналізації. Проте, принцип роботи охоронної сигналізації практично не змінився, хіба що на зміну дзвонику прийшли різного виду охоронні датчики. Встановлені на об'єкті, що охороняється датчики, певним чином налаштовані, при попаданні об'єкта в поле їх дії, видають електричний сигнал, який за допомогою мікропроцесорів і потрапляє на пульт управління різними пристроями. Сповіщувачі бувають різних типів: реагують на рух – інфрачервоні, налаштовані на випромінювання тілом теплових променів. На розбивання скла – акустичні. На підкоп або ж на тряску – вібраційні, а так само суміщені, комбіновані і багато інших. Призначення системи охоронної сигналізації полягає у виявленні несанкціонованого проникнення на об'єкт, що охороняється і формування відповідного оповіщення. Своєчасно інформувати про проникнення сторонніх осіб на територію, що охороняється і сприяти їх затриманню – це завдання систем охоронної сигналізації. Охоронні системи дають можливість вести постійний контроль над обстановкою в житлових, офісних, виробничих приміщеннях, зовнішнього і внутрішнього периметра, незалежно від площі об'єкту, що охороняється, часу доби та присутності охоронного персоналу [1].

Частина процесів будь-якої охоронної сигналізації автоматизована. Це стосується питань виявлення спроб проникнення, обробки сигналу і формування тривожного сповіщення. Однак, існують системи з таким ступенем автоматизації, що їх цілком можна назвати інтелектуальними. Наприклад, однозначно корисною можна вважати опцію автоматичного контролю працездатності системи. Самотестування датчиків і передача інформації про їх стан (працездатності) реалізується в цифрових (адресних) варіантах виконання обладнання. Треба сказати, що такі системи працюють на програмно-апаратному рівні. Наявність програмної складової дозволяє реалізовувати такі інтелектуальні функції як: автоматичне керування обладнанням за заданим розкладом або події; розмежування прав доступу користувачів для роботи з системою; можливість інтеграції сигналізації з

іншими системами безпеки. Більшість сучасних датчиків охоронної сигналізації в своїй роботі використовують алгоритми, що дозволяють аналізувати сукупність факторів, що впливає на сповіщувачі. Це дозволяє значно скоротити кількість помилкових спрацьовувань системи, що підвищить надійність і ефективність [2].

Для забезпечення охоронної безпеки різних об'єктів широко застосовуються автоматизовані засоби охоронного спостереження. Звичайно ці засоби включають встановлені на об'єкті прилади, що одержують інформацію від всіх охоронних сповіщувачів (датчиків), і пульта централізованого спостереження (ПЦС), що знаходиться в організації, що відповідає за охорону об'єкта.

Така структура системи найбільш зручна для клієнтів, що користуються послугами автоматизованої охорони, оскільки дозволяє їм не створювати локальні пульти і не організовувати чергування на самому об'єкті. У разі установки автоматизованої системи охоронної сигналізації на великому об'єкті, наприклад, складах, підприємствах, бізнес-центрах, відомствах, коли вона охоплює велику кількість приміщень, локальний пульт нагляду може застосовуватися спільно з підключенням до пульта централізованого спостереження охоронної організації.

Надходження повідомлень про тривоги на великому розподіленому об’єкті, що має безліч приміщень, безпосередньо особі, що несе чергування на локальному пульті цього об’єкта та одночасне дублювання повідомлень про тривоги на пульті централізованого охоронного спостереження, забезпечує більш швидку реакцію і вживання негайних заходів по запобіганню та виявленню крадіжки. Крім того застосування локального автоматизованого пульту безпосередньо на об’єкті дозволяє значно скоротити кількість виїздів груп реагування при помилкових спрацьовуваннях системи, тим самим підвищив надійність і ефективність охорони на великому розподіленому об’єкті. Програмне забезпечення сучасних пультів охоронної сигналізації мають функцію, яка автоматично проводе фільтрування та відокремлення повідомлень про тривоги конкретного розподіленого об’єкту із загального потоку повідомлень, та може здійснювати синхронне дублювання таких тривог на локальний пульт об’єкту від устаткування по каналам зв’язку.

Метою магістерської роботи є проектування та програмна реалізація спеціалізованої підсистеми охоронної безпеки об’єктів, що реалізує функції віддаленого робочого місця чергового локального пульта охорони,

розташованого безпосередньо на об’єкті. Реалізація такої підсистеми забезпечить реалізацію розподіленої архітектури пульту централізованого спостереження.

1 АНАЛІЗ ТА ДОСЛІДЖЕННЯ ФУНКЦІОНАЛЬНИХ МОЖЛИВОСТЕЙ ПУЛЬТІВ ЦЕНТРАЛІЗОВАНОЇ ОХОРОНИ ОБ'ЄКТІВ

1.1 Призначення та основні функції систем охоронної сигналізації об'єктів

Розробка ефективної схеми охоронної сигналізації, як і будь-який інший системи безпеки, здійснюється з урахуванням поставлених цілей і особливостей кожного конкретного об'єкта. Сучасні технічні охоронні засоби дозволяють забезпечувати мінімальність помилкових спрацьовувань, вибирати найбільш зручні схеми постановки / зняття під охорону і отримання сигналу тривоги, можливість інтеграції з системами контролю доступу і відеоспостереження.

Об'єктове обладнання, що входить до складу системи охорони має елементи (охоронні датчики, контрольна панель, сигнальні пристрої і пристрої управління сигналізацією), що дозволяють встановлювати найбільш ефективні для кожного конкретного об'єкта параметри сигналу тривоги і реакції системи. Різні види датчиків дозволяють враховувати такі фактори, як несанкціонований рух, звук розбитого скла, руйнування. Сигнальні пристрої, в свою чергу, можуть задіяти слухові (звукові сирени) і / або зорові канали (спалахи), а також передавати заздалегідь записаний сигнал тривоги на телефонні номери довірених осіб, посилати SMS-повідомлення. Для постановки і зняття з охоронної сигналізації також можуть використовуватися різні пристрої: сенсорна клавіатура, зручний радіо-брелок, електронні ключі (пластикові картки або ключі touch memory) [3].

Залежно від того, куди надходить сигнал тривоги, охоронні системи підрозділяються на автономні і пультові. У першому випадку сигнал тривоги подається безпосередньо на сигнальні пристрої, розташовані на території об'єкту, що охороняється, а в другому – на пульт централізованого спостереження черговому оператору, який обробляє тривогу та приймає рішення о видах реагування. Одним з елементів комплексного забезпечення безпеки об'єкта є оснащення об'єкта системою охоронної сигналізації. Система охоронної сигналізації складається з [1]:

- набору охоронних датчиків різного принципу дії (контактні, об'ємні, вібраційні і т. д.);
- центрального контролера системи, який займається обробкою сповіщень, що надходять;

– засобів оповіщення (звукових і світлових). Використання датчиків різного принципу виявлення дозволяє звести до мінімуму помилкові сигнали тривог і підвищити ймовірність виявлення спроб проникнення на об'єкт.

До системи охоронної сигналізації можуть бути підключені не тільки охоронні сповіщувачі, але і датчики витоку води, аналізатори газу і інші пристрої побутового призначення. Системи охоронної сигналізації можна умовно розділити на два типи, взявши за основу такий критерій, як спосіб оповіщення про тривогу. Перший тип – автономна система охоронної сигналізації. У такій системі передбачений потужний сигнал тривоги – сирена або ревун, що реагують гучним звуковим сигналом при вторгненні на територію приміщення. При цьому сигнал тривоги не передається ні на пост охорони, ні повідомляється власникові. Такі охоронні системи є пасивними і діють на злочинця чисто психологічно. Хоча цього в більшості випадків буває достатньо. Другий тип системи охорони – активний. Сигнал про проникнення на територію, що охороняється надходить або на стаціонарний пост охорони, або в спеціалізовану охоронну структуру, обладнану пультом [1]. Такі системи зазвичай називають пультова система охоронної сигналізації (рис. 1.1).

Рисунок 1.1 – Схема функціонування пультової системи сигналізації

По всьому об'єкту, що охороняється розташовані датчики, які передають сигнал тривоги на контрольну панель, а звідти сигнал надходить на пульт централізованого спостереження (ПЦС). Цей пульт може збирати сигнали з декількох об'єктів. Охоронна структура в найкоротші терміни зобов'язана прибути на об'єкт і припинити протиправні дії зловмисника (рис 1.2).

Рисунок 1.2 – Складові охоронної системи сигналізації

Тільки в системі пультової охорони є тривожна кнопка. Варто натиснути її, і сигнал тривоги тут же поступає на пульт управління. Ця кнопка в основному розрахована на захист від розбійних нападів і хуліганів. Є різні способи реалізації цієї «кнопки тривоги». Вона може знаходитися під столом співробітника, бути замаскованою під декоративну прикрасу або бути виконаною зовні як брелок для ключів або мобільного телефону. Так само в пультовій охороні, можливо, здійснити захист від «входу під примусом». Це коли зловмисники нападають на власника та змушують силою зняти з охорони сигналізацію. У цьому випадку власник вводить спеціальний код, який нібито знімає об'єкт з охорони, при цьому охоронна система дійсно як би відключається, але потай передає на ПЦС охорони повідомлення про проникнення [2].

Для передачі тривожного сигналу система охоронної сигналізації може бути оснащена таким устаткуванням, як радіопередавач, дозваніватель, GSM-модуль. Радіопередавач передає тривожний сигнал, використовуючи радіоканал. Цей спосіб досить універсальний, але потрібен спеціальний дозвіл на радіочастоту. Та й дальність таких передавачів обмежена рельєфом місцевості і висотою приймаючої антени. До того ж обладнання досить дороге і чутливе до перешкод. Автоматичний дозвон використовуючи дротову телефонну лінію, оповіщає про тривогу. Він передає кодовий тоновий сигнал або заздалегідь підготовлений запис на запрограмовані в нього телефонні номери. Це найнадійніший і найдешевший варіант. Але і у нього є недоліки. Телефонна лінія, на жаль, є не скрізь, та й з ладу її вивести досить просто.

За допомогою GSM-модуля можна оповіщати про тривогу за допомогою коротких sms-повідомлень, використовуючи GSM канал стільникового оператора. Спосіб простий, але не дуже надійний. І в операторів сотового зв'язку бувають збої, до того ж, не в одному тарифному плані, не передбачено гарантованого часу доставки SMS. Останнім часом, з розвитком технологій, до способів передачі тривожного сигналу також зараховують Інтернет. Знаючи ці особливості, багато охоронних фірм пропонують дублювання каналів зв'язку. Це здорожує систему.

Застосування будь-якого з перерахованих варіантів в конкретній схемі має свої переваги і недоліки. Основний недолік використання радіоканалу – велика кількість перешкоджаючих чинників, є ймовірність заглушення сигналу, постановці навмисних перешкод. Основний недолік використання міської телефонної лінії – є ймовірність перерізання телефонного кабелю, не завжди є можливість підключення міського телефонного номера. Основний недолік використання GSM-каналу – оператор мережі не несе відповідальності за якість надання послуг і в разі втрати тривожного сигналу відповідальність несе тільки охоронна організація. Тому найбільш оптимальним є застосування комбінованого варіанту передачі сигналів оповіщення на ПЦС, що забезпечує багаторазове резервування каналів і високу надійність передачі сигналів. Пультова охоронна сигналізація має на увазі повний цілодобовий контроль над приміщенням і всім, що в ньому відбувається. На пульті ведеться постійний облік всіх подій на об'єкті, що охороняється з фіксацією точного часу.

1.2 Принципи функціонування пультів централізованого нагляду за

безпекою об’єктів

Охоронно-пожежна сигналізація (ОПС) – це одержання, обробка, передача і уявлення в заданому вигляді споживачам інформації про проникнення на об'єкти, що охороняються і пожежі на них за допомогою технічних засобів. Споживачем інформації є персонал, на який покладено функції реагування на тривожні і службові повідомлення, що надходять з об'єктів, що охороняються [3].

Об'єктом, що охороняється називається окреме приміщення, яке містить матеріальні чи інші цінності, обладнане технічними засобами ОПС, або комплекс приміщень, розосереджених в межах одного або декількох будівель, об'єднаних загальною територією і охоронюваних підрозділами охорони. Місця можливого проникнення на ГО або окремі зони, що охороняються обладнуються різними сповіщувачами, які включаються в шлейф сигналізації. Повідомленням в техніці ОПС називається повідомлення, що несе інформацію про контрольовані зміни стану об'єкту, що охороняється або технічного засобу ОПВ і передане за допомогою електромагнітних, електричних, світлових і (або) звукових сигналів. Повідомлення діляться на тривожні і службові. Тривожне повідомлення містить інформацію про проникнення пожежі, службове – про «взяття» під охорону, «зняття» з охорони, несправності апаратури та ін.

Комплекс охоронно-пожежної сигналізації – це сукупність спільно діючих технічних засобів охоронної, пожежної і (або) охоронно-пожежної сигналізації, встановлених на об'єкті, що охороняється і об'єднаних системою інженерних мереж і комунікацій [1].

Система передачі сповіщень (СПІ) – це сукупність спільно діючих технічних засобів для передачі по каналах зв'язку і прийому в пункті централізованої охорони сповіщень про проникнення на об'єкти, що охороняються пожежі, засобами службових та контрольно-діагностичних повідомлень, а також для передачі та прийому команд управління.

Пункт централізованої охорони (ПЦО) – це диспетчерський пункт централізованої охорони ряду розосереджених об'єктів від проникнення і пожежі з використанням СПІ.

Призначення системи охоронної сигналізації визначає її загальну структуру, а саме, наявність чотирьох складових системи, що виконують різні функції [1]:

- виявлення несанкціонованого доступу здійснюється автоматичними охоронними сповіщувачами з різними принципами виявлення і різними методами обробки і обміну інформацією;
- обробка інформації, що поступає з сповіщувачів, і видача результатів операторові виконуються центральною станцією і пультом управління;
- виконання, певних дій для сповіщення персоналу і груп швидкого реагування для усунення порушення, виконується центральною станцією;
- швидке і точне реагування груп швидкого реагування і локальних постів охорони.

Всі чотири складових тісно взаємозв'язані між собою, і ефективність роботи системи охоронної сигналізації в цілому залежить від надійності і стабільності роботи кожної її складової [3].

Проте, основну роль при створенні професійних систем охоронної безпеки об'єктів відіграють сповіщувачі. Саме вони повинні забезпечити швидке і надійне виявлення несанкціонованого доступу на об’єкті. Види охоронних сповіщувачів, що можуть бути використані при проектуванні систем сигналізації на об’єкті представлено на рис. 1.3

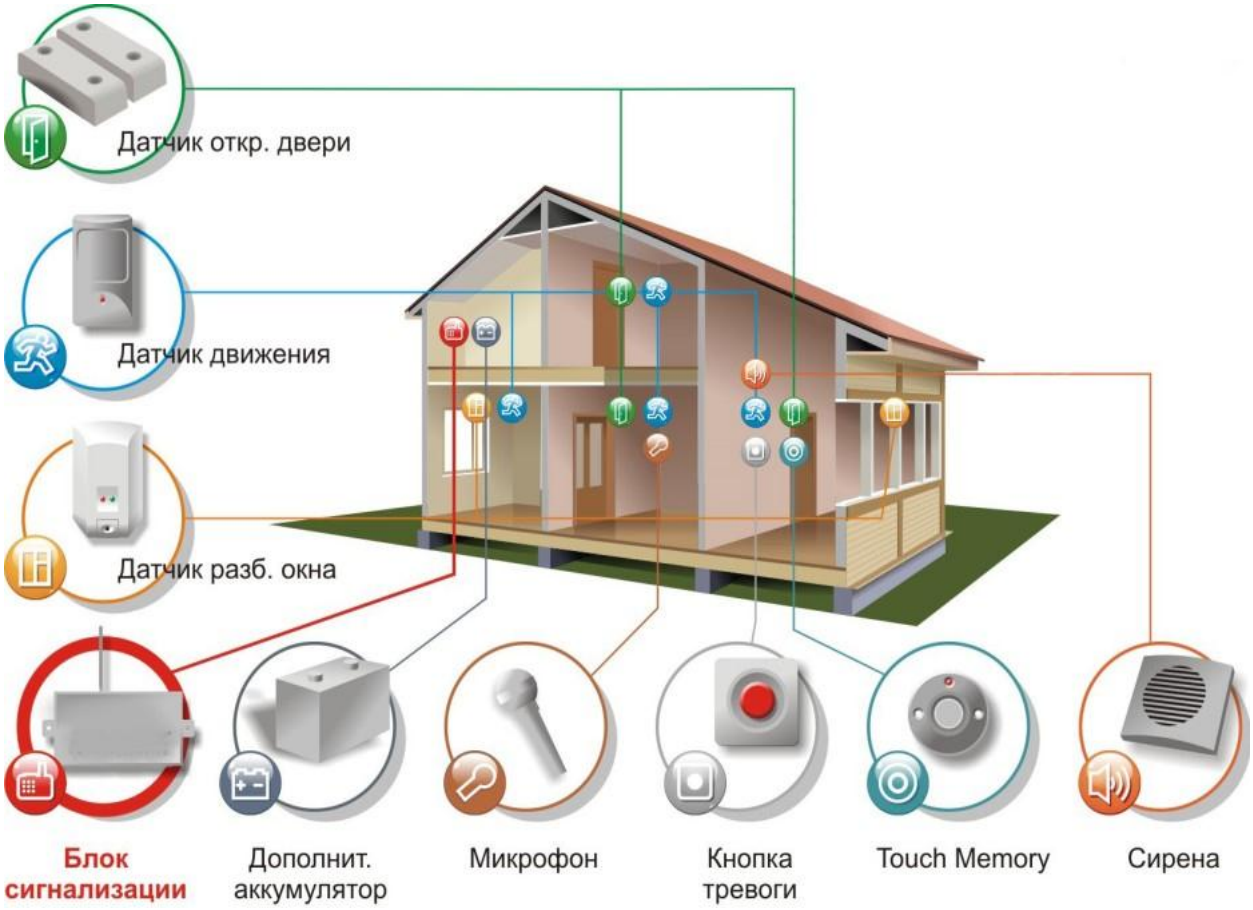

Рисунок 1.3 – Види охоронних сповіщувачів на об’єкті

Технічні засоби (ТЗ) охоронної сигналізації, призначені для отримання інформації про стан контрольованих параметрів на об'єкті, що охороняється, приймання, перетворення, передачі, зберігання, відображення цієї інформації у вигляді звукової та світлової сигналізації та класифікується за двома ознаками: області застосування і функціональному призначенню.

По області застосування ТЗ діляться на охоронні, охоронно-пожежні; за функціональним призначенням – на технічні засоби виявлення (сповіщувачі), призначені для отримання інформації про стан контрольованих параметрів і ТЗ оповіщення, призначені для прийому, перетворення, передачі, зберігання, обробки і відображення інформації.

Автоматизована система зв'язку й оперативного управління охоронного нагляду призначена для удосконалювання діяльності підрозділів швидкого реагування та охорони шляхом автоматизації вирішення завдань управління, прийняття рішень і оптимізації існуючої організаційної структури системи управління і вхідних у неї підрозділів.

Автоматизована спеціалізована підсистема охоронної безпеки об’єктів повинна забезпечувати виконання наступних функцій:

- прийом заявок про несанкціоніроване порушення безпеки чи проникнення на об’єкті, що надходять по каналах зв'язку або безпосередньо від заявників;
- прийому електричних сигналів від систем охоронного спостереження;
- мобілізації спеціалізованих служб та швидке реагування на протиправні дії зловмисниками;
- забезпечення необхідною інформацією груп швидкого реагування;
- забезпечення оперативного зв'язку між персоналом управління охороною об’єктів та групою швидкого реагування;
- збору, хронометрування, сортування, накопичення, збереження, відображення, документування і видачі зацікавленим відомствам відомостей про порушення безпеки на об’єкті.

1.3 Огляд та аналіз пультовських охоронних систем централізованого спостереження за безпекою об’єктів

Сигналізація – звучить просто, але в дійсності пультова охорона являє собою комплекс засобів, як фізичних, так і технічних, які забезпечують охорону об'єктів всіх форм власності. Цілодобово проводиться моніторинг всіх засобів охорони, у разі необхідності реагування, висилається мобільна група, а також оповіщається клієнт. При необхідності постановки об'єкта під пультової охорону, будівля комплектується датчиками. Датчики бувають різних типів: внутрішні та зовнішні, на відкривання вікон і дверей, руху, розбиття скла і з імунітетом на тварин. На центральний пульт надходять всі види тривог [4]. При спрацьовуванні датчиків тривоги, формується пакет інформації, який передається на централізований диспетчерський пульт охорони (рис. 1.4).

Рисунок 1.4 – Схема розташування пультів централізованого нагляду

Пультова охорона складається з наступних компонентів [2]:

- централізований диспетчерський пульт охорони;
- приймально-контрольний прилад;
- датчики;
- мобільні екіпажі реагування.

За типом сигналу тривоги диспетчер визначає тип спрацювання і приймає рішення щодо реагування. При необхідності, диспетчер відправляє на об'єкт групу мобільного реагування, яка прибуває на місце протягом 5 – 7 хвилин і затримує зловмисника.

Диспетчерські пульти централізованої охорони розміщені у всіх великих містах України. При виконанні магістерської роботи виконано дослідження та аналіз сучасного стану використання пультів централізованого нагляду за безпекою об’єктів різними охоронними структурами України.

Першою розглянута система пульта централізовано спостереження – Система передачі тривожних сповіщень (СПТС). Призначена для реалізації охоронного моніторингу (ПЦС) на об'єктах різного розміру і складності. СПТС «МОСТ» забезпечує прийом, обробку та відображення інформації про стан об'єктів, пультового і станційного обладнання. СПТС «МОСТ» підтримує наступні канали зв'язку з об'єктовими ППК: GSM-канал, формат GPRS; GSM-канал, формат CSD; Ethernet-канал.

СПТС «Мост» можна розділити на три основні частини [1]:

- ПЦС «Мост» з встановленим програмним забезпеченням «Мост»;
- додаткове пультове обладнання для ПЦС «Мост» (модеми, приймачі);
- об'єктові прилади серії «Оріон» і додаткове обладнання для них – ретранслятори і мультиплексори т.п.;

Серед особливостей програмного забезпечення пульта централізованого нагляду за безпекою об’єктів можливо відзначити наступне:

- мережева структура з можливістю реалізації до 12 робочих місць;
- резервування бази даних на будь-якому ПК, який входить в мережу;
- збір статистики по об'єктах, які обслуговуються на будь-якому з робочих місць;
- можливість обслуговування до 16 COM-портів, для підключення пультового обладнання (модеми, приймачі, передавачі і т.д.).

Програмне забезпечення «Мост» для пультів централізованого нагляду за безпекою об’єктів є комерційним програмним продуктом, яке необхідно придбати у компанія ТОВ «Тірас-12», яка є відомим українським виробником систем протипожежного та охоронного призначення.

Наступною системою пульта централізованого спостереження для забезпечення охорони об’єктів розглянуто програмного забезпечення «ФЕНІКС-4». Слід відзначити про надзвичайну гнучкість в конфігурації системи. Передбачено можливість фізичного підключення приймальних пристроїв до одного комп'ютера, запуск програми для операторів на іншому комп'ютері, – при цьому сама база даних може знаходитися на третьому, а інженер пульта (редактор бази даних) – на четвертому комп'ютері. Подібна конфігурація, крім зручності в роботі, сприяє підвищенню відмовостійкості системи в цілому [1]. Вигляд системи пульта централізованого забезпечення «ФЕНІКС-4» представлено на рис. 1.5.

