МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"
Завідувач кафедри ІУСТ
_____
к.т.н, доц. Михелєв І.Л.
"____" _____ 2025 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "магістр"

на тему:

Дослідження та розробка інформаційної системи обліку паливних матеріалів на підприємстві

Виконав: студент групи 6141 м
_____ ***Дібрівний О.М.***
(підпис)
Керівник роботи:
_____ к.т.н.,професор
(посада, науковий ступінь вчене звання)
_____ ***Слободян С.О.***
(підпис)

м. Миколаїв – 2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова

Навчально-науковий інститут комп'ютерних наук та управління проєктами

Кафедра інформаційних управляючих систем та технологій

Спеціальність 122 "Комп'ютерні науки"

Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"
Гарант освітньої програми
_____
к.т.н, доц. Ажищев В.Ф.
"____" _____ 2025 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ
на здобуття ступеня вищої освіти "магістр"

Студенту Дібрівному Олегу Михайловичу
(прізвище, ім'я, по батькові)

1. Тема роботи: Дослідження та розробка інформаційної системи обліку паливних матеріалів на підприємстві

Керівник роботи к.т.н., професор Слободян С.О.

Затверджені наказом ректора № 1355-уч. від "14" листопада 2025 року.

2. Термін подання роботи: 15 грудня 2025 р.

3. Вихідні дані до проєкту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем, матеріали наукового стажування

4. Перелік питань, що належать до розробки (найменування розділів) Вступ. 1 Аналіз предметної області і постановка задачі. 1.1 Загальний опис проблеми. 1.2 Аналіз підприємства для якого виконується автоматизація. 1.3 Аналіз існуючих рішень. 2 Розробка проектних рішень. 2.1 Інформаційне забезпечення. 2.2 Програмне забезпечення. 3 Реалізація проекту автоматизованої системи. 4. Розрахунок економічного ефекту. 5 Охорона праці. 6. Охорона навколишнього середовища. Додатки

5. Перелік презентаційних матеріалів Структура підприємства, Схема комп'ютерної мережі, Функціональна модель інформаційної системи, Концептуальна модель даних, Фізична структура БД, Схема взаємодії модулів ПЗ, Знімки екранних форм.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	К.т.н., проф. Слободян С.О.	05.09.2024	22.12.2024
2	К.т.н., проф. Слободян С.О.	23.12.2024	15.06.2025
3	К.т.н., проф. Слободян С.О.	15.06.2025	25.09.2025
4		25.09.2025	14.11.2025
5		25.09.2025	14.11.2025
6		25.09.2025	14.11.2025

7. Дата видачі завдання 05 вересня 2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів роботи	Примітка
1	Дослідження та аналіз предметної галузі	22.12.2024	
2	Розробка концепції	15.03.2025	
3	Розділ з розробки проєктних рішень по системі (та її частинах)	15.06.2025	
4	Реалізація проєкту та розробка робочої документації	25.09.2025	
5	Обчислення економічної ефективності	14.11.2025	
6	Вирішення питань охорони праці та навколишнього середовища	14.11.2025	
7	Оформлення пояснювальної записки	14.11.2025	
8	Подання магістерської роботи на перевірку на плагіат	16.11.2025	
9	Подання магістерської роботи рецензенту	20.11.2025	
10	Подання магістерської роботи на державний захист	15.12.2025	

Студент _____ **Дібрівний О.М.**
(підпис) (ПІБ)

Керівник роботи _____ **Слободян С.О.**
(підпис) (ПІБ)

АНОТАЦІЯ

Дипломна робота присвячена розробці інформаційної системи обліку паливних матеріалів на підприємстві «Південь-Транс». У даній роботі проведений аналіз сучасних інформаційних технологій, які використовують для розробки інформаційних систем; сформовані вимоги до інформаційної системи; розроблена її концепція; проведене техніко-економічне обґрунтування розробки системи; розроблене технічне завдання; розроблені загальносистемні рішення, рішення з інформаційного, організаційного, технічного та програмного забезпечення; розроблена відповідна документація. Також у роботі виконані розділи з економіки, охорони праці, охорони навколишнього середовища та цивільної оборони.

Дипломна робота виконана на __133___ сторінках машинописного тексту, містить рисунків: _17____, таблиць: __7___, додатків __4___ і список використаних джерел з _11___ найменувань.

Ключові слова: База даних, Корпоративна система, паливні матеріали, інформаційна система

ABSTRACT

This diploma work is dedicated to developing information system of the fuel supplies in the company «Pivden-Trans» on driving. The work includes: analysis of modern information technologies, which are used for information system development; formulation of system requirements; system conception development; technical and economical proof of system development benefits; technical task formulation; development of system basic solutions, solutions in informational, organizational, technical and software aspects of system maintenance; development of appropriate documentation. Also, this work has economics, labor protection, environment protection, and civil defense sections.

The degree work is stated on _133___ pages of typescript, contains __17__ figures, __17__ tables, __4__ enclosures, and the list of __11___ original sources.

Keywords: Database, Enterprise System, fuel supplies, information system

Зміст

Вступ……....
1 Формування вимог до інформаційної системи обліку паливних матеріалів на підприємстві Південь-Транс, розробка її концепції і постановка задачі……
1.1 Опис інформаційної системи обліку паливних матеріалів на підприємстві..
1.2 Формування вимог до інформаційної системи обліку паливних матеріалів …….
1.2.1 Характеристика інформаційної системи обліку паливних матеріалів на підприємстві …….
1.2.2 Опис інформаційної системи підприємства ……...
1.3 Дослідження технологій розробки систем обліку паливних матеріалів на підприємстві . …….....
1.3.1 Опис результатів вивчення системи підприємства …….……
1.3.2 Аналіз вимог до інформаційної системи обліку паливних матеріалів на підприємстві ……..……
1.3.3 Вибір логічної архітектури програмного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві ……...……
1.3.4 Вибір фізичної архітектури програмного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві ……
1.3.5 Вибір технологій для розробки інформаційної системи на підприємстві ……...
1.3.6 Розробка варіантів концепції інформаційної підсистеми обліку паливних матеріалів на підприємстві …….
1.3.7 Вимоги, що впливають на якість інформаційної системи обліку паливних матеріалів на підприємстві …….
1.4 Постановка задачі…….
1.4.1 Загальні відомості……..
1.4.2 Призначення і мета створення інформаційної системи обліку паливних матеріалів на підприємстві ……..
1.4.3 Вимоги до інформаційної системи обліку паливних матеріалів на підприємстві...
1.4.4 Вимоги до інформаційного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві ……
1.4.5 Вимоги до лінгвістичного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві …….
1.4.6 Вимоги до програмного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві …….

1.4.7 Вимоги до технічного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві …….
1.4.8 Вимоги до документації інформаційної системи обліку паливних матеріалів на підприємстві ……..…….
1.4.9 Порядок контролю і приймання інформаційної системи обліку паливних матеріалів на підприємстві ……..…….
2 Розробка проектних рішень по системі та її частинах…….
2.1 Загальносистемні рішення…….
2.1.1 Схема організаційної структури…….
2.1.2 Схема функціональної структури……...
2.1.3 Загальний опис системи…….
2.1.4 Захист системи від несанкціонованого доступу……...
2.2 Рішення з організаційного забезпечення……..
2.3 Рішення з інформаційного забезпечення……..
2.3.1 Опис інформаційного забезпечення системи……
2.3.2 Концептуальна модель даних…….
2.3.3 Логічна структура бази даних……
2.3.4 Фізична структура бази даних……
2.3.5 Оцінка розміру бази даних……..
2.4 Рішення з технічного забезпечення……...
2.5 Рішення з програмного забезпечення……....
2.5.1 Структура програмного забезпечення……
2.5.2 Функції частин програмного забезпечення……...
2.5.4 Операційна система…….
3 Розробка робочої документації…….
4 Розрахунок економічного ефекту від впровадження системи обліку паливних матеріалів…….
5 Охорона праці при роботі з ПК…….
6 Охорона навколишнього середовища…….
Висновки .....
Список використаних джерел.....
Додаток А. Технічне завдання.....
Додаток Б. Опис застосування.....
Додаток В. Керівництво користувача.....
Додаток Г. Текст програмних модулів .....

Вступ

Попри значну кількість програмних продуктів, призначених для автоматизації бухгалтерського обліку, створення системи, яка б одночасно швидко окупалася та повністю відповідала вимогам користувача, залишається складним завданням. Потужні інтегровані програмні комплекси вирізняються високою вартістю та складністю впровадження. Крім того, специфічні особливості окремих підприємств часто є економічно недоцільними для врахування у масових універсальних системах.

На сучасному вітчизняному ринку представлено широкий спектр програмних рішень, орієнтованих на автоматизацію бухгалтерського обліку та управління підприємством. Проте прості програмні продукти, як правило, реалізують лише загальні положення обліку і не завжди здатні врахувати специфічні умови діяльності конкретного суб'єкта господарювання. Водночас підприємства прагнуть зберегти інвестиції, вкладені в автоматизацію, і тому потребують сумісності нового програмного забезпечення з уже наявними продуктами інших виробників, що нерідко є технічно неможливим. Додатковою проблемою більшості простих систем є недостатній рівень фізичного та логічного захисту даних, а також відсутність механізмів авторизації доступу.

Слід зазначити, що багато з перелічених недоліків відсутні у потужних інтегрованих комплексах, однак їх висока вартість значною мірою нівелює переваги. Навіть середні підприємства не завжди мають можливість придбати подібні системи.

У зв'язку з цим під час автоматизації обліку на середніх підприємствах із розвиненою індивідуальною інфраструктурою часто виникає потреба у створенні програмного забезпечення «на замовлення» — відносно недорогих інтегрованих рішень, адаптованих до специфічних вимог конкретного замовника.

Метою даної дипломної роботи є розроблення системи, що забезпечує автоматизацію бухгалтерського обліку паливних матеріалів на автотранспортному підприємстві «Південь-Транс».

Як серверну платформу для роботи з базою даних обрано Microsoft SQL Server 2022, що зумовлено його порівняно невисокою вартістю та високою продуктивністю.

1 Формування вимог до інформаційної системи обліку паливних матеріалів на підприємстві Південь-транс, розробка її концепції і постановка задачі

1.1 Опис інформаційної системи обліку паливних матеріалів на підприємстві

Транспортно-експедиційне підприємство «Південь-Транс» надає послуги власникам вантажів з перевозки та супроводженню вантажу. Підприємство володіє парком автомобілів різного призначення (тентовані машини, рефрижератори), які мають різні вантажопідйомність та витрату палива. Підприємство має філії у м. Одеса, м. Миколаєві та у м. Краматорськ (Донецька філія). Головний офіс знаходиться у м. Миколаєві. Підприємство має наступну структуру:

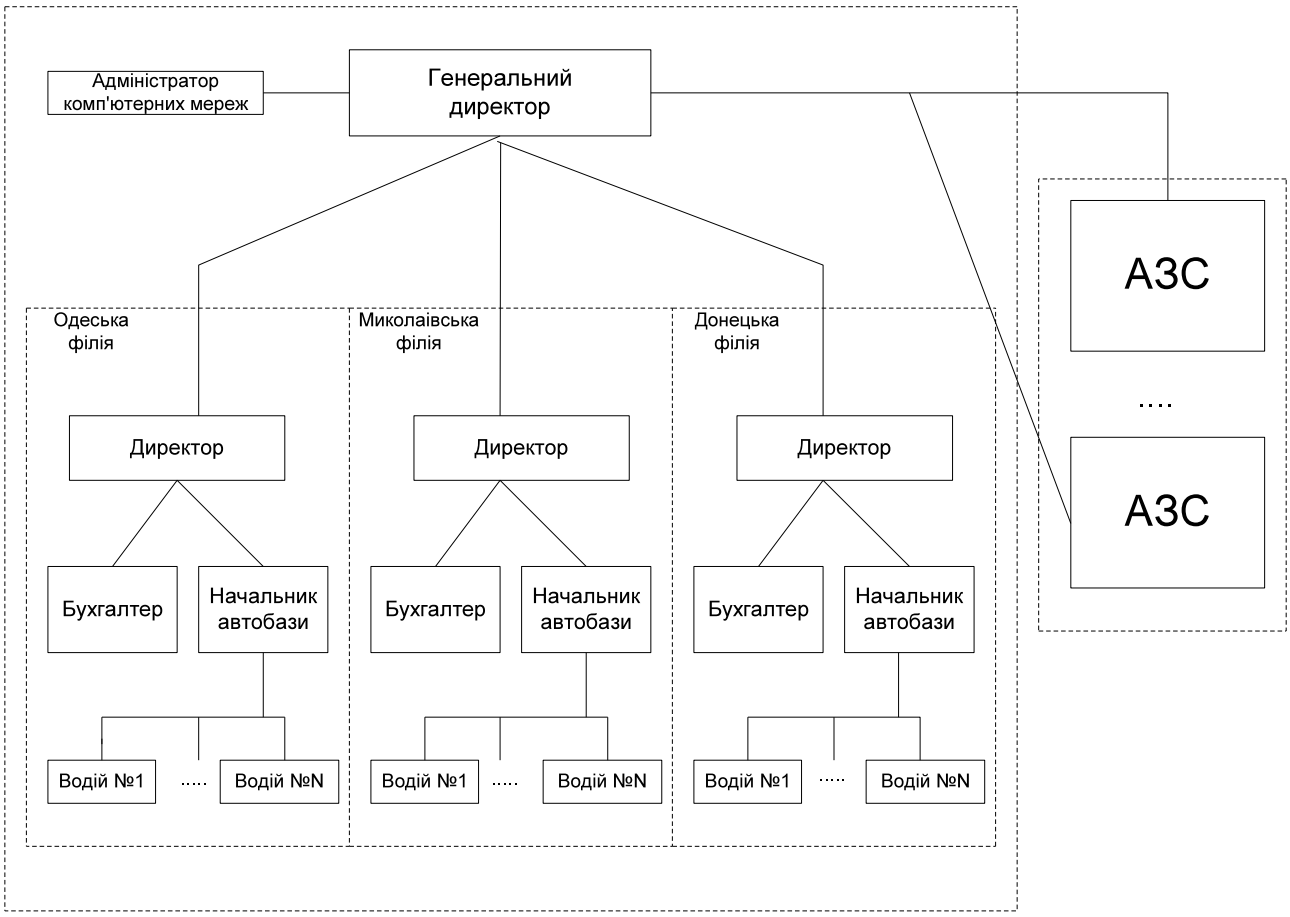


Рисунок 1 – Структура підприємства Південь-Транс

Автомобіль у поєднанні з причепом утворює єдиний транспортний засіб. Кожний автомобіль має індивідуальний номер шасі, а причіп — номер рами. Транспортний засіб характеризується власним серійним номером, об’ємом паливного бака та набором запчастин, які підбираються відповідно до номера шасі або рами згідно з каталогом виробника. Крім того, кожний автомобіль має

індивідуальну норму витрати палива на один кілометр пробігу та оснащений спідометром, що фіксує загальний пробіг.

Під час реєстрації транспортного засобу в органах Патрульної поліції України та сервісних центрах йому присвоюється державний реєстраційний номер та видається технічний талон. У технічному талоні зазначаються державний номерний знак, дані про власника, а також технічні характеристики транспортного засобу, зокрема номер шасі або рами, об'єм двигуна та повна маса з вантажем.

Перед виїздом транспортного засобу з гаража водієві видається шляховий лист. Шляховий лист підприємства є офіційним документом, що підтверджує виконання перевезення вантажу та може використовуватися для моніторингу транспортних послуг, які надає підприємство.

У шляховому листі фіксуються такі обов'язкові реквізити:

- номер шляхового листа;
- марка транспортного засобу;
- державний реєстраційний номер транспортного засобу;
- державний реєстраційний номер причепа;
- прізвище, ім'я та по батькові водія;
- дата виїзду транспортного засобу з гаража;
- дата повернення транспортного засобу до гаража;
- залишок пального на момент виїзду;
- залишок пального на момент повернення;
- кількість палива, заправленого зі складу;
- кількість палива, заправленого за готівковий розрахунок;
- кількість здійснених поїздок;
- пробіг транспортного засобу з вантажем;
- обсяг перевезеного вантажу;
- балансовий рахунок поїздки.

Після повернення транспортного засобу до гаража шляхові листи передаються до бухгалтерії. Бухгалтер, відповідальний за їх облік, на підставі даних шляхових листів формує до завершення звітного періоду (місяця або року) такі документи:

- Відомість по транспортному засобу, що містить узагальнені дані всіх шляхових листів, які стосуються конкретного транспортного засобу. У відомості враховуються відпрацьовані години, обсяг перевезеного вантажу, загальний пробіг, пробіг з вантажем, а також кількість заправленого палива. Документ формується у

розрізі окремих шляхових листів, після чого підсумкові дані передаються до зведення по транспортних засобах.

- Зведення по транспортних засобах, яке містить підсумкову інформацію всіх відомостей по транспортних засобах підприємства.
- Відомість заправки транспортного засобу зі складу, що відображає кількість палива, виданого зі складу за звітний період відповідно до марок пального.
- Відомість списання палива, яка містить фактичні місячні витрати палива за звітний період.
- Відомість пробігу за балансовими рахунками, що включає дані про пробіг та обсяг перевезеного вантажу у розрізі балансових рахунків поїздок.
- Відомість по причепах, яка містить інформацію про пробіг та кількість перевезеного вантажу причепами.
- Лицевий рахунок транспортного засобу, що включає дані, пов’язані з конкретним транспортним засобом, зокрема дату виїзду з гаража та номер шляхового листа.

Значний обсяг шляхових листів зумовлює високу трудомісткість обробки первинної документації. Це спричиняє збільшення кількості помилок і необхідність частих коригувань під час підрахунку даних, що, у свою чергу, може призводити до порушення строків подання звітності та накладення штрафних санкцій.

Після завершення рейсу водій подає до бухгалтерії звіт, у якому зазначає маршрут руху, кількість заправленого пального на маршруті та надає підтвердні документи (чеки із зазначенням дати, найменування АЗС та обсягу заправленого пального). Бухгалтер у шляховому листі фіксує показання спідометра, залишок пального та дату повернення.

Інформаційна система забезпечуватиме контроль відповідності фактичних витрат палива розрахунковим нормам і даним, поданим водієм. Це дозволить запобігти перевитраті пального, прискорити обробку даних, зменшити навантаження на бухгалтерію та мінімізувати кількість помилок.

1.2 Формування вимог до інформаційної системи обліку паливних матеріалів на підприємстві Південь-Транс

1.2.1 Характеристика інформаційної системи обліку паливних матеріалів на підприємстві

Об'єктом автоматизації є система обліку паливних матеріалів, пов'язана з роботою розрахункової групи бухгалтерського відділу. Існуюча інформаційна

система обліку паливних матеріалів (її комп'ютеризована версія перебуває в стадії розробки) має потребу в створення єдиної бази даних для централізованого збереження інформації. Об'єкт автоматизації сам по собі є інформаційною системою, для автоматизації роботи якої потрібно створення власної бази даних і програмного забезпечення для бухгалтера. Об'єкт автоматизації вирішує задачі пов'язані з контролем за витратами кількості розрахункового палива та даними, які надані водієм, дозволяє автоматично розраховувати витрату палива за шляховим листом, проводить звірку з АЗС та аналізує, чи не є використання палива поза нормою.