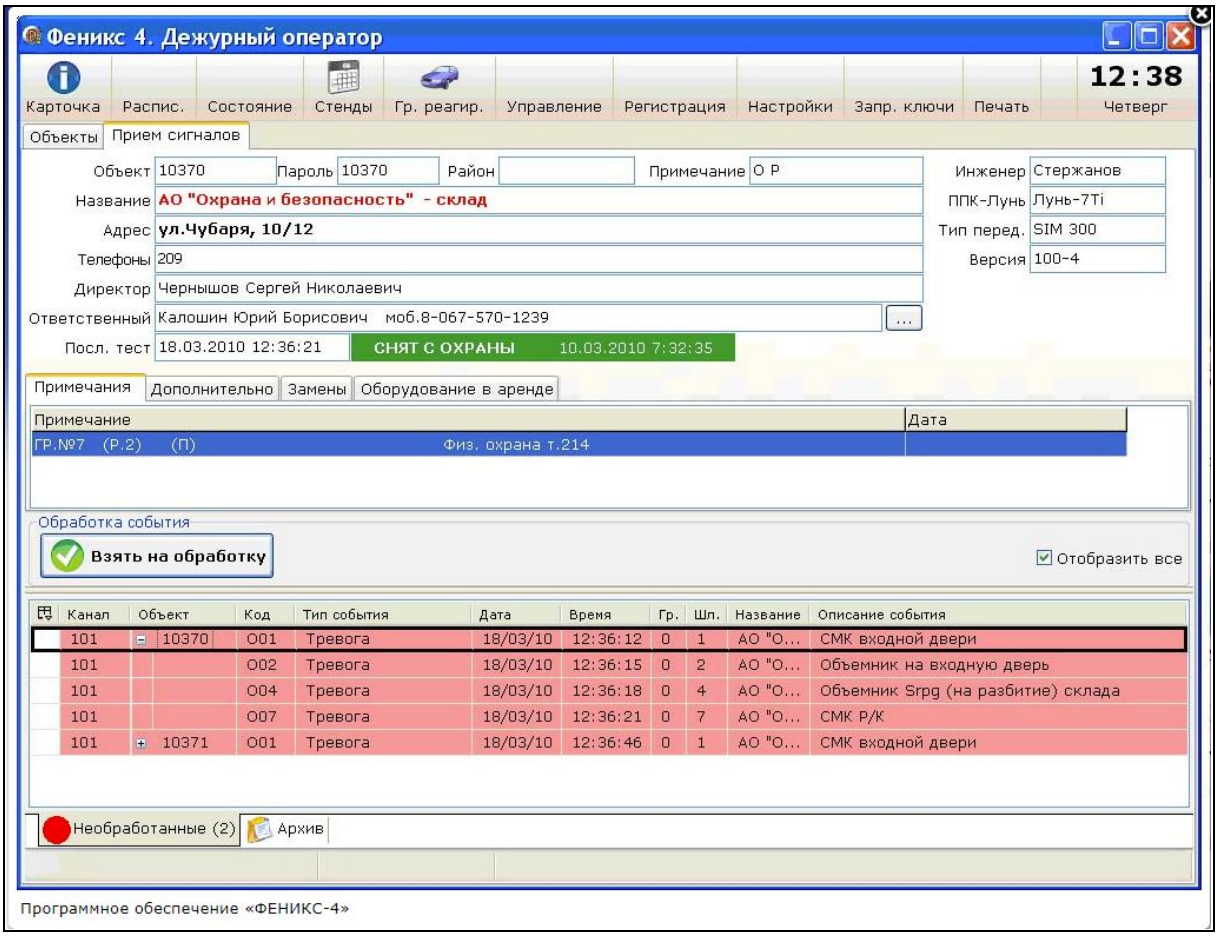

Рисунок 1.5 – ПЗ пульта централізованого спостереження «ФЕНІКС-4»

До основних переваг системи можливо віднести наступні:

- мережева модель обробки операторами подій, що надходять: кожна подія відображається відразу на всіх робочих місцях;
- обробка події проводиться в кілька етапів з реєструванням часу на кожному етапі: подія надійшла, подія взято на обробку оператором на конкретному комп'ютері, робота з об'єктом, група реагування вислана, прибуття групи реагування або скасування виклику, вказівка причини тривоги (як вибір зі списку, так і можливість ручного введення), подія оброблено (в архів);
- буфер тривог з угрупованням за типом і по об'єкту (при багаторазових спрацьовування);
- всі тривоги відображаються на всіх робочих місцях операторів із зазначенням етапу їх обробки в даний момент часу;
- для створення об'єкта використовується майстер з контекстної допомогою;

- створення в базі даних груп реагування, стеження за виїздами на тривогу, на СТО, на заправку;
- службова інформація прихована від користувача, наприклад, коди подій, протокол від централей та інше.

Програмне забезпечення «ФЕНІКС-4» для пультів централізованого нагляду за безпекою об’єктів є комерційним програмним продуктом, яке необхідно придбати у українського виробника АО «Охрана и Безопасность».

В ході виконання магістерської роботи було розглянуто один з найбільш використованих охоронними компаніями України систем пультів централізованого спостереження (ПЦС) «Дунай». ПЦН «Дунай» – це цілий апаратний комплекс програмного і технічного забезпечення, який забезпечує цілодобову охорону об'єктів, від виробника компанії «Венбест». Програмне забезпечення Дунай, використовує більш ніж 500 охоронних фірм.

Принцип роботи пульта централізованого спостереження «Дунай» забезпечує виконання наступних функцій [4]:

- збір і обробку сигналів, що надходять від об'єктів, формування і видачу на екран пульта централізованого спостереження тривожної, службової, довідкової та графічної інформації;
- передачу інформації по лініях і каналах зв'язку: радіоканал, канал зв'язку GSM / GPRS, телефонна лінія – на сьогоднішній день найнадійніший канал зв'язку, через провідний Інтернет.
- автоматичне взяття (зняття) об'єктів під спостереження за допомогою електронного замка, клавіатури або електронної картки з видачею користувачеві відповідного підтвердження;
- можливість дублювати інформацію по інших каналах зв'язку;
- високі діагностичні властивості блоків і вузлів системи;
- повну інформативність, достовірність і надійність інформації, що передається;
- резервування всіх вузлів і блоків системи від знеструмлення мережі живлення 220 В;
- повне збереження оперативної інформації в незалежній пам'яті приладів при будь-яких позаштатних ситуаціях в системі.

ПЦС «Дунай», виробництва НВФ «ВЕНБЕСТ ЛТД» забезпечує цілодобовий прийом інформації з охоронюваних об'єктів та обробку і передачу тривог групам мобільного реагування (ГМР) для реагування. Система ПЦС «Дунай» забезпечує ведення аудиту подій в системі для

подальшого аналізу причин, якості реагування на події, стану, як складових частин, так і працездатності системи в цілому.

Для розгортання системи пульта централізованого спостереження «Дунай» повинно забезпечити наявність наступних складових [4]:

- персональний комп'ютер для організації сервера БД, сервера драйверів і терміналів користувачів, мережеве обладнання (якщо ПК більше 2-х);
- до 24-х ПОУ в будь-якому поєднанні:
- радіомультіплексор «ВБД6-13» – для організації роботи радіомережі на виділеній фіксованій частоті (до 1024 ППК);
- комутатор пакетів «КОП1» – для організації мережі з 1024 ППК або 49 аналогічних «КОП1» по телефонних лініях зв'язку міської телефонної мережі (ГТС);
- стандартний телефонний модем для організації зв'язку по телефонних лініях зв'язку ГТС з віддаленим ретранслятором «Дунай-Р1000», до якого можуть бути підключені ретранслятори «Дунай-Р», «SPIN ATU» або «Каштан»;
- стандартний GSM-модем або модем «Радіо СРП» – для прийому / передачі повідомлень через мережу GSM або пакетну радіомережу bkcNET».
- мультиплексор GSM «Дунай-GMX1» – для контролю каналу зв'язку і прийому / передачі повідомлень через мережу GSM;
- мультиплексор «Дунай-ECOM» – для транслювання IP-пакетів, контролю каналу зв'язку і прийому / передачі повідомлень через мережу GPRS / Інтернет;
- програмне забезпечення ПЦС «Дунай-ПРО»;
- принтер, джерела безперебійного живлення для комп'ютерів і пультових кінцевих пристроїв.

1.4 Постановка завдання та вимоги до розробки підсистеми

Системи охоронної сигналізації на територіально розподілених об'єктах, підключені до пульта централізованого спостереження, вимагають збільшення швидкості реагування на тривоги. Доцільно передавати тривоги з великих об'єктів не тільки на міські пульти централізованого спостереження, а й на локальні пости чергових на цих об'єктах. Розгортання на таких об'єктах локальних пультів вимагає великих витрат, тому що пульт централізованого

спостереження – це комплекс як програмного забезпечення так і апаратно-технічних засобів. Тому стає актуальною задача розробки спеціалізованої підсистеми охоронної безпеки для розподілених об'єктів, яка б виконувала функції віддаленого пульта спостереження і була інтегрована в систему централізованого спостереження «Дунай», виконуючи функцію дублювання передачі тривог на локальний пост охорони цього об'єкта (рис. 1.6).

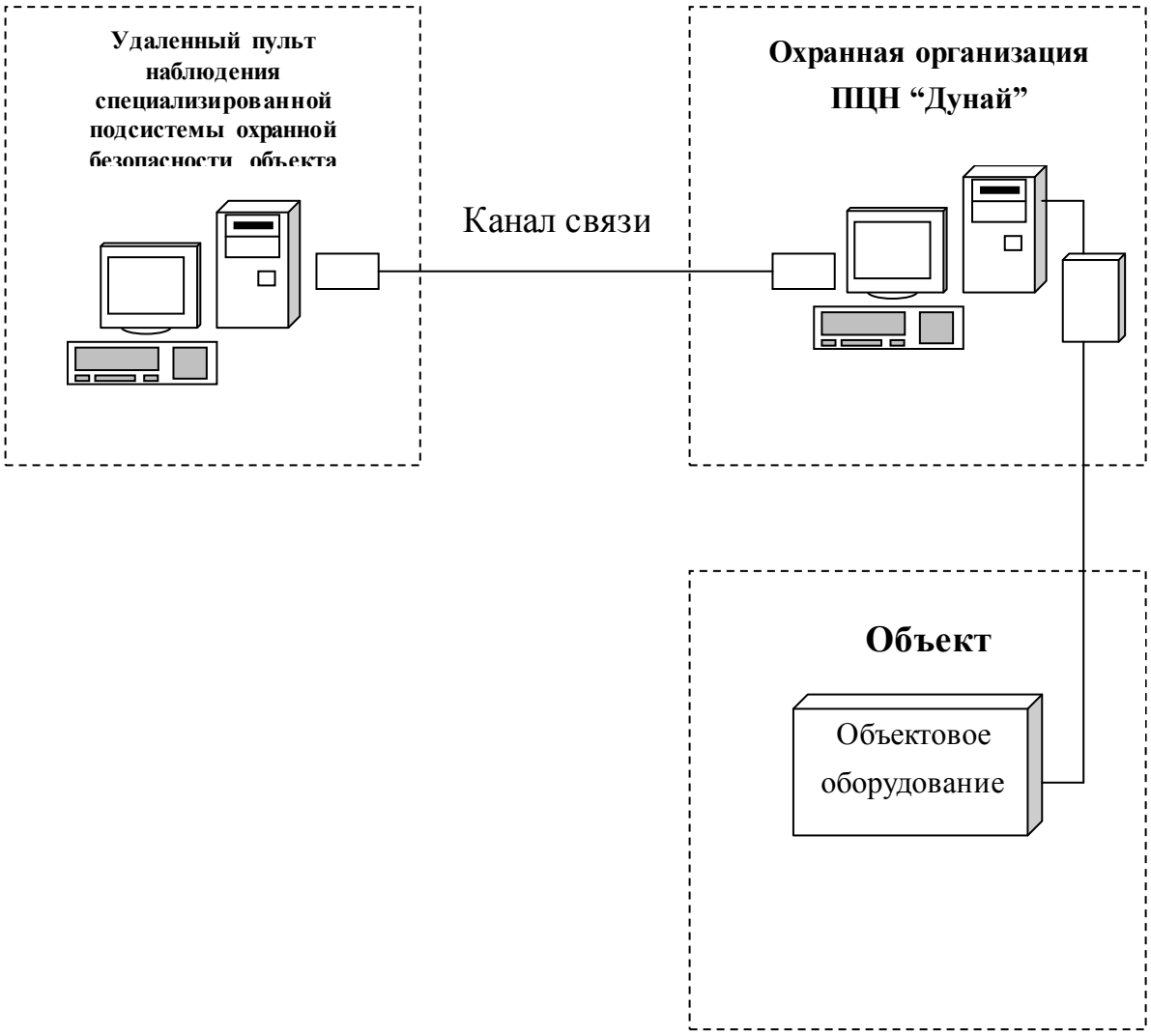

Рисунок 1.6 – Структурна схема функціонування спеціалізованої підсистеми

При такій структурі реалізації підсистеми, повідомлення про тривоги з територіально розподілених об'єктів, надходять на пульт централізованого спостереження, і повинні бути передані безпосередньо черговому посту охорони цього об'єкта на віддалене робоче місце. Програмне забезпечення системи «Дунай» має вбудовану функцію автоматичної фільтрації тривог по конкретно обраному об'єкту із загального потоку тривог. Основною

функцією спеціалізованою підсистеми охоронної безпеки об'єктів, що розробляється, є реалізація функцій передачі тривог від ПЗ системи ПЦН «Дунай» на віддалений пост охорони територіально розподіленого об'єкта.

Підтримка віддаленого робочого місця є подальшим розвитком КІСЦО «Дунай», дозволяючи розширювати систему, додаючи в неї нове апаратне або програмне забезпечення, яке виконує нові функції і взаємодіє з уже функціонуючим ПЗ та обладнанням через стандартні протоколи.

Програмне забезпечення спеціалізованої підсистеми охоронної безпеки територіально розподілених об'єктів, що реалізує функції віддаленого робочого місця чергового поста охорони об'єкту, має відповідати двом основним вимогам:

- взаємодіяти з пультом централізованого спостереження «Дунай» за протоколом, який визначає формат передачі повідомлень про тривоги;
- забезпечити черговому посту охорони об'єкта можливість максимально оперативно і безпомилково реагувати на повідомлення, що приходять, видаючи йому всю необхідну інформацію о структурі підрозділів об'єкта, з якого надійшла тривога.

2 ПРИНЦИПИ ФУНКЦІОНУВАННЯ ТА ОСНОВНІ СКЛАДОВІ СИСТЕМИ ПЦН «ДУНАЙ»

2.1 Апаратне забезпечення та технічні складові комплексної інтегрованої системи «Дунай»

Сучасна практика показує, що найбільш перспективним і раціональним шляхом організації охорони є застосування інтегрованих систем, які включають в себе кілька напрямків технічних засобів, що взаємодіють між собою в єдиній базі даних або в складі єдиного комплексу, що складається з декількох підсистем. Однією з таких інтегрованих систем, успішно застосовується в підрозділах охорони з 1991 року, стала інтегрована система централізованої охорони «Дунай» [4].

Система «Дунай» складається з:

- пульта централізованого спостереження;
- пультових крайових пристроїв;
- програмного забезпечення;
- устаткування, що ретранслює інформацію від об'єктових приладів на пульт централізованого спостереження;
- об'єктових приладів з різними протоколами функціонування;
- сервісного устаткування.

Принципи побудови КІСЦО «Дунай» [5] дозволяють користувачу дістати дуже великі можливості по конфігурації комплексу технічних засобів безпеки. Пульт централізованого спостереження – це ланка структури, призначена для збору і обробки інформації тієї, що поступає від інших ланок. У системі «Дунай» пульт централізованого спостереження, звичайно, є ПЕОМ, до якої, через інтерфейси RS-232 підключаються пультові крайові пристрої (ПКП) різних типів, які призначені для роботи з устаткуванням іншої ланки структури.

Архітектура комплексної інтегрованої системи централізованого спостереження «Дунай», як системи збору і обробки інформації, в переважній більшості є класичною триланковою структурою (рис. 2.1):

- ПЦС (пульт централізованого спостереження);
- АТС (автоматична телефонна станція);
- об'єкт.

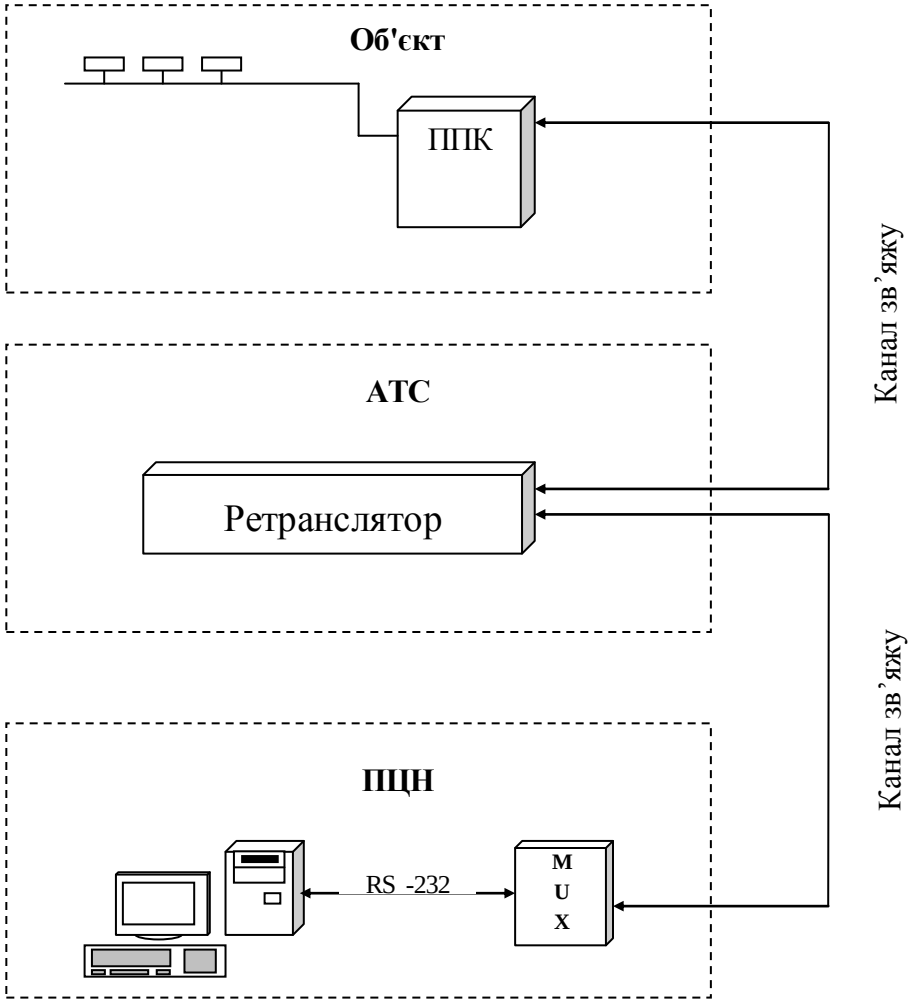

Рисунок 2.1 – Архітектура комплексної інтегрованої охоронної системи централізованого спостереження

Об'єкт – це ланка структури КІСЦО «Дунай» призначена для безпосереднього збору інформації з датчиків про їх стан і для здійснення передачі цієї інформації іншим ланкам.

АТС – це ланка структури, призначена для передачі інформації від різноманітного устаткування ланки «Об'єкт» до устаткування ланки ПЦС. В деяких випадках дана ланка може бути відсутньою. При територіальному розподілі устаткування «Об'єктів» устаткування «АТС» виконує роль локального об'єднання, збору і трансляції інформації на ПЦС.

ПЦН – пульт централізованого спостереження «Дунай» це цілий апаратний комплекс програмного і технічного забезпечення, який забезпечує цілодобову охорону об'єкта.

Дунай» окремо або в складі комплексу дозволяє вирішувати завдання не тільки охорони і пожежної безпеки, а й здійснювати спостереження за станом самих об'єктів охорони, запобігаючи аваріям і техногенним подіям.

Структура інтегрованої системи «Дунай» така, що один пульт охорони, створений на її основі, забезпечує охорону (моніторинг) невеликого або великого об'єкта і безлічі об'єктів одночасно. ІС «Дунай» може впроваджуватися послідовно, починаючи з однієї необхідної підсистеми [4].

Моніторинг проводиться з будь-якої кількості автоматизованих робочих місць (АРМ), встановлених в приміщеннях служби безпеки, технічного персоналу, що супроводжує і обслуговує систему, а також на «віддалених АРМ», встановлених на будь-якій відстані від пульта охорони.

Використання сучасних каналів зв'язку і передачі даних спільно з обладнанням «Дунай» дозволяє відмовитися від стандартних принципів надання охоронних послуг, розширюючи можливості реалізації. Схеми організації охоронних комплексів можуть ґрунтуватися на об'єктової системі передачі сповіщень.

Залежно від кількості (місткості) контрольованих об'єктових приладів змінюється побудова пультів централізованого спостереження і застосування на ньому того або іншого устаткування.

З міркувань економічної доцільності пульти малої місткості (до 200 – 300) об'єктів будуються на базі контрольної панелі SPIN CP. Такі пульти можуть працювати на вибір, або по радіоканалу, або по телефонних лініях [4].

При збільшенні місткості пульта він будується на базі персональної ЕОМ. При цьому кількість автоматизованих робочих місць ПЦС може змінюватися від одного шістнадцяти.

Програмне забезпечення комплексу дозволяє гнучко розподілити повноваження автоматизованих робочих місць з метою виконання різних функцій операторами системи «Дунай».

2.2 Функціональні можливості та обладнання ПЦН «Дунай»

Одним з найважливіших елементів системи «Дунай» є об'єктове устаткування – приймально-контрольних приймач (ППК) або централі. Це пояснюється декількома причинами. ППК – це прилади, з якими мають справу безпосередньо користувачі, тому вони повинні, володіючи всією необхідною функціональністю, бути достатньо простими для використання і настройки користувачем; по-друге, саме від цього приладу залежить ухвалення рішення або про реальне проникнення на об'єкт, або про просте відхилення параметрів шлейфів сигналізації від норми; по-третє, саме

об'єктовий прилад на 99% визначає вартість всієї системи в цілому, оскільки це наймасовіший прилад в системі централізованого спостереження [5].

У системі «Дунай» можуть бути використані різні види ППК, сигналізації, що відрізняються по максимальній кількості шлейфів, що підключаються, часу роботи від резервного джерела живлення за відсутності або неправильних параметрах напруги побутової електромережі, можливості роботи в мережі з іншими ППК для забезпечення охорони великої кількості приміщень, ціні.

Якщо не вимагається підключення до системи «Дунай» вже наявного об'єктового устаткування, на об'єктах звичайно встановлюються сучасні прилади «ВБД-4», «ВБД-6», «Дунай», «SPIN VB» [5].

Будь-який прилад приймально-контрольний (ППК) виконує наступні функції [4]:

- збір інформації від датчиків, що знаходяться на шлейфах, підключених до ППК і формування сповіщення про тривогу при відхиленні сигналу від стандартних параметрів;
- передача сповіщень про тривоги на пульт централізованого спостереження через устаткування зв'язку (ретранслятор, мультиплексор каналів) або на локальний пульт спостереження; електроживлення від мережі змінного струму напругою 220 В і автоматичне перемикання на роботу від встановленого в корпус ППК резервного джерела (акумулятора) у разі неможливості роботи від мережі.

Конфігурація і режими роботи ППК програмуються за допомогою спеціальної клавіатури або персонального комп'ютера.