1.2.2 Опис інформаційної системи підприємства

Придбання палива може здійснюватися шляхом:

1) безготівкового розрахунку:

а) перерахування суми передоплати з рахунку автотранспортного підприємства АТП на рахунок роздрібного продавця - визначеної автозаправочної станції (АЗС):

без придбання талонів;

з придбанням талонів;

б) використання електронних карток;

в) купівлі палива в оптових постачальників при наявності власних чи орендованих ємкостей і устаткування для його збереження і заправлення автомобілів.

2) розрахунку готівкою:

а) купівлі палива водієм АТП у дорозі на різних АЗС за рахунок виданих йому під звіт коштів;

б) купівлі палива (як правило, оптової) підзвітною особою підприємства за наявності власних чи орендованих ємкостей і устаткування для його збереження і заправлення автомобілів.

Первинний облік

1) Безготівковий розрахунок. Підставою для придбання палива в постачальника таким шляхом є договір, укладений між АТП і АЗС чи оптовим постачальником, винятком є використання електронних карток при розрахунках. На сьогодні передбачений типовий договір за формою № 23-нп (додаток 23 до Інструкції № 81).

У цьому випадку можливі два варіанти купівлі нафтопродуктів на АЗС: без придбання талонів і з придбанням талонів.

Без використання талонів. Якщо автопідприємство отримує паливо по безготівковому розрахунку і не використовує при цьому талони, то воно зобов'язане використовувати відомість на одиничний відпуск нафтопродуктів за формою № 24-нп . Дана відомість призначена для обліку на одноразовий відпуск однорідних палив. Заповнення відомості за ф. № 24-нп здійснюється в декілька етапів.

Так, першим етапом є первинне заповнення відомості на АТП на момент укладення договору (чи протягом декількох днів після його укладення) - а саме проставляння штампа на даних відомостей, після чого вони передаються на АЗС. Кількість даних відомостей не обмежена, виходячи з цього, вона може бути визначена сторонами, які домовляються на їхній розсуд. У більшості випадків, при визначенні кількості відомостей, використовуються дані про середню кількість заправлень автомобіля за місяць і кількість самих автомобілів.

Другим етапом є вторинне заповнення (але вже на АЗС) усіх граф відомості, які залишилися. При цьому відомість заповнюється в двох екземплярах. Оператор АЗС, при відпуску палива по відомості, зобов'язаний поставити штамп про кількість відпущеного палива також і в шляховий лист водія. При цьому необхідно пам'ятати, що відомість може заповнюватись на кілька автомобілів, але тільки за умови, що вони прибули на АЗС одночасно. Наприкінці місяця АЗС зобов'язана передати другий екземпляр відомостей на АТП. Крім того, наприкінці звітного періоду бухгалтерією АЗС виписується видаткова накладна на загальну вартість виданого палива за місяць із вказанням найменувань, кількості і вартості по марках палива.

При заправленні по талонах, підприємство-покупець закуповує визначену кількість талонів в АЗС. Потім ці талони видають водіям. Згідно п. 4.1.7.27 Інструкції № 81 використовувати талони як засіб платежу на АЗС за паливо

заборонено. Але, якщо аналізувати дану ситуацію, то талони і не є засобом платежу, вони надають тільки право на одержання палива на визначеній АЗС, у визначеному обсязі, оскільки оплата за дані палива уже була здійснена (передоплата).

Відомість по обліку талонів також повинна вестися на АЗС у двох екземплярах у (розрізі АТП), один із яких повинен бути переданий у бухгалтерію АТП для звірки даних про видачу талонів.

Дані відомості менеджери АТП використовують для зустрічного контролю за кількістю палива, отриманого від постачальника за місяць. При цьому порівнюють окремо в акті підсумки про видачу паливо у літрах з підсумковою товарною накладною за місяць, отриманою від постачальника. Причинами розбіжностей можуть бути: невірно виведений залишок палива в баках транспортних засобів на кінець місяця в обліку АТП (погрішності при вимірі залишків), втрата талонів водіями чи використання їхній не за призначенням (присвоєння), загальна занедбаність в обліку палива.

Використання електронних карток. АТП може при покупці палива використовувати електронні картки (дебетові та/або кредитні). У даному випадку для:

1. АЗС – необхідно вести спеціальну відомість. Оскільки дана відомість ведеться тільки АЗС і не має ніякого відношення до АТП.

2. АТП – необхідно вести спеціальну відомість по обліку виданих електронних карток.

Відомості щодо обліку електронних паливних карток також використовуються для звірки витрачених коштів із даними, зазначеними у банківських виписках за картковими рахунками. На підставі цих відомостей бухгалтерія або матеріально відповідальна особа здійснює вибірку використаних коштів у розрізі кожного водія та кожної картки. Отримані дані застосовуються для відображення надходження палива до паливних баків транспортних засобів. Залежно від потреб автотранспортного підприємства (АТП), відомість може вестися за видами робіт, гаражами, водіями тощо.

Облік придбання палива в оптових постачальників

Особливістю обліку придбання палива в оптових постачальників є те, що у товарних накладних постачальника обсяг палива зазначається у вагових одиницях, зазвичай у тоннах. Оскільки облік відпуску палива в АТП ведеться в літрах,

виникає необхідність перерахунку вагових показників в об’ємні. Для цього застосовуються спеціальні коефіцієнти перерахунку.

Облік заправки на власних АЗС

У разі заправки на власних автозаправних станціях або роздавальних колонках АТП заправник чи інша матеріально відповідальна особа веде або аналогічну відомість, або лімітно-забірні картки. У будь-якому випадку використовується накопичувальний первинний документ, у якому фіксується кількість палива, виданого під час кожної заправки транспортного засобу. Ліміт відпуску бензину чи дизельного палива на кожен автомобіль на день або місяць затверджується керівником або головним інженером АТП.

У цьому випадку одиницею обліку є транспортний засіб, а не водій. Якщо водій закріплений за конкретним автомобілем, обсяг відпущеного палива автоматично пов’язується з його прізвищем. У разі змінності водіїв або їх заміни виникає потреба у розподілі отриманого палива між кожним водієм окремо. Це необхідно для коректного визначення перевитрати або економії пального, оскільки преміювання чи утримання із заробітної плати здійснюється індивідуально. Вибір конкретної системи обліку залежить від організації контролю за витратами палива та наявності затвердженої системи стимулювання або санкцій.

Податкові аспекти

Після завершення місяця або після кожного надходження товарної накладної бухгалтерія повинна отримати від постачальника податкову накладну, яка є підставою для відображення податкового кредиту за отриманим паливом.

Придбання палива за готівку

У разі придбання палива водіями за готівку первинними документами є фіскальні чеки РРО автозаправної станції та авансовий звіт водія.

Водій зобов’язаний протягом трьох робочих днів після повернення з відрядження або наприкінці робочого дня (чи вранці наступного дня), якщо кошти були видані для придбання палива протягом робочого дня, подати авансовий звіт і повернути невикористані кошти. У великих АТП кількість водіїв, які отримують кошти під звіт, може становити кілька десятків осіб, що призводить до значної кількості авансових звітів — інколи кількох сотень на місяць.

Перевірка авансових звітів

Під час перевірки авансових звітів необхідно встановити наявність і коректність документів, що підтверджують придбання палива та його зв’язок із виконанням виробничого завдання, а саме:

- фіскальні чеки РРО;

- податкові накладні (у разі, якщо сума в чеку перевищує 20 грн відповідно до пп. 7.2.6 Закону про ПДВ);
- товарні накладні на перевезений вантаж або шляховий лист.

Також перевіряється відповідність маршруту руху, зазначеного у шляховому листі, адресам АЗС, вказаним у чеках. Це особливо важливо при міжміських та міжнародних перевезеннях.

За наявності належним чином оформлених документів бухгалтер має право віднести витрати на придбання палива до валових витрат (податковий облік) та оприбуткувати паливо на рахунок 20 «Виробничі запаси» (бухгалтерський облік). Коректно оформлена податкова накладна, підтверджена фіскальними чеками, надає право на відображення операції у книзі обліку придбань та формування податкового кредиту з ПДВ.

Покупка підзвітною особою. При використанні даного варіанта придбання ПММ, бухгалтеру необхідно пам’ятати, що він здійснюється у такому ж порядку, що і придбання палива водіями. Тому:

1. Вимоги до касових і авансових документів ті ж самі, що і при придбанні водіями за готівку.

2. Паливо від підзвітної особи оприбутковується комірником і надалі видається водіям. Документальне оформлення видачі палива аналогічне видачі палива, куплених в оптових постачальників (дивися вище).

Бухгалтерський і податковий облік придбання палива.

Наявність та рух палива в бухгалтерському обліку відображаються на субрахунку 203 «Паливо». Оскільки автотранспортні підприємства (далі — АТП) у своїй діяльності використовують різні види палива та, як правило, мають спеціально обладнані місця для його зберігання, виникає потреба у відкритті за субрахунком 203 субрахунків другого, а за необхідності — і третього порядку. Наприклад, субрахунок 2031 «Бензин АІ-95», 2032 «Дизельне паливо» тощо.

Для обліку палива за місцями його зберігання відкриваються субрахунки третього порядку, зокрема: 20311 «Бензин АІ-95 на складі», 20312 «Бензин АІ-95 в автозаправниках», 20313 «Бензин АІ-95 у баках автомобілів» та інші, відповідно до наявних місць зберігання. Аналогічні субрахунки можуть бути відкриті для всіх видів палива, якщо це необхідно для забезпечення належного рівня аналітичного обліку та відповідає обліковій політиці підприємства.

Надмірна деталізація, наприклад відкриття за субрахунком 20313 субрахунків четвертого порядку (за кожним автомобілем окремо), не завжди є доцільною та

ефективною. Рішення щодо застосування такої деталізації приймається керівником та головним бухгалтером АТП (за наявності відповідних повноважень) з урахуванням специфіки діяльності підприємства, кількості транспортних засобів, обсягів наданих послуг та інших організаційних чинників.

Особливості оподаткування палива, придбаного водіями у нерезидентів

Під час здійснення міжнародних перевезень водії нерідко змушені придбавати паливо не лише на території України, а й за кордоном. Вартість такого палива включається до складу валових витрат на підставі первинних документів, доданих до авансового звіту водія. Проте постає питання щодо оподаткування таких операцій податком на додану вартість (ПДВ).

Відповідно до п.п. 3.1.2 Закону України про ПДВ, операції з ввезення товарів на митну територію України, а також отримання робіт (послуг) від нерезидентів для їх використання на митній території України є об'єктом оподаткування ПДВ. Звідси випливає, що на вартість палива, придбаного за межами України та не використаного до моменту перетину кордону, підприємство повинно нарахувати та сплатити податкове зобов'язання з ПДВ.

Податковий кредит за такими операціями не формується, оскільки залишок палива, як правило, не декларується під час перетину кордону. Водночас обов'язок нарахування податкових зобов'язань виникає лише за умови використання ввезеного палива на митній території України. Якщо в баку транспортного засобу міститься паливо, придбане як в Україні, так і за кордоном, фактично неможливо визначити, яка його частина була використана за межами України, а яка — на її території.

З огляду на це доцільно, у разі якщо залишок палива при в'їзді на митну територію України перевищує залишок при виїзді, суму такого перевищення оподатковувати ПДВ за ставкою 20 %, не очікуючи результатів податкової перевірки (п.п. 3.1.2 Закону про ПДВ).

Визначення різниці залишків палива

Різниця між залишками паливно-мастильних матеріалів у баках автомобіля на момент виїзду з України та на момент повернення визначається розрахунковим шляхом. Для цього використовуються такі дані:

- залишок палива при виїзді з АТП;
- залишок палива при поверненні;
- загальний пробіг автомобіля за рейс;
- пробіг по території іноземної держави;

- норми витрати палива.

Отримані розрахунки дозволяють визначити обсяг палива, що підлягає оподаткуванню при ввезенні на митну територію України.

Списання ПММ

Облік витрати ПММ здійснюється на підставі даних шляхових листів. Списання палива проводиться відповідно до Норм № 43.

Види норм витрати ПММ і їх застосування

Нормування витрат палива - це встановлення припустимої міри його споживання у визначених умовах експлуатації автомобіля, для чого застосовуються базові лінійні норми, установлені по моделях (модифікаціям) автомобілів, і система нормативів і коректувальних коефіцієнтів, що дозволяють враховувати виконувану транспортну роботу, кліматичні, дорожні й інші умови експлуатації.

Облік дорожніх, кліматичних і інших експлуатаційних факторів проводиться за допомогою ряду поправочних коефіцієнтів, приведених у формі відсотків чи підвищення зниження базового значення норми. Право встановлення конкретних величин коефіцієнтів у регламентованих межах і термін їх дії надається керівникам підприємств і затверджується наказом по підприємству.

У випадку застосування одночасно декількох коректувальних коефіцієнтів розраховується сумарний коефіцієнт коректування, що дорівнює сумі цих надбавок (зменшувальні коректувальні коефіцієнти віднімаються): Кс = К1 5 + К2 + К3 +...+ Кn.

Крім нормативних витрат палива, дозволяється додаткове його споживання:

1. На міжгаражні роз'їзди і технічні потреби (технічні огляди, регулювальні роботи, приробляння деталей двигунів і автомобіля після ремонту і т.п.) - не більш 1 % від загальної кількості палива, спожитого підприємством;

2. Не більш 10 л палива на місяць на один газобалонний (газодизельний) автомобіль для його запуску в холодний період року (температура навколишнього середовища нижче 0 °С). Зазначена надбавка не застосовується при експлуатації газобалонних (газодизельних) автомобілів на рідкому паливі;

3. При тривалих простоях автомобілів під навантаженнямрозвантаженням у пунктах, де за умовами пожежної безпеки заборонено виключати двигун (нафтобази, спеціальні склади і т.п.), а також при тривалих простоях автомобілів у

зимовий і холодний час року з працюючим двигуном (чекання хворих, інвалідів, клієнтів, нічліг у дорожніх умовах і т.п.) чи під час перевезення спеціальних вантажів, що мають потребу в постійному обігріві салону (кузова) автомобіля, дозволяється додаткова витрата палива з розрахунку: одна година простою відповідає витраті палива по лінійній нормі на 5 км пробігу автомобіля.

Тон - тривалість роботи обігрівача, година (розраховується в залежності від тривалості роботи автомобіля на лінії і температури навколишнього середовища).

Списання ПММ і відображення його в бухгалтерському та податковому обліку

Бухгалтерський облік. Як і будь-які інші матеріальні цінності, ПММ можуть мати нормативні та фактичні витрати, що, з цілком зрозумілих причин, не завжди збігаються.

Визначення нормативної кількості використаного ПММ, зокрема пального, здійснюється на підставі даних шляхових аркушів, установлених норм (Норми № 43) і по визначеній формулі. При цьому використовується шляховий лист як джерело наступної інформації про:

1. наявність пального в баці автомобіля на початок зміни (рейсу) за даними вимірювальних приладів автомобіля ;
2. кількість залитого пального (на автозаправній станції (АЗС) чи з автозаправника автотранспортного підприємства, по безготівковому розрахунку, у тому числі по картках, чи за готівку на АЗС сторонніх підприємств та інше) під час зміни (виконання рейсу);
3. пробіг автомобіля за час зміни (рейсу).

Як визначається витрата пального по нормі ми докладно розповіли на минулому уроці (дивися "ШБ" №23/2003 р).

Маючи дані про залишок пального в баці автомобіля на початок зміни (рейсу), кількість залитого пального в бак протягом зміни (рейсу) і витрата пального по нормі можна визначити залишок пального на момент закінчення зміни (рейсу), виконавши нескладні розрахунки. Такий залишок можна розрахувати по наступній формулі

$$Zkr = Zn + R - Qн$$

Де: Zn - залишок пального в баці автомобіля на момент початку зміни (рейсу);

R - залито пального в бак протягом зміни (рейсу);

Qн - витрата пального протягом зміни (рейсу) по нормі.

Zkr - залишок пального в баці автомобіля на момент закінчення зміни (рейсу) визначений розрахунковим шляхом (по нормі).

Для визначення фактичної витрати пального необхідно по закінченню зміни (рейсу) визначити залишок пального в баці автомобіля. Його можна визначити двома способами:

1. за даними вимірювальних приладів автомобіля;
2. по фактичній наявності пального в баці 2 .

Слід зазначити, що другий спосіб обліку застосовується лише у виняткових випадках, зокрема під час проведення інвентаризації.

Податковий облік паливно-мастильних матеріалів

Паливно-мастильні матеріали (ПММ), придбані у постачальників за безготівковим розрахунком і такі, що надходять безпосередньо на склад підприємства (у цистерни, автоцистерни тощо), включаються до складу валових витрат на підставі платіжних документів або прибуткових накладних — залежно від того, яка подія відбулася раніше: оплата чи надходження. ПММ, придбані за безготівковим розрахунком і отримані безпосередньо у баки або заправні ємності транспортних засобів, відносяться до валових витрат, як правило, на підставі платіжних документів, оскільки заправлення за талонами зазвичай здійснюється після їх оплати. Якщо оплата талонів відбувається після фактичного отримання ПММ на АЗС сторонніх організацій, такі витрати включаються до валових на підставі звітів водіїв про використані талони.

У разі придбання палива за електронними картками валові витрати автотранспортного підприємства збільшуються на дату перерахування коштів на рахунок сторонньої організації (АЗС) як оплату електронних карток. Якщо ПММ придбані підзвітною особою та передані на склад підприємства або придбані водієм і оприбутковані безпосередньо в бак транспортного засобу, датою збільшення валових витрат є дата затвердження авансового звіту відповідної особи. Незалежно від способу придбання ПММ та місця їх зберігання (склад підприємства

чи баки транспортних засобів), їх залишок на кінець звітного кварталу підлягає включенню до розрахунку приросту (убутку) запасів відповідно до п. 5.9 ст. 5 Закону України «Про оподаткування прибутку підприємств» з метою коригування валових витрат.

Питання віднесення ПММ до валових витрат

Важливо враховувати, що фактичні витрати ПММ не завжди збігаються з нормативними. Якщо економія не впливає на валові витрати та податковий кредит, то питання впливу перевитрати на податковий облік є предметом фахових дискусій і нерідко призводить до помилкових рішень бухгалтерів автотранспортних підприємств.

На думку окремих фахівців, зокрема представників податкових органів, наднормативну перевитрату ПММ слід:

1. виключати зі складу валових витрат;
2. зменшувати відповідний податковий кредит з ПДВ;
3. включати до сукупного оподатковуваного доходу працівника-водія з подальшим утриманням прибуткового податку та внесків до соціальних фондів.

Такі підходи ґрунтуються на припущенні, що недотримання норм витрати не пов’язане з господарською діяльністю підприємства, а працівник, який допустив перевитрату, отримав особистий дохід. Проте сам факт перевитрати не може бути підставою для таких висновків. Перевитрата або економія ПММ не є аналогами нестач чи лишків; визначальним є встановлення причин їх виникнення. Лише після з’ясування причин можна робити висновок щодо зв’язку витрат із виробничою діяльністю.

Встановлення причин перевитрати

Для визначення причин перевитрати ПММ на автотранспортних підприємствах створюються спеціальні комісії. До можливих причин належать: технічні несправності транспортного засобу, несприятливі погодні умови, недбалість водія, привласнення пального тощо.

Якщо комісія встановлює, що перевитрата не пов’язана з господарською діяльністю підприємства, вартість наднормативної витрати підлягає виключенню зі складу запасів на початок періоду у податковому обліку. Крім того, необхідно сторнувати податковий кредит з ПДВ у частині пального, використаного не для господарської діяльності. Якщо перевитрата виникла з вини працівника і підлягає відшкодуванню, відшкодування здійснюється відповідно до Постанови Кабінету Міністрів України № 116.

Якщо ж комісія встановлює, що перевитрата ПММ була спричинена непередбаченими обставинами та пов’язана з виробничою діяльністю підприємства, податкові наслідки у вигляді коригувань не виникають.

1.3 Дослідження технологій розробки системи обліку паливних матеріалів на підприємстві Південь-транс

1.3.1 Опис результатів вивчення системи підприємства

Інформаційна система обліку паливно-мастильних матеріалів призначена для автоматизації процесів розрахунку та контролю витрат палива. Створення такої системи передбачає розроблення бази даних, програмного забезпечення для бухгалтерського обліку, а також визначення вимог до інших компонентів інформаційного забезпечення.

Основні функції інформаційної системи полягають у забезпеченні контролю за витратами розрахункової кількості палива, обробленні даних, наданих водіями, автоматизованому розрахунку фактичної витрати палива за даними шляхових листів, здійсненні звірки з даними автозаправних станцій та аналізі відповідності фактичного використання палива встановленим нормам.

Запровадження автоматизованої інформаційної системи дає змогу усунути випадки перевитрати палива, підвищити оперативність розрахунків, скоротити трудомісткість роботи бухгалтера та мінімізувати ймовірність помилок у процесі обліку.

1.3.2 Аналіз вимог до інформаційної системи обліку паливних матеріалів на підприємстві

необхідно врахувати вимоги до системи в цілому:

- Простий і зручний інтерфейс користувача.
- Доступ до системи через Internet з будь-якого комп'ютера глобальної мережі.
- Багатокористувачева робота.
- Висока масштабуємість системи.
- Висока продуктивність.
- Надійність.
- Конфіденційність інформації.

Аналізуючи зазначені вище вимоги, можна зробити висновок, що інформаційна система повинна мати/підтримувати:

- Web-інтерфейс.
- Багаторівневу архітектуру.
- Сучасні стандарти в області інформаційних технологій (протоколи передачі даних, стандарти збереження даних і т.д.).
- Сучасні серверні платформи і програмне забезпечення.
- Сучасні рішення в області інформаційних технологій по розробці розподілених систем (CGI, ISAPI, DCOM/COM+, CORBA, RMI і інші).
- Ефективні методи доступу до даних.
- Забезпечити простоту модернізації, можливість повторного використання програмного забезпечення (коду).

1.3.3 Вибір логічної архітектури програмного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві

Логічна архітектура програмного забезпечення інформаційної системи являє собою концептуальний опис структури програмного забезпечення системи. Кожна система має три рівні: рівень користувача, прикладний і рівень даних. Крім того, в усіх системах мається презентаційне програмне забезпечення (ПЗ), ПЗ обробки бізнес-правил і даних, а також ПЗ, що відповідає за збереження інформації. Відрізняються ж архітектури тільки організацією програмного забезпечення (програмного коду), індивідуальної для конкретної системи.

Багато сучасних інформаційних систем побудовані на базі дворівневої архітектури «клієнт/сервер» [11]. При цьому клієнтське програмне забезпечення відповідає за обробку і відображення даних. Самі ж дані централізовано зберігаються на серверах, до яких підключаються клієнти. Часто час використання системи обмежується тривалістю такого з'єднання.

Клієнт-серверні системи добре працюють тільки в контрольованих середовищах, коли число користувачів передбачуване (це необхідно для виділення ресурсів). Однак ця архітектура стає неефективною, при великому числі користувачів або якщо їхнє число невідомо. Крім цього, масштабованість таких систем невисока, тому що кількість безпосередніх з'єднань із сервером обмежена. Невеликі і можливості повторного застосування, оскільки користувачі прив'язані до визначеного формату бази даних. А тому що клієнтське програмне забезпечення містить блок обробки інформації, загальний обсяг програмного забезпечення досить великий (Такий тип клієнтських програм іноді називають «товстим»

клієнтом). У цьому випадку при зміні алгоритмів обробки даних приходиться встановлювати нове програмне забезпечення на кожен клієнтський комп'ютер [11].

Можна отримати невелике поліпшення шляхом переміщення прикладних алгоритмів і блоків обробки даних на сервери даних (наприклад, за допомогою збережених процедур сервера баз даних). Таку архітектуру іноді називають 2,5-рівневою. Масштабованість подібних систем трохи краща, але все рівно мала для виконання вимог інформаційної системи. Крім того, можливість повторного використання програмного забезпечення (коду) залишається на невисокому рівні [8].

Масштабованість і ступінь повторного використання можна помітно поліпшити, додавши в архітектуру системи третій рівень. У такій багаторівневій архітектурі (її також називають N-рівневою) усі рівні – рівень користувача, прикладний і рівень даних – логічно розділені (рисунок 1.3). Багаторівнева архітектура задовольняє вимогам до інформаційної системи, що розробляється.

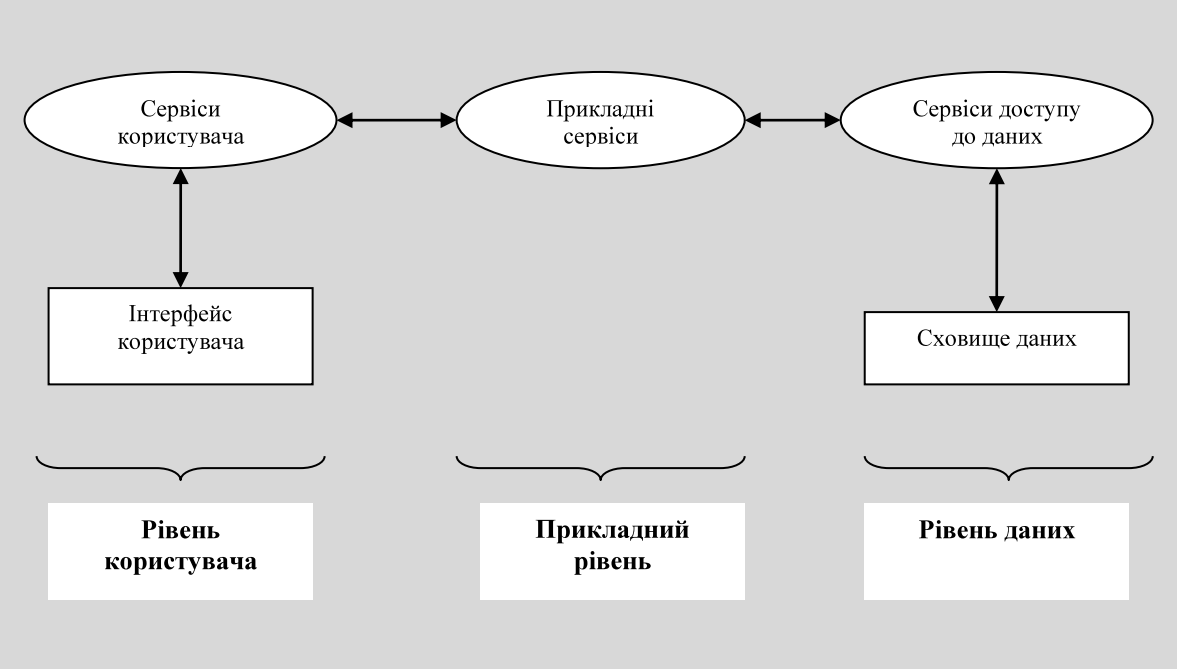


Рисунок 1.3 - Логічна архітектура
програмного забезпечення інформаційної системи

Функціональні можливості кожного рівня багаторівневої архітектури можна охарактеризувати таким чином:

- Рівень користувача. На цьому рівні здійснюється відображення даних та забезпечується можливість їх редагування користувачем. Існують два основних типи інтерфейсів: віконні, що реалізуються засобами підсистеми графічного інтерфейсу операційної системи, та веб-інтерфейси. Останні базуються на

технологіях HTML, DHTML, XHTML або XML, що забезпечує їх коректне відображення у будь-якому веб-браузері незалежно від платформи.

- Прикладний рівень. На цьому рівні реалізуються бізнес-правила та обмеження щодо даних, зокрема механізми перевірки їх коректності. Хоча сервіси прикладного рівня використовуються презентаційним рівнем, вони не прив'язані до конкретного клієнта та можуть бути доступні будь-якому клієнтському застосунку. Бізнес-правила формалізуються у вигляді прикладних алгоритмів, корпоративних регламентів та інших логічних конструкцій. Обмеження на дані забезпечують точність і цілісність інформації. Зазвичай бізнес-логіка реалізується у вигляді окремого модуля на централізованому сервері, що дозволяє одночасний доступ до нього кільком клієнтам.
- Рівень даних. Прикладний рівень не має інформації про фізичне розташування чи спосіб зберігання даних, які він обробляє. Усі операції з отримання, збереження та передавання даних виконуються сервісами доступу до даних, що реалізуються у вигляді ізольованих модулів, які містять знання про структуру та місце розташування інформації. Кожен модуль доступу до даних, як правило, відповідає також за підтримання цілісності сховища, наприклад реляційної бази даних. У багаторівневих програмних системах як сховище інформації використовуються сервери баз даних (MS SQL Server, Oracle, Sybase, MySQL, PostgreSQL тощо), які забезпечують зберігання даних у таблицях та високопродуктивне виконання запитів.

1.3.4 Вибір фізичної архітектури програмного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві

Виходячи з логічної архітектури програмного забезпечення, можливо сформувати його фізичну архітектуру. Фізична архітектура являє собою опис структури програмного забезпечення з позиції наявних у системі фізичних компонентів, зокрема серверів та клієнтських робочих станцій.

Відповідно до визначеної логічної архітектури можуть бути побудовані дві варіації фізичної архітектури, кожна з яких у певній мірі відповідає вимогам до системи. Перша з них має трирівневу структуру (рисунок 1.4), тоді як друга реалізує чотирирівневу модель (рисунок 1.5).



Рисунок 1.4 – Триланкова фізична архітектура програмного забезпечення ІС

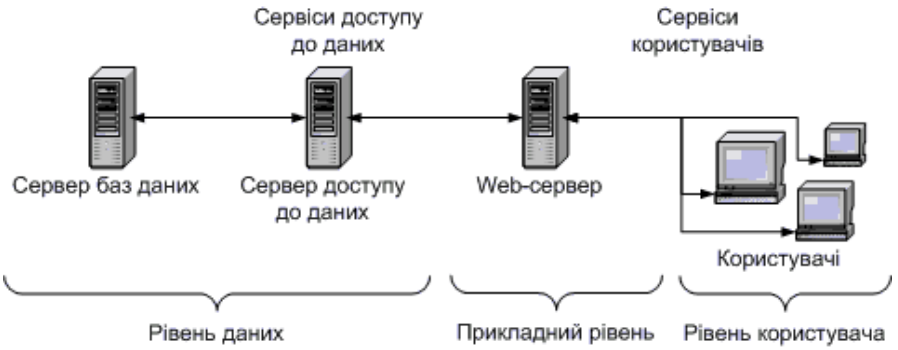


Рисунок 1.5 – Чотириланкова фізична архітектура програмного забезпечення ІС

Порівняємо отримані фізичні архітектури для інформаційної системи.

Переваги триланкової архітектури над чотириланковою:

- Краща продуктивність при невеликому числі користувачів.
- Порівняльна простота адміністрування системи.

Переваги чотири ланкової архітектури над триланковою:

- Краща продуктивність при великому числі користувачів.
- Краща масштабуємість.
- Краща надійність.
- Кращі можливості по модернізації і повторному використанню.

Таким чином, чотириланкова архітектура є більш перспективною. Вона в більшій мері задовольняє вимогам до інформаційної системи.

1.3.5 Вибір технологій для розробки інформаційної системи на підприємстві

Система є розподіленою і має чотири ланки: сервер баз даних, сервер доступу до даних, Web-сервер і клієнт.

Існують наступні технології для створення розподілених систем:

- DCOM/COM+, CORBA, RMI (об’єктно-орієнтовані технології).
- CGI, ISAPI/NSAPI, Servlets і інші.

Всі ці технології тим чи іншим способом організують виклик віддалених процедур (методів) чи його аналог.

Взаємодію між клієнтськими застосунками та Web-сервером доцільно організовувати за допомогою технологій CGI, ISAPI/NSAPI та/або Servlets, оскільки вони спеціально орієнтовані на роботу з веб-інтерфейсами. Використання технологій DCOM/COM+, CORBA чи RMI у цьому контексті суттєво підвищило б вимоги до програмного забезпечення клієнтських машин, що суперечить вимогам до інформаційної системи, яка повинна забезпечувати доступ через Internet з будь-якого комп’ютера глобальної мережі.

Усі зазначені технології (CGI, ISAPI/NSAPI, Servlets) розв'язують аналогічні задачі та характеризуються близькими показниками продуктивності. Серед них CGI є найбільш універсальною та поширеною: ISAPI обмежена використанням Windows-платформи, підтримку NSAPI припинено, а Servlets прив'язані до мови Java. З огляду на це вибір CGI є найбільш раціональним.

Для організації взаємодії між серверами необхідно застосовувати більш розвинуті технології, такі як DCOM/COM+, CORBA або RMI. Найбільш перспективною та універсальною вважається CORBA, однак вона має низку недоліків. CORBA є відносно молодою технологією, що перебуває у стадії активного розвитку, тому характеризується наявністю невирішених технічних проблем. Крім того, не всі стандартні сервіси CORBA реалізовані на належному рівні, а використання технології є платним, що відрізняє її від COM.

Технологія RMI орієнтована виключно на Java-середовище та потребує використання мови Java. Оскільки Java є інтерпретованою мовою, це може негативно позначатися на продуктивності системи, хоча й забезпечує її переносимість.

Основним недоліком DCOM/COM+ є залежність від Windows-платформи. Водночас DCOM/COM+ довела свою ефективність у розробленні інформаційних систем, є безкоштовною та слугує основою для низки інших технологій, зокрема OLE DB, яка забезпечує один із найпродуктивніших способів доступу до даних. З огляду на практичну перевіреність, безкоштовність та відповідність вимогам до системи, використання DCOM/COM+ для взаємодії серверів є доцільним.

Як зазначалося раніше, на рівні користувача необхідно застосовувати веб-інтерфейс, реалізований засобами HTML/DHTML, CSS, JavaScript, DOM або Java-аплетів. Клієнтським програмним забезпеченням у цьому випадку виступає веб-браузер.

Доступ до даних може здійснюватися за допомогою технологій ODBC, OLE DB та/або ADO. Оскільки ADO базується на OLE DB, безпосереднє використання OLE DB забезпечує вищу продуктивність. Крім того, OLE DB є універсальнішою технологією порівняно з ODBC, оскільки дозволяє працювати з даними різних типів.

Обрані технології, що застосовуються під час розроблення інформаційної системи обліку паливних матеріалів на підприємстві «Південь-Транс», наведено на рисунку 1.6.

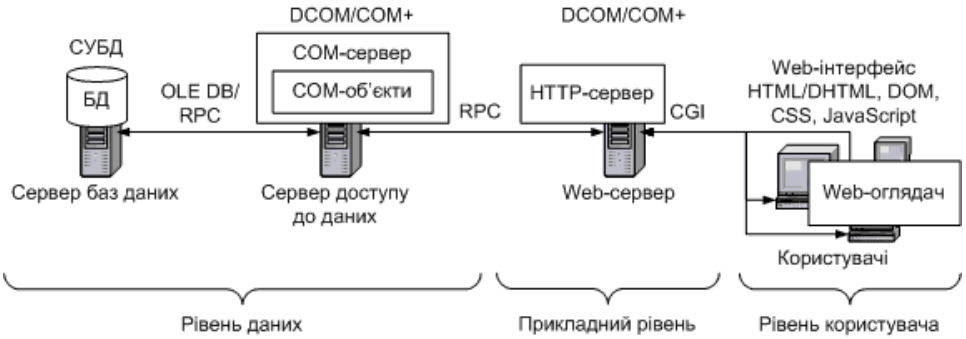


Рисунок 1.6 – Технології для розробки ІС обліку паливних матеріалів на підприємстві

1.3.6 Розробка варіантів концепції інформаційної підсистеми обліку паливних матеріалів на підприємстві

Виходячи з аналізу вимог до інформаційної системи, її функціональних характеристик, фізичної архітектури і технологій для розробки, можна розробити концепцію інформаційної системи. Основними компонентами системи є:

- серверна платформа і серверне програмне забезпечення;
- система управління базами даних;
- клієнтська платформа і клієнтське програмне забезпечення;
- засоби розробки;
- мережне середовище;
- технічні засоби.

Від вибору перерахованих компонентів залежить ефективність інформаційної системи. Розглянемо кожен компонент інформаційної системи.

Серверна платформа і серверне програмне забезпечення

Оскільки для розробки системи використовується технологія DCOM/COM+, то на всіх трьох серверах повинна використовуватися Windows-платформа. Відзначимо, що на сервері баз даних можливо використання і UNIX. Але при цьому адміністрування системи сильно ускладниться, а також може погіршитися продуктивність (якщо немає відповідного OLE DB драйвера).

Необхідну функціональність, надійність, безпеку і продуктивність інформаційної системи можуть забезпечити наступні операційні системи (у порядку збільшення продуктивності і функціональності):

- Windows NT Server 4.0 SP 5.
- Windows 2000 Server.
- Windows 2008 Server.
- Windows 2008 Advanced Server.

Ці мережні операційні системи включають підтримку DCOM/COM+.

Випуск Windows 8 Ultimate уже припинений, підтримка обмежена, тому використовувати цю операційну систему в нових розробках нераціонально.

Windows 10 перевершує Windows 8 по продуктивності і задовольняє усім вимогам інформаційної системи.

Windows 11 Ultimate має всі можливості Windows 11 Pro, а також додаткові можливості (підтримка кластерів, підтримка більшої кількості оперативної пам'яті і мікропроцесорів). Windows 10 Ultimate має кращу продуктивність і надійність, але можливості цієї операційної системи набагато перевищують максимальні вимоги до інформаційної системи, що розробляється, тому немає необхідності купувати більш дорогу інформаційну систему, якщо її можливості не будуть використані повною мірою.

Те ж саме можна сказати і про Windows Server 2016. Це сама могутня операційна система, що пропонує компанія Microsoft на сьогоднішній день.