Додатковими можливостями для приладів приймально-контрольних є: можливість підключення інших ППК для організації мережі на об'єкті, що охороняється, подача сигналу на світлові і звукові оповісники у разі виникнення тривоги, виведення повідомлень про зафіксовані ППК події на принтер, що підтримує протокол Centronics.

Відкрита архітектура комплексної інтегрованої системи централізованої охорони «Дунай» дозволяє використовувати різне устаткування і схеми його підключення, але найчастіше застосовується стандартна схема з трьох ланок.

Розглянемо технічне обладнання необхідне для функціонування ПЦС на базі програмного забезпечення «Дунай», на базі GSM / GPRS каналу зв'язку.

Для такого пульта необхідний конвертор ECOM 18 (мультиплексор призначений для транслювання IP-пакетів від ППКОП через драйвер P1000, через RS-232 інтерфейс на пульт централізованого спостереження «Дунай». Потрібно підключити Інтернет-кабель в маршрутизатор ECOM 18 , а вже сам провід з роз'ємом DB-9 потрібно підключити в COM порт комп'ютера. Необхідним обладнання є маршрутизатор DRY TEK VIGOR, який забезпечить дублювання каналів зв'язку [4]. В разі якщо один канал зв'язку Інтернет пропадає, маршрутизатор автоматично перемикається на резервний канал звязку і таким чином забезпечується надійність функціонування ПЦС: на вхід маршрутизатора подається два канали зв'язку з двох різних Інтернет-провайдерів (рис. 2.2).

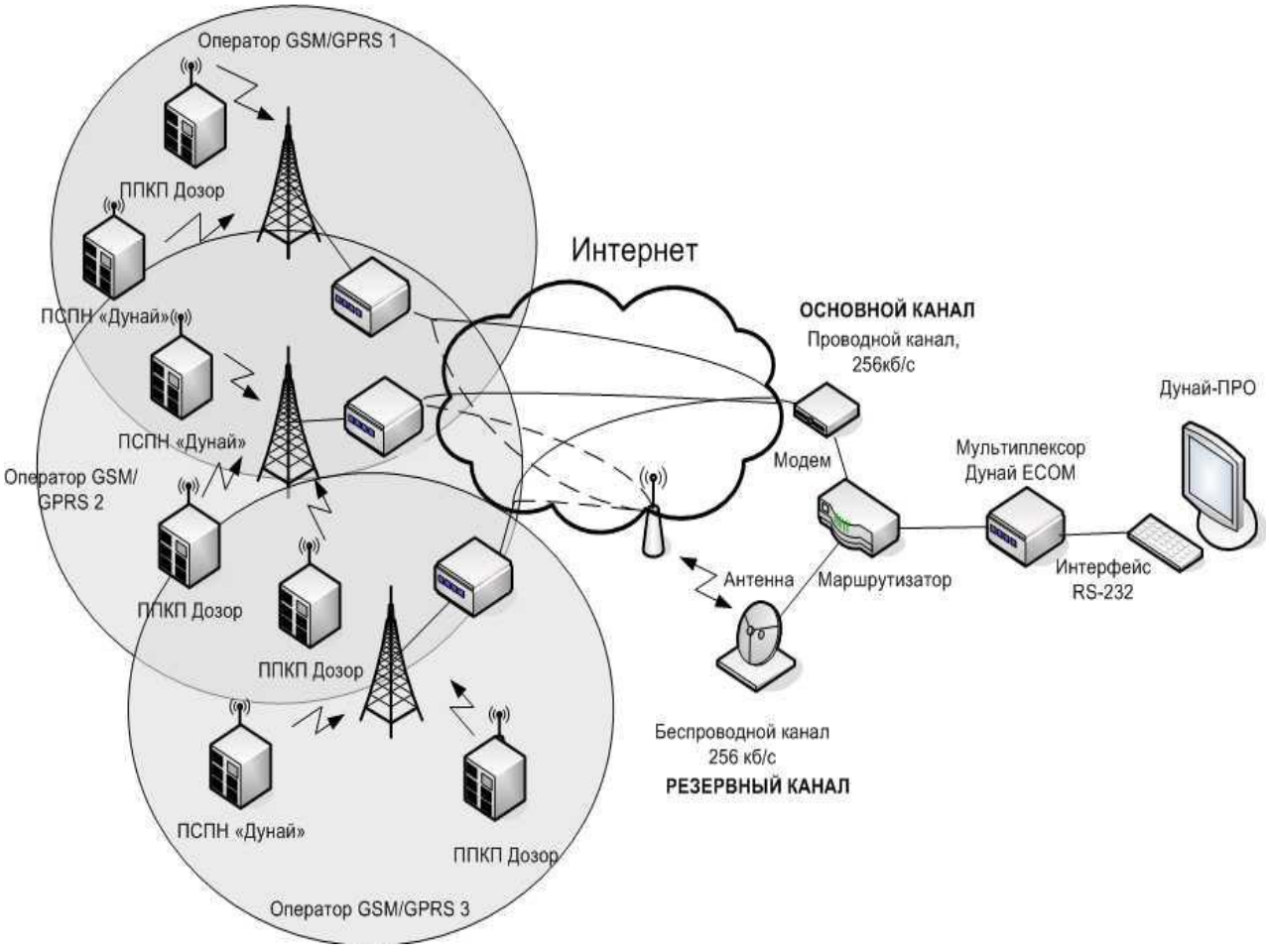

Рисунок 2.2 – Схема функціонування ПЦН «Дунай» на базі GPRS-зв'язку

Відкрита архітектура комплексної інтегрованої системи централізованої охорони «Дунай» дозволяє використовувати різне устаткування і схеми його підключення.

У цій схемі як пультові крайові пристрої, сполучені з комп’ютером, використовуючи мультиплексор, до якого по виділених лініях зв'язку можна підключити до восьми комплектів ретрансляторів. До кожного комплекту ретрансляторів можливо підключити по зайнятій телефонній лінії до 512 різноманітних приладів встановлюваних на об'єктах [5].

Крім стандартної схеми підключення устаткування до комплексної інтегрованої системи централізованої охорони, можливі інші способи організації зв'язку з пультом централізованого спостереження залежно від вживаних пристроїв і їх функціональних можливостей [4]:

- ретранслятор SPIN ATU може використовуватися як пультовий крайовий пристрій без застосування при цьому мультиплексора SPIN MUX. В цьому випадку ланка АТС виключається, а об'єктове устаткування, як і в стандартній схемі, по зайнятих або виділених лініях зв'язку підключається до ретранслятора;
- якщо на об'єкті встановлені ППК SPIN URB, здатні працювати по радіо каналу, інформація може передаватися з них на такий же прилад, встановлений на пульті централізованого спостереження;
- система «Дунай» може працювати з пакетною радіомережею. При цьому як апаратура, що утворює канал зв’язку застосовується модем РАДИО-СРП. Як об'єктове устаткування можуть використовуватися різні ППК, що забезпечують взаємодію з модемами РАДИО-СРП в протоколі Х.25. Така схема зазвичай застосовується на місцевості де вже розгорнена пакетна радіомережа.

Мультиплексор каналів SPIN MUX призначений для сполучення ПЕВМ пульта централізованого спостереження по стику RS-232 з одного боку і каналів зв'язку, що сполучають до восьми видалених ретрансляторів з іншого боку. Зв'язок між мультиплексором і ПЕОМ здійснюється із швидкістю 9600 бит/с. Мультиплексор може бути підключений кабелями до двох комп'ютерів, і перемикання для роботи з вибраним комп'ютером здійснюється у ручному режимі. Спосіб передачі повідомлень по телефонному каналу – амплітудна модуляція, швидкість передачі сповіщень 200 бод. Прилад SPIN MUX приймає від ретрансляторів пакети про стан об'єктів, що охороняються, і встановлені на них технічні засоби охорони і передає їх в ПЕВМ пульта централізованого спостереження. Також він передає команди, що здійснюють управління з ПЦС на ретранслятор. Цей прилад здійснює контроль за справністю каналів зв'язку на ділянці

«ретранслятор – ПЦС». У разі порушення зв'язку з ретранслятором мультиплексор SPIN MUX передає на ПЕОМ пульта централізованого спостереження спеціальне сповіщення про це, а у разі порушення зв'язку з ПЕВМ – включає вбудований звуковий сигнал, що оповіщає черговий персонал про несправність [4].

Як проміжна ланка в структурі передачі сповіщень з об'єкту на пульт централізованого спостереження, при використанні телефонних ліній, застосовуються ретранслятори SPIN ATU. Вони встановлюються на АТС або на об'єктах при організації локальних ПЦС. Ретранслятор призначений для сполучення каналів зв'язку, що сполучають видалені ППК з одного боку, з мультиплексором SPIN MUX з іншого боку; реалізації протоколу автоматизованої і ручний тактик охорони на ділянці «ретранслятор – об'єкт»; прийому–передачі сповіщень від ППК, встановлених на об'єктах; прийому–передачі пакетів від (на) ПЦН (мультиплексор SPIN MUX); контролю за справністю каналів зв'язку на ділянці «ретранслятор – об'єкт» і видачі на пульт централізованого спостереження сповіщень про несправність; забезпечення оптимізації часу циклу опиту залежно від кількості приписаних з ПЦС приладів приймально-контрольних.

Функціонально ретранслятор є універсальним приладом, обслуговуючим до 512 телефонних ліній (напрямів). Конструктивно до одного корпусу ретранслятора може бути підключено до 64 телефонних ліній (напрямів). Місткість ретранслятора збільшується до максимальної шляхом установки додаткових корпусів ретрансляторів (без модулів управління) і з'єднання їх з першим ретранслятором стрічковим кабелем, що входить в комплект. Знов встановлені ретранслятори працюють під управлінням першого ретранслятора [5].

Окрім підключення до мультиплексора SPIN MUX по виділеній лінії зв'язку, ретранслятор SPIN ATU може бути підключений:

- до мультиплексора системи «Каштан» по виділеній лінії зв'язку;
- до ПЕОМ ПЦС через модеми «Радіо СРП»;
- до ПЕОМ ПЦС через модеми типу ZYXEL, MOTOROLA, MULTITECH і інші по виділених або зайнятих лініях зв'язку;
- до радіо-мультиплексора SPIN PCO (використовуючи радіо-НС) або до ПЕОМ ПЦС через послідовний порт RS-232 на відстань до 15 м. При використанні «конвертерів» сигналів стику RS-232 – до 1 км.;

– одночасно до мультиплексора SPIN MUX або мультиплексору системи «Каштан» по виділеній лінії зв'язку і через послідовний порт RS-232 до ПЕОМ ПЦС або контрольної панелі SPIN CP.

2.3 Програмне забезпечення ПЦН «Дунай» та його функції інтеграції

Сучасний розвиток і розповсюдження комп’ютерної техніки зумовили використання сучасної комп’ютерної техніки для реалізації пультів централізованого спостереження великої місткості, оскільки вона володіє великим об'ємом постійної (дискової) і оперативної пам'яті для зберігання всієї необхідної інформації, зручним графічним пристроєм виведення інформації, при цьому всі функції, передбачені вимогами до пульта централізованого спостереження великої місткості реалізуються програмним забезпеченням, що дозволяє досягти найбільшої гнучкості системи.

Пульт централізованого спостереження великої місткості «Дунай» в загальному випадку може складатися з [4]:

– не менш ніж двох комп’ютерів типу IBM PC (у разі використання мережевого варіанту – до 16 комп’ютерів)
– принтера (єдиного для мережі);
– джерела безперебійного живлення типу SMART UPS;
– до 8-ми мультиплексорів каналів SPIN MUX, що дозволяють підключити до одного комп’ютера від 1 до 64 ретрансляторів SPIN ATU;
– до 8-ми мультиплксоров каналів SPIN MUX, що дозволяють підключити до одного комп’ютера від 1 до 64 ретрансляторів SPIN ATU;
– до 8-ми радіо мультиплексорів SPIN PCO, що дозволяють організовувати роботу радіомережі, зокрема через маршрутизатори, за допомогою якої може охоронятися до 512 об'єктів в одній радіомережі;
– до 8-ми радіо мультиплексорів SPIN PCO, що дозволяють організовувати роботу 8-ми радіомереж, в кожній з яких може бути включено до 8-ми ретрансляторів SPIN ATU;

- до 8-ми модемів «Радіо СРП», що забезпечують прийом – передачу сповіщень в середовищі пакетної радіомережі від централей SPIN URB через модем «Радіо СРП» або від ретрансляторів SPIN ATU;
- до 8-ми базових комплектів SUR-GARD, забезпечуючи прийом сповіщень від видалених централей;
- програмного забезпечення.

Використання що найменш двох комп'ютерів передбачене тому, що один з них працюватиме в режимі «майстра» (тобто виконує завдання прийняття сповіщення від апаратури), інший – в режимі «клієнта» (у «гарячому» резерві), повністю дублюючи інформацію комп'ютера-майстра.

Працювати з апаратурою (обробляти сповіщення, знімати з охорони, брати під охорону, проводити опит, редагувати базу даних і ін.) можна на будь-якому комп'ютері. Таким чином, виходить два робочих місця, але на комп'ютері-клієнті, без припинення охорони, можна вийти з програми «Дунай» і запустити іншу програму з пакету «Дунай», працювати в інших програмах або вимкнути його, якщо він не потрібен (але на якийсь час, коли немає адміністратора, другу машину краще тримати включеною з програмою «Дунай», тобто в «гарячому» резерві). При виході з ладу комп'ютера-майстра, комп'ютер-клієнт автоматично переходить в режим майстра і продовжує охорону, зберігаючи всю інформацію, яка була на комп'ютері-майстра.

Кожен мультиплексор слід підключити на два комп'ютери і встановити перемикач, що виконує комутацію всіх мультиплексорів з одного комп'ютера на інший. Таким чином, у разі виходу з ладу комп'ютера-майстра, користувачу необхідно лише перемкнути одним перемикачем всі мультиплексори на резервний комп'ютер, який, тим часом, автоматично переходить в режим майстра [5].

Система «Дунай» підтримує роботу з мультиплексорами в кількості до 8 одиниць. Кількість робочих місць (комп'ютерів) передбачена від 2 до 16 одиниць. З цих комп'ютерів необхідно вибрати два комп'ютери, які матимуть можливість працювати з апаратурою, тобто мультиплексорами. Вони повинні мати кількість послідовних портів (COM-портів), що дозволяє підключати потрібну кількість мультиплексорів (по одному мультиплексору на порт), при необхідності, можна встановити плату розширення COM-портів.

Кожен мультиплексор повинен бути підключений до двох комп'ютерів (майстра і резервного клієнта), враховуючи, що кожен мультиплексор необхідно підключати на однаковий номер COM-порту кожного з двох

комп'ютерів (щоб уникнути плутанини). Решта комп'ютерів працюватиме в режимі клієнтів, «спілкуючись» з апаратурою (мультиплексорами) засобами локальної мережі.

Програмне забезпечення ПЦС гнучко настроюється, що дає можливість оператору контролювати ситуацію і в разі необхідності швидко приймати рішення, на основному сервері зберігатимуться всі необхідні данні та інформація по об'єкту, а саме: назва об'єкту, що охороняється, план приміщення, де чітко вказано входи і виходи, яке обладнання встановлено в кожній кімнаті (кабінеті, приміщенні) і які датчики спрацювали. Так само в базі ПЦС зберігатися особиста інформація людей які мають право знімати або ставити об'єкт під охорону (довірені особи). ПЦН «Дунай» має різні рівні доступу – оператор, адміністратор, програміст.

Програмне забезпечення (ПО) пульта комплексної інтегрованої системи «Дунай» призначено для інформаційного і функціонального забезпечення дій користувачів системи. Структура ПО і дружній інтерфейс користувача орієнтовані на те, щоб в системі мали змогу функціонувати різні прилади. Програмне забезпечення системи підтримує роботу приладів по різним каналам зв’язку [4].

З урахуванням незамкнутої архітектури апаратних засобів і досить універсального підходу в програмному забезпеченні вирішене завдання отримання відкритої системи за допомогою створення драйверів апаратури з відкритим інтерфейсом. Всі рішення, які реалізовані в програмному забезпеченні пульта централізованого спостереження, направлені на досягнення головної мети – дати можливість оператору (черговому) максимально швидко і безпомилково реагувати на сповіщення, що поступили. Багато в чому це досягається наочністю і продуманістю до дрібниць призначеного для користувача інтерфейсу.

Програмне забезпечення, яке функціонує у КІСЦО «Дунай» реалізоване у середовищі інтегрованої розробки Boralnd Delphi та Microsoft DDK. Система управління базою даних, яка надає змогу функціонування та виконання всіх завдань системи «Дунай» реалізована у СУБД InterBase. Програмне забезпечення КІСЦО «Дунай» детально протестоване і може надійно функціонувати в системах сімейства Windows. Застосування цього програмного забезпечення надає системі [5]:

- простій і зручний графічний інтерфейс користувача;
- гнучку реляційну базу даних з можливістю швидкого пошуку і запитів;

- вбудовану роботу з картами і планами об'єктів;
- гнучку систему розмежування доступу до функцій меню системи;
- отримання комплекту готових і формування нових звітних форм про роботу пульта централізованого спостереження, додавання і редагування інформації про реальні причини тривог, формування добової відомості, відомостей для напряму електромонтерів по заявках та інше;
- зберігання бази даних на всіх комп'ютерах, її редагування в реальному режимі часу;
- повну взаємозамінюванність і резервування комп'ютерів за рахунок мережевих можливостей програмного забезпечення і 100%-ного дублювання інформації в реальному режимі часу;
- простий спосіб організації декількох робочих місць на пульті централізованого спостереження з паралельною обробкою сповіщень;
- можливість сполучення пультових крайових пристроїв і компостерів за допомогою зовнішнього програмного драйвера.

ПЗ КІСЦО «Дунай» складається з декількох програм (пакету програм): автоматизованого робочого місця оператора (програма «Дунай»), програми огляду журналів подій, створення звітів і відомостей (програма «Журнали і Звіти»), програми «Резервного копіювання БД». Кожна з програм виконує свої специфічні функції.

Для надійного функціонування програмного забезпечення системи «Дунай» на платформі операційної системи Windows бажано мати комп'ютери, що забезпечені наступними технічними характеристиками: процесор Intel Core i5 і вище, оперативна пам'ять 4 Гбайт, SVGA відеоадаптер з дозволом 1280x1024 і від 256 Мб ОЗУ, мережеву плату із швидкістю передачі даних не менш 100 Мбайт/сек., жорсткий диск місткістю від 500 Гбайт, декілька портів послідовного обміну (COM-портів) в стандарті RS-232 (кількість портів залежить від планованої кількості мультиплексорів), клавіатура, маніпулятор, звукова плата і акустична система, DVD-ROM привід з будь-якою швидкістю читання даних. Додаткове устаткування: лазерний або струменевий принтер. При виборі конфігурації комп'ютера слід враховувати кількість об'єктів, що охороняються.

3 ПРОЕКТУВАННЯ СПЕЦІАЛІЗОВАНОЇ ПІДСИСТЕМИ ОХОРОННОЇ БЕЗПЕКИ ОБ'ЄКТІВ

3.1 Визначення функціональних можливостей підсистеми

Для здійснення проектування спеціалізованої підсистеми охоронної безпеки об'єктів, яка забезпечує підтримку віддаленого робочого місця ПЦС «Дунай», повинна перш за все забезпечити інтеграцію з системою «Дунай», а саме використовувати протоколи та програмні засоби, які дозволять функціонувати підсистемі з ПЦН «Дунай» як одне ціле. Спеціалізована підсистема охоронної безпеки об'єктів дозволяє розширювати систему «Дунай», додаючи в неї нове програмне забезпечення, яке виконує нові функції і взаємодіє з уже функціонуючим програмним забезпеченням та обладнанням через стандартні протоколи.

Програмне забезпечення «Дунай» працює в локальній мережі по протоколу TCP/IP. При цьому всі потоки інформації від устаткування, яке встановлене на кожному окремому об'єкті, поступають на один комп'ютер – «майстер». Копії програми, запущені на інших комп'ютерах, працюють у режимі клієнтів. Протокол зв'язку майстер-комп'ютера і клієнтів надійно функціонує в локальній мережі, що забезпечує достатньо велику швидкість передачі інформації. При використанні в локальній мережі виправдано те, що при підключенні будь-якого клієнта база даних переписується на нього з «майстра», що забезпечує резервування інформації і спрощує роботу програми «Дунай», оскільки, незалежно від того де вона запущена – на «майстер-комп'ютері», або на комп'ютері-клієнті, робота з базою даних здійснюється однаково: в мережі розповсюджуються тільки внесені зміни. Але встановити програму «Дунай» на комп'ютері, пов'язаному з «майстром» модемним з'єднанням виявляється проблематичним. Для підключення таких клієнтів програмне забезпечення «Дунай» підтримує спеціальний протокол, призначений для зв'язку через GSM-модем [6].

Спеціалізована підсистема охоронної безпеки об'єктів в автоматичному режимі одержує від програми «Дунай» сповіщення про тривоги, що виникають на конкретному територіально розподіленому об'єкті, на якому встановлене устаткування для здійснення охоронного спостереження, підключене до КІСЦО «Дунай». Програмне забезпечення КІСЦО «Дунай» має функціональну можливість здійснення розподілу охоронних тривог від конкретно визначеного об'єкту від всіх інших тривог з інших об'єктів, що

охороняються. При цьому сигнали від об'єктового устаткування приходять по призначених для цього каналах зв'язку (зайнята або виділена телефонна лінія, виділена радіо частота, пакетний радіозв'язок) на пульт централізованого спостереження системи «Дунай» та якщо вони відносяться до визначеного об’єкта , то здійснюється передача тривог спеціалізованій підсистеми охоронної безпеки об’єктів Здійснив аналіз сигналів, які поступають від датчиків спостереження, програма «Дунай» формує спеціальне сповіщення, що посилається на пульт спеціалізованій підсистеми охоронної безпеки об’єктів для його обробки черговим оператором безпосередньо на цьому об’єкті [4].

Підтримка віддаленого робочого місця чергового оператора розподіленого об’єкта охорони та реалізація інтегрованої спеціалізованої підсистеми охоронної безпеки об’єкту є подальшим розвитком КІСЦО «Дунай» у напрямі відкритості системи.

Введення такої підсистеми забезпечить можливість розширювати систему, додаючи в її склад нового програмного забезпечення, що виконує нові функції і взаємодіє з вже існуючим програмним забезпеченням і устаткуванням через стандартні протоколи (рис. 3.1).