Таким чином, Windows 11 Pro – це найбільш оптимальний вибір для інформаційної системи, що розробляється. Windows 10 Pro може підтримувати оперативну пам'ять до 64 Гб (2 Гб – мінімальна; 4 Гб – рекомендується), до 16 мікропроцесорів (мінімальний CPU – 2000 Мгц) і жорсткий диск від 32 Гб і більше. Усе це додасть інформаційній системі широкі можливості по нарощуванню продуктивності і забезпечить високу масштабуємість.

На Web-сервері повинен бути встановлений HTTP-сервер з підтримкою CGI, HTTP 1.1 і можливістю аутентифікації користувача. Найрозповсюдженими HTTP-сервера, що задовольняючі зазначеним вимогам і мають високу продуктивність, є:

- Apache Server 2.2 і вище.
- MS IIS 5.0 (Internet Information Services).

На Web-сервері також повинно бути встановлене програмне забезпечення (CGI-сценарії), що реалізує бізнес-правила інформаційної системи; його необхідно розробити.

На сервері доступу до даних необхідно установити програмне забезпечення у виді COM-серверів, яке необхідно розробити.

На сервері баз даних повинна бути встановлена серверна система управління базами даних (СУБД).

Система управління базами даних

Необхідна сучасна серверна реляційна СУБД, що має високу продуктивність, здатна працювати з великими обсягами інформації і підтримує стандарт SQL-92. Крім того, СУБД повинна працювати під управлінням Windows. Серверними СУБД, що задовольняють зазначеним вимогам, на сьогодні є:

- MS SQL Server 2008.
- MySQL 8.x.
- PostgreSQL 15.
- Sybase (SQL Anyware Studio 18).
- Oracle 9i (Standard Edition чи Enterprise Edition).
- IBM Informix Dynamic Server 11.7 (Standard Engine).

Клієнтська платформа і клієнтське програмне забезпечення

На клієнтських машинах може використовуватися будь-яка операційна система з можливістю встановлення віддаленого мережного з'єднання і графічним інтерфейсом. Операційні системи, що рекомендуються – MS Windows 10/ Windows 7, Linux.

Програмне забезпечення, з яким буде працювати користувач, являє собою Web-оглядач з підтримкою HTML 4.0, JavaScript 1.0, CSS 2 (наприклад, MS Edge, Mozilla Firefox, Google Chrome, Opera).

Засоби розробки

Для розробки програмного забезпечення сервера доступу до даних (COM-серверів і COM-об'єктів) необхідно використовувати мови програмування і засоби розробки, що підтримують COM-технологію. Такими мовами є C++, Visual Basic і Embarcadero Delphi. Visual Basic і Embarcadero Delphi не мають стандарту і не підтримуються ніким, крім Microsoft і Embarcadero, відповідно. Це не говорить про те, що ці мови використовувати не можна – це говорить про слабкість підтримки даної технології на рівні відповідності затвердженим промисловим стандартам. Крім того, C++ є найрозповсюдженою мовою програмування, що спростить розробку і супровід програмного забезпечення; програмне забезпечення, розроблене на C++, має високу продуктивність [6]. Отже, у якості мови програмування для розробки COM-серверів і COM-об'єктів необхідно використовувати C++, а засіб розробки – Visual C++ 12i вище. Можна використовувати і C++ Builder, що має аналогічні можливості. Однак Visual C++ підтримується компанією Microsoft (розробником COM), тому Visual C++ має більш повну підтримку COM-технології, ніж C++ Builder [4].

Для розробки програмного забезпечення Web-сервера (CGI-сценаріїв) можна використовувати різні мови: C, C++, Perl, PHP, Pascal і інші (CGI-сценарії можна писати на будь-якій універсальній мові програмування, багатьох спеціалізованих типу Prolog і навіть на мовах оболонок типу bash). Кожна мова має свої переваги і недоліки. Perl уступає мові C/C++ у швидкості виконання програм, але перевершує його в наявності могутніх вбудованих засобів обробки текстової інформації й у зручності розробки (не потрібно компілювати програми). PHP має аналогічні можливості що і Perl, але PHP не є універсальною мовою програмування і має незрівнянно більш слабку підтримку, чим Perl (CPAN містить велику кількість модулів, документацію, версії для різних платформ, первісні тексти й інше). Pascal і інші мови програмування або менш придатні для рішення наших задач, або уступають по своїх характеристиках мовам C, C++, Perl і PHP. Не можна також забувати, що CGI-сценарії повинні взаємодіяти з COM-об'єктами, а значить підтримувати COM-технологію. Таким чином, для розробки CGI-сценаріїв необхідно використовувати мову Perl (Active Perl 5.2) завдяки наявності підтримки

COM, могутніх вбудованих засобів обробки текстової інформації (регулярні вираження) і зручності розробки і супроводу.

Мережне середовище

Підприємство Південь-Транс має локальну комп'ютерну мережу Ethernet, яку можна використовувати як мережне середовище інформаційної системи. Можливо, потрібно буде внести зміни в конфігурацію існуючої комп'ютерної мережі. Основна вимога до мережі – швидкість передачі даних не менше 100 Мбіт/с.

Технічні засоби

Технічні засоби серверів повинні задовольняти вимогам операційної системи і СУБД до обладнання. Конфігурація кожного сервера може бути наступною:

- Мікропроцесор класу Intel Core i5 і вище.
- Оперативна пам'ять RAM 4096 Мб і більше.
- Жорсткий диск з вільним місцем 50 Гб і більше (сервер баз даних повинен мати жорсткий диск з більшим обсягом вільного місця – 80 Гб і більше).
- Мережна карта Ethernet (не менше 100 Мбіт/с).

Технічні засоби клієнтських машин повинні відповідати вимогам операційної системи, що використовується. Конфігурація комп'ютера користувача може бути наступною:

- Мікропроцесор класу Intel Core i3 і вище.
- Оперативна пам'ять RAM 2048 Мб і більше.
- Жорсткий диск з вільним місцем 50 Гб і більше.
- Мережна карта Ethernet (не менше 100 Мбіт/с).

Варіанти концепції інформаційної системи

Таким чином, розглянувши компоненти системи і вибравши найбільш оптимальні, можна визначити кілька варіантів концепції інформаційної системи обліку паливних матеріалів на підприємстві «Південь-Транс» (див. таблицю 1.1).

У таблиці використані наступні скорочення:

- ПЗ – програмне забезпечення.
- ОС – операційна система.
- СУБД – система управління базами даних.

Таблиця 1.1 – Варіанти концепції інформаційної системи

№	ПЗ Web-сервера (ОС, HTTP-сервер)	ПЗ сервера доступу до даних (ОС)	ПЗ сервера баз даних (ОС, СУБД)	Засоби розробки (CGI, COM-сервера)
1	MS Windows 11 Pro, MS IIS 5.0	MS Windows 11 Ultimate	MS Windows 11 Ultimate, IBM Informix 11.7	MS Visual C++, Delphi XE5
2	MS Windows 11 Pro, Apache Server 2.1	MS Windows 11 Pro	MS Windows 11 Ultimate, MS SQL Server 2022	MS Visual C++, Delphi XE5
3	MS Windows 11 Pro, MS IIS 5.0	MS Windows 11 Ultimate	MS Windows 11 Ultimate, Sybase (SQL AS 12)	Delphi XE5, Delphi XE5
4	MS Windows 11 Ultimate, MS IIS 5.0	MS Windows 11 Pro	MS Windows 11 Ultimate, MS SQL Server 2022	MS Visual C++, Delphi XE5

Крім того, кожен варіант концепції системи має відповідне технічне (описане вище) і організаційне забезпечення. Причому, і технічне, і організаційне забезпечення будуть однаковими для усіх варіантів, оскільки обране інформаційне забезпечення має приблизно однакові вимоги до апаратних ресурсів і обслуговування.

При виборі варіантів концепції інформаційної системи, були враховані наступні особливості:

- На Web-сервері та сервері доступу до даних повинна функціонувати операційна система з повною підтримкою технологій DCOM/COM+, тобто одна з Windows-систем. Сервер баз даних може працювати як під управлінням Windows, так і UNIX. Якщо сервер баз даних не є виділеним (наприклад, поєднаний із сервером доступу до даних), використання Windows-системи є обов'язковим.
- Доцільно встановлювати однакову операційну систему на всіх серверах, оскільки це спрощує їх адміністрування та технічну підтримку. Знайти фахівця, який однаково добре володіє кількома серверними операційними системами, значно складніше.
- СУБД MySQL у стандартному дистрибутиві не підтримує вкладених запитів і демонструє деяке зниження продуктивності під управлінням Windows-систем.
- Використання СУБД Oracle є недоцільним через високі вимоги до системних ресурсів та неповне використання її функціональних можливостей у межах даної системи.

- Використання ОС Windows 11 Enterprise є нераціональним, оскільки її розширені можливості не будуть задіяні повною мірою, а вартість придбання є значною.
- Операційна система Windows 10 Ultimate більше не випускається, а її підтримка обмежена, тому застосування цієї ОС у нових розробках є недоцільним.
- Використання ОС Windows 11 Enterprise може забезпечити лише незначне підвищення продуктивності порівняно з Windows 11 Pro, оскільки додаткові можливості старшої редакції не будуть затребувані в межах даної інформаційної системи.

1.3.7 Вимоги, що впливають на якість інформаційної системи обліку паливних матеріалів на підприємстві

Якість інформаційних систем та програмного забезпечення не піддається однозначному визначенню чи точному вимірюванню. На сучасному етапі розвитку інформаційних технологій не існує ані усталеної метрології якості інформаційних систем, ані повноцінних методик оцінювання якості програмного забезпечення, ані стандартизованих підходів до оцінювання якості процесів проєктування, розроблення та впровадження.

Попри це, проблема залишається актуальною та потребує системного вирішення. У зв'язку з цим доцільно сформувати перелік вимог (критеріїв), що можуть впливати на якість інформаційної системи. Частина таких критеріїв ґрунтується на положеннях міжнародного стандарту ISO 9126:

- Здатність підвищити ефективність роботи підприємства.
- Функціональність.
  - Відповідність призначенню.
  - Точність.
  - Повнота інформації.
  - Здатність взаємодіяти із середовищем.
  - Відповідність нормам.
  - Безпека (захист від взлому даних і інших злочинних дій).
- Надійність.
  - Зрілість ("обкатаність").
  - Стійкість до відмов.
  - Стійкість до помилок користувача.
  - Здатність відновлюватися після збоїв.
- Масштабованість.
- Придатність до використання.
  - Зрозумілість.
  - Легкість у вивченні.

    - Зручність і простота в роботі.
- Ефективність.
    - Продуктивність і час відгуку.
    - Споживання ресурсів.
- Легкість у супроводженні.
    - Придатність до аналізації (діагностика причин помилок і зіставлення з первісним кодом).
    - Придатність до змін.
    - Стабільність.
    - Легкість у тестуванні.

Потрібно відзначити, що виконання всіх зазначених вимог ще не гарантує якість інформаційної системи. Також, невідповідність системи якій-небудь вимозі не говорить про те, що система неякісна. Як уже говорилося вище, не існує формальних методів оцінки якості інформаційної системи.

1.4 Постановка задачі

1.4.1 Загальні відомості

Потрібно створити автоматизовану інформаційну систему, призначену для підтримки та автоматизації функцій, що виконуються бухгалтером. Замовником розробки є автотранспортне підприємство «Південь-Транс». Проєкт має назву «Інформаційна система обліку паливних матеріалів на підприємстві “Південь-Транс”».

1.4.2 Призначення і мета створення інформаційної системи обліку паливних матеріалів на підприємстві

Призначення системи полягає в автоматизації функцій, що виконуються бухгалтером, а також у формуванні єдиної бази даних, у якій зберігатиметься повний комплекс інформації про шляхові листи, водіїв, транспортні засоби, види палива, поїздки та інші пов’язані дані. У перспективі така база даних може бути використана й для розв’язання адміністративних завдань.

Метою створення системи є підвищення продуктивності праці, поліпшення умов роботи персоналу, прискорення процесів розрахунку, скорочення трудомісткості облікових операцій та мінімізація ймовірності помилок у бухгалтерському обліку.

1.4.3 Вимоги до інформаційної системи обліку паливних матеріалів на підприємстві

Режим функціонування системи має відповідати режиму роботи працівників підприємства «Південь-Транс», тоді як засоби віддаленого доступу повинні

забезпечувати безперервну, цілодобову роботу. У графіку експлуатації системи необхідно передбачити інтервали для проведення профілактичних та технічних робіт.

Впровадження інформаційної системи не потребує створення додаткових робочих місць. Кваліфікація користувачів повинна відповідати їх посадовим обов'язкам, а рівень комп'ютерної підготовки — не нижче рівня «користувач ПЕОМ». Технічний супровід системи має здійснюватися штатними спеціалістами підприємства «Південь-Транс».

Компоненти системи повинні мати потенціал для модернізації та подальшого розвитку в межах обраних технологій або можливість їх часткової заміни без порушення цілісності системи. Цільове призначення системи має зберігатися протягом усього періоду експлуатації — до моменту повного морального та технічного застарівання її апаратного, програмного, інформаційного та організаційного забезпечення.

Інформаційна система повинна відповідати таким функціональним характеристикам:

1) Аутентифікація користувача.
2) Ідентифікація користувача (надання користувачу тільки тих даних, які необхідні йому для виконання своєї роботи).
3) Конфіденційність передачі даних по комп'ютерній мережі.
4) Ведення обліку паливних матеріалів за шляховим листом:
   - введення шляхового листа
   - пошук шляхового листа за номером
   - пошук, довведення, зміни та видалення водія з довідника
   - пошук, довведення, зміни та видалення автомобіля з довідника
   - довведення, зміни та видалення причепа з довідника
   - довведення, зміни та видалення палива з довідника
5) Надання користувачу інформації про його обов'язки.
6) Надання зручного Web-інтерфейсу:
   - однакове представлення інформації незалежно від платформи і програмного забезпечення машини користувача;
   - сортування даних;
   - проста навігація;
   - можливості налаштування шрифтів інтерфейсу й інших параметрів інтерфейсу;
   - можливість збереження параметрів інтерфейсу між сеансами роботи;
   - можливість автоматичного заповнення полів форм.
7) Ведення записної книжки користувача.
   - додавання запису;
   - зміна запису;
   - вилучення запису.
8) Можливість віддаленої роботи з інформаційною системою як у локальній мережі, так і в Internet.

1.4.4 Вимоги до інформаційного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві

Система повинна бути відкритою, тобто такою, що використовує загальноприйняті стандарти та забезпечує мобільність і інтероперабельність програмних компонентів.

Система має підтримувати розподілену багаторівневу архітектуру, що гарантує можливість її масштабування відповідно до зростання обсягів даних і кількості користувачів.

Система повинна забезпечувати комплексний захист даних, який включає механізми аутентифікації та ідентифікації користувачів, захист каналів передавання інформації шляхом шифрування, а також фізичний захист обладнання.

Web-сервер та CGI-сценарії повинні реалізовувати бізнес-логіку системи та формувати веб-інтерфейс, який забезпечує доступ до інформації через мережі Intranet/Internet. Серверні компоненти мають генерувати документи та форми мовами HTML/DHTML на основі даних, що зберігаються у базі даних інформаційної системи, а також на основі інформації з файлів записної книжки. Вхідними даними слугують, зокрема, поля форм, заповнені користувачем.

Сервіси доступу до бази даних повинні бути винесені на окремий сервер з метою розвантаження Web-сервера та підвищення масштабованості системи.

Сервер бази даних також має бути виділеним, що забезпечує зменшення навантаження на інші сервери та підвищує рівень безпеки зберігання інформації. Як програмне забезпечення сервера бази даних необхідно використовувати серверну СУБД, яка підтримує стандарт SQL-92 та універсальні засоби доступу до даних. Використання стандартних можливостей СУБД є необхідним для забезпечення переносимості системи.

1.4.5 Вимоги до лінгвістичного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві

Web-інтерфейс системи має бути реалізований із використанням технологій HTML/DHTML, CSS та JavaScript. Функціональність на стороні клієнта («тонкий клієнт») повинна розроблятися мовою JavaScript, що забезпечує необхідний рівень інтерактивності та підтримку сучасних веб-стандартів.

Для створення клієнтської частини програмного забезпечення доцільним є використання мови програмування Delphi XE5 компанії Embarcadero, яка є об’єктно-орієнтованою та повною мірою відповідає вимогам методології ООП.

База даних інформаційної системи буде реалізована на основі серверної СУБД MS SQL Server 2022, що забезпечує високу продуктивність, масштабованість та надійність зберігання даних.

1.4.6 Вимоги до програмного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві

Інформаційна система включає кілька серверів (Web-сервер з бізнес логікою; сервер із сервісами доступу до даних; сервер баз даних) і клієнтські машини.

Вимоги до програмного забезпечення Web-сервера:

- Операційна система: Windows 11 Professional, Windows 11 Ultimate, Windows 2022 Advanced Server. Операційна система, що рекомендується – MS Windows 11 Ultimate.
- Реалізація DCOM/COM+.
- Web-сервер (HTTP-сервер): будь-який із підтримкою HTTP 1.1, CGI і можливістю аутентифікації користувача. Web-сервер, що рекомендується – Apache Server 2.1 і вище.
- Інтерпретатор мови Perl: ActivePerl 5.2 і вище.

Вимоги до програмного забезпечення сервера доступу до даних:

- Операційна система: Windows 10 Professional, Windows 2022 Server, Windows 2022 Advanced Server, Windows 2022 Datacenter Server, Windows .NET Server. Операційна система, що рекомендується – MS Windows 2022 Server.
- Реалізація DCOM/COM+.

Вимоги до програмного забезпечення сервера баз даних:

- Операційна система: Windows 11 Professional, Windows 11 Ultimate. Операційна система, що рекомендується – MS Windows 11 Professional. Можливо використання СУБД на платформі Unix, якщо вона цілком підтримує стандарт SQL 92 і має OLE DB-драйвер.
- Сервер баз даних: рекомендується використання MS SQL Server 2022. Інші можливі сервера: MySQL, PostgreSQL, Sybase, Oracle, Informix.
- OLE DB та/або ODBC драйвера.

Вимоги до програмного забезпечення клієнтської машини:

- Операційна система: будь-яка операційна система з можливістю установки віддаленого мережного з'єднання і графічним інтерфейсом. Операційні системи, що рекомендуються – MS Windows 10 / Windows 11, Linux.
- Web-оглядач: будь-який Web-оглядач із підтримкою HTML 5.0, JavaScript, CSS 3. Web-оглядачі, що рекомендуються: MS Edge, Mozilla Firefox, Google Chrome.

1.4.7 Вимоги до технічного забезпечення інформаційної системи обліку паливних матеріалів на підприємстві

Інформаційна система складається з кількох серверних компонентів: Web-сервера, на якому реалізовано бізнес-логіку; сервера, що забезпечує сервіси доступу до даних; а також сервера баз даних. Крім того, до її складу входять клієнтські робочі станції.

Вимоги до технічного забезпечення Web-сервера

- Мікропроцесор класу Intel Core i5 або потужніший
- Оперативна пам’ять обсягом не менше 16 ГБ
- Жорсткий диск із вільним простором не менше 150 ГБ
- Мережева карта Ethernet зі швидкістю не нижче 100 Мбіт/с

Вимоги до технічного забезпечення сервера доступу до даних

- Мікропроцесор класу Intel Core i5 або потужніший
- Оперативна пам’ять обсягом не менше 16 ГБ
- Жорсткий диск із вільним простором не менше 150 ГБ
- Мережева карта Ethernet зі швидкістю не нижче 100 Мбіт/с

Вимоги до технічного забезпечення сервера баз даних

- Мікропроцесор класу Intel Core i5 або потужніший
- Оперативна пам’ять обсягом не менше 16 ГБ
- Жорсткий диск із вільним простором не менше 80 ГБ
- Мережева карта Ethernet зі швидкістю не нижче 100 Мбіт/с

Вимоги до технічного забезпечення клієнтських машин

- Мікропроцесор класу Intel Core i3 або потужніший
- Оперативна пам’ять обсягом не менше 8192 МБ
- Жорсткий диск із вільним простором не менше 50 ГБ
- Мережева карта Ethernet зі швидкістю не нижче 100 Мбіт/с

Для організації мережного середовища можуть використовуватися кабелі типу «вита пара» категорії 5 або вище. Мережне обладнання повинно забезпечувати пропускну здатність не менше 100 Мбіт/с.

1.4.8 Вимоги до документації інформаційної системи обліку паливних матеріалів на підприємстві

Необхідна наявність наступних документів:

- Загальний опис системи.
- Керівництво користувача.
- Інструкція з формування і ведення бази даних.
- Керівництво адміністратора з установки програмного забезпечення на серверах.
- Програма і методика випробувань.

- Текст програми.

1.4.9 Порядок контролю і приймання інформаційної системи обліку паливних матеріалів на підприємстві

Перевірка документації інформаційної системи здійснюється замовником з метою встановлення відповідності або невідповідності розробленої системи вимогам технічного завдання (постановці задачі).

Тестування системи виконується розробником у процесі її створення без залучення замовника. Тестування передбачає експлуатацію системи в контрольованих умовах та аналіз отриманих результатів. У ході тестування перевіряється коректність роботи системи як із валідними, так і з помилковими даними та подіями. Тестування розподілених систем є особливо складним через значну кількість взаємозв'язків між компонентами. Кожна залежність, веб-сторінка, програмний компонент, база даних, елементи графічного інтерфейсу, програмне забезпечення проміжного рівня та мережна інфраструктура повинні перевірятися не лише на коректність функціонування, а й на сумісність у різних конфігураціях.

Оптимальним підходом до тестування розподілених систем є метод «знизу догори». На першому етапі розробник тестує кожен компонент окремо; на другому — перевіряє взаємодію компонента з іншими елементами системи на автономному комп'ютері; на третьому — здійснює тестування всієї системи в розподіленому середовищі. Усі зазначені етапи виконуються розробником без участі замовника.

Приймання системи здійснюється замовником. Система вважається придатною до експлуатації, якщо вона відповідає всім вимогам технічного завдання. Замовник або його уповноважені представники проводять тестування функціональних можливостей системи методом «чорної скриньки» у присутності розробника. У разі виявлення помилок формується їх перелік та узгоджується строк їх усунення. Після виправлення недоліків замовник проводить повторне тестування.

Після усунення всіх виявлених помилок або за умови їх відсутності, а також за підтвердження відповідності системи вимогам технічного завдання, оформлюється та підписується акт приймання системи.

Висновок

У розділі здійснено комплексне формування вимог до інформаційної системи обліку паливних матеріалів підприємства «Південь-Транс», що включає аналіз існуючих процесів, характеристику наявної системи та визначення напрямів її вдосконалення. На

основі дослідження технологій розробки, логічної та фізичної архітектури, а також вибору відповідних програмних засобів сформовано концепцію майбутньої інформаційної підсистеми. Визначено повний перелік функціональних, інформаційних, лінгвістичних, програмних і технічних вимог, які забезпечують якість, надійність і ефективність системи. Сформульована постановка задачі узагальнює призначення, мету створення та критерії приймання системи, що створює методологічну основу для подальшого проєктування та реалізації інформаційної системи обліку паливних матеріалів на підприємстві.

2 Розробка проектних рішень по системі та її частинах

2.1 Загальносистемні рішення

2.1.1 Схема організаційної структури

Інформаційна система обліку паливних матеріалів розробляється для автотранспортного підприємства

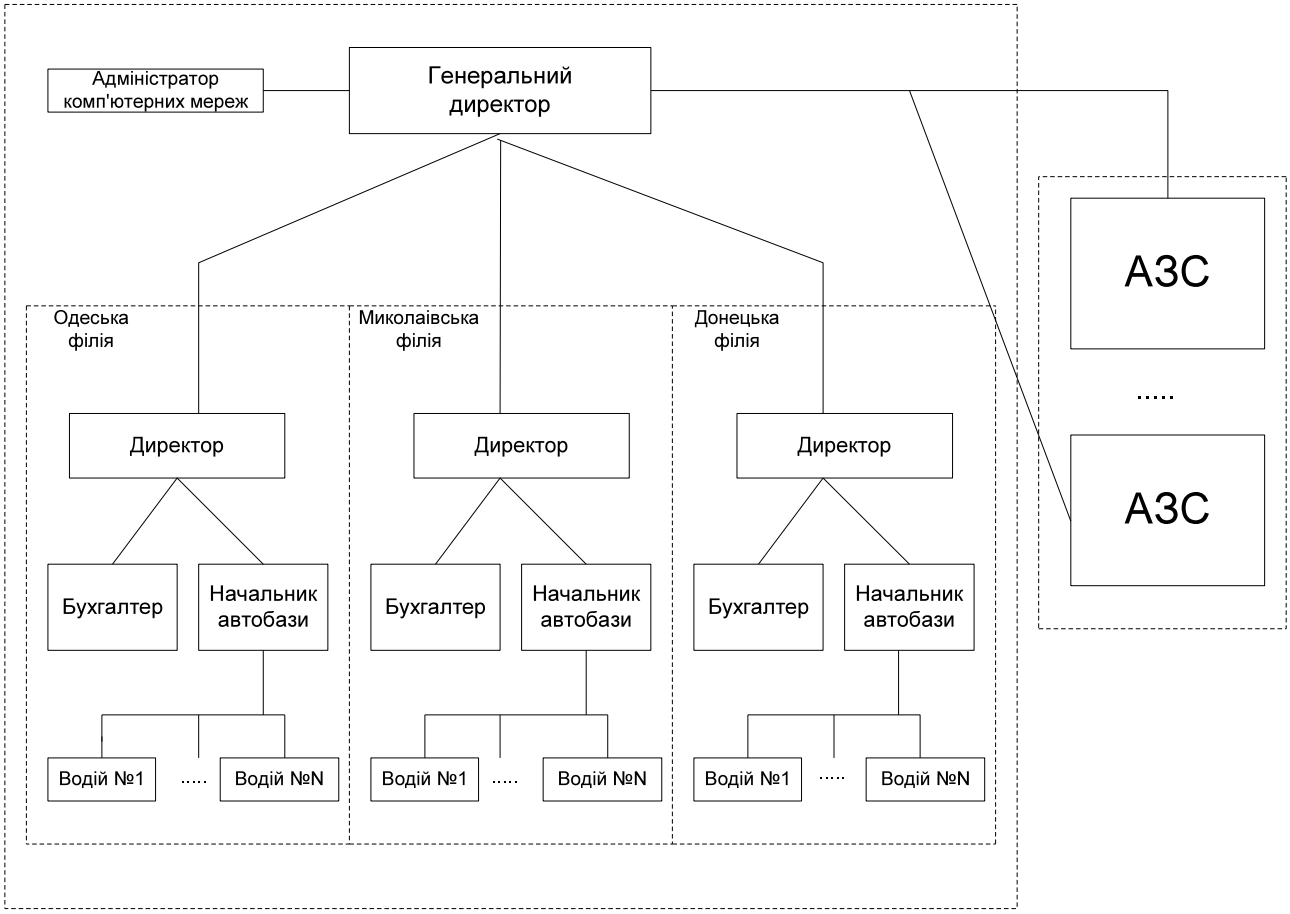


Рисунок 2.1 – Схема організаційної структури підприємства «Південь-Транс»

Користувачами системи, що розробляється, будуть директор, бухгалтер та начальник автобази.

2.1.2 Схема функціональної структури

Для уточнення функціональних можливостей інформаційної системи, визначення її функціональної структури, окреслення підсистем (модулів), що входять до її складу, а також для аналізу динаміки роботи системи в цілому необхідним є побудова функціональної моделі.

Функціональна модель відображає процеси обчислення в системі. Вона демонструє, яким чином вихідні дані формуються на основі вхідних, не деталізуючи порядок або спосіб реалізації цих обчислень. Така модель складається з набору діаграм потоків даних, які відображають рух інформації від зовнішніх джерел через операції та внутрішні сховища даних до зовнішніх споживачів.

Діаграми потоків даних (Data Flow Diagrams, DFD) є ключовим інструментом моделювання функціональних вимог під час проєктування системи. За їх допомогою вимоги декомпозуються на окремі функціональні компоненти (процеси) та подаються у вигляді мережі, пов'язаної потоками даних. Основною метою використання DFD є демонстрація того, як кожен процес трансформує свої вхідні дані у вихідні, а також виявлення взаємозв'язків між процесами. У подальшому для побудови DFD застосовуватиметься нотація Гейна–Сарсона (Gane–Sarson).

Зовнішні сутності (джерела інформації) генерують інформаційні потоки, які передають дані до підсистем або процесів. Останні здійснюють перетворення інформації та формують нові потоки, що спрямовуються до інших процесів, підсистем, накопичувачів даних або зовнішніх споживачів інформації. Таким чином, основними компонентами діаграм потоків даних є:

- зовнішні сутності;
- процеси;
- накопичувачі даних;
- потоки даних.

Зовнішня сутність у контексті діаграм потоків даних (DFD) розглядається як елемент, що перебуває поза межами інформаційної системи, але взаємодіє з нею шляхом передавання або отримання інформації. Це може бути як фізична особа, так і матеріальний об'єкт чи зовнішня організація, що виступає джерелом або приймачем даних. У межах розроблюваної системи зовнішніми сутностями є директор та головний бухгалтер, оскільки саме вони ініціюють інформаційні потоки (запити, команди, дані) та отримують результати роботи системи (звітність, аналітичні дані, довідкову інформацію).

Процес у DFD визначається як логічне перетворення вхідних потоків даних у вихідні відповідно до певного алгоритму або правила обробки. Процес не прив'язаний до конкретної технології реалізації: він може бути втілений програмним модулем,

функцією, процедурою, апаратним пристроєм або навіть ручною операцією. У розроблюваній системі основні процеси пов’язані з аутентифікацією користувачів, веденням обліку, обробкою інформації про шляхові листи, водіїв, транспортні засоби та паливні матеріали, формуванням звітів для друку, а також виконанням адміністративних функцій — підтримкою бази даних, налаштуванням параметрів системи та забезпеченням коректної роботи механізмів доступу.

Накопичувач даних у DFD є абстрактним логічним сховищем, призначеним для довготривалого або тимчасового зберігання інформації. Він не описує фізичний спосіб збереження даних, а лише фіксує факт їх наявності та можливість доступу до них у будь-який момент. Запис і зчитування можуть здійснюватися будь-якими технічними засобами, що не впливає на логічну структуру моделі. У межах даної системи накопичувачами даних є:

- База даних системи, що містить повний обсяг інформації підприємства, включаючи дані про транспортні засоби, водіїв, поїздки, паливо та інші облікові записи.
- Файли аутентифікації, у яких зберігаються облікові дані користувачів (логіни та паролі), необхідні для забезпечення контролю доступу.

Потік даних визначає інформацію, що передається від одного елемента системи до іншого. Він може відображати як логічний рух даних, так і фізичну передачу інформації різними каналами: через мережеве з’єднання, електронну пошту, зовнішні носії тощо. Потік даних у DFD фіксує лише зміст і напрям передавання інформації, не деталізуючи технічні механізми її транспортування.

Особливе значення у функціональному моделюванні має контекстна діаграма — найвищий рівень абстракції DFD. Вона подає систему як єдиний процес і відображає її взаємодію із зовнішнім середовищем. Контекстна діаграма демонструє всі зовнішні сутності, з якими система обмінюється інформацією, а також основні потоки даних між ними. Таким чином, вона визначає межі системи, її призначення та ключові інформаційні взаємозв’язки.

Контекстна діаграма інформаційної системи обліку паливних матеріалів на підприємстві «Південь-Транс» відображає загальну структуру взаємодії між системою, директором та головним бухгалтером, а також основні інформаційні потоки, що забезпечують функціонування облікових процесів.

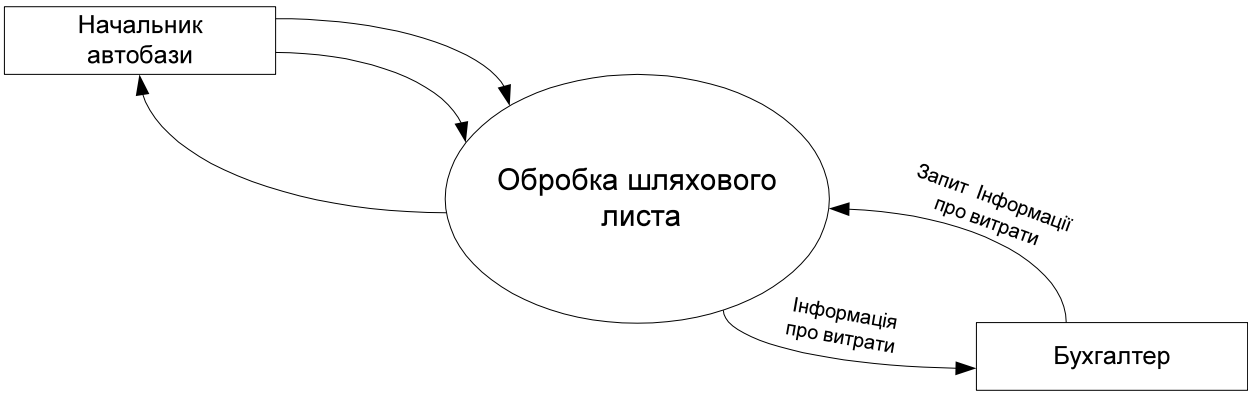


Рисунок 2.2 — Контекстна діаграма системи обліку паливних матеріалів на підприємстві «Південь-Транс»

У наведеній на рисунку 2.2 діаграмі, що відображає процес ОБРОБКА ШЛЯХОВОГО ЛИСТА, ідентифіковано такі зовнішні сутності: КОРИСТУВАЧ, СЕРВЕР, МОНІТОР та ПРИНТЕР.

Сутність КОРИСТУВАЧ представляє реальну фізичну особу, яка взаємодіє з програмним забезпеченням.

Сутність СЕРВЕР інтерпретується як віддалена обчислювальна система, на якій розміщено базу даних, що забезпечує обробку запитів та містить структуровані таблиці.

Сутності МОНІТОР і ПРИНТЕР відображають апаратні компоненти комп'ютера, призначені відповідно для візуалізації інформації та її матеріального відтворення у вигляді друкованих документів.

Потоки даних, що циркулюють між системою та зовнішніми об'єктами, мають такий характер. Для виконання своїх функцій КОРИСТУВАЧ вводить інформацію щодо кожного шляхового листа. Після завершення введення даних користувач формує запит, задаючи ключові параметри для вибірки необхідної інформації. Результати обробки запиту подаються у вигляді сформованого звіту, який може бути відображений на МОНІТОРІ або виведений на друк за допомогою ПРИНТЕРА.

Побудована контекстна діаграма має один процес, який може бути декомпозований в DFD першого рівня.

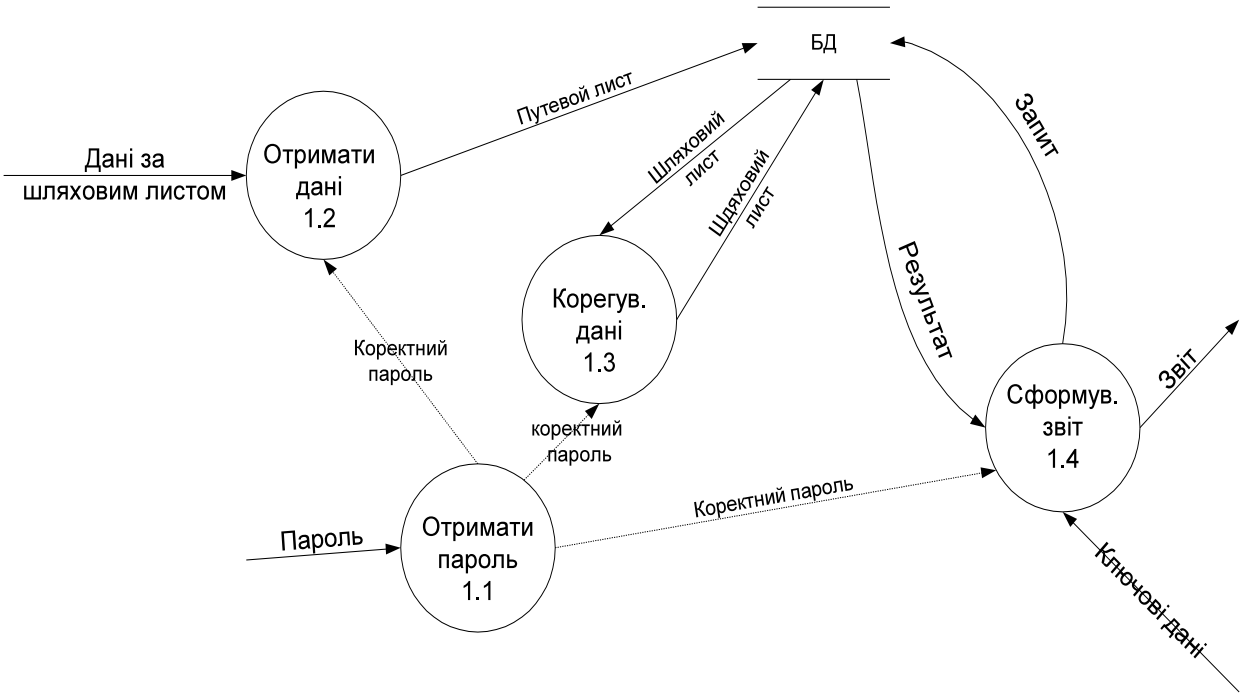


Рисунок 2.3 – деталізація процесу ОБРОБИТИ ШЛЯХОВІ ЛИСТИ

DFD-діаграма деталізації процесу ОБРОБИТИ ШЛЯХОВІ ЛИСТИ (рисунок 2.3) містить чотири основні процеси: ОТРИМАТИ ПАРОЛЬ, ОТРИМАТИ ДАНІ, КОРЕГУВАТИ ДАНІ та СФОРМУВАТИ ЗВІТ.