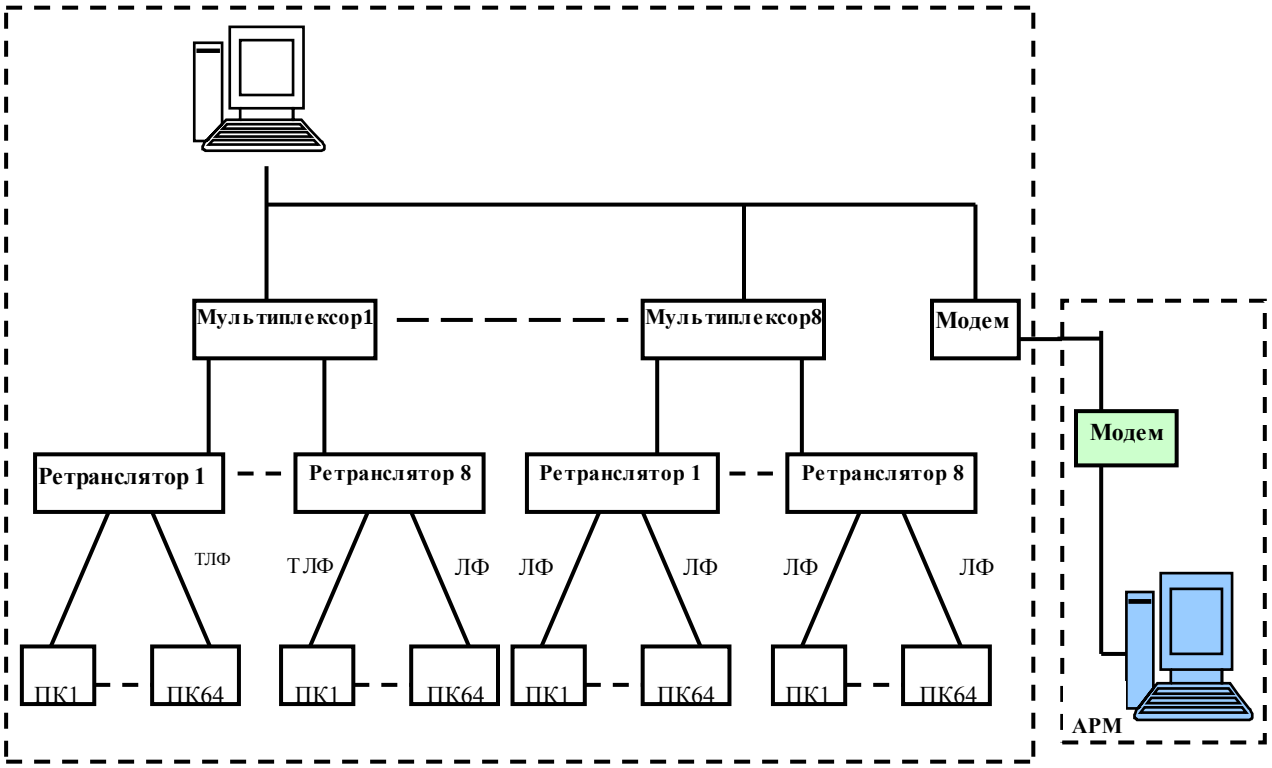

Рисунок 3.1 – Віддалене робоче місце оператора підсистеми охоронної безпеки об’єктів

Програмне забезпечення спеціалізованої підсистеми охоронної безпеки об’єктів здійснює взаємодію з пультом централізованого спостереження КІСЦО «Дунай» по протоколу, що визначає формат передачі повідомлень про тривоги і забезпечити черговому оператору віддаленого розподіленого об’єкту охорони можливість максимально оперативно і безпомилково реагувати на сповіщення. Введення такої підсистеми забезпечити оперативний доступ до інформації, що до стану об’єкта спостереження і надасть можливість швидко реагувати на тривоги безпосередньо на об’єкті.

3.2 Вибір та обґрунтування архітектури підсистеми

Архітектура «клієнт-сервер» визначає загальні принципи організації взаємодії в мережі, де є сервери, вузли-постачальники деяких специфічних функцій (сервісів) і клієнти, споживачі цих функцій. Практичні реалізації такої архітектури називаються клієнт-серверними технологіями. Кожна технологія визначає власні або використовує наявні правила взаємодії між клієнтом і сервером, які називаються протоколом обміну (протоколом взаємодії).

Під клієнт-серверним додатком визначаємо інформаційну систему, засновану на використанні серверів баз даних. На стороні клієнта виконується код додатка, в який обов'язково входять компоненти, що підтримують інтерфейс з кінцевим користувачем, що виробляють звіти, виконують інші специфічні для докладання функції. Клієнтська частина програми взаємодіє з клієнтською частиною програмного забезпечення управління базами даних, яка, фактично, є індивідуальним представником СУБД для додатка.

Клієнт-серверна система складається з безлічі комп'ютерів, об'єднаних в мережу. Комп'ютери звані клієнтами, займаються обробкою прикладних програм. Комп'ютери, звані серверами, займаються обробкою БД [7].

У будь-якій мережі, побудованої на сучасних мережевих технологіях, присутні елементи клієнт-серверної взаємодії, найчастіше на основі двухзвенной архітектури. Двухзвенною (two-tier) вона називається через необхідність розподілу трьох базових компонентів між двома вузлами – клієнтом і сервером. Функції клієнтів виконують майже завжди ПК. У ролі сервера може виступати комп'ютер який має більш високу продуктивність.

На сервері мережі розміщується БД і встановлюється потужна серверна СУБД – сервер баз даних. Сервер БД – це програмний компонент, що

забезпечує зберігання великих обсягів інформації, її обробку та подання необхідної інформації користувачам в мережевому режимі.

На комп'ютері-клієнті додаток-клієнт формує запит до БД. Серверна СУБД забезпечує інтерпретацію запиту, його виконання, формування результату запиту і пересилання його по мережі на клієнтський комп'ютер.

Двухзвена архітектура використовується в клієнт-серверних системах, де сервер відповідає на клієнтські запити безпосередньо і в повному обсязі, при цьому використовуючи тільки власні ресурси. Тобто сервер не викликає сторонні мережеві програми, але не звертається до сторонніх ресурсів для виконання будь-якої частини запиту (рис. 3.2).

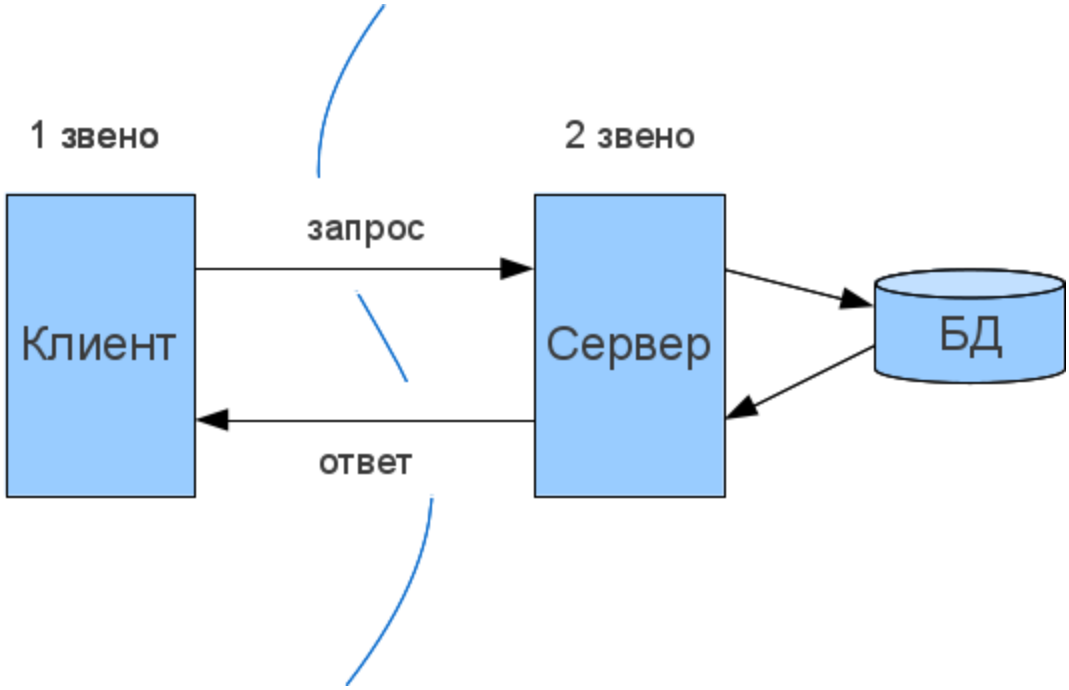

Рисунок 3.2 – Схема функціонування архітектури «клієнт-сервер»

В архітектурі клієнт-сервер функції клієнтського додатка і серверної СУБД розділені. Функції клієнтської програми розбиваються на наступні групи [8]:

- введення-виведення даних (презентаційна логіка) – це частина коду клієнтської програми, яка визначає, що користувач бачить на екрані, коли працює з додатком;
- бізнес-логіка – це частина коду клієнтської програми, яка визначає алгоритм вирішення конкретних завдань програми;
- обробка даних всередині програми (логіка бази даних) – це частина коду клієнтської програми, яка пов'язує дані сервера з додатком.

Для зв'язку з цим використовується процедурна мова запитів SQL, за допомогою якого здійснюється вибірка і модифікація даних в серверних СУБД.

Сервер баз даних в загальному випадку здійснює цілий комплекс дій по управлінню даними. Основними серед них є такі:

- виконання запитів користувачів на вибір і модифікацію даних і метаданих, одержуваних від клієнтських додатків, що функціонують на ПК локальної мережі;
- зберігання та резервне копіювання даних;
- підтримка посилальної цілісності даних згідно з визначеними в БД правилам;
- забезпечення авторизованого доступу до даних на основі перевірки прав і привілеїв користувача;
- протоколювання операцій і ведення журналу транзакцій.

При здійсненні клієнт-серверної обробки зменшується мережевий трафік, завдяки здійснення передачі тільки результатів запитів. Вантаж файлових операцій лягає в основному на сервер, який має більші системні ресурси, ніж у комп'ютерів-клієнтів, і тому здатний швидше обслуговувати запити. Як наслідок цього, зменшується потреба клієнтських додатків в оперативній пам'яті. Оскільки сервери здатні зберігати велику кількість даних, то на комп'ютерах-клієнтах звільняється значний обсяг дискового простору для інших додатків.

Підвищується рівень несуперечності даних і суттєво підвищується ступінь безпеки БД, так як правила цілісності даних визначаються в серверній СУБД і є єдиними для всіх додатків, що використовують цю БД. Є можливість зберігання бізнес-правил (наприклад правил посилальної цілісності або обмежень на значення даних) на сервері, що дозволяє уникнути дублювання коду в різних клієнтських додатках, що використовують загальну базу даних.

Всі серверні СУБД мають клієнтську частину, яка звертається до БД за допомогою СУБД. Між клієнтським додатком і СУБД не існує прямого зв'язку і додатково вбудовуються програмні модулі, що дозволяють клієнтському застосуванню отримувати доступ до БД, який створюється за допомогою різних СУБД. Такі модулі називаються механізмами доступу до даних. Існує два основних способи доступу до даних з клієнтських додатків: використання прикладного інтерфейсу і використання універсального програмного інтерфейсу [9].

Прикладний програмний інтерфейс (API – Application Programming Interface) являє собою набір функцій, що викликаються з клієнтського застосування. Він може працювати тільки з СУБД даного виробника і при її заміні доведеться переписувати значну частину коду клієнтського додатку. Прикладний програмний інтерфейс різний для різних СУБД.

Універсальний механізм доступу до даних забезпечує можливість використання одного і того ж інтерфейсу для доступу до різних типів СУБД. Зазвичай він реалізований у вигляді спеціальних додаткових модулів, які називаються драйверами [10].

Найбільш поширеним програмним інтерфейсом, що забезпечує доступ до даних конкретної бази даних є ODBC (Open Database Connectivity) фірми Microsoft. В рамках ODBC програмний додаток безпосередньо взаємодіє з диспетчером драйвером, посилаючи йому ODBC-виклики. Диспетчер драйверів відповідає за динамічну завантаження потрібного ODBC-драйвера, через який звертається з сервера баз даних. ODBC-драйвер виконує всі виклики ODBC-функцій і «переводить» їх на мову джерела даних. СУБД зберігає і виводить дані у відповідь на запити з боку ODBC-драйвера.

Завдання ODBC-джерела даних є дією, яка здійснюється засобами операційної системи, що управляє комп’ютером. Призначене для користувача DSN-джерело даних, доступний тільки активному користувачу на поточному комп'ютері. Файловий DSN – джерело даних, які можуть застосовувати спільно різні користувачі, у яких встановлені однакові ODBC-драйвери. Системний DSN – джерело даних, доступне всім користувачам і службам поточного комп'ютера [9].

Застосування архітектури клієнт-сервер дає можливість поступового нарощування інформаційні системи будь-якого призначення. Поділ загальної бази даних на корпоративну і персональні дозволяє зменшити складність проектування баз даних в порівнянні з централізованим варіантом, а значить знизити ймовірність помилок при проектуванні і вартість проектування.

Найважливішою метою застосування бази даних в інформаційних системах є забезпечення незалежності даних від прикладних програм, це дає можливість користувачам не займатися проблемами представлення даних на фізичному рівні: розміщення даних в пам'яті, методів доступу до них.

Така незалежність досягається підтримуваним СУБД багаторівневим поданням даних в базі даних на логічному (призначеному для користувача) і фізичному рівнях. Завдяки СУБД і наявності логічного рівня представлення

даних забезпечується відділення концептуальної (понятійної) моделі бази даних від її фізичного представлення в пам'яті ПЕОМ [9].

Найважливішим параметром великої інформаційної системи є швидкодія при значній кількості користувачів, а також надійність, масштабованість і безпеку. Все це забезпечує архітектура «клієнт-сервер». Така архітектура дозволяє оптимально розподілити роботу між клієнтськими і серверної частинами системи: тепер додаток, що працює на робочій станції, не читає записи бази даних «прямо», а посилає запити на сервер, де вони приймаються і послідовно відпрацьовуються спеціальними програмами. В результаті на робочу станцію надходять тільки оброблені дані, що радикально скорочує інформаційні потоки в обчислювальній мережі.

3.3 Середовище розробки інтерфейсу підсистеми

Для здійснення програмної реалізації спеціалізованої підсистеми охоронної безпеки об’єктів було обрано інтегроване середовище розробки Lazarus. Так як ПЗ системи «Дунай» виконано у середовищі Delphi, мовою Object Pascal, то доцільно застосування сучасних програмних середовищ з мовою Object Pascal для подальшого інтегрування розробляэмоъ підсистеми з системою «Дунай». Lazarus э відкритим середовищем розробки програмного забезпечення на мові Object Pascal для компілятора Free Pascal (Free Pascal Compiler), який э безкоштовно розповсюджуваний компілятор мови програмування Pascal. Інтегроване середовище розробки надає можливість кросплатформової розробки додатків в Delphi-подібному оточенні, що э важливою умовою для інтегрування розробленоъ підсистеми з системою «Дунай» [4].

Lazarus використовує компілятор FPC (Free Pascal Compiler), редактори коду, форм, інспектор об'єктів, відладчик і багато інших інструментів. Середовище Lazarus – це RAD (Rapid Application Development) – середовище швидкої розробки додатків [11].

Інтегроване середовище розробки програм Lazarus став першою (і поки єдиною) безкоштовною IDE, доступною освітнім і державним установам абсолютно безкоштовно. Lazarus є проектом Open Source – проектом з відкритим вихідним кодом. Lazarus має підтримку безлічі мов, в тому числі і російською та українською, що вигідно відрізняє його від інших IDE. FPC – кросплатформовий інструмент, що підтримує величезну кількість платформ. Серед них – AmigaOS, DOS, Linux, BSD, OS/2, MacOS і Win32.

IDE Lazarus поширюється під вільними ліцензіями GNU General Public License (GNU GPL) і GNU Lesser General Public License (GNU LGPL). Це середовище з графічним інтерфейсом для розробки програм, що базується на кросплатформій бібліотеці візуальних і не візуальних компонентів LCL (Lazarus Component Library), сумісних з VCL Delphi. Такого набору компонентів Lazarus досить для створення додатків з графічним інтерфейсом, які працюють з базами даних та з мережею Інтернет. В даний час Lazarus працює під управлінням операційних систем (ОС) Linux, Mac OS X, UNIX-подібних, Windows, Android. Використовуючи середу Lazarus, можна створювати кросплатформові додатки.

Lazarus повністю Юнікодний інтерфейс (UTF-8) і редактор, тому відсутні проблеми з імпортом коду, що містить національні символи. У Lazarus є потужний редактор коду, що включає систему підказок, гіпертекстову навігацію по вихідним текстам, а також форматування коду з використанням механізмів Jedi Code Format. У Lazarus здійснена підтримка безлічі типів синтаксису Pascal: Object Pascal, Turbo Pascal, Mac Pascal, Delphi, підтримуваних з боку компілятора. Він має власний формат управління пакетами. Недоліками середовища Lazarus є відсутність повної сумісності з Delphi, хоча надає можливість створювати більш кросплатформені додатки [11].

Lazarus дозволяє працювати з більшістю СУБД, для цього в середовище необхідно встановити компоненти для роботи з відповідними СУБД. Стандартний набір компонентів для роботи з базою даних в Lazarus можливо застосувати з компонентом SQLdb, який містить набір для роботи з різними СУБД [9]. TIBConnection – компонент для підключення к СУБД Interbase. TFBEventMonitor – компонент для роботи з подіями Interbase. TSQLDBLibraryLoader – компонент для завантаження бібліотеки доступу до даних. TSQLScript – компонент для роботи з великою кількістю SQL коду.

Для підключення до будь-якої СУБД на форму слід розмістити необхідний компонент, що дозволить налаштувати підключення до необхідної бази даних. Так само слід додати на форму компонент TSQLTransaction, який забезпечує цілісність роботи з даними [12].

3.4 Проектування логічної моделі бази даних підсистеми охоронної безпеки об’єктів

Для здійснення зберігання всієї необхідної інформації для забезпечення функціонування підсистеми охоронної безпеки об’єктів та її інтеграції з базою даних системи»Дунай» була обрана СУБД InterBase. Сервер InterBase – це кросплатформова СУБД, що підтримує більшість операційних систем: Windows, Linux, Unix, Solaris, Mac OS та інші [13].

СУБД InterBase побудована на версійної архітектурі зберігання даних. Цей підхід має ряд переваг перед блокувальними СУБД: для відновлення баз даних InterBase після системного збою, немає необхідності підтримки низки транзакцій; клієнти, які читають дані ніколи не блокують клієнтів, які здійснюють запис даних.

Сервер InterBase має цілу низку переваг, що вигідно відрізняють його від інших СУБД: оновлювані уявлення View; двофазне підтвердження транзакцій; ефективний механізм тригерів; серверна обробка Blob-полів (Blob-filters); події (повідомлення).

Для збереження та швидкого доступу до даних в спеціалізованій підсистемі охоронної безпеки об’єктів використана повнофункціональна СУБД InterBase, що забезпечує легку масштабованість, не вимагає складного адміністрування, підтримує Unicode, відповідає стандартам SQL, кросплатформова. СУБД InterBase перевірена в таких критично важливих системах до яких і відноситься підсистема охоронної безпеки об’єктів, задовольняючи необхідний рівень надійності і безвідмовності [14].

У порівнянні з багатьма іншими СУБД, InterBase надає дуже ефективний механізм тригерів: кожна таблиця може мати велику кількість тригерів, які виконуються автоматично при вставці, зміні або видаленні кожного окремого запису, до або після цих подій. Важливою особливістю сервера InterBase є можливість розширення стандартного набору SQL-функцій за допомогою призначених для користувача бібліотек – User Defined Functions, а також механізми обробки BLOB-полів на сервері за допомогою BLOB-фільтрів.

При проектуванні бази даних спеціалізованої підсистеми охоронної безпеки обїектів, в першу чергу, йдеться про об'єкти, за якими здійснюється спостереження, головна суть в базі даних – об'єкт охорони. Оскільки об'єкт володіє довільною кількістю шлейфів, за кодом яких здійснюється передача тривог, для ліквідації множинних атрибутів вводиться суть шлейф сигналізації (зона об'єкту), пов'язана з об'єктом відношенням багато до одного. Охоронна організація (пульт) працюють з устаткуванням на всіх об'єктах. Сутність Обслуговуюча організація, пов'язана з об'єктами

відношенням один до багатьох. У окрему суть виділені атрибути вулиця, галузь (до якої належить об'єкт), тип устаткування, яке установлене на об'єкті. Ця сутність визначена постановкою завдання: у базі даних пульта, встановленого в деякому районі міста можуть бути введені всі вулиці району, незалежно від того знаходяться на них об'єкти, на яких здійснюється спостереження чи ні. Приналежність об'єкту до тієї або іншої галузі може впливати на умови надання послуг, і список галузей повинен бути складений наперед.

Сутність Галузь визначається наступними перерахованими атрибутами. Назва об'єкту, що ідентифікує об'єкт, яка і є ключовим атрибутом для нього («Склад №1», «Ліцей «Приморський»). Об'єкт характеризується галуззю до якої він належить («Торгівля», «Освіта»,). Адреса об'єкту – це важливий атрибут, що включається в повідомлення про тривогу, яке видається базою даних і відображається у формі вікна підсистеми охоронної безпеки об’єкту. Адреса об’єкта складається з: назви вулиці, номеру будинку, номеру під'їзду, поверху, номера приміщення, або номерів кабінетів в приміщенні. Ці поля заповнюються у разі потреби (наприклад, поверх, під'їзд, номер приміщення можуть бути не обов'язкові). Якщо об’єкт має велику територіально розподілену структуру то для ідентификування конкретного приміщення додається більш детальна інформація.

По будь-якому об'єкту нагляду зберігається інформація про організації або власників цього об'єкту, це наступні атрибути: назва організації-власника, керівник організації чи підприємства, чи особа власник приміщень, його службовий телефон, мобільний телефон, домашній телефон; заступник керівника, його службовий телефон, мобільний телефон, домашній телефон; Телефон чергового (якщо такий є) в організації-власника об'єкту. Цей телефон використовується перш за все для сповіщення власника про тривогу на його об'єкті. Це може бути, наприклад, телефон чергового, що знаходиться в тій же будівлі, що і об'єкт, з якого поступила тривога.

Для гнучкішого уявлення інформації про об'єкт, який може бути державною, комунальною або приватною власністю вводиться поле орендодавець. Наприклад, як орендодавець об'єкту «Склад №1» виступає фірма «Метро». Організація-орендодавець характеризується наступними атрибутами: назва організації, керівник організації орендодавця, його службовий телефон, мобільний телефон, домашній телефон. Додаткова інформація – про заступника керівника. Також передбачена можливість зберігати його робочий телефон, мобільний телефон, номер домашнього

телефону. Потрібно відзначити, що вся ця інформація не обов'язково повинна бути присутньою в базі даних. Наприклад, якщо якісь питання розв'язуються тільки із заступником, не потрібно вводити дані по керівнику, тим більше, включаючи його домашній телефон. У полі керівник або заступник взагалі можна внести іншу відповідальну особу, що відповідає за контакти організації власника (орендодавця) з пожежною охороною. Черговий телефон орендодавця може використовуватися для тих же цілей, що і черговий телефон організації – власника.

Наступними були визначені дві важливі сутності – Обслуговуюча організація, що відповідає за працездатність системи охорони і Устаткування, яке встановлене на об'єкті. Атрибутами сутності Устаткування є інформація про встановлене на об'єкті устаткування – це тип обладнання (прибори), до якого підключені датчики сигналізації. Обслуговуюча організація – організація, що встановила це устаткування і займається його обслуговуванням та підтримкою стану працездатності. Сутність обслуговуюча організація повинна зберігати наступну інформацію: керівник; робочий, мобільний і домашній телефони керівника. Замість імені і координат керівника обслуговуючої організації, в цих полях можуть зберігатися дані про іншого співробітника обслуговуючої організації, з яким підтримуються контакти.

Система сигналізація на територіально розподіленому об'єкті може охоплювати багато приміщень, тому про тривогу потрібно ідентифікувати по зонах об'єкту (наприклад, складське приміщення №12, кабінет №7і т.п.). Звичайно сповіщувачі, контролюючі окреме приміщення (або невеликі суміжні приміщення) підключені до одного шлейфу. Сигнал тривоги, проходячи через все проміжне устаткування, приходить на ПЦН з якогось конкретного шлейфу об'єктного устаткування. Таким чином, на кожному об'єкті є окремі контрольовані шлейфами ділянки – зони. Кожен шлейф має код і найменування (наприклад, «Датчик руху у кімнаті №3»), яке є описом місця розташування. Код шлейфу грає важливу роль в протоколі обміну інформацією з програмою «Дунай» і має формат «М.Р.ППК.Группа.Шлейф», де М – номер мультиплексора, Р – номер ретранслятора, ППК – ППК, Група – група шлейфів, підключених до ППК, Шлейф – номер шлейфу.