Процес ОТРИМАТИ ПАРОЛЬ забезпечує автентифікацію користувача та відкриває доступ до даних і функцій системи за умови коректного введення пароля.

Процес ОТРИМАТИ ДАНІ призначений для введення нового шляхового листа до сховища даних бази даних.

Процес СФОРМУВАТИ ЗВІТ генерує потік ЗАПИТ, що містить SQL-інструкцію, сформовану відповідно до КЛЮЧОВИХ ДАНИХ, і передає його до сховища БД. На основі потоку РЕЗУЛЬТАТ, отриманого від бази даних, процес формує вихідний потік ЗВІТ.

Процес КОРЕГУВАННЯ ДАНИХ забезпечує можливість редагування інформації, що надходить у потоці ШЛЯХОВИЙ ЛИСТ, та запису оновлених даних назад до бази даних.

Декомпозиція процесу 1.2 ОТРИМАТИ ДАНІ наведена на наступному рисунку.

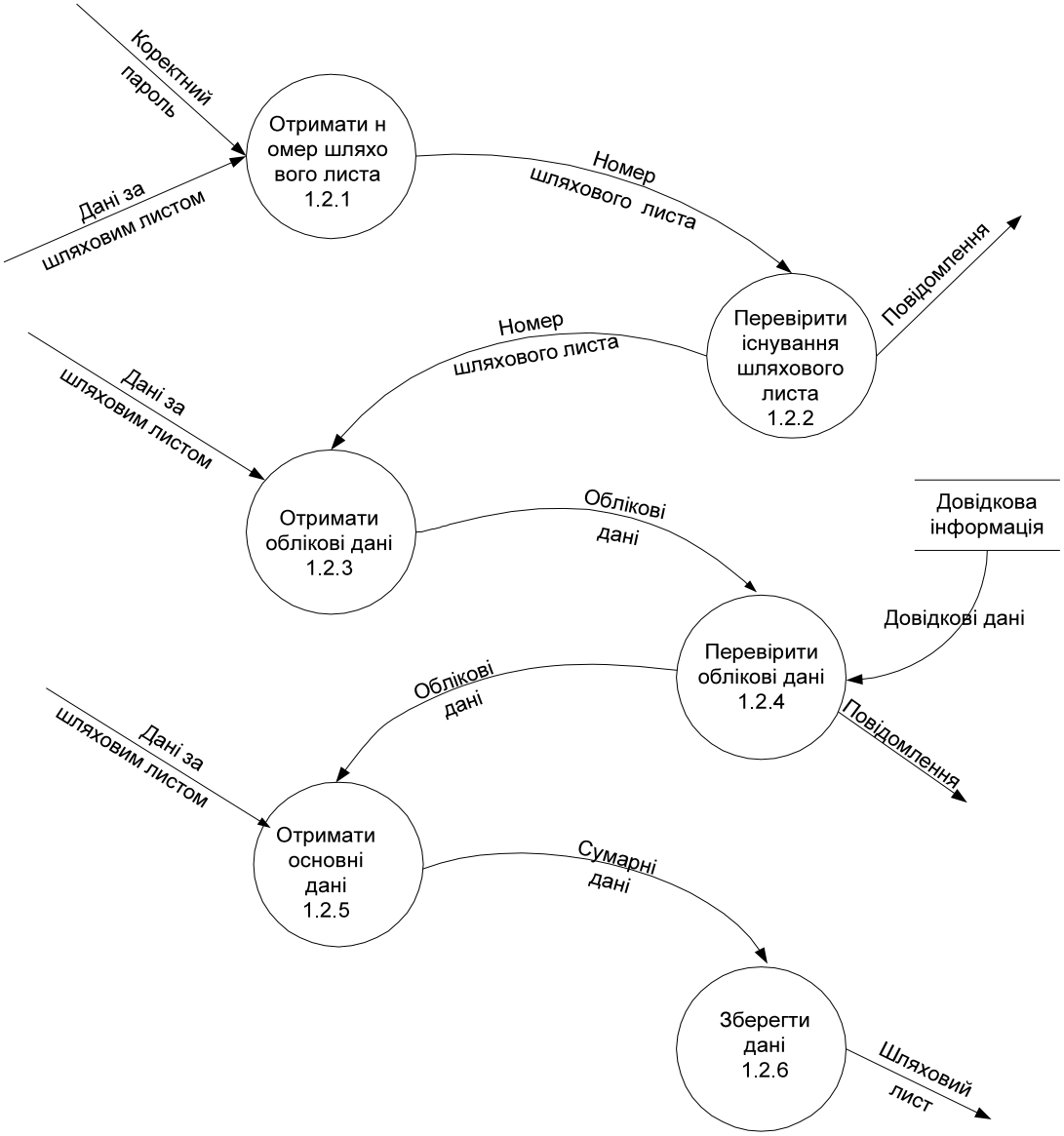


Рисунок 2.4 – Деталізація процесу ОТРИМАТИ ДАНІ

Деталізація процесу 1.2 ОТРИМАТИ ДАНІ (рисунок 2.4) описує послідовність дій, що виконуються під час введення нового шляхового листа до системи. На першому етапі здійснюється перевірка наявності запису з указаним номером у сховищі даних. Якщо шляховий лист із таким номером уже існує, система формує та відображає відповідне ПОВІДОМЛЕННЯ.

Далі з потоку ДАНІ ЗА ШЛЯХОВИМ ЛИСТОМ виділяються ОБЛІКОВІ ДАНІ, які проходять перевірку на відповідність довідковій інформації, що зберігається у відповідних накопичувачах даних. У разі успішної перевірки формується потік ОСНОВНІ ДАНІ.

Після цього ОБЛІКОВІ ДАНІ та ОСНОВНІ ДАНІ об’єднуються у потік СУМАРНІ ДАНІ, який передається до процесу ЗБЕРЕГТИ ДАНІ. Зазначений процес здійснює запис інформації до сховища та формує вихідний потік ШЛЯХОВИЙ ЛИСТ, що відображає успішне збереження нового запису.

Декомпозуємо процес 1.5 DFD - діаграми першого рівня – СФОРМУВАТИ ЗВІТ.

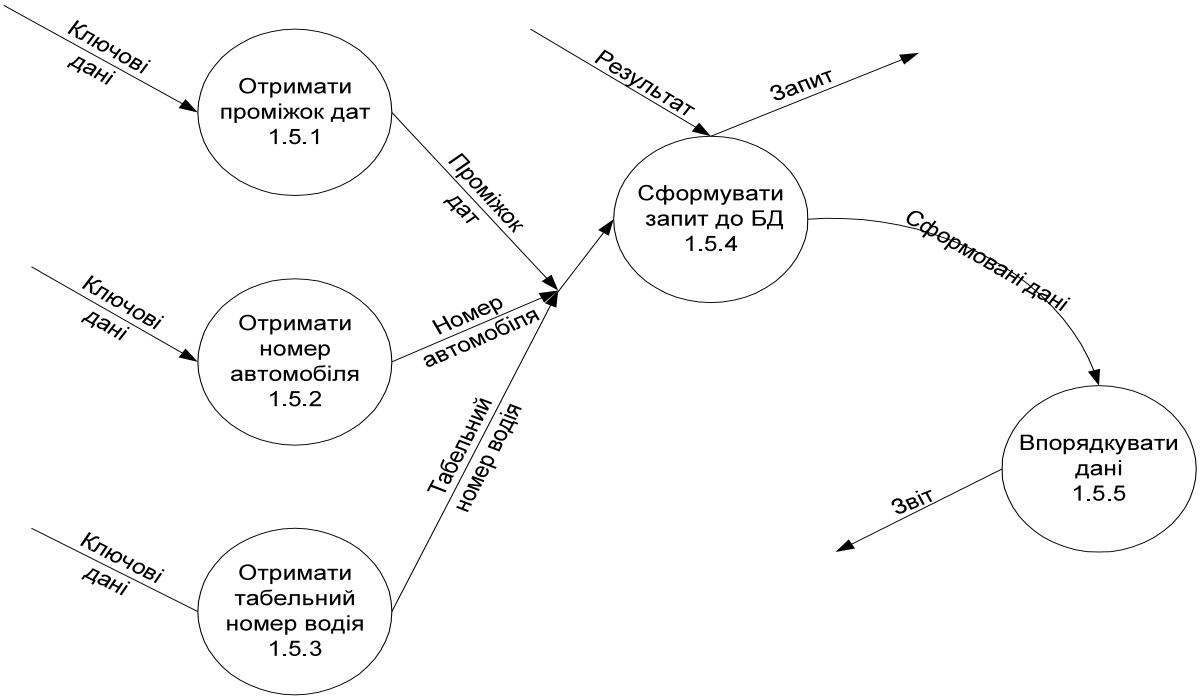


Рисунок 2.5 – Деталізація процесу СФОРМУВАТИ ЗВІТ

Декомпозиція процесу СФОРМУВАТИ ЗВІТ (рисунок 2.5) відображає послідовність операцій, необхідних для формування вибірки даних та підготовки звітної інформації. З потоку КЛЮЧОВІ ДАНІ процеси ОТРИМАТИ ПРОМІЖОК ДАТ, ОТРИМАТИ НОМЕР АВТОМОБІЛЯ та ОТРИМАТИ ТАБЕЛЬНИЙ НОМЕР ВОДІЯ відповідно виділяють окремі потоки ПРОМІЖОК ДАТ, НОМЕР АВТОМОБІЛЯ та ТАБЕЛЬНИЙ НОМЕР ВОДІЯ, які використовуються для уточнення параметрів вибірки.

Процес СФОРМУВАТИ ЗАПИТ генерує потік, що містить SQL-інструкцію, сформовану на основі зазначених параметрів, та передає його до сховища даних. Після отримання результатів вибірки формується потік СФОРМОВАНІ ДАНІ, який надходить до процесу ВПОРЯДКУВАТИ ДАНІ. Зазначений процес здійснює впорядкування та структуризацію інформації, подаючи її у форматі, зручному для подальшого перегляду та аналізу.

Декомпозиція процеса 1.2.4 DFD – ПЕРЕВІРИТИ ОБЛІКОВІ ДАНІ

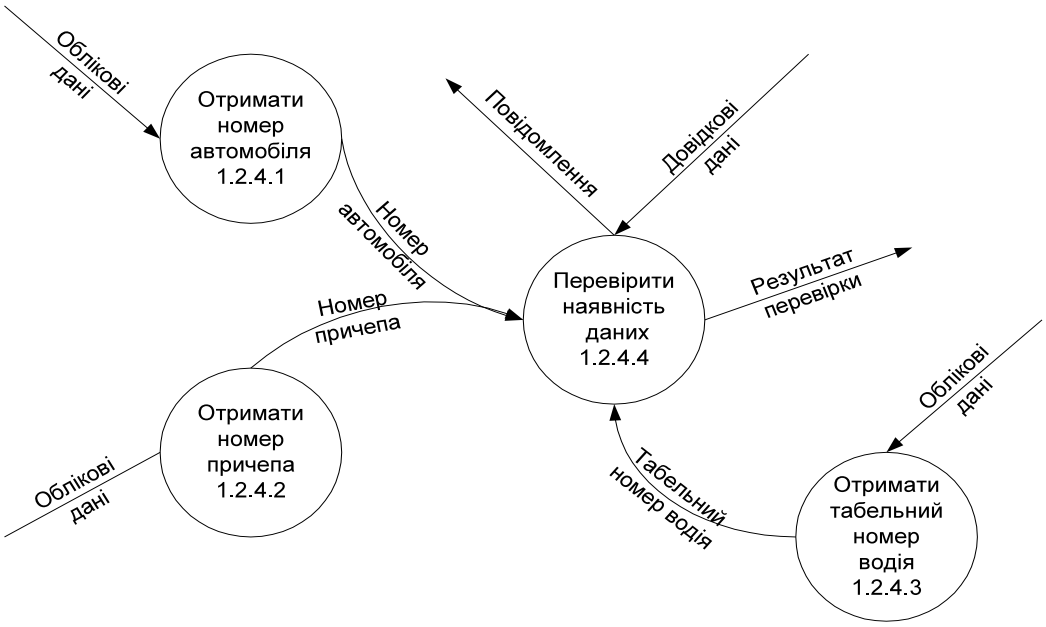


Рисунок 2.6 – Деталізація процесу ПЕРЕВІРИТИ ДАНІ

Процес ПЕРЕВІРИТИ ДАНІ, поданий у вигляді DFD-діаграми (рисунок 2.6), відображає механізм перевірки коректності облікової інформації. Процес ОТРИМАТИ НОМЕР АВТОМОБІЛЯ виділяє з потоку ОБЛІКОВІ ДАНІ окремий потік НОМЕР АВТОМОБІЛЯ та передає його до процесу ПЕРЕВІРИТИ НАЯВНІСТЬ ДАНИХ. Під час виконання цього процесу здійснюється порівняння отриманого значення з потоком ДОВІДКОВІ ДАНІ, у результаті чого формується потік РЕЗУЛЬТАТ ПЕРЕВІРКИ, що містить інформацію про наявність або відсутність відповідного номера автомобіля у сховищі даних.

Аналогічним чином функціонують процеси ОТРИМАТИ ТАБЕЛЬНИЙ НОМЕР ВОДІЯ та ОТРИМАТИ НОМЕР ПРИЧЕПА, які відповідно виділяють із потоку ОБЛІКОВІ ДАНІ потоки ТАБЕЛЬНИЙ НОМЕР ВОДІЯ та НОМЕР ПРИЧЕПА для подальшої перевірки їх наявності у довідкових даних.

2.1.3 Загальний опис системи

Призначення системи

Інформаційна система обліку паливних матеріалів призначена для автоматизації задач, що виконують директор та головний бухгалтер. Система розробляється для підприємства «Південь-Транс».

Інформаційна система повинна володіти наступними функціональними характеристиками:

1) Аутентифікація користувача.
2) Ідентифікація користувача (надання користувачу тільки тих даних, які необхідні йому для виконання своєї роботи).
3) Конфіденційність передачі даних по комп'ютерній мережі.
4) Ведення обліку паливних матеріалів за шляховим листом:
   - введення шляхового листа
   - пошук шляхового листа за номером
   - пошук, довведення, зміни та видалення водія з довідника
   - пошук, довведення, зміни та видалення автомобіля з довідника
   - довведення, зміни та видалення причепа з довідника
   - довведення, зміни та видалення палива з довідника
5) Надання користувачу інформації про його обов’язки.
6) Надання зручного Web-інтерфейсу:
   - однакове представлення інформації незалежно від платформи і програмного забезпечення машини користувача;
   - сортування даних;
   - проста навігація;
   - можливості налаштування шрифтів інтерфейсу й інших параметрів інтерфейсу;
   - можливість збереження параметрів інтерфейсу між сеансами роботи;
   - можливість автоматичного заповнення полів форм.
7) Ведення записної книжки користувача.
   - додавання запису;

- зміна запису;
- вилучення запису.

8) Можливість віддаленої роботи з інформаційною системою як у локальній мережі, так і в Internet.

Опис системи

Користувачами системи є директор та головний бухгалтер. Також головний бухгалтер повинен виконувати функції адміністратора – підтримувати роботу системи..

Інформаційне забезпечення системи складається з:

1) Бази даних, у якій зберігається вся інформація про ведення обліку паливних матеріалів, шляхових листів.
2) Файлів електронної записної книжки, у яких зберігаються записи, зроблені головним бухгалтером та директором.
3) Файлів аутентифікації, у яких зберігаються логіни та паролі (у зашифрованому виді) користувачів системи. З цими файлами безпосередньо працює тільки адміністратор системи.

Технічне забезпечення системи включає один або кілька серверів та робочі станції, розміщені в окремому приміщенні. Для забезпечення високої продуктивності та масштабованості доцільно використовувати щонайменше два сервери (хоча для мінімальної працездатності можливе розгортання всієї системи на одному сервері):

- Web-сервер
- Сервер прикладної логіки / сервер API-сервісів
- Сервер баз даних

Програмне забезпечення системи складається з кількох програмних пакетів:

- Пакет програм, що забезпечує функціональність, необхідну працівникам підприємства
- Пакет програм для адміністратора системи

Кожен пакет містить набір модулів, що реалізують окремі функції системи.

Бізнес-процеси реалізуються за допомогою сучасних серверних технологій, таких як RESTful API, Node.js, ASP.NET Core, Java Spring Boot або інші фреймворки, що розгортаються на Web-сервері та забезпечують обробку запитів користувачів.

Доступ до бази даних здійснюється через сучасні засоби взаємодії з даними, зокрема ORM-фреймворки (Entity Framework Core, Hibernate, Sequelize тощо) або нативні драйвери СУБД. Сервер прикладної логіки виконує роль проміжної ланки між

Web-інтерфейсом та базою даних, забезпечуючи безпечний і структурований доступ до інформації.

База даних є реляційною та функціонує під управлінням сучасної серверної СУБД, що підтримує стандарт SQL (наприклад, MS SQL Server 2022, PostgreSQL, MySQL/MariaDB). Взаємодія з базою даних здійснюється через офіційні драйвери або ORM-рішення.

Операційна система, встановлена на Web-сервері та сервері прикладної логіки, повинна забезпечувати стабільну роботу серверних застосунків і підтримку сучасних мережевих протоколів. Рекомендованими є Windows Server 2022 або сучасні дистрибутиви Linux (Ubuntu Server, Debian, CentOS Stream). Сервер баз даних працює під управлінням ОС, сумісної з обраною СУБД.

На Web-сервері має бути встановлений сучасний HTTP-сервер, наприклад Nginx або Apache, з підтримкою протоколу HTTP/1.1 або HTTP/2, механізмів аутентифікації користувачів та можливістю інтеграції з серверними застосунками.

2.1.4 Захист системи від несанкціонованого доступу

Аутентифікація

Аутентифікація – це процес, у ході якого на підставі пароля, ключа чи якої-небудь іншої інформації користувач підтверджує, що є саме тим, за кого себе видає.

При вході користувача у систему першою справою відбувається його аутентифікація. Якщо введені логін і пароль збігаються з тими, що зберігаються в системі на сервері, то користувач успішно входить у систему, інакше йому відмовляється в доступі. Тут доречно контролювати кількість спроб, щоб уникнути підбора паролів.

Логін і пароль необхідно передавати винятково методом POST. При методі GET дані кешуються проксі-серверами і Web-оглядачами, тобто імовірність вкрасти секретний пароль значно вище.

Наступним важливим питанням Web-аутентифікації є проблема аутентифікації при виклику кожного нового CGI-сценарію. Проблема полягає в тім, що пароль користувач вводить тільки один раз при першому вході в систему, а аутентифікацію треба робити щораз, коли він звертається до якого-небудь CGI-сценарію.

Найбільше часто використовуються наступні способи Web-аутентифікації:

1) Передавати логін і пароль кожному CGI-сценарію («носити за собою»).

2) Записати логін і пароль у файл cookie.

3) Згенерувати ключ з певним терміном дії і передавати його кожному CGI-сценарію.