Хоча атрибути назва об'єкту, вулиця, галузь, устаткування є природними ключовими атрибутами відповідних відносин, для них, як для первинних ключів, вводиться числовий ідентифікатор. Це скорочує розмір зовнішніх ключів в підлеглих відносинах.

3.5 Проектування фізичної моделі бази даних підсистеми

Виходячи з вимог до розробки та зберігання інформації у спеціалізованій підсистемі охоронної безпеки об’єктів, було отримано ряд таблиць реляційної бази даних.

На підставі врахування всіх вимог до підсистеми та визначених вимог до сутностей бази даних, були розроблені наступні сутності (табл. 3.1 – 3.6).

Таблиця 3.1 – Сутність Zones – Шлейфи (Зони)

Поле	Тип	Розмір	Опис
Code	Alpha	14	Код шлейфу. Первинний ключ
Name	Alpha	254	Назва шлейфу
Code_Obj	Short		Зовнішній ключ об'єкту
Location	Memo	240	Додаткова інформація

Таблиця 3.2 – Сутність Branche – Галузь

Поле	Тип	Розмір	Опис
Code	Short		Унікальний ID. Первинний ключ
Name	Alpha	100	Назва галузі

Таблиця 3.3 – Сутність Equipment – Устаткування

Поле	Тип	Розмір	Опис
Code	Short		Унікальний ID. Первинний ключ
Name	Alpha	250	Назва устаткування

Таблиця 3.4 –Сутність Servises – Обслуговуюча організація

Поле	Тип	Розмір	Опис
Code	Short		Унікальний ID. Первинний ключ
Name	Alpha	255	Назва обслуговуючої організації
Chif_Name	Alpha	100	Ім'я керівника організації
Chif_PhoneW	Alpha	20	Робочий телефон
Chif_PhoneH	Alpha	20	Домашній телефон
Chif_PhoneM	Alpha	20	Мобільний телефон

Таблиця 3.5 – Сутність Street – Вулиця

Поле	Тип	Розмір	Опис

Code	Short		Унікальний ID. Первинний ключ
Name	Alpha	100	Назва вулиці

Таблиця Object індексована по полю Full_Name (назва об’єкту) , завдяки чому , об'єкти охоронної безпеки відображаються у програмі підсистеми охоронної безпеки об’єктів у алфавітному порядку.

Таблиця 3.6 – Сутність Object – Об’єкт

Поле	Тип	Розмір	Опис
Code	Autoincr		Унікальний ID, первинний ключ
Full_Name	Alpha	255	Назва об'єкту
Branch	Short		Галузь (Branch.Code)
A_Street	Short		Вулиця (Street.Code)
A_Home	Alpha	20	Будинок
A_Apartment	Alpha	10	Приміщення
A_Doorway	Alpha	10	Під'їзд
A_Floor	Alpha	10	Поверх
Owner	Alpha	200	Власник
Chif_Name	Alpha	100	Ім'я керівника організації-власника
Chif_PhoneW	Alpha	20	Робочий телефон
Chif_PhoneH	Alpha	20	Домашній телефон
Chif_PhoneM	Alpha	20	Мобільний телефон
Depu_Name	Alpha	100	Ім'я заступника організації-власника
Depu_PhoneW	Alpha	20	Робочий телефон
Depu_PhoneM	Alpha	20	Мобільний телефон
Duty_Phone	Alpha	20	Телефон чергового орг.-власника
Rentor	Alpha	255	Орендодавець
RStreet	Short		Вулиця орендодавця
RHome	Alpha	10	Будинок орендодавця
RApartment	Alpha	10	Приміщення орендодавця
RChif_Name	Alpha	100	Ім'я керівника орг.-орендодавця
RChif_PhoneW	Alpha	20	Робочий телефон
RChif_PhoneH	Alpha	20	Домашній телефон
RChif_PhoneM	Alpha	20	Мобільний телефон
RDepu_Name	Alpha	100	Ім'я заступника орг.-орендодавця
RDepu_PhoneW	Alpha	20	Робочий телефон
RDepu_PhoneH	Alpha	20	Домашній телефон

RDepu_PhoneM	Alpha	20	Мобільний телефон
Rduty_Phone	Alpha	20	Телефон чергового орг.-орендодавця
Additional	Memo	240	Додаткова інформація
Equipment	Short		Устаткування
Servises	Short		Обслуговуюча організація

Для збереження інформації про всі подій перелічені вище, у інтерфейсі підсистеми в наочній області визначена ще одна сутність – Сповіщення (Тривога), що відбулася.

Сутність Сповіщення характеризується такими атрибутами: час, коли подія відбулася; об'єкт, шлейф і його розташування (якщо подія – тривога, що поступила); чи вимагає до тепер увагу оператора (оброблена чи ні). Цю сутність реалізує таблиця Alerts (табл. 3.7).

Таблиця 3.7 – Сутність Alerts – Сповіщення (тривоги)

Поле	Тип	Розмір	Опис
Kod	+Autoincr	14	Унікальний ID. Первинний ключ
Object	Alpha	100	Тривога або подія
Location	Alpha	240	Розміщення шлейфу
Zone	Alpha	20	Шлейф
DatTime	Timestamp		Час виникнення
Obrabotana	Alpha	1	Оброблена чи ні (‘y’/’n’)

Вона не зв'язана відношенням багато до одного з таблицею Zones і Object. У цих сутностях зберігається інформація про об'єкти, за якими здійснюється спостереження віддаленим пультом спостереження, засобами спеціалізованої підсистеми охоронної безпеки об’єктів.

Якщо спостереження за об'єктом більше не ведеться – він видаляється з бази даних. Інформація, яка зберігається у таблиці Alerts служить журналом і повинна зберігати інформацію про тривоги достатньо довго (півроку, декілька років).

Об'єкт за цей час може бути просто видалений з бази даних (наприклад, по причині припинення договору надання охоронних послуг). Тому назва об'єкту і шлейфу повинні бути копійовані у відповідні поля таблиці Alerts.

Розрахований на багато користувачів режим доступу до спеціалізованої підсистеми охоронної безпеки об’єктів зумовив використання у базі даних системи Сутності Access – «Права користувачів». Ця сутність забезпечує

розмежування прав доступу та повноважень осіб, що працюють з базою даних засобами спеціалізованої підсистеми охоронної безпеки об’єктів.

Ідентифікація користувача у підсистеми необхідна для занесення в журнал дій оператора, для контролю можливих його помилок при роботі з програмою. Крім того, розмежування прав доступу в системі гарантує захист даних від навмисної спроби дістати несанкціонований доступ до них (табл. 3.8).

Таблиця 3.8 – Сутність Access – Права доступу користувачів

Поле	Тип	Розмір	Опис
UserName	Alpha	100	Ім'я користувача
Password	Alpha	10	Пароль
Root	Logical		Ознака користувача-адміністратора
Right1	Alpha	30	База даних «read», «write»
Right2	Alpha	30	Журнал (Alerts) «read», «write»
Right3	Alpha	30	Розрив з'єднання «yes»
Right4	Alpha	30	Зарезервовано
Right5	Alpha	30	Зарезервовано
Setting	Alpha	255	Настройки інтерфейсу користувача

Під час надходження тривоги, інтерфейс спеціалізованої підсистеми охоронної безпеки об’єктів відображає інформацію про тривогу у вікні спостереження, але, окрім цього, десь повинно зробити запис про надходження тривоги з часом її надходження. Якщо у вікні спостереження були не оброблені тривоги, або робота підсистеми була припинена аварійно, при наступному запуску програми ці тривоги повинні знову бути показані оператору.

Окрім тривог підсистема фіксує наступні події: «програма запущена користувачем», «вихід з програми», «установка з'єднання з ПО «Дунай», «з'єднання розірвано користувачем», «аварійний розрив з'єднання».

У реальному проектуванні структури бази даних застосовується метод – так зване, семантичне моделювання. Семантичне моделювання є моделювання структури даних, спираючись на зміст цих даних. Як інструмент семантичного моделювання використовуються різні варіанти діаграм сутність-зв'язок (ER – Entity-Relationship. Всі такі діаграми використовують графічне зображення сутностей предметної області, їх властивостей (атрибутів), і взаємозв'язків між сутностями [6].

Ключ сутності – це ненадлишковий набір атрибутів, значення яких у сукупності є унікальними для кожного екземпляра сутності. Ненадлишковість полягає в тому, що видалення будь-якого атрибута з ключа порушується його унікальність. Сутність може мати кілька різних ключів.

Зв'язок – це деяка асоціація між двома сутностями. Одна сутність може бути пов'язана з іншою сутністю або сама з собою. Зв'язки дозволяють по одній сутності знаходити інші суті, пов'язані з нею.

При розробці ER-моделей треба отримати наступну інформацію про предметну область: список сутностей предметної області; список атрибутів сутностей; опис взаємозв'язків між сутностями.

Діаграма «Сутність – Зв'язок» дозволяє використовувати наочні графічні позначення для моделювання сутностей та їх взаємозв'язків [9].

Схема сутностей бази даних спеціалізованої підсистеми охоронної безпеки об’єктів і зв'язків між ними, представлені на рисунку 3.1.

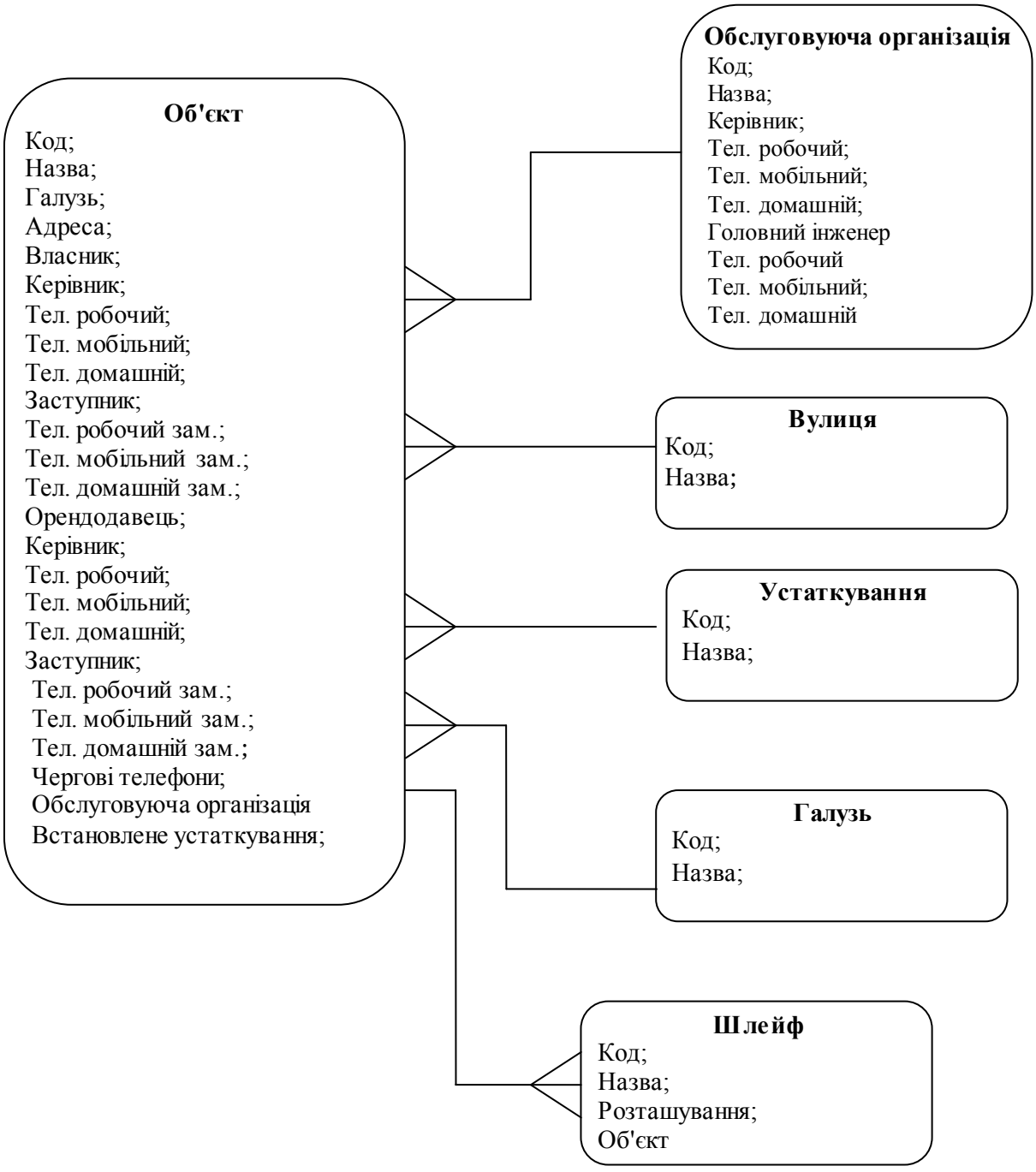

Рисунок 3.3 – Діаграма «Сутність – Зв'язок» бази даних спеціалізованої підсистеми охоронної безпеки об’єктів

3.6 Протокол передачі даних між підсистемою та ПЗ системи «Дунай»

Розробка спеціалізованої підсистеми охоронної безпеки об’єктів забезпечить функціонування віддаленого робочого місця оператора пульта територіально розподіленого об’єкту охорони, що реалізує спостереженні безпосередньо на об’єкті черговим оператором (охороною).

Ця підсистема інтегрована у КІСЦО «Дунай» та реалізує робоче місце чергового охорони віддаленого обєкту, на який надходять повідомлення про тривоги з структурних підрозділів цього і тільки цього об'єкту, де встановлене охоронне устаткування (охоронна сигналізація), підключена до обладнання системи «Дунай».

Програмне забезпечення спеціалізованої підсистеми охоронної безпеки, призначене для забезпечення надійного і максимально швидкого реагування чергового охоронної структури об’єкту, яка територіально віддалено від базового пульта охорони, на тривоги, що надходять з структурних підрозділів тільки цього об'єкту.

Для виконання з’єднання спеціалізованої підсистеми охоронної безпеки об’єктів з системою «Дунай» необхідно визначити протокол передачі даних. Реалізація протоколу зв'язку з системою «Дунай» означає можливість виконання програмним забезпеченням спеціалізованої підсистеми охоронної безпеки об’єктів наступних функцій:

- встановлювати з'єднання з програмою «Дунай» засобами GSM-мережі та її каналів зв’язку з використанням GSM-модему. З'єднання вважається встановленим після того, як спеціалізована підсистема охоронної безпеки об’єктів перший раз посилає підтвердження на контрольний запит, який приходить кожну хвилину.
- контролювати наявність з'єднання (прийом повідомлень типу «контроль лінії» від програми «Дунай» і підтвердження їх прийому посилкою повідомлень у відповідь). Контрольне повідомлення складається з тринадцяти байт: FF000000001E<13h>. Підтвердження – з одного байта з кодом 25.
- одержувати повідомлення про тривогу на об'єкті від пульта централізованого спостереження «Дунай». Повідомлення складається з унікального номеру ППК, що функціонує у складі комплексної інтегрованої системи «Дунай» і номеру шлейфу, підключеного до цього ППК, з якого прийшло сповіщення про тривогу. Формат повідомлення представлений на рис. 3.4.

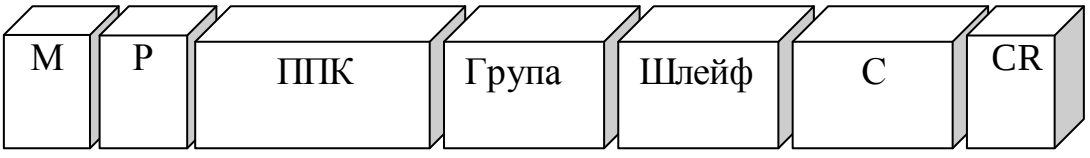

Рисунок 3.4 – Формат протоколу зв'язку ПЦН «Дунай» з підсистемою охоронної безпеки об’єктів

М – номер мультиплексора (1 байт)

Р – номер ретранслятора (1 байт)

ППК – номер ППК (4 байти)

Група – номер групи шлейфів (2 байти)

Шлейф – номер шлейфу (2 байти)

С – контрольна сума (2 байти)

CR – байт з кодом 13

Всі поля в цьому повідомленні протоколу зв’язку є ASCII записом шістнадцятирічних чисел. Наприклад, номер мультиплексору – 4 (34h). ППК з номером шістнадцять – 0010 (30h, 30h, 31h, 30h), таким чином старша цифра знаходиться за молодшою адресою. Одержане повідомлення про тривогу повинно бути підтверджено посилкою однобайтового повідомлення з кодом 25. Для забезпечення контролю помилок передачі в повідомленні про тривогу, яке передане пультом централізованого спостереження «Дунай», включена контрольна сума, що є сумою числових значень всієї решти полів повідомлення (окрім завершального байта з кодом 13). Спеціалізована підсистема охоронної безпеки об’єктів обчислює контрольну суму наново і у разі не збігання її з контрольною сумою, яка передана в повідомленні, не посилає підтвердження програмі «Дунай», що примушує її повторити передачу повідомлення про тривогу.

4 ПРОГРАМНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ ОХОРОННОЇ БЕЗПЕКИ ОБ’ЄКТІВ

4.1 Реалізація інформаційних потреб користувачів системи

Для здійснення розробки програмного забезпечення для спеціалізованої підсистема охоронної безпеки об’єктів необхідно забезпечити виконання основних умов: надійно взаємодіяти з пультом централізованого спостереження «Дунай» по протоколу, що визначає формат передачі повідомлень про тривоги і забезпечити оператору (черговому) можливість максимально оперативно і безпомилково реагувати на сповіщення, що приходять, видаючи йому всю необхідну інформацію про об'єкт, з якого поступила тривога. В якості СУБД, для бази даних інтегрованої інформаційної підсистеми протипожежного нагляду буде використовуватися SQL-сервер InterBase, якій дозволяє зберігати і обробляти великі обсяги інформації в умовах одночасної роботи з базою даних множини клієнтських додатків.

Це програмне забезпечення, яке є частиною системи «Дунай», призначене для розміщення на віддаленому робочому місці чергового оператора охорони на віддаленому територіально розподіленому об’єкті і відповідає вимогам пульта, які включають в себе зберігання бази даних про структурні підрозділи об’єкту, включені в базову систему спостереження «Дунай». Робота розробленого програмного забезпечення з програмою «Дунай» забезпечується стандартним протоколом зв'язку системи «Дунай» з віддаленим робочим місцем оператора.

Повідомлення, які передаються спеціалізованій підсистемі охоронної безпеки об’єктів від КІСЦО «Дунай», мають мінімальний розмір і включають тільки код, що однозначно ідентифікує шлейф сигналізації на об'єкті, тому інформаційне забезпечення спеціалізованої підсистеми охоронної безпеки об’єктів повинне включати всі необхідні дані про об'єкти, які визначає система «Дунай» (ту її частину, яка має відношення до видаленого робочого місця). Це означає, що спеціалізована підсистема охоронної безпеки повинна зберігати дані в реляційній базі даних і уміти швидко виконувати всі передбачені запити.

Програмне забезпечення пульта системи «Дунай» призначено для інформаційного і функціонального забезпечення дій користувачів системи.

Структура ПО і дружній інтерфейс користувача орієнтовані на те, щоб в системі могли функціонувати різне устаткування та прилади [4].

З урахуванням незамкнутої архітектури апаратних засобів і досить універсального підходу в програмному забезпеченні системи «Дунай» для реалізації спеціалізованої підсистеми охоронної безпеки об’єкту були обрані наступні програмні засоби реалізації:

- кросплатформовє середовище розробки інтерфейсу Lazarus, за допомогою якого здійснено розробку інтерфейсу підсистеми, який виконує функції забезпечення реагування чергового оператора охорони територіально розподіленого об’єкту і надає можливість максимально швидко і безпомилково реагувати на сповіщення, що поступили;
- СУБД InterBase, яка дозволилаь створити гнучку реляційну базу даних з можливістю швидкого пошуку і запитів, а також гнучку систему розмежування доступу до функцій меню підсистеми та зберігання, редагування бази в реальному режимі часу.

4.2 Реалізація інтерфейсу користувача-оператора підсистеми

Головне завдання спеціалізованої підсистеми охоронної безпеки об’єктів – надання зручного та функціонального графічного інтерфейсу для роботи користувача-оператора пульта централізованого спостереження за станом об’єктів, що охороняються. Основні фактори, за допомогою яких можливо оцінити зручність використання системи, такі. Адекватність призначеного для користувача інтерфейсу підсистеми – це його відповідність тим завданням, які користувачі повинні вирішувати з його допомогою. Інтерфейс підсистеми повинен сприяти запобіганню та подоланню можливих помилок користувачів. Відсутність помилок операторів-користувачів при роботі з даним інтерфейсом та зменшення часу та зусиль для подолання наслідків вже зроблених помилок забезпечує гарно розроблений інтерфейс. Крім того робота з підсистемою повинна бути настільки зрозумілою, щоб оператор-користувач, який вважається фахівцем в системах нагляду за безпекою об’єктів мав змогу легко виконувати всі необхідні дії. Підсистема не повинна перешкоджати ефективній роботі досвідчених користувачів, які працюють з нею довгий час. Інтерфейс підсистеми повинен забезпечувати зручність використання, звертаючи увагу на спеціальне призначення даного програмного забезпечення. Однією з основних деталей, що впливають на

враження користувача, не був переповнений графічними об’єктами та зайвою інформацією. Інтерфейс також повинен бути функціональним. По його вигляду і структурі користувач повинен одержати уявлення про всі функції підтримуваних програмою.

В реалізованому програмному забезпеченню спеціалізованої підсистеми охоронної безпеки об’єктів реалізована система розмежування прав доступу, що працюють з базою даних підсистеми. Це забезпечує змогу занесення в журнал всіх дій оператора, для контролю можливих його помилок при роботі з програмою. Цей режим не призначений для гарантованого захисту даних від навмисної спроби дістати несанкціонований доступ до них. Розрахований на багато користувачів режим служить для захисту бази даних від ненавмисного псування користувачем, оскільки, якщо можливість редагувати, переглядати, видаляти інформацію присутня в інтерфейсі програми і не заблокована, черговий, робота якого достатньо монотонна може зайнятися «вивченням» інтерфейсу програми і щось ненавмисно видалити. Крім того процедура аутентифікації чергового оператора (і користувача з розширеними правами) призводить до того, що він відповідальніше відноситься до всіх своїх дій при роботі з підсистемою. Для кожного користувача в базі даних користувачів зберігається його ім'я і пароль, що ідентифікує кожного користувача. Воно використовується при занесенні в журнал записів про дії користувачів.

При вході користувача в програму вводиться тільки його пароль (тому паролі повинні бути унікальними для кожного користувача і програма стежить за цим). Існує один користувач – адміністратор, який володіє правом редагувати права доступу інших користувачів на перегляд, зміну об'єктів охорони бази даних. Інформація про події з об'єктів поступає в режимі реального часу, і повинна бути максимально швидко оброблена оператором. При здійсненні програмної реалізації інтегрованої інформаційної підсистеми протипожежного нагляду було виділено ряд вимог, які буди забезпечені.

У базі даних потрібно зберігати всю необхідну інформацію, підтримувати її цілісність, несуперечність і актуальність. Так само передбачені різні корисні вибірки з бази даних, що дозволяють користувачу одержувати відповіді на питання, що цікавлять його, оперативно і легко.

Апаратне забезпечення, яке застосоване для робочого місця чергового оператора пульта централізованого спостереження повинно бути перш за все надійним. Для забезпечення захисту персонального комп'ютера від нестабільної напруги в електричній мережі і можливості його роботи при

аварійному відключенні живлення необхідно використовувати джерела безперебійного живлення (UPS). Оскільки черговий оператор пульта цілодобово працює з інформацією, яку він отримує з монітора, необхідно приділити увагу правильному вибору цього пристрою. Однією з вимог, що пред'являються до комп'ютера пульта централізованого спостереження, є можливість його цілодобової роботи без виключень.

Модульна структура програмного забезпечення спеціалізованої підсистеми охоронного спостереження за безпекою об’єктів дозволяє додавати до нього новий програмний код, який реалізує ті чи інші функціональні можливості без переробки і перекомпіляції вже створених і працюючих програмних модулів.

Інтерфейс спеціалізованої підсистеми охоронної безпеки об’єктів має одновіконну форму, яка реалізує можливість здійснення всіх необхідних функцій обробки сповіщень користувачем-оператором та здійснення редагування необхідної інформації користувачем-адміністратором. Головна форма розкривається в повноекранному режимі (рис. 4.1).

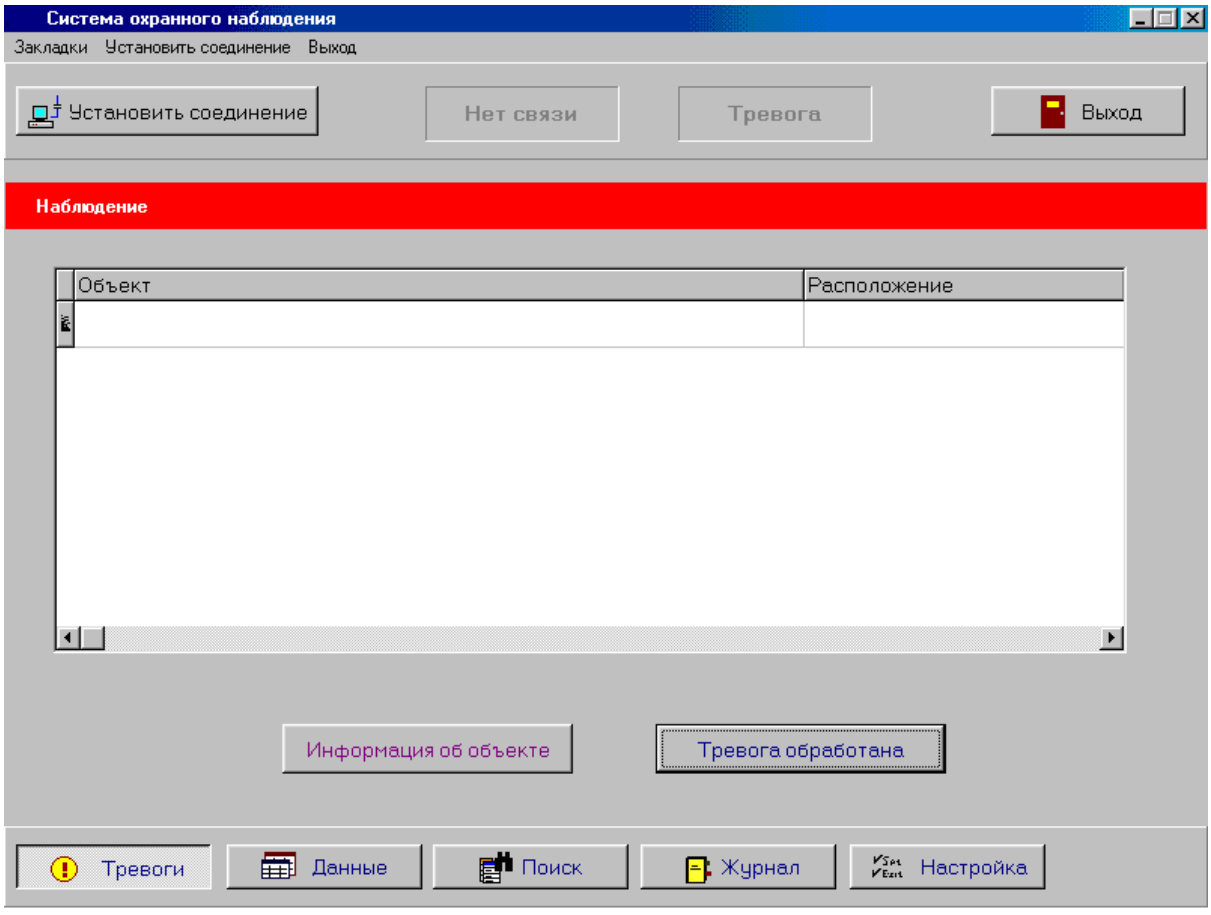

Рисунок 4.1 – Головна форма підсистеми охоронної безпеки об’єктів

Більшу частину форми займає активна вкладка, яка перемикається кнопками внизу екрану. Всього є п'ять вкладок: «Тривоги», «Дані», «Пошук», «Журнал», «Настройка». Залежно від прав користувача, який здійснив успішну ідентифікацію у підсистемі, деякі з них можуть бути недоступні. Кнопки перемикання закладок, як і інші кнопки, що знаходяться на головній формі, дублюються пунктами головного меню, що дозволяє вибирати їх за допомогою клавіатури.

У верхній частині головної форми знаходяться кнопка «Встановити з'єднання», яка міняється на кнопку «Розірвати з'єднання», якщо зв'язок по GSM-каналу з пультом централізованого спостереження «Дунай» в даний момент встановлено. Кнопка «Вихід» реалізує можливість завершення роботи з спеціалізованою підсистемою. Окрім цього, в центрі верхньої частини форми, знаходяться два індикатори «Немає зв'язку» і «Тривога», які становляться активними при виникненні цих подій. Підсистема передбачає реалізацію функції звукового сповіщення, якщо стався аварійний розрив зв'язку чи прийшло сповіщення о тривозі.

Спостереження за охоронною безпекою на об'єктах відбувається впродовж цілої доби або постійно, і, отже, з спеціалізованою підсистемою охоронної безпеки об’єкту будуть працювати декілька чергових користувачів-операторів, обов'язки яких передбачають здійснення реагування на повідомлення, що надходять з пульта централізованого спостереження «Дунай». Крім того, водночас інший користувач, який має повноваження адміністратора системи, має можливість редагування бази даних об’єкту.

Реалізація механізму аутентифікації в підсистемі забезпечує крім розмежування прав доступу ще і для підвищення відповідальності операторів при роботі зпідсистемою, оскільки основні дії оператора чи адміністратора підсистеми, такі як запуск і вихід з програми, установка з'єднання з ПЦС «Дунай» і його розрив протоколюються у журналі (рис. 4.2).

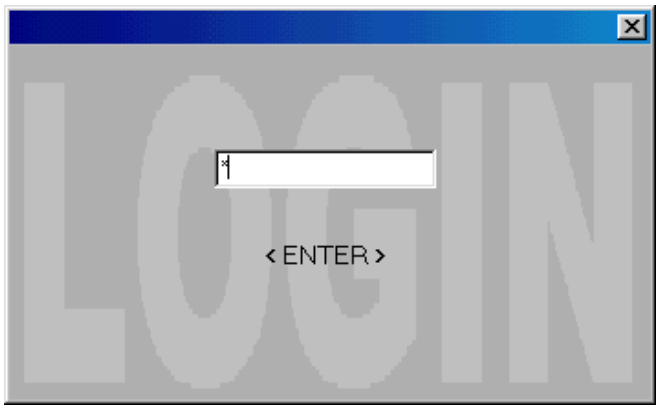

Рисунок 4.2– Вікно аутентифікації користувачів у підсистемі

Для здійснення редагування самих користувачів та розподілу прав доступу у системі одному з них привласнюється ознака «root». Користувач з цими правами може додавати користувачів і змінювати їх паролі.

Здійснюючи запуск програмного забезпечення спеціалізованої підсистеми охоронної безпеки об’єктів з'являється вікно аутентифікації користувача у системі, в якому користувач вводить свій пароль (паролі кожного користувача унікальні). Якщо пароль співпадає з тим, що зберігається в базі даних, відкривається головна форма програми, де згідно з розподілом прав доступу до інформації можуть бути недоступні деякі функціональні кнопки системи.

4.3 Реалізація спільної роботи спеціалізованої підсистеми з ПЗ «Дунай»

Першим кроком, після успішної ідентифікації користувачів у підсистемі, є запуск механізму, що реалізує інтеграцію розробленої спеціалізованої підсистеми з ПЦС «Дунай» для успішної передачі сповіщень. Користувач повинен встановити з’єднання з пультом централізованого спостереження «Дунай», щоб на робоче місце чергового користувача-оператора охори об’єкту приходили тривоги з структурних підрозділів цього територіально розподіленого об'єкту. При натисненні на кнопку «Встановити з'єднання», яка знаходиться на формі підсистеми, відкривається вікно в якому показані вільні COM-порти наявні в системі, з яких потрібно вибрати той, до якого підключений GSM-модем (рис. 4.3).

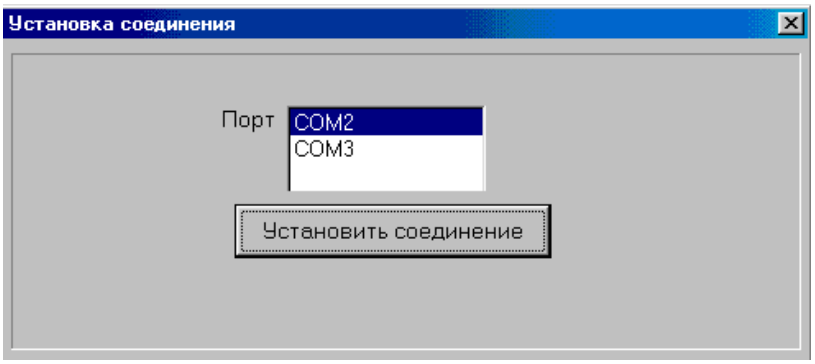

Рисунок. 4.3 – Установка з'єднання підсистеми з ПЦС «Дунай»

Після цього спеціалізована підсистема, яка віддалена від основного пульту, чекає контрольне повідомлення від пульта централізованого спостереження «Дунай» і посилає відповідь на нього. Механізм інтеграції спеціалізованої підсистеми також ілюструється відповідним вікном (рис. 4.4).

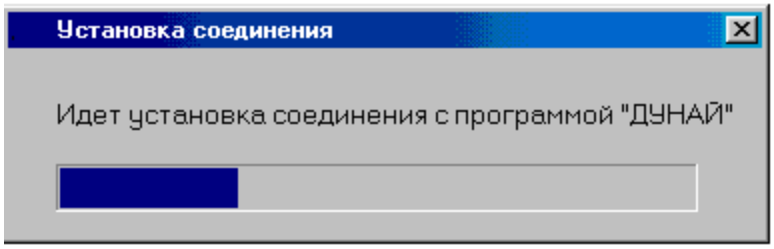

Рисунок. 4.4 – Очікування з’єднання з ПЦС «Дунай»

Якщо з'єднання успішно встановлене, кнопка «Встановити з'єднання» змінюється на «Розірвати з'єднання», яка активна, якщо користувач, який здійснює з’єднання, має повноваження в підсистемі здійснювати розрив з'єднання.

4.4 Реалізація функцій підсистеми

Спеціалізована підсистема охоронної безпеки об’єктів створена для здійснення передачі тривог з зазначеного об’єкту ПЦС «Дунай» для надання можливості швидкого реагування віддаленим територіально розподіленим черговим охорони цього об’єкту та тривоги, які надходять з структурних підрозділів цього об’єкту. Коли тривога від ПЦС «Дунай» поступає на робоче місце чергового оператора підсистеми, то підсистема здійснює пошук за кодом шлейфу (зони) в своїй базі даних об'єкт, якому належить цей шлейф. Якщо об'єкт знайдений, сповіщення додається у вікно тривог на

вкладці «Тривоги» і отримання тривоги відображається індикатором на головній формі (рис. 4.5).

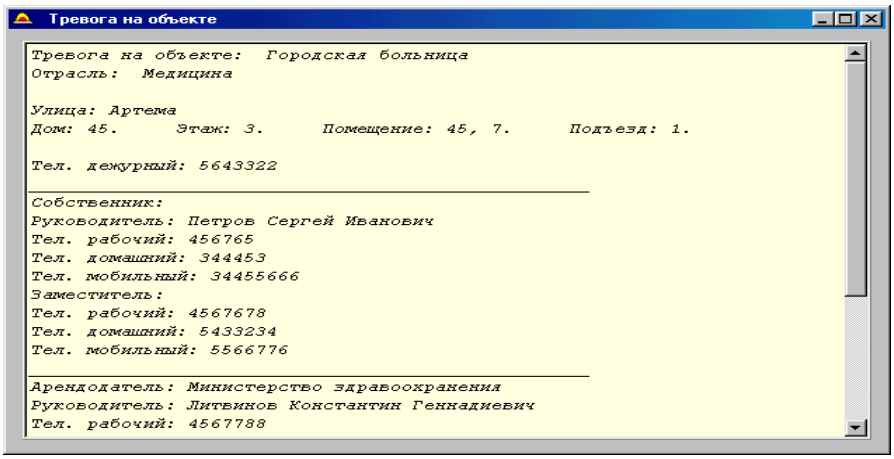

Рисунок. 4.5 – Інформаційне повідомлення про тривогу на об'єкті

Натиснувши на кнопку «Інформація про об'єкт» оператор може переглянути в окремому вікні всю інформацію про об'єкт, на якому відбулася тривога. Коли оператор зробив всі дії по обробці тривоги, він указує на закінчення обробки, натисненням на кнопку «Тривога оброблена».

Всі дії оператора з обов’язковим зазначенням часу записуються у журнал, який в змозі переглянути користувач-адміністратор.

Вкладка «Дані» доступна, якщо користувач, принаймні, має право здійснення перегляду бази даних. На цій вкладці можна обрати, яку з таблиць потрібно редагувати.

Після цього, для переходу до редагування потрібного запису таблиці, використовується кнопка «Редагування», для додавання – «Додати», для видалення – «Видалити». Під час внесення змін інші кнопки у вкладці недоступні, а перехід на іншу вкладку викликає вихід з режиму редагування або додавання. Для того, щоб правильно встановлювати зовнішні ключі (наприклад, вулицю для об'єкту) використовуються поля підстановки (рис. 4.6).

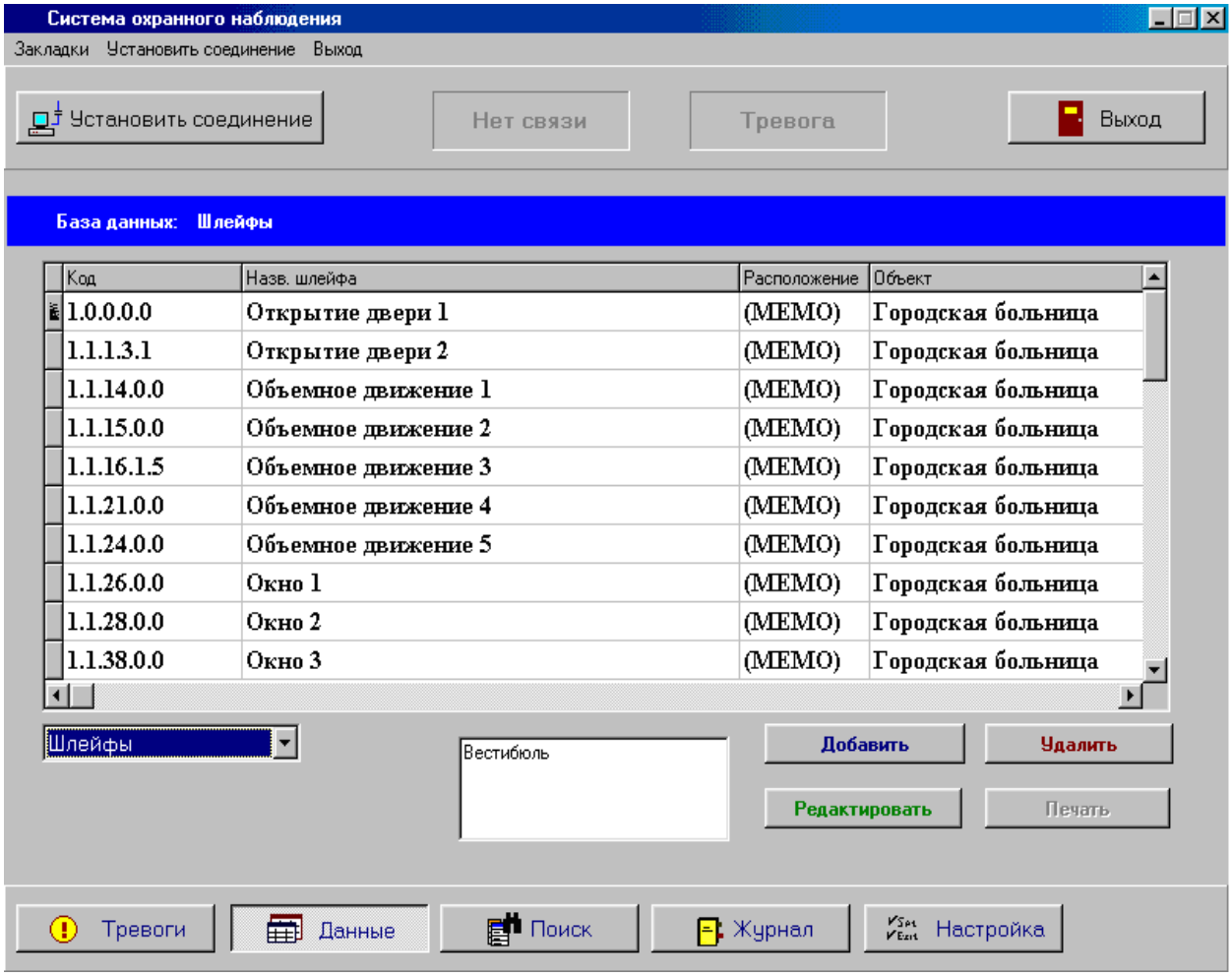

Рисунок. 4.6 – Редагування таблиці «Шлейфи» БД підсистеми

При виборі таблиці «Шлейфи» на вкладку додається багаторядковий редактор для відображення MEMO-поля додаткової інформації про шлейф.

За допомогою кнопки «Друк» можна роздрукувати дані з різних таблиць. Для таблиці «Об'єкти» обирається, яка ж сама інформація потрібна для друку: поточний об'єкт, всі об'єкти, або об'єкти, що відносяться до конкретної галузі. Разом з об'єктом обираються всі шлейфи сигналізації, встановлені на ньому. Після натиснення на кнопку «Друк» відкривається вікно, в якому можна переглянути, що саме буде роздруковане, встановити розмір шрифту, вибрати принтер і отримати доступ до настройок друку Windows.

Якщо права користувача підсистеми обмежені і не передбачають можливості здійснення яких би то дій з базою даних крім перегляду, то користувачу кнопки «Додати», «Редагувати» і «Видалити» будуть недоступні.

В спеціалізованій підсистемі охоронної безпеки об’єктів реалізована функція здійснення пошуку в базі даних для всіх таблиць і необхідних полів.

Початковою інформацією для пошуку є рядок, для якого перевіряється входження в поля пошуку таблиць без урахування позиції і регістра символів (рис. 4.7).

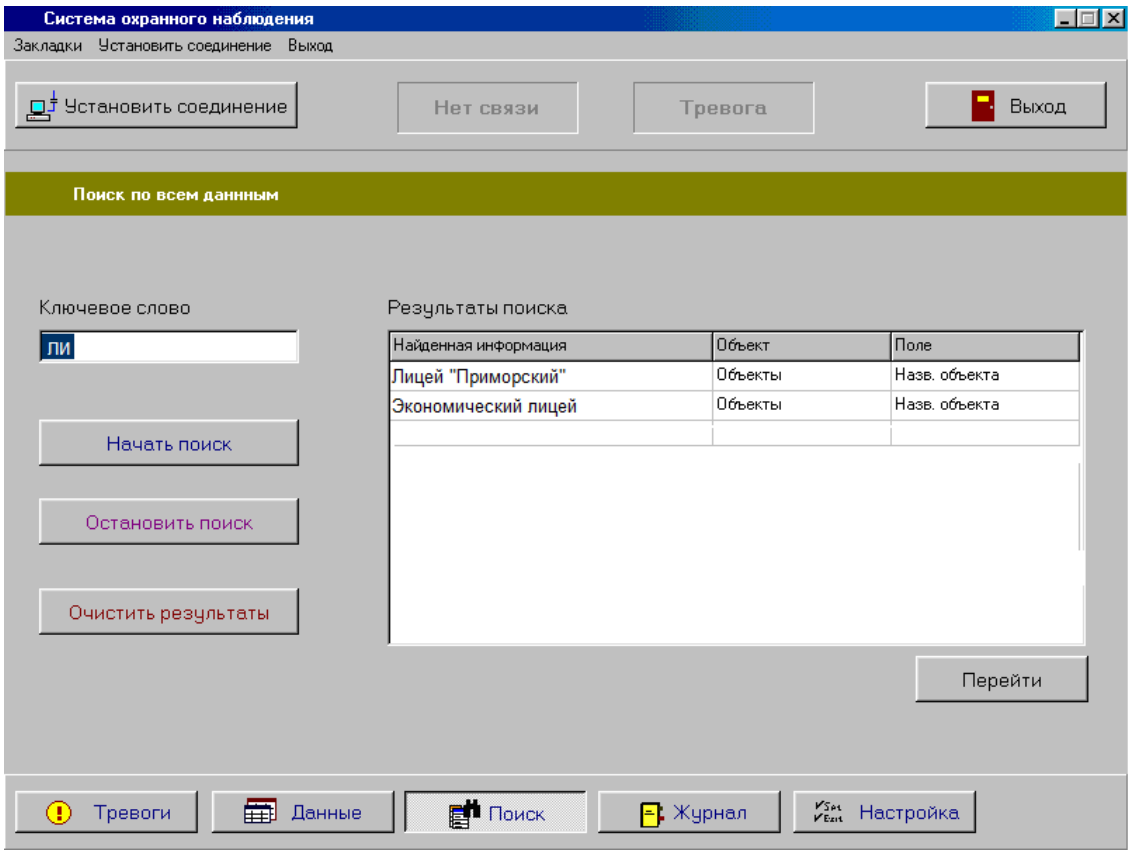

Рисунок. 4.7 – Організація пошуку в спеціалізованій підсистемі

Рядок пошуку і його результати відображаються у вкладці «Пошук». У вікні «Результати пошуку» крім значення поля, в якому знайдений підрядок представлена інформація про таблицю і саме поле в якому цей підрядок був знайдений. При виборі рядка в таблиці, що відображена у вікні «Результатів пошуку» і натисненні на кнопку «Перейти» відкриється вкладка «Дані» з тією таблицею, в якій була знайдена інформація. Курсор, в цій таблиці, стоятиме на потрібному рядку і полі. Якщо пошук займає тривалий час, що може бути тільки при дуже великих розмірах бази даних, його можна зупинити за допомогою кнопки «Зупинити пошук».