4) Аутентифікація засобами HTTP-сервера.

У першому способі виникають суттєві обмеження, оскільки частина HTML-сторінок може приймати параметри виключно методом GET, що призводить до кешування логіна та пароля у відкритому вигляді. Перетворення всіх HTML-сторінок на серверні сценарії та примусове використання лише методу POST є недоцільним, оскільки це штучно знижує продуктивність сервера. З огляду на це, перший спосіб вважається ненадійним.

У другому та третьому способах логін і пароль передаються лише один раз, після чого створюються cookie або ключ, що діють протягом визначеного часу. Cookie — це дані з обмеженим строком життя, які зберігаються на стороні клієнта; ключ виконує аналогічну функцію, але зберігається на сервері. Використання cookie має два принципові недоліки. По-перше, вони можуть бути вимкнені або не підтримуватися браузером. По-друге, у разі роботи на загальнодоступному комп'ютері існує ризик несанкціонованого доступу до системи після завершення роботи користувача. Тому cookie можуть застосовуватися лише на персональному комп'ютері, доступ до якого гарантовано обмежений. У ситуаціях, коли користувач працює на спільному обладнанні, цей спосіб також є ненадійним.

Третій спосіб забезпечує вищий рівень захищеності. Після введення логіна та пароля сервер генерує ключ із визначеним строком дії. Надалі між сценаріями або HTML-документами передається лише ключ. Кожен серверний сценарій, отримавши ключ, може за ним визначити логін і пароль користувача. Попри надійність, цей спосіб є складним у реалізації, оскільки потребує створення спеціальної бази даних для зберігання ключів та облікових даних, а також механізмів генерації випадкових і унікальних ключів. Крім того, продуктивність сценаріїв знижується через необхідність постійного звернення до бази даних. Строк дії ключа також створює компроміс: надто короткий інтервал змушує

користувача повторно вводити логін і пароль, а надто довгий — підвищує ризик несанкціонованого доступу після завершення роботи.

Четвертий спосіб ґрунтується на використанні вбудованих механізмів HTTP-сервера. Сервер автоматично формує діалогове вікно для введення логіна та пароля, які передаються у зашифрованому вигляді. Після порівняння отриманих даних із тими, що зберігаються у конфігураційних файлах сервера, користувач отримує доступ до всіх ресурсів визначеної директорії (серверних сценаріїв, HTML-документів тощо). Для завершення роботи достатньо закрити вікно браузера. Цей спосіб є надійним, простим у впровадженні та не створює додаткового навантаження на серверні сценарії, оскільки не потребує звернень до бази даних. Дані для подальшої ідентифікації можуть бути отримані зі стандартних змінних оточення HTTP-сервера.

Отже, з погляду захищеності, ефективності та простоти реалізації найбільш доцільним є четвертий спосіб — аутентифікація засобами HTTP-сервера. Саме цей механізм використовується у підсистемі «Зарплата + Кадри».

Окрім аутентифікації на рівні HTTP-сервера, необхідно забезпечити аутентифікацію на сервері доступу до даних та сервері баз даних. На сервері доступу до даних налаштування прав доступу здійснюється засобами конфігурації компонентів, які визначають дозволені місця запуску та коло користувачів, що мають право доступу. Аутентифікація на сервері баз даних реалізується вбудованими механізмами відповідної СУБД.

Ідентифікація

Ідентифікація — це процес встановлення прав доступу, привілеїв, а також визначення властивостей і характеристик користувача на підставі його імені, логіна або іншої доступної інформації про нього.

Після проходження аутентифікації, що виконується вбудованими засобами HTTP-сервера, будь-який серверний сценарій може отримати логін користувача (наприклад, працівника чи викладача) зі змінних оточення сервера та здійснити процедуру ідентифікації.

Доступ до даних у підсистемі «Зарплата + Кадри» (за винятком загальнодоступної інформації) здійснюється шляхом вибірки даних із відповідних

джерел із використанням логіна працівника, якому надано відповідні привілеї доступу. Таким чином, за відсутності логіна отримання даних є неможливим, а за його наявності користувач може отримати лише ті відомості, доступ до яких йому дозволено.

Захист каналів зв'язку

Логін і пароль повинні передаватися виключно через захищене з'єднання, сформоване HTTP-сервером, у зашифрованому вигляді. Це є обов'язковою умовою для забезпечення конфіденційності облікових даних користувачів.

Інша інформація, що циркулює в системі, зазвичай має низьку чутливість і складно піддається інтерпретації сторонніми особами, тому вимоги до її захисту є менш критичними. Крім того, частина цих даних може бути доступною для ознайомлення іншим уповноваженим особам, зокрема керівництву підприємства. Водночас необхідно гарантувати, що змінювати, додавати або вилучати дані в базі можуть лише працівники, які мають відповідні повноваження. Це забезпечується механізмами аутентифікації та ідентифікації.

Слід враховувати, що в майбутньому система може бути розширена новими функціями та даними, які потребуватимуть підвищеного рівня захисту. Уже зараз у системі наявні персональні відомості (наприклад, дані з записної книжки працівників бухгалтерії чи відділу кадрів), які мають конфіденційний характер і повинні бути захищені від несанкціонованого перегляду.

Умовно канали зв'язку в системі можна поділити на такі типи:

- Між користувачами та Web-сервером. Рекомендований захист — використання протоколу SSL, який забезпечує встановлення зашифрованого каналу шляхом узгодження методу шифрування та обміну сертифікатами й ключами між сервером і браузером.
- Між Web-сервером і сервером доступу до даних. Рекомендований захист — використання механізмів безпеки, що надаються інфраструктурою компонентів, а також додаткових програмних засобів контролю доступу.
- Між сервером доступу до даних і сервером баз даних. Рекомендований захист — застосування вбудованих механізмів безпеки СУБД, включно з підтримкою SSL-шифрування.

Необхідно усвідомлювати, що захист каналів зв'язку потребує виконання додаткових операцій, пов'язаних із шифруванням, обміном ключами та сертифікатами. Це може призвести до певного зниження продуктивності системи, що слід враховувати під час її проєктування та експлуатації.

2.2 Рішення з організаційного забезпечення

Інформаційна система обліку паливних матеріалів розробляється для підприємства «Південь-Транс». Організаційна структура підприємства подана на рисунку 2.1.

Користувачами створюваної підсистеми є директор та головний бухгалтер, при цьому функції адміністратора системи покладаються на головного бухгалтера.

Отже, впровадження інформаційної системи не передбачає змін в існуючій організаційній структурі управління підприємства.

2.3 Рішення з інформаційного забезпечення

2.3.1 Опис інформаційного забезпечення системи

Склад інформаційного забезпечення

Інформаційне забезпечення системи складається з:

- Бази даних, у якій зберігається вся інформація про водія, транспортний засіб, паливо, шляховий лист, поїздки.
- Файлів аутентифікації, у яких зберігаються логіни та паролі. З цими файлами безпосередньо працює тільки адміністратор системи.

Організація інформаційного забезпечення

База даних повинна зберігатися в реляційній СУБД. Рекомендується використання СУБД MS SQL Server 2022. База даних розташовується на жорсткому диску сервера баз даних. Розмір вільного місця жорсткого диска повинен бути не менше 150 Гб (оцінка отримана шляхом проведення розрахунку).

Організація збору і передачі інформації

Введення інформації в базу даних системи повинно здійснюватися за допомогою програмного забезпечення системи.

Наступна інформації повинна бути введена в систему до початку її експлуатації:

- Інформація про водіїв, що працюють на підприємстві, про транспортні засоби. За збір цієї інформації відповідає адміністратор системи.

- Інформація про аутентифікацію. Адміністратор повинен ввести інформацію про логіни та паролі. За збір і збереження цієї інформації відповідає адміністратор системи.

Інша інформація вводиться бухгалтером та директором протягом всього часу роботи системи .

2.3.2 Концептуальна модель даних

З аналізу функціональної моделі системи (дивись пункт 2.1.2) можна побудувати концептуальну модель даних системи.

Найбільш розповсюдженим засобом представлення концептуальної моделі даних є діаграми "сутність-зв'язок" (ERD). З їхньою допомогою визначаються важливі для предметної області об'єкти (сутності), їхні властивості (атрибути) і відносини один з одним (зв'язки). ERD безпосередньо використовуються для проектування реляційних баз даних.

Для побудови концептуальної моделі даних системи необхідно виділити сутності і визначити їх атрибути. Діаграма представлена в нотації Чена (рисунок 2.7).

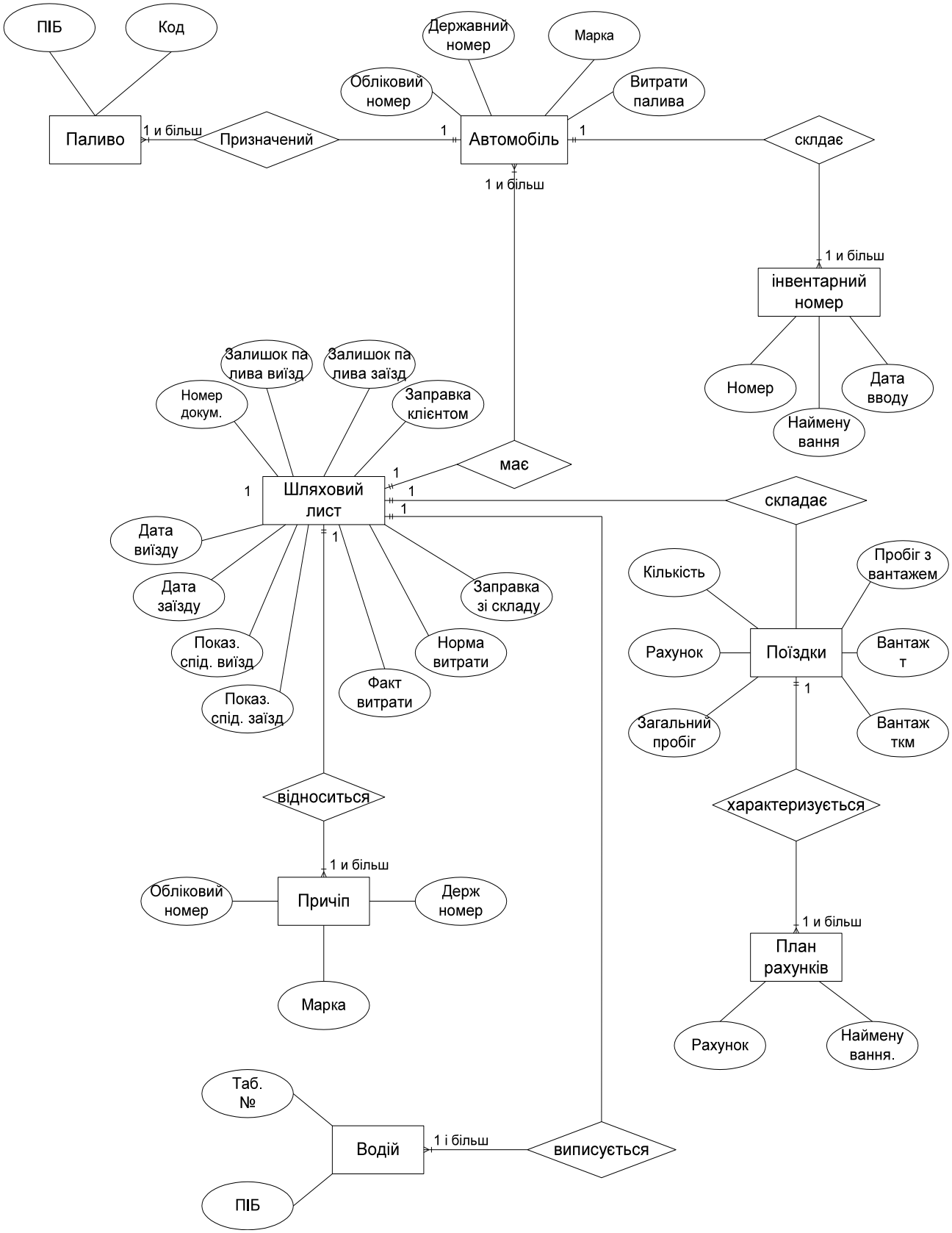


Рисунок 2.7 — Діаграма «Сутність – зв’язок» системи обліку паливних матеріалів на підприємстві «Південь-Транс»

Згідно до діаграми кожен АВТОМОБІЛЬ може мати декілька ШЛЯХОВИХ ЛИСТІВ, і кожен АВТОМОБІЛЬ призначений для одного виду ПАЛИВА і має один ІНВЕНТАРНИЙ НОМЕР.

За ШЛЯХОВИМ ЛИСТОМ отримують ПАЛИВО. До ШЛЯХОВОГО ЛИСТА також відноситься один ПРИЧІП. Кожен ВОДІЙ може мати декілька ШЛЯХОВИХ ЛИСТІВ.

ШЛЯХОВИЙ ЛИСТ складає ПОЇЗДКИ, які характеризуються ПЛАНОМ РАХУНКІВ.

2.3.3 Логічна структура бази даних

Виходячи з аналізу концептуальної моделі даних системи, розробимо логічну структуру бази даних. Логічна структура бази даних системи представлена на рисунку 2.8.

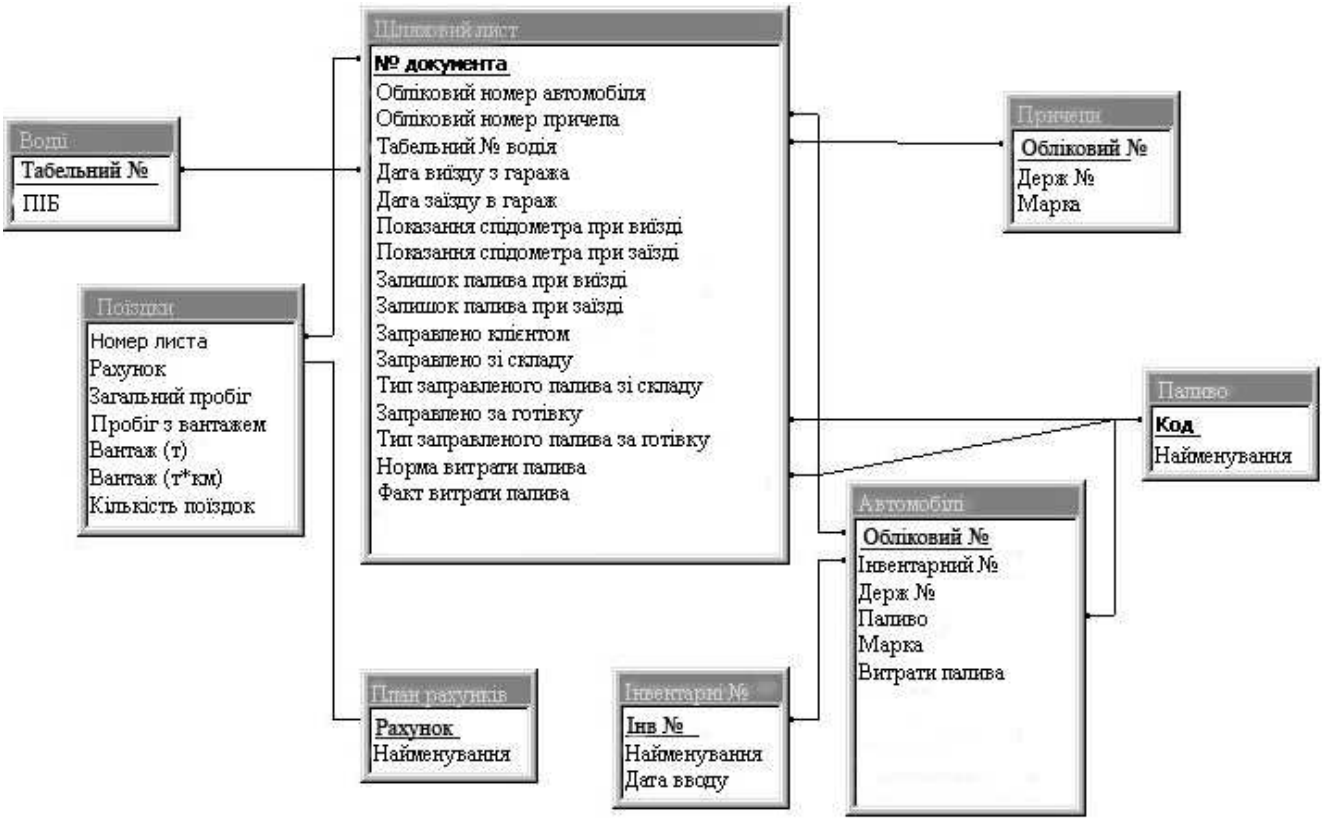


Рисунок 2.8 – Логічна структура бази даних системи

2.3.4 Фізична структура бази даних

На підставі логічної структури, проаналізувавши відносини між таблицями та зміст кожної з них можна створити остаточну структуру кожної таблиці.

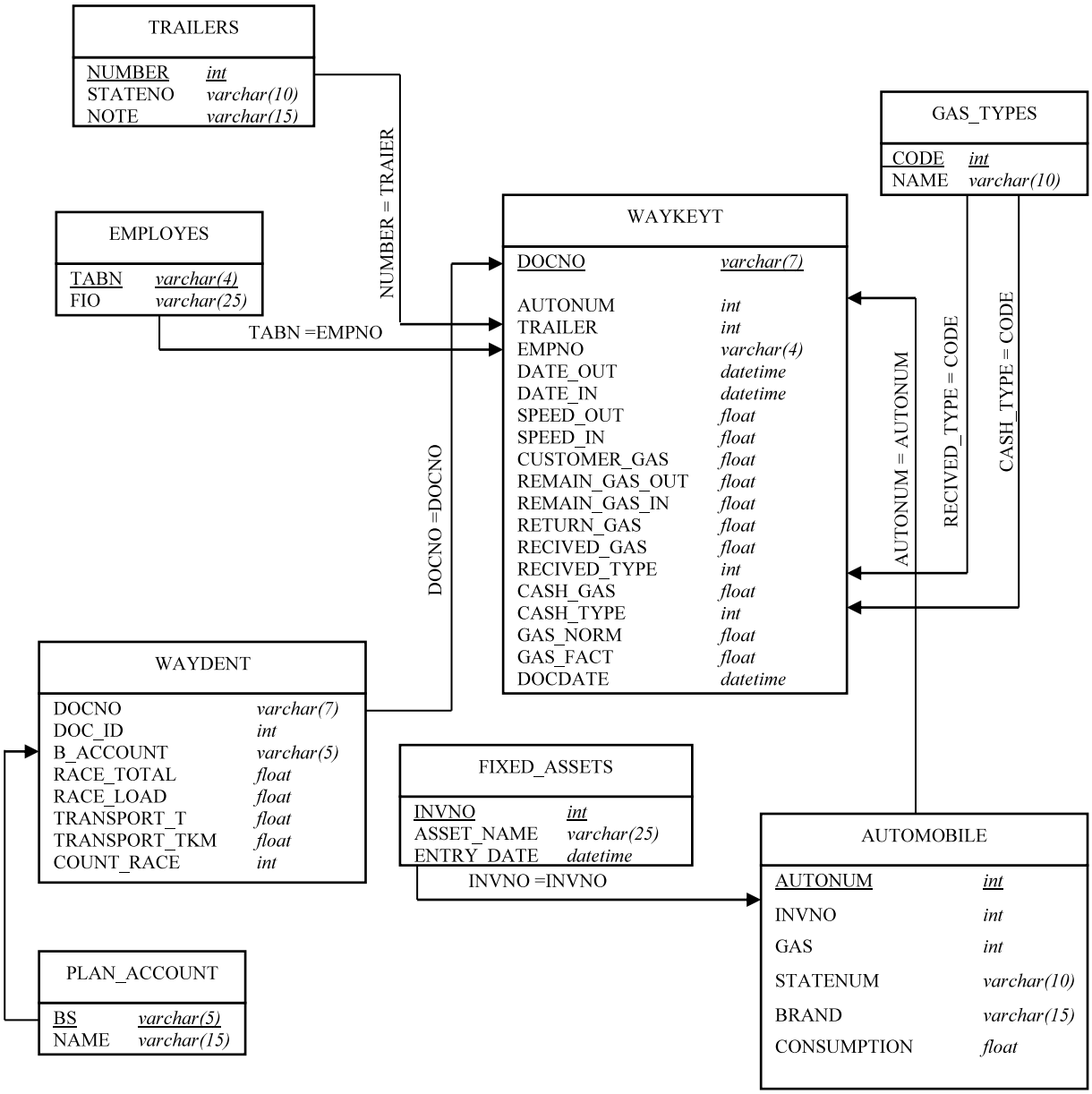


Рисунок 2.9 – Фізична модель даних

На рисунку 2.9 зображена фізична модель даних. Більш докладно поля таблиць і їх призначення описані в таблиці 2.1.