4.5 Реалізація обліку подій і тривог у підсистемі

Для забезпечення функціональних можливостей реалізації спеціалізованої підсистеми охоронної безпеки об’єктів, була здійснена програмна реалізація журналу обліку подій і тривог у підсистемі. Основним

призначенням цього журналу є зберігання з точним зазначення часу (з урахуванням секунд), всіх подій і тривог на обраному віддаленому територіально розподіленому об’єкті у системі. Це забезпечить протоколювання як всіх тривог, так і всіх дій користувачів у системі, які забезпечують швидку реакцію на подію (тривоги, втрата зв’язку і т.п.). У журнал заносяться різні події зафіксовані програмою. Їх можна розділити на кілька категорій:

- тривоги, які надійшли на пульт охоронного спостереження. Як тільки підсистема нагляду за охоронною безпекою об’єкту отримує повідомлення про тривогу від ПЦС «Дунай», вона видає повідомлення на екран монітора і вносить в журнал запис такого чину: «об'єкт, на якому сталася тривога; шлейф – ініціатор тривоги; розташування шлейфа; час (і дата) надходження тривоги;
- втрати і автоматичні відновлення зв'язку з ПЦС «Дунай». Такі події записуються в вигляді записів наступного формату: подія («втрата зв'язку з пультом центрального спостереження» / «відновлення зв'язку з пультом центрального спостереження») та час, в який відбулася подія;
- дії користувачів: «запуск програми одним з користувачів» (користувач ідентифікується по введенню пароля), «вихід з програми». При запуску або закриттю спеціалізованої підсистеми в журналі з'являється запис виду: «користувач «ІМ'Я» запустив програму» / «вихід з програми» та час запуску або виходу.

Здійснивши запуск спеціалізованої підсистеми охоронної безпеки об’єкту, користувач зазвичай встановлює з'єднання з пультом централізованого спостереження «Дунай» (якщо він не збирається, наприклад, просто редагувати базу даних або переглядати якусь статистику по журналу). Підсистема забезпечує спробу встановити з'єднання, і, якщо це вдається, вносить в журнал запис: «встановлено з'єднання з ПЦС «Дунай» та зазначає час, коли подія наступила. Якщо з'єднання встановити не вдалося, підсистема вносить в журнал запис протилежного змісту: «не вдалося встановити з'єднання з ПЦС «Дунай» з зазначенням часу. При здійсненні програмної реалізація журналу обліку подій та тривог, реалізована можливість здійснення користувачами підсистеми перегляду у цьому журналі повідомлення окремих груп, за окремий період (наприклад, добу чи місяць), а також друкування обраної інформації (рис. 4.8) .

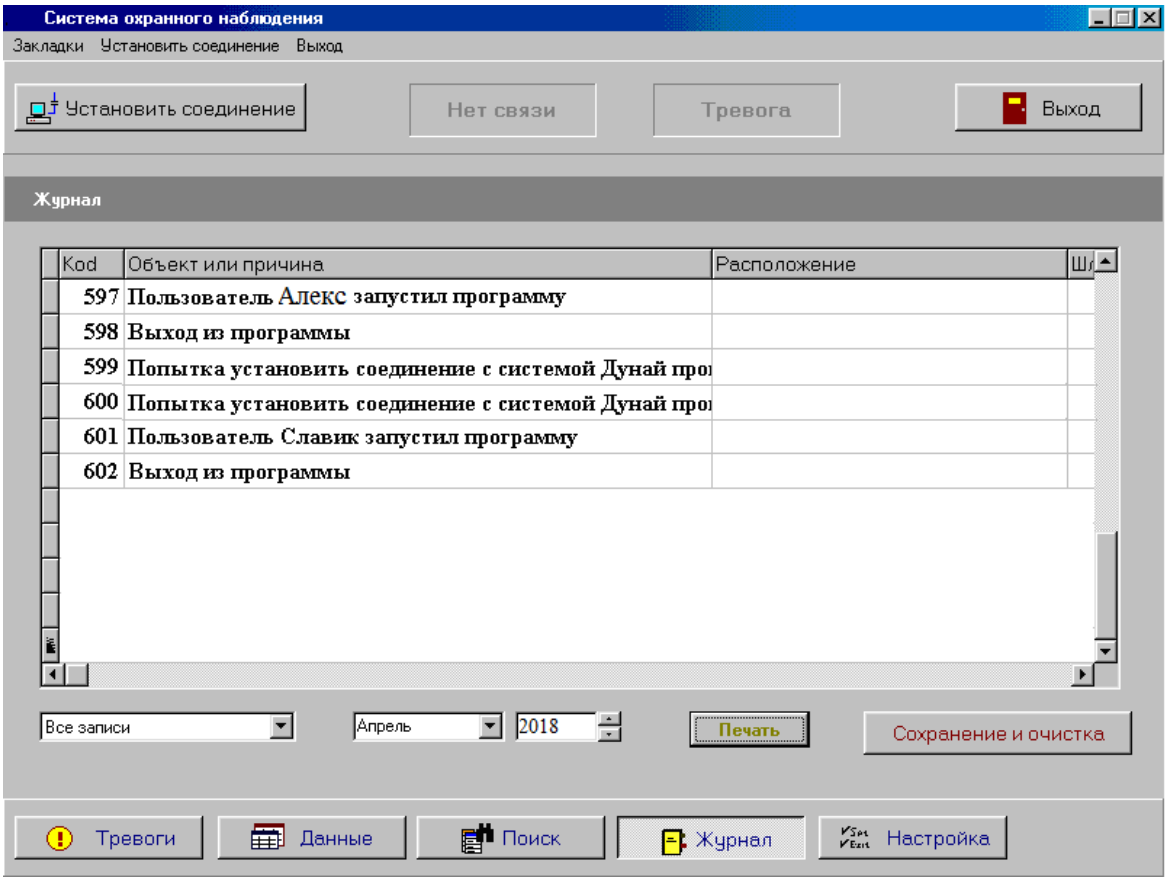

Рисунок. 4.8 – Журнал подій та тривог спеціалізованої підсистеми

Всі тривоги, що поступили з пульта централізованого спостереження «Дунай», і події, типу запуску, виходу, установки з'єднання, зберігаються в базі даних. Відкривши вкладку основної форми програми «Журнал» можна дістати доступ до цієї інформації. Слід зазначити, що ця вкладка доступна, якщо користувач має право поглядання журналу, якщо ж він має право редагування журналу, на цій вкладці стає доступною кнопка «Збереження і очищення». При виконанні цієї операції викликається програма Backup з підкаталогу Plugins каталогу програми, яка дозволяє зберегти журнал (у тому числі і в архіві), очистити його від застарілої інформації, зберегти всю базу даних. Інтерфейс спеціалізованої підсистеми охоронної безпеки об’єктів забезпечує можливість здійснення перегляду журналу, обравши потрібну категорію, записи з якої будуть відображені: «всі записи», «тривоги», «розриви чи установки з'єднання», «користувачі». Також можна вибрати за який місяць і рік потрібно проглянути записи (за умовчанням стоїть поточний місяць поточного року). Обрані записи можна роздрукувати, аналогічно інформації з бази даних.

В інтерфейсі спеціалізованої підсистеми охоронної безпеки об’єктів реалізована вкладка «Налаштування», яка надає деякі можливості настройки призначеного для користувача інтерфейсу окремо для кожного користувача, і забезпечує можливість здійснення додавання і редагування користувачів адміністратором (рис. 4.9).

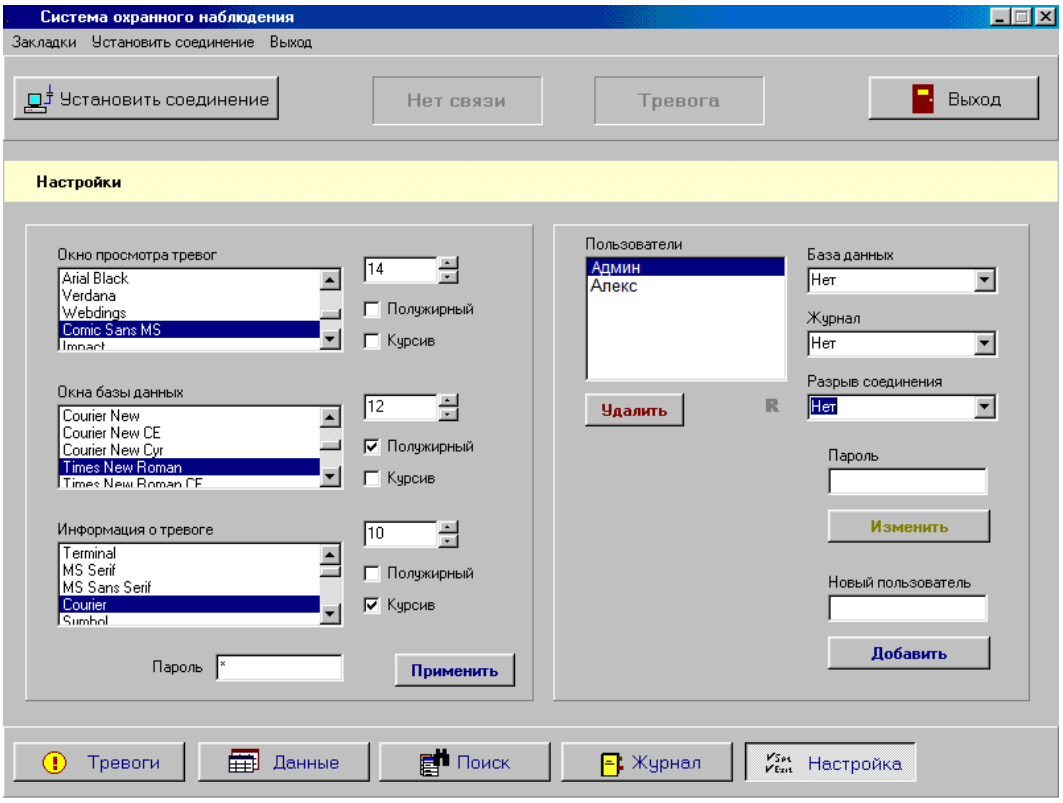

Рисунок. 4.9 – Налаштування параметрів інтерфейсу підсистеми

Вкладка розділена на дві частини – ліву і праву. Ліва частина – наявна та завжди показує доступні опції налаштування інтерфейсу для поточного користувача. Для вкладки «Перегляд тривог», вкладки «Вікно бази даних», і вікна «Перегляд тривог» про об'єкт, на якому відбулася тривога, можна встановити свій шрифт, його розмір і відображення.

Права частина головної форми наявна тільки якщо з програмою працює користувач-адміністратор. У цій частині реалізована можливість додавання або видалення користувача, крім того реалізована функція зміни паролів користувачів у системі.

Реалізований інтерфейс спеціалізованої підсистеми охоронної безпеки об’єктів забезпечує можливість надання наступних прав.

Для управління базою даних: «зміна / перегляд» – доступна вкладка «Дані» і всі елементи управління на ній; «тільки перегляд» – на вкладці

«Дані» доступні кнопки «Додати», «Редагувати», «Видалити»; «Немає» – відкрити вкладку «Дані» неможливо.

Для налаштування журналу подій: «Зміна / перегляд» – доступна вкладка «Журнал» і всі елементи управління на ній; «тільки перегляд» – вкладка «Журнал» доступна, кнопка «Збереження і очищення» – недоступна, «Немає» – відкрити вкладку «Журнал» неможливо.

Для управління з’єднанням з ПЦН «Дунай»: «Розрив з'єднання» – якщо дозволений, то користувач може розірвати з'єднання з ПЦН «Дунай».

Програмна реалізація спеціалізованої підсистеми охоронної безпеки об’єктів, здійснена в рамках магістерської роботи, забезпечує виконання всіх визначених функції. У складі програмного забезпечення спеціалізованої підсистеми охоронної безпеки об’єктів працюють додаткові програми, які знаходяться у підкаталозі Plugins каталогу підсистеми. Програма prsetup.exe – призначена для настройки принтерів, встановлених в системі. Виклик здійснюється засобами натиснення на кнопку «Налуштування» у вікні друку.

ВИСНОВКИ

В результаті виконання магістерської роботи здійснено проектування та програмна реалізація спеціалізованої підсистеми охоронної безпеки об’єктів, що реалізує функції віддаленого робочого місця чергового локального пульта охорони, розташованого безпосередньо на об’єкті. В роботі проведено дослідження та ананаліз предмета, цілей і особливостей функціонування пультів централізованої охорони об’єктів, визначено основні функції систем охоронної безпеки об’єктів.

На підставі здійсненого аналізу сформульовані вимоги до розробки програмного забезпечення спеціалізованої підсистеми охоронної безпеки об’єктів. Розроблена спеціалізована підсистема охоронної безпеки об’єктів забезпечує отримання тривог з зазначеного об’єкту через устаткування пульта централізованого спостереження «Дунай», що забезпечує можливість здійснення швидкого реагування на тривогу черговим підрозділом охорони цього об’єкту.

Підтимка функцій формування та зберігання журналу обліку подій і тривог у підсистемі забезпечує зберігання з точним зазначення часу, всіх подій і тривог на обраному віддаленому територіально розподіленому об’єкті. Спеціалізована підсистема охоронної безпеки об’єктів забезпечує спільне функціонування з програмним забезпеченням системи «Дунай», засобами протоколу зв’язку, що встановлює формат передачі подій та тривог з теріторіально віддаленого об’єкту охорони. Використання спеціалізованої підсистеми охоронної безпеки об’єктів надає можливість оператору чергового підрозділа охорони зазначеного об’єкта максимально оперативно і безпомилково реагувати на тривоги, що надходять, надаючи всю необхідну інформацію про об'єкт охорони, з якого поступила тривога.

Розроблена спеціалізована підсистема охоронної безпеки об’єктів відповідає всім основним вимогам, а програмного забезпечення дозволяє додавати нові функції без переробки вже створених підпрограм.

ПЕРЕЛІК ПОСИЛАНЬ

1. Современные системы сигнализации. [Электронный ресурс] – Режим доступа: https: // bezpeka.com.ua /reviews /sovremennye – sistemy -signaliza tsii?gclid=EAIaI/
2. Системы охранной сигнализации. [Электронный ресурс] – Режим доступа: http://www.firma-rubezh.com.ua.
3. Обслуживание охранной сигнализации. [Электронный ресурс] – Режим доступа: http://alarmtelecom.net/safe.htm
4. Руководство пользователя программного обеспечения комплексной интегрированной системы централизованной охраны «Дунай». – Киев: НВФ «Венбест Л.т.д.» 2007. – 64 с.
5. Каталог разработок научно-внедренческой фирмы «Венбест». – Киев: НВФ «Венбест Л.т.д.», 2010. – 70 с.
6. Прибор приемно-контрольный охранно-пожарный «Дунай». Руководство по эксплуатации. – Киев: НВФ «Венбест», 2015. – 53 с.
7. Деревянко А.С., Солощук М.Н. Операционные системы. – Харьков: НТУ «ХПИ», 2003 – 573 с.
8. Грофф Дж. Р., Вайнберг П.Н., Оппелъ Э. Дж. SQL. Полное руководство. – Издательство «Вильямс», 2015. – 959 с.
9. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: «Питер», 2001. – 304 с.
10. Тереза Нейл, Билл Скотт. Проектирование веб-интерфейсов. – СПб.: «Символ-Плюс», 2010. – 352 с.
11. Алексеев Е. Р., Чеснокова О. В., Кучер Т. В. Free Pascal и Lazarus: Учебник по программированию. – М.: Издательство «ДМК-пресс», 2010. – 442 с.
12. Мансуров К.Т. Основы программирования в среде Lazarus. – М.: Издательство «Вильямс», 2010. – 772с.
13. Скляр А. Я. Введение в InterBase. – М.: Издательство «Горячая Линия – Телеком», 2002. – 520 с.
14. Ковязин А., Востриков С. Мир InterBase. Архитектура, администрирование и разработка приложений баз данных в InterBase/Firebird/Yaffil. Издание 2-е, дополненное – М.: КУДИЦ-ОБРАЗ, 2002. – 496 с.

ДОДАТОК А ОСНОВНІ КОДИ ПРОГРАМНИХ МОДУЛІВ

```
int CALLBACK EnumFontFamProc1( LOGFONT* lpelf,  TEXTMETRIC* lpntm, int
FontType, LPARAM  lParam)
{
   Form1->ListBox1->Items->Add((char*)(lpelf->lfFaceName));
   Form1->ListBox2->Items->Add((char*)(lpelf->lfFaceName));
   Form1->ListBox3->Items->Add((char*)(lpelf->lfFaceName));
   return 1;
}

void TForm1::SetSetting()
{
  DBGrid2->Font->Name = ListBox1->Items->Strings[ListBox1->ItemIndex];
  DBGrid1->Font->Name = ListBox2->Items->Strings[ListBox2->ItemIndex];
  DBGrid3->Font->Name = ListBox2->Items->Strings[ListBox2->ItemIndex];
  Alert->Memo1->Font->Name     =     ListBox3->Items->Strings[ListBox3-
>ItemIndex];
  DBGrid2->Font->Size = UpDown1->Position;
  DBGrid1->Font->Size = UpDown2->Position;
  DBGrid3->Font->Size = UpDown2->Position;
  Alert->Memo1->Font->Size = UpDown3->Position;
  DBGrid1->Font->Style = TFontStyles();
  DBGrid3->Font->Style = TFontStyles();
  if (CheckBox3->Checked) {
    DBGrid1->Font->Style = TFontStyles()<< fsBold;
    DBGrid3->Font->Style = TFontStyles()<< fsBold;
  }
  if (CheckBox4->Checked) {
    DBGrid1->Font->Style = TFontStyles()<< fsItalic;
    DBGrid3->Font->Style = TFontStyles()<< fsItalic;
  }
  DBGrid2->Font->Style = TFontStyles();
  if   (CheckBox1->Checked)   DBGrid2->Font->Style   =   TFontStyles()<<
fsBold;
  if   (CheckBox2->Checked)   DBGrid2->Font->Style   =   TFontStyles()<<
fsItalic;
  Alert->Memo1->Font->Style = TFontStyles();
  if  (CheckBox5->Checked)  Alert->Memo1->Font->Style  =  TFontStyles()<<
fsBold;
  if  (CheckBox6->Checked)  Alert->Memo1->Font->Style  =  TFontStyles()<<
fsItalic;
}
void TForm1::LoadSetting()
{
  int l,k;
  AnsiString sa[12];
  tAccess->SetKey();
  tAccess->FieldByName("UserName")->AsString = CurrentUser;
  tAccess->GotoKey();
  AnsiString s = tAccess->FieldByName("Setting")->AsString;
  for (int i=0; i<12; i++) {
    l = s.Pos("(");
    k = s.Pos(")");
    sa[i] = s.SubString(l, k - l + 1);
    s.Delete(l, k - l + 1);
    sa[i].Delete(1,1);
```

```
    sa[i].Delete(sa[i].Length(), 1);
  }
  for (int i=0; i<ListBox1->Items->Count; i++)
f (ListBox1->Items->Strings[i] == sa[0]) ListBox1->ItemIndex = i;
for (int i=0; i<ListBox2->Items->Count; i++)
if (ListBox2->Items->Strings[i] == sa[4])ListBox2->ItemIndex = i;
for (int i=0; i<ListBox3->Items->Count; i++)
if (ListBox3->Items->Strings[i] == sa[8])ListBox3->ItemIndex = i;
  UpDown1->Position    =    sa[1].ToInt();    UpDown2->Position    =
sa[5].ToInt(); UpDown3->Position = sa[9].ToInt();
  if (sa[2] == "yes") CheckBox1->Checked = true; else CheckBox1-
>Checked = false;
  if (sa[3] == "yes") CheckBox2->Checked = true; else CheckBox2-
>Checked = false;
  if (sa[6] == "yes") CheckBox3->Checked = true; else CheckBox3-
>Checked = false;
  if (sa[7] == "yes") CheckBox4->Checked = true; else CheckBox4-
>Checked = false;
  if (sa[10] == "yes") CheckBox5->Checked = true; else CheckBox5-
>Checked = false;
  if (sa[11] == "yes") CheckBox6->Checked = true; else CheckBox6-
>Checked = false;

  Edit4->Text = tAccess->FieldByName("Password")->AsString;
  Edit4Exit(Edit4);
  SetSetting();
}
void TForm1::SaveSetting()
{
  int l,k;
  AnsiString sa[12];
  tAccess->SetKey();
  tAccess->FieldByName("UserName")->AsString = CurrentUser;
  tAccess->GotoKey();
  AnsiString s = "";
  s += "(" + ListBox1->Items->Strings[ListBox1->ItemIndex] + ")";
  s += "(" + Edit5->Text + ") ";
  if (CheckBox1->Checked) s += "(yes) "; else s += "(no) ";
  if (CheckBox2->Checked) s += "(yes) "; else s += "(no) ";
  s += " ";
  s += "(" + ListBox2->Items->Strings[ListBox2->ItemIndex] + ")";
  s += "(" + Edit6->Text + ") ";
  if (CheckBox3->Checked) s += "(yes) "; else s += "(no) ";
  if (CheckBox4->Checked) s += "(yes) "; else s += "(no) ";
  s += " ";
  s += "(" + ListBox3->Items->Strings[ListBox3->ItemIndex] + ")";
  s += "(" + Edit7->Text + ") ";
  if (CheckBox5->Checked) s += "(yes) "; else s += "(no) ";
  if (CheckBox6->Checked) s += "(yes) "; else s += "(no) ";
  tAccess->Edit();
  tAccess->FieldByName("Setting")->AsString = s;
  tAccess->Post();
  tAccess->Edit();
  tAccess->FieldByName("Password")->AsString = Edit4->Text;
  tAccess->Post();
    SetSetting();
}
__fastcall TForm1::TForm1(TComponent* Owner)
```

```
        : TForm(Owner)
{
}
void __fastcall TForm1::ComboBox1Change(TObject *Sender)
{
  BitBtn15->Enabled = true;
  if (ComboBox1->Text == "Объекты")
    DataSource1->DataSet = tObject;
  if (ComboBox1->Text == "Улицы")
    DataSource1->DataSet = tStreet;
  if (ComboBox1->Text == "Оборудование")
    DataSource1->DataSet = tEquipment;
  if (ComboBox1->Text == "Тип объекта")
    DataSource1->DataSet = tBranche;
  if (ComboBox1->Text == "Обслуживающая организация")
    DataSource1->DataSet = tServices;
  if (ComboBox1->Text == "Шлейфы")
    DataSource1->DataSet = tZones;
  if (DataSource1->DataSet == tZones) {
        Memo1->Show();
        tZones->First();
        BitBtn15->Enabled = false;
  }
  else Memo1->Hide();
  Panel3->Caption="        База данных:   " + ComboBox1->Text + " ";
}
void __fastcall TForm1::FormShow(TObject *Sender)
{
  HDC hdc = GetDC(Form1->Handle);
  EnumFonts(hdc, 0 , (int(*)())EnumFontFamProc1,0);
  Login->ShowModal();
  CurrentUser = tAccess->FieldByName("UserName")->AsString;
  LoadSetting();
  DBLookupListBox1->KeyValue    =    tAccess->FieldByName("UserName")-
>AsString;
  if (!tAccess->FieldByName("Root")->AsBoolean) Panel10->Hide();
  Query1->SQL->Text = "INSERT INTO Alert (Object, DatTime, Obrabotana)
VALUES (\"Пользователь " +  tAccess->FieldByName("UserName")->AsString
+ " запустил программу\", \"" + Now().DateTimeString() + "\", \"y\")";
  Query1->ExecSQL();
  Finded1 = new TStringList;
  Finded2 = new TStringList;
  DbPanel->Left = 1;
  DbPanel->Top = 80;
  SettingPanel->Left = 1;
  SettingPanel->Top = 80;
  AlertPanel->Left = 1;
  AlertPanel->Top = 80;
  FindPanel->Left = 1;
  FindPanel->Top = 80;
  LogPanel->Left = 1;
  LogPanel->Top = 80;
  ComboBox1->ItemIndex = 0;
  Panel3->Caption="        База данных:   " + ComboBox1->Text + " ";
  SpeedButton1->Click();
  SpeedButton1->Down = true;

  TDateTime dt;
```

```
  dt = Now();
  unsigned short year, month, day;
  dt.DecodeDate(&year, &month, &day);
  ComboBox6->ItemIndex = month - 1;
  UpDown4->Position = year;
  ComboBox5->ItemIndex = 0;
  ComboBox6Change(ComboBox6);
  ComboBox5Change(ComboBox5);
}
void __fastcall TForm1::SpeedButton2Click(TObject *Sender)
{
  AlertPanel->Visible = false;
  SettingPanel->Visible = false;
  FindPanel->Visible = false;
  LogPanel->Visible = false;
  DbPanel->Visible= true;
}
void __fastcall TForm1::SpeedButton3Click(TObject *Sender)
{
  AlertPanel->Visible = false;
  DbPanel->Visible= false;
  FindPanel->Visible = false;
  LogPanel->Visible = false;
  Edit3->Text = "";
  SettingPanel->Visible = true;
}
void __fastcall TForm1::SpeedButton1Click(TObject *Sender)
{
  SettingPanel->Visible = false;
  DbPanel->Visible= false;
  FindPanel->Visible = false;
  LogPanel->Visible = false;
  AlertPanel->Visible = true;
}
void __fastcall TForm1::SpeedButton4Click(TObject *Sender)
{
  ExitWhile = false;
  if (SpeedButton4->Caption == "Установить соединение") {
    Form2->ShowModal();
    if (Form2->ModalResult == mrOk) {
        hCom    =    CreateFile((Form2->ListBox1->Items->Strings[Form2-
>ListBox1->ItemIndex]).c_str(),
                GENERIC_READ , 0, NULL,
                OPEN_EXISTING, FILE_FLAG_OVERLAPPED, NULL);
        if (hCom == INVALID_HANDLE_VALUE) return;
        GetCommState(hCom, &dd);
        dd.BaudRate = 9600;
        SetCommState(hCom, &dd);
        o.hEvent = CreateEvent(NULL, FALSE, FALSE, NULL);
        ReadFile(hCom, buf, 13, &res, &o);
        Form4->ProgressBar1->Position = 0;
        Form1->Enabled = false;
        Form4->Show();
        Application->ProcessMessages();
        for (int i=1; i <= 650; i++) {
          Sleep(100);
          if (GetOverlappedResult(hCom, &o, &res, 0)) {
            Form1->Enabled = true;
```

```
            Form4->Hide();
            Query1->SQL->Text = "INSERT INTO Alert (Object, DatTime,
Obrabotana) VALUES (\"Установлено соединение с системой Дунай\" , \""
+ Now().DateTimeString() + "\", \"y\")";
            Query1->ExecSQL();
            break;
          };
          Form4->ProgressBar1->Position = (i*1000) / 650;
          Application->ProcessMessages();
          if (i == 650) {
            FileClose((int)hCom);
            ExitWhile = true;
            Form1->Enabled = true;
            Form4->Hide();
            Query1->SQL->Text = "INSERT INTO Alert (Object, DatTime,
Obrabotana) VALUES (\"Попытка установить соединение с системой Дунай
провалилась\" , \"" + Now().DateTimeString() + "\", \"y\")";
            Query1->ExecSQL();
            return ;
          }
        }
        ReadFile(hCom, buf, 13, &res, &o);
        Timer1->Enabled = true;
        SpeedButton4->Caption = "Разорвать соединение";
        N5->Caption = SpeedButton4->Caption;
        if (!RightDisconnect) {
          SpeedButton4->Enabled = false;
          N5->Enabled = false;
        }
       while(1) {
         if (ExitWhile) break;
         if (ExitProgram) {
           FileClose((int)hCom);
           return;
         }
         Application->ProcessMessages();
         if (GetOverlappedResult(hCom, &o, &res, 0)) {
         if (CheckBox7->Checked) {
           CheckBox7->Checked = false;
           buf[0]='1'; buf[1]='1'; buf[2]='0'; buf[3]='0';
           buf[4]='1'; buf[5]='0'; buf[6]='0'; buf[7]='1';
           buf[8]='0'; buf[9]='5'; buf[10]='1'; buf[11]='8';
           buf[12]=13;*/
           if (AnsiString(buf) != "FF000000001E\0xD") {
                AnsiString s, s1, s2, s3, s4, s5;
                s1 = AnsiString("0x")+buf[0];
                s2 = AnsiString("0x")+buf[1];
                s3 = AnsiString("0x")+buf[2]+buf[3]+buf[4]+buf[5];
                s4 = AnsiString("0x")+buf[6]+buf[7];
                s5 = AnsiString("0x")+buf[8]+buf[9];
                s = IntToStr(StrToInt(s1))+"."+
                IntToStr(StrToInt(s2))+"."+
                IntToStr(StrToInt(s3))+"."+
                IntToStr(StrToInt(s4))+"."+
                IntToStr(StrToInt(s5));
                tZones->Close();
                tZones->Filter = "Code='"+s+"'";
                tZones->Filtered = true;
```

```
                tZones->Open();
                if (tZones->RecordCount != 0) {
                  AnsiString s = "INSERT INTO Alert (Object, Location,
Zone,     DatTime,     Obrabotana)     VALUES     (\""     +     tZones-
>FieldByName("lokObject")->AsString +
                                  "\",        \""        +        tZones-
>FieldByName("Lacation")->AsString +
                                  "\",        \""        +        tZones-
>FieldByName("Code")->AsString +
                                  "\",  \""  +  Now().DateTimeString()  +
"\", \"n\")";
                  Query1->SQL->Text = s;
                  Query1->ExecSQL();
                  qAlerts->DisableControls();
                  qAlerts->Close();
                  qAlerts->Open();
                  qAlerts->EnableControls();
                  TrevogaON();
                  SpeedButton1->Click();
                  SpeedButton1->Down = true;
                }
                tZones->Close();
                tZones->Filter = "";
                tZones->Filtered = false;
                tZones->Open();
            }
//                        if  (!WriteFile(hCom,  &buf2,  1,  &res,  0))
MessageBox(0,"error","error",0);;
            WriteFile(hCom, &buf2, 1, &res, &o);
            while (!GetOverlappedResult(hCom, &o, &res, 0)) ;
            ReadFile(hCom, buf, 13, &res, &o);
            Timer1->Enabled = false;
            Timer1->Enabled = true;
            if (Panel6->Font->Color == clWhite) {
              Query1->SQL->Text = "INSERT INTO Alert (Object, DatTime,
Obrabotana) VALUES (\"Связь с системой Дунай восстановилась\" , \"" +
Now().DateTimeString() + "\", \"y\")";
              Query1->ExecSQL();
            }
            NoconnectOFF();
          }
        }
      }
    }
    else {
      if (!RightDisconnect) return;
      FileClose((int)hCom);
      NoconnectOFF();
      SpeedButton4->Caption = "Установить соединение";
      N5->Caption = SpeedButton4->Caption;
      Query1->SQL->Text   =   "INSERT   INTO   Alert   (Object,   DatTime,
Obrabotana) VALUES (\"Пользователь разорвал связь с системой Дунай\" ,
\"" + Now().DateTimeString() + "\", \"y\")";
      Query1->ExecSQL();
      Timer1->Enabled = false;
      ExitWhile = true;
    }
}
```

```
void __fastcall TForm1::BitBtn3Click(TObject *Sender)
{
  if (qAlerts->RecordCount != 0) {
    if (Application->MessageBox("Тревога обработана ?", "Подтвердите",
1) != IDOK)
      return;
    Query1->SQL->Text = "UPDATE Alert SET Obrabotana = \"y\" WHERE Kod
="+qAlerts->FieldByName("Kod")->AsString;
    Query1->ExecSQL();
    qAlerts->DisableControls();
    qAlerts->Close();
    qAlerts->Open();
    qAlerts->EnableControls();
  }
  if (qAlerts->RecordCount == 0)
    TrevogaOFF();

}
void __fastcall TForm1::SpeedButton5Click(TObject *Sender)
{
  Query1->SQL->Text = "INSERT INTO Alert (Object, DatTime, Obrabotana)
VALUES (\"Выход из программы\", \"" + Now().DateTimeString() + "\",
\"y\")";
  Query1->ExecSQL();
  ExitProgram = true;
  Close();
}
void __fastcall TForm1::N3Click(TObject *Sender)
{
  SpeedButton2->Click();
  SpeedButton2->Down = true;
}
void __fastcall TForm1::N2Click(TObject *Sender)
{
  SpeedButton1->Click();
  SpeedButton1->Down = true;
}
void __fastcall TForm1::N4Click(TObject *Sender)
{
  SpeedButton3->Click();
  SpeedButton3->Down = true;
}
void __fastcall TForm1::N6Click(TObject *Sender)
{
  SpeedButton5Click(SpeedButton5);
}
void __fastcall TForm1::N5Click(TObject *Sender)
{
  SpeedButton4Click(SpeedButton4);
}
void __fastcall TForm1::Timer1Timer(TObject *Sender)
{
  if (Panel6->Font->Color == clWhite) return;
  Query1->SQL->Text = "INSERT INTO Alert (Object, DatTime, Obrabotana)
VALUES (\"Потеряна связь с системой Дунай\" , \"" +
Now().DateTimeString() + "\", \"y\")";
  Query1->ExecSQL();
  NoconnectON();
```

```
}
void __fastcall TForm1::BitBtn1Click(TObject *Sender)
{
  DataSource1->DataSet->Insert();
  BitBtn14Click(BitBtn14);
}
void __fastcall TForm1::BitBtn2Click(TObject *Sender)
{
  DataSource1->DataSet->Delete();
}
void __fastcall TForm1::tZonesAfterScroll(TDataSet *DataSet)
{
        Memo1->Lines->Clear();
        Memo1->Lines->Text      =      tZones->FieldByName("Lacation")-
>AsString;
}
void __fastcall TForm1::tZonesBeforeScroll(TDataSet *DataSet)
{
        if (Memo1->Lines->Text == "") return;
        tZones->Edit();
        tZones->FieldByName("Lacation")->AsString    =    Memo1->Lines-
>Text;
        tZones->Post();
}
void __fastcall TForm1::vcdc(TObject *Sender)
{
  SettingPanel->Visible = false;
  DbPanel->Visible= false;
  AlertPanel->Visible = false;
  LogPanel->Visible = false;
  FindPanel->Visible = true;
}
void __fastcall TForm1::BitBtn7Click(TObject *Sender)
{
  Finded1->Clear();
  Finded2->Clear();
  StringGrid1->RowCount = 2;
  StringGrid1->Cells[0][0] = "Найденная информация";
  StringGrid1->Cells[1][0] = "Объект";
  StringGrid1->Cells[2][0] = "Поле";
  StringGrid1->Cells[0][1] = "";
  StringGrid1->Cells[1][1] = "";
  StringGrid1->Cells[2][1] = "";
}
void  TForm1::FindFields(TDataSet*  dataset,  AnsiString  datasetName,
AnsiString field, bool like, AnsiString Key)
{
  AnsiString sField;
  Key = Key.UpperCase();
  AnsiString s = dataset->FieldByName(field)->DisplayLabel;
  dataset->First();
  while (!dataset->Eof) {
    Application->ProcessMessages();
    if (StopFind) return;
    sField = dataset->FieldByName(field)->AsString;
    sField = sField.UpperCase();
    if (!like && sField == Key || like && sField.Pos(Key) !=0) {
```

```
      StringGrid1->Cells[0][StringGrid1->RowCount-1]     =     dataset-
>FieldByName(field)->AsString;
      StringGrid1->Cells[1][StringGrid1->RowCount-1] = datasetName;
      StringGrid1->Cells[2][StringGrid1->RowCount-1] = s;
      int k = dataset->Fields->IndexOf(dataset->FieldByName(field));
      Finded1->Add(AnsiString(dataset->RecNo - 1));
      Finded2->Add(IntToStr(k));
      StringGrid1->RowCount++;
      StringGrid1->Cells[0][StringGrid1->RowCount-1] = "";
      StringGrid1->Cells[1][StringGrid1->RowCount-1] = "";
      StringGrid1->Cells[2][StringGrid1->RowCount-1] = "";
    }
    dataset->Next();
  }
  return;
}
void __fastcall TForm1::BitBtn5Click(TObject *Sender)
{
  StopFind = false;
  BitBtn5->Enabled = false;
  BitBtn7Click(BitBtn7);
  FindFields(tObject, "Объекты", "Full_Name", 1, Edit2->Text);
  FindFields(tObject, "Объекты", "Owner", 1, Edit2->Text);
  FindFields(tObject, "Объекты", "Chif_Name", 1, Edit2->Text);
  FindFields(tObject, "Объекты", "Chif_PhoneW", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "Chif_PhoneH", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "Chif_PhoneM", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "Depu_Name", 1, Edit2->Text);
  FindFields(tObject, "Объекты", "Depu_PhoneW", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "Depu_PhoneH", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "Depu_PhoneM", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "Rentor", 1, Edit2->Text);
  FindFields(tObject, "Объекты", "RChif_Name", 1, Edit2->Text);
  FindFields(tObject, "Объекты", "RChif_PhoneW", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "RChif_PhoneH", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "RChif_PhoneM", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "RDepu_Name", 1, Edit2->Text);
  FindFields(tObject, "Объекты", "RDChif_PhoneW", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "RDChif_PhoneH", 0, Edit2->Text);
  FindFields(tObject, "Объекты", "RDChif_PhoneM", 0, Edit2->Text);
  FindFields(tBranche, "Тип объекта", "Name", 1, Edit2->Text);
  FindFields(tStreet, "Улицы", "Name", 1, Edit2->Text);
  FindFields(tEquipment, "Оборудование", "Name", 1, Edit2->Text);
  FindFields(tServices, "Обслуживающая организация", "Name", 1, Edit2-
>Text);
  FindFields(tServices, "Обслуживающая организация", "Chif_Name", 1,
Edit2->Text);
  FindFields(tServices, "Обслуживающая организация", "Chif_PhoneW", 0,
Edit2->Text);
  FindFields(tServices, "Обслуживающая организация", "Chif_PhoneH", 0,
Edit2->Text);
  FindFields(tServices, "Обслуживающая организация", "Chif_PhoneM", 0,
Edit2->Text);
  BitBtn5->Enabled = true;
}
void __fastcall TForm1::BitBtn6Click(TObject *Sender)
{
  StopFind = true;
```

```
}
void __fastcall TForm1::BitBtn8Click(TObject *Sender)
{
  __try  {
    TTable* t1 = NULL;
    AnsiString s1 = StringGrid1->Cells[1][StringGrid1->Row];
    if (s1 == "Объекты")
      t1 = tObject;
    if (s1 == "Улицы")
      t1 = tStreet;
    if (s1 == "Оборудование")
      t1 = tEquipment;
    if (s1 == "Тип объекта")
      t1 = tBranche;
    if (s1 == "Обслуживающая организация")
      t1 = tServices;
    if (s1 == "Шлейфы")
      t1 = tZones;
    if (t1 == NULL) return;
    SpeedButton2->Click();
    SpeedButton2->Down = true;
    ComboBox1->ItemIndex = ComboBox1->Items->IndexOf(s1);
    ComboBox1Change(ComboBox1);
         t1->First();
         t1->MoveBy(StrToInt(Finded1->Strings[StringGrid1->Row-1]));
         int k = StrToInt(Finded2->Strings[StringGrid1->Row-1]);
         DBGrid1->SelectedIndex = k;
         DBGrid1->SetFocus();
       }
       __finally {};
     }
     void __fastcall TForm1::SpeedButton7Click(TObject *Sender)
     {
       qLog->Close();
       qLog->Open();
       SettingPanel->Visible = false;
       DbPanel->Visible= false;
       AlertPanel->Visible = false;
       FindPanel->Visible = false;
       LogPanel->Visible = true;
     }
     void __fastcall TForm1::BitBtn9Click(TObject *Sender)
     {
     //  Query1->SQL->Text = "DELETE FROM Alert WHERE DatTime <= \"" +
DateTimePicker1->Date.DateString() + "\"";
     /*  Query1->ExecSQL();
       qLog->DisableControls();
       qLog->Close();
       qLog->Open();
       qLog->EnableControls(); */
       int   i   =   (int)ShellExecute(0,"open",   "Backup.exe",   ""   ,
"Plugins", SW_SHOW);
     //  if (i==ERROR_FILE_NOT_FOUND)
     //    MessageBox(0,"dfdfd","dfdfdf",0);
     }
     void __fastcall TForm1::BitBtn4Click(TObject *Sender)
     {
       AnsiString s;
```

```
        if(qAlerts->RecordCount == 0) return;
        Alert->Memo1->Lines->Clear();
        tObject->Close();
        tObject->Filter      =      "Full_Name='"      +      qAlerts-
>FieldByName("Object")->AsString + "'";
        tObject->Filtered = true;
        tObject->Open();
        s    =    "Тревога    на    объекте:        "    +    tObject-
>FieldByName("Full_Name")->AsString;
        Alert->Memo1->Lines->Add(s);
        s   =   "Отрасль:     "   +   tObject->FieldByName("BranchLook")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        Alert->Memo1->Lines->Add("");
        s = "Улица: " + tObject->FieldByName("StreetLook")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Дом: " + tObject->FieldByName("A_Home")->AsString +
            ".     Этаж: " + tObject->FieldByName("A_Flow")->AsString +
            ".      Помещение: " + tObject->FieldByName("A_Apartment")-
>AsString +
            ".        Подъезд:  "  +  tObject->FieldByName("A_Doorway")-
>AsString + ".";
        Alert->Memo1->Lines->Add(s);
        Alert->Memo1->Lines->Add("");
        s  =  "Тел.  дежурный:  "  +  tObject->FieldByName("Duty_Phone")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        Alert->Memo1->Lines-
>Add("_____________________________________________");
        s = "Собственник: " + tObject->FieldByName("Owner")->AsString;
        Alert->Memo1->Lines->Add(s);
        s   =   "Руководитель:   "   +   tObject->FieldByName("Chif_Name")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        s  =  "Тел.  рабочий:  "  +  tObject->FieldByName("Chif_PhoneW")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        s  =  "Тел.  домашний:  "  +  tObject->FieldByName("Chif_PhoneH")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        s  =  "Тел.  мобильный:  "  +  tObject->FieldByName("Chif_PhoneM")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        s   =   "Заместитель:   "   +   tObject->FieldByName("Depu_Name")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        s  =  "Тел.  рабочий:  "  +  tObject->FieldByName("Depu_PhoneW")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        s  =  "Тел.  домашний:  "  +  tObject->FieldByName("Depu_PhoneH")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        s  =  "Тел.  мобильный:  "  +  tObject->FieldByName("Depu_PhoneM")-
>AsString;
        Alert->Memo1->Lines->Add(s);
        Alert->Memo1->Lines-
>Add("_____________________________________________");
```

```
        s = "Арендодатель: " + tObject->FieldByName("Rentor")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Руководитель: " + tObject->FieldByName("RChif_Name")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Тел. рабочий: " + tObject->FieldByName("RChif_PhoneW")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Тел. домашний: " + tObject->FieldByName("RChif_PhoneH")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Тел. мобильный: " + tObject->FieldByName("RChif_PhoneM")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Заместитель: " + tObject->FieldByName("RDepu_Name")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Тел. рабочий: " + tObject->FieldByName("RDChif_PhoneW")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Тел. домашний: " + tObject->FieldByName("RDChif_PhoneH")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Тел. мобильный: " + tObject->FieldByName("RDChif_PhoneM")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Тел. дежурный: " + tObject->FieldByName("RDuty_Phone")->AsString;
        Alert->Memo1->Lines->Add(s);
        Alert->Memo1->Lines->Add("_____________________________________________");
        s = "Обслуживающая организация: " + tObject->FieldByName("ServicesLook")->AsString;
        Alert->Memo1->Lines->Add(s);
        s = "Оборудование: " + tObject->FieldByName("EquipmentLook")->AsString;
        Alert->Memo1->Lines->Add(s);
        Alert->Show();
        tObject->Close();
        tObject->Filter = "";
        tObject->Filtered = false;
        tObject->Open();
      }
      void __fastcall TForm1::N7Click(TObject *Sender)
      {
        SpeedButton6->Click();
        SpeedButton6->Down = true;
      }
      void __fastcall TForm1::N8Click(TObject *Sender)
      {
        SpeedButton7->Click();
        SpeedButton7->Down = true;
      }
      void __fastcall TForm1::tAccessAfterScroll(TDataSet *DataSet)
      {
        if (tAccess->FieldByName("Right1")->AsString == "write")
          ComboBox2->ItemIndex = 0;
```

```
        if (tAccess->FieldByName("Right1")->AsString == "read")
          ComboBox2->ItemIndex = 1;
        if (tAccess->FieldByName("Right1")->AsString == "")
          ComboBox2->ItemIndex = 2;
        if (tAccess->FieldByName("Right2")->AsString == "write")
          ComboBox3->ItemIndex = 0;
        if (tAccess->FieldByName("Right2")->AsString == "read")
          ComboBox3->ItemIndex = 1;
        if (tAccess->FieldByName("Right2")->AsString == "")
          ComboBox3->ItemIndex = 2;
        if (tAccess->FieldByName("Right3")->AsString == "yes")
          ComboBox4->ItemIndex = 0;
        if (tAccess->FieldByName("Right3")->AsString == "")
          ComboBox4->ItemIndex = 1;
        Edit1->Text = tAccess->FieldByName("Password")->AsString;
        if    (tAccess->FieldByName("Root")->AsBoolean)    Label7->Font-
>Color = clMaroon; else Label7->Font->Color = clGray;
      }
      void __fastcall TForm1::ComboBox2Change(TObject *Sender)
      {
        AnsiString f;
        tAccess->Edit();
        if (Sender == ComboBox2) f = "Right1"; else f = "Right2";
        if (((TComboBox*)(Sender))->ItemIndex == 0)
          tAccess->FieldByName(f)->AsString = "write";
        if (((TComboBox*)(Sender))->ItemIndex == 1)
          tAccess->FieldByName(f)->AsString = "read";
        if (((TComboBox*)(Sender))->ItemIndex == 2)
          tAccess->FieldByName(f)->AsString = "";
        tAccess->Post();
      }
      void __fastcall TForm1::BitBtn11Click(TObject *Sender)
      {
        if (tAccess->FieldByName("Root")->AsBoolean) return;
        tAccess->Delete();
        DBLookupListBox1->KeyValue  =  tAccess->FieldByName("UserName")-
>AsString;
      }
      void __fastcall TForm1::BitBtn10Click(TObject *Sender)
      {
        if (Edit3->Text == "") return;
        tAccess->SetKey();
        tAccess->FieldByName("UserName")->AsString = CurrentUser;
        tAccess->GotoKey();
        AnsiString s = tAccess->FieldByName("Setting")->AsString;
        tAccess->Insert();
        tAccess->FieldByName("UserName")->AsString = Edit3->Text;
        tAccess->FieldByName("Password")->AsString = "";
        tAccess->FieldByName("Root")->AsBoolean = false;
        tAccess->FieldByName("Setting")->AsString = s;
        tAccess->Post();
        DBLookupListBox1->KeyValue  =  tAccess->FieldByName("UserName")-
>AsString;
        Edit3->Text = "";
      }
```