Таблиця 2.1 - опис таблиць бази даних

Назва таблиці				
I'мя поля	**Тип даних**	**Довжина**	**Ключ**	**Примітка**
WAYKEYT				
DocNo	Текстове	7	*	Номер документу
AutoNum	Ціле	4		Обліковий номер автомобілю
Trailer	Ціле	4		Обліковий номер причепа
EmpNo	Строкове	4		Табельний номер водія
Date_Out	Дата/Час	8		Дата виїзду
Date_In	Дата/ Час	8		Дата заїзду
Speed_Out	Дійсне	8		Показання спідометра при виїзді
Speed_In	Дійсне	8		Показання спідометру при заїзді
Customer_Gas	Дійсне	8		Кількість пального, заправленого клієнтом
Remain_Gas_Out	Дійсне	8		Залишок пального при виїзді
Remain_Gas_In	Дійсне	8		Залишок пального при заїзді
Return_Gas	Дійсне	8		Повернення пального
Recived_Gas	Дійсне	8		Кількість пального, заправленого зі складу

Продовження таблиці 2.1

1	2	3	4	5
Recived_Type	Ціле	4		Тип пального заправленого зі складу
Cash_Gas	Дійсне	8		Кількість пального заправленого за готівку
Cash_Type	Ціле	4		Тип пального заправленого за готівку
Gas_Norm	Дійсне	8		Норма витрат палива
Gas_Fact	Дійсне	8		Фактична витрата палива
DocDate	Дата/ Час	8		Дата виписки листа
WAYDENT				
DocNo	Текстовий	7		Номер документу
B_Account	Текстовий	5		Балансовий рахунок
Race_Total	Дійсне	8		Загальний пробіг
Race_Load	Дійсне	8		Пробіг з вантажем
Transport_T	Дійсне	8		Кількість перевезеного вантажу
Transport_TKM	Дійсне	8		Кількість тоно-кілометрів
TRAILERS				
Number	Ціле	4	*	Обліковий номер
StateNo	Текстовий	15		Держ номер
Note	Текстовий	25		Примітка

Продовження таблиці 2.1

1	2	3	4	5
AUTOMMOBILE				
AutoNum	Ціле	4	*	Обліковий номер
InvNo	Ціле	4		Інвентарний номер
Gas	Ціле	4		Тип палива
State_Num	Текстовий	10		Держ номер
Brand	Текстовий	15		Марка
Consumption	Дійсне	8		Витрати палива на 100 км
GAS_TYPES				
CODE	Ціле	4	*	Код палива
NAME	Текстовий	10		Марка палива
PLAN_ACCOUNT				
BS	Текстовий	5	*	Балансовий рахунок
NAME	Текстовий	25		Найменування рахунку
EMPLOYES				
TabN	Ціле	4	*	Табельний номер водія
FIO	Текстовий	25		ПІБ водія
FIXED_ASSETS				
InvNo	Ціле	4	*	Інв. номер
Asset_Name	Текстовий	25		Наименование

2.3.5 Оцінка розміру бази даних

Оцінювання обсягу бази даних є необхідним етапом для обґрунтованого вибору технічного забезпечення сервера баз даних. Зазначена оцінка має наближений характер.

Враховуючи, що на підприємстві експлуатується близько 50 транспортних засобів і працює приблизно 50 водіїв, а більшість таблиць містять записи, пов’язані з кожним водієм і транспортним засобом, початковий обсяг бази даних становитиме орієнтовно 1 МБ. Надалі її розмір щомісяця збільшуватиметься. Такий обсяг є порівняно невеликим для бази даних із подібною кількістю записів.

Водночас слід ураховувати, що фактичний розмір бази даних на жорсткому диску буде у кілька разів більшим. Це зумовлено тим, що система керування базами даних додатково зберігає індекси, тимчасові таблиці та іншу службову інформацію, необхідну для забезпечення ефективної роботи СУБД.

2.4 Рішення з технічного забезпечення

Загальні положення

Програмне забезпечення інформаційної системи обліку паливних матеріалів має розподілену архітектуру, що передбачає використання кількох серверів та клієнтських робочих станцій. Такий підхід забезпечує підвищену продуктивність, надійність та можливість масштабування системи.

Для досягнення оптимальних експлуатаційних характеристик доцільно використовувати три окремі сервери (хоча за мінімальної конфігурації вся система може бути розгорнута й на одному сервері):

- Web-сервер, що забезпечує обробку запитів користувачів та функціонування інтерфейсу;
- Сервер доступу до даних / сервер прикладної логіки, який виконує бізнес-процеси та забезпечує взаємодію між Web-інтерфейсом і базою даних;
- Сервер баз даних, на якому розміщується реляційна СУБД та зберігаються всі дані системи.

Така структурна організація дозволяє розподілити навантаження між різними рівнями системи та забезпечити її ефективне функціонування в умовах зростання обсягів даних і кількості користувачів.

Засоби обчислювальної техніки

Вимоги до технічного забезпечення Web-сервера:

- Мікропроцесор класу Intel core i5 і вище.
- Оперативна пам'ять RAM 16 Гб і більше.
- Жорсткий диск з вільним місцем 150 Гб і більше.
- Мережна карта Ethernet (не менше 100 Мбіт/с).

Вимоги до технічного забезпечення сервера доступу до даних:

- Мікропроцесор класу Intel Core i5 і вище.
- Оперативна пам'ять RAM 16 Гб і більше.
- Жорсткий диск з вільним місцем 100 Гб і більше.
- Мережна карта Ethernet (не менше 100 Мбіт/с).

Вимоги до технічного забезпечення сервера баз даних:

- Мікропроцесор класу Intel Core i5 і вище.
- Оперативна пам'ять RAM 8192 Мб і більше.
- Жорсткий диск з вільним місцем 100 Гб і більше.
- Мережна карта Ethernet (не менше 100 Мбіт/с).

Вимоги до технічного забезпечення клієнтської машини:

- Мікропроцесор класу Intel Core i3 і вище.
- Оперативна пам'ять RAM 8192 Мб і більше.
- Жорсткий диск з вільним місцем 50 Гб і більше.
- Мережна карта Ethernet (не менше 100 Мбіт/с).

У якості мережного середовища можна використовувати стандартний кабель «кручена пара» категорії 5е і вище. Мережне обладнання повинне підтримувати швидкість не менше 100 Мбіт/сек.

Апаратура передачі даних

Підприємство «Південь-Транс» має приміщення, оснащені персональними комп’ютерами, які можуть бути використані як робочі станції для функціонування розроблюваної інформаційної системи, зокрема:

- робочий кабінет;
- кабінет директора.

Сервери та робочі станції повинні бути об’єднані в єдину локальну комп’ютерну мережу. Пропускна здатність мережі має становити не менше 100 Мбіт/с. На

серверному сегменті для забезпечення підвищеної продуктивності доцільно використовувати мережне обладнання, що підтримує швидкість 1000 Мбіт/с (мережеві адаптери, комутатори, маршрутизатори, кабельна інфраструктура). Як фізичне середовище передавання даних може застосовуватися неекранована кручена пара категорії 5 (100Base-T) або 5e (1000Base-TX).

На рисунку 2.10 наведено схему комп'ютерної мережі підприємства «Південь-Транс». Діаметр мережі не перевищує 120 метрів, а максимальна довжина окремого сегмента – 60 метрів. Використання комутаторів дозволяє розділити мережу на окремі домени колізій, що забезпечує коректне та стабільне функціонування мережевої інфраструктури.

На даний момент підприємство володіє всім необхідним обладнанням для розгортання зазначеної мережі. За потреби можлива реалізація альтернативної конфігурації з підвищеною пропускною здатністю.

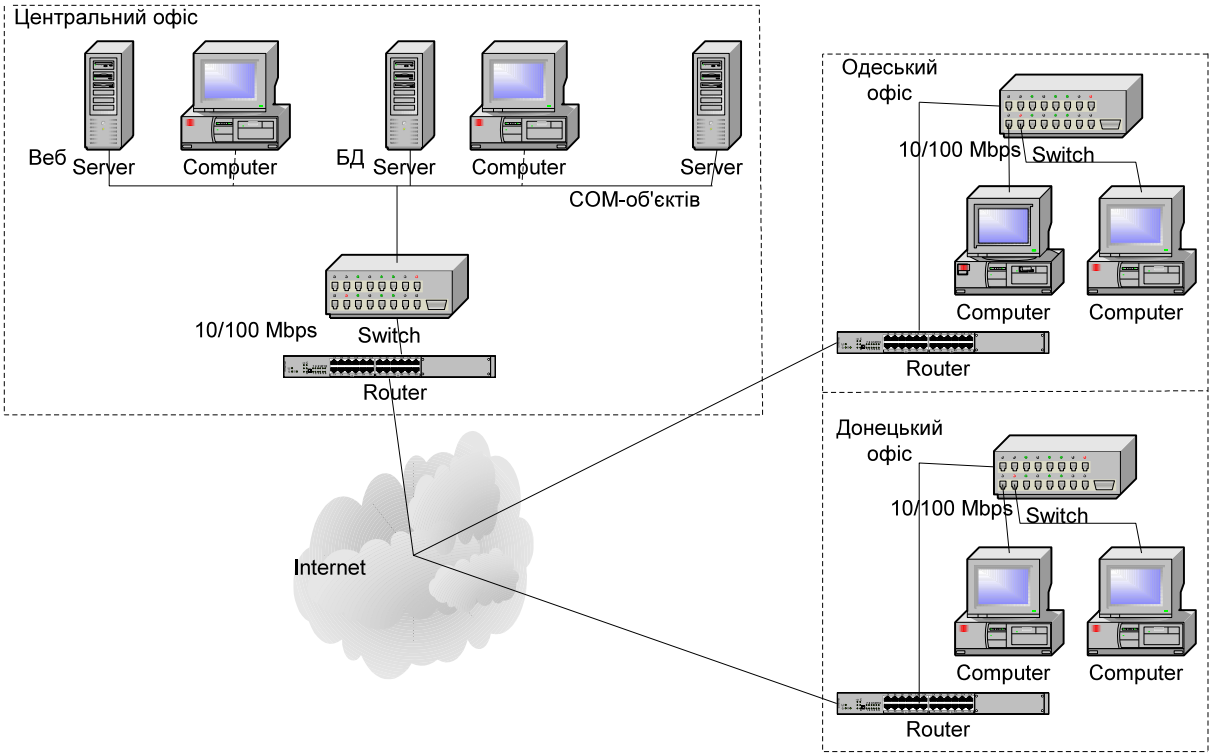


Рисунок 2.10 - Схема комп'ютерної мережі підприємства «Південь-Транс»

На підприємстві вже встановлено мережу, вона працює коректно, відповідає усім вимогам. Також дана мережа може доповнюватись, якщо підприємство буде розширюватись.

2.5 Рішення з програмного забезпечення

Призначенням інформаційної системи є автоматизація функцій, що виконуються бухгалтером, а також формування єдиної бази даних, у якій зберігатиметься повний обсяг інформації про шляхові листи, водіїв, транспортні засоби, види палива, поїздки та інші пов'язані дані. У перспективі ця база даних може бути використана також для розв'язання адміністративних завдань.

Метою створення системи є підвищення продуктивності праці, покращення умов роботи персоналу, прискорення виконання розрахункових операцій, скорочення часу обробки інформації та мінімізація кількості помилок, що виникають під час ведення обліку.

Інформаційне забезпечення системи включає:

- Базу даних, у якій зберігається вся інформація про діяльність підприємства;
- Файли аутентифікації, що містять логіни та паролі користувачів і доступні виключно адміністратору системи.

Програмне забезпечення інформаційної системи обліку паливних матеріалів має розподілену архітектуру, що передбачає використання кількох серверів і клієнтських робочих станцій. Для забезпечення максимальної продуктивності та можливості масштабування доцільно використовувати три сервери (хоча за мінімальної конфігурації система може функціонувати і на одному сервері):

- Web-сервер;
- Сервер доступу до даних / сервер COM-об'єктів;
- Сервер баз даних.

Програмне забезпечення системи складається з кількох пакетів:

- пакета програм, що забезпечує функціональність, необхідну бухгалтеру та директору;
- пакета програм для адміністратора системи.

Кожен пакет містить набір модулів, які реалізують окремі функції системи. У даній роботі розглядається розробка програмного пакета, призначеного для бухгалтера та директора.

Бізнес-процеси реалізуються за допомогою CGI-сценаріїв, що виконуються на Web-сервері.

Доступ до бази даних здійснюється через COM-об’єкти, які надаються COM-серверами, із використанням технології OLE DB. Відповідні COM-компоненти встановлюються на сервер доступу до даних.

База даних є реляційною та функціонує під управлінням СУБД, що підтримує стандарт SQL-92 і має драйвери OLE DB та/або ODBC.

Операційна система, встановлена на Web-сервері та сервері доступу до даних, повинна підтримувати технології DCOM/COM+ (рекомендована ОС — MS Windows Server 2008). Сервер баз даних не має додаткових обмежень, окрім тих, що визначаються обраною СУБД.

Web-сервер повинен мати встановлений HTTP-сервер із підтримкою протоколу HTTP/1.1, виконання CGI-сценаріїв та механізмів аутентифікації користувачів.

2.5.1 Структура програмного забезпечення

На підставі аналізу функціональної моделі інформаційної системи можливо сформувати структуру програмного забезпечення. У даній роботі розглядається розроблення програмного забезпечення, призначеного для використання бухгалтером та директором. З огляду на те, що програмне забезпечення має розподілену архітектуру та функціонує на кількох обчислювальних вузлах, його структура повинна бути модульною, що забезпечує гнучкість, масштабованість і можливість подальшого розширення.

Вихідні тексти програмних модулів подано у додатку Г.

На рисунку 2.11 наведено схему програмних модулів системи.

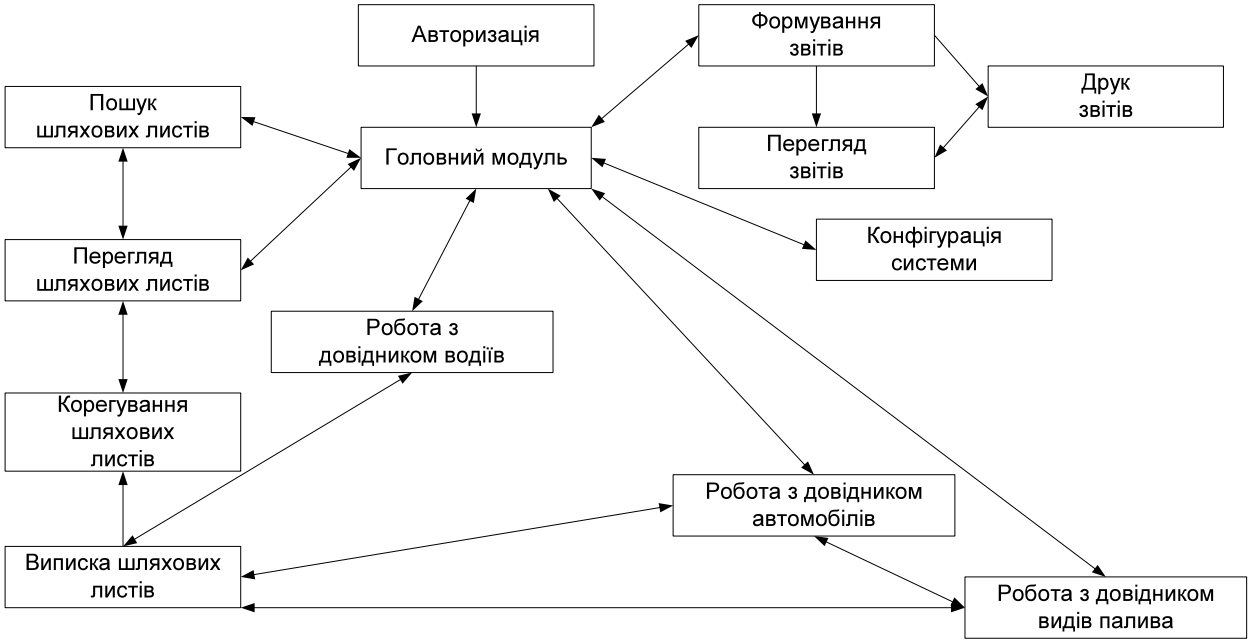


Рисунок 2.10 – Схема програмних модулів системи

Модуль авторизації забезпечує введення користувачем облікових даних (логіна та пароля) з метою отримання доступу до системи.

- Модуль формування звітів призначений для створення звітів на основі заданих ключових параметрів.
- Модуль перегляду звітів забезпечує відображення сформованих звітів на екрані монітора.
- Модуль друку звітів надає можливість виведення сформованих звітів на друк.
- Модуль пошуку шляхових листів дозволяє здійснювати пошук необхідних шляхових листів за визначеними критеріями, зокрема за періодом їх виписки, номером автомобіля або табельним номером водія.
- Модуль перегляду шляхових листів забезпечує доступ до перегляду змісту вже існуючих шляхових листів.
- Модуль коригування шляхових листів призначений для внесення змін до наявних шляхових листів.
- Модуль виписки шляхових листів забезпечує створення нових шляхових листів та їх збереження у відповідному сховищі даних.
- Модуль налаштування конфігурації системи використовується для первинного налаштування або зміни загальних параметрів системи та постійних величин; викликається з головного модуля програми.

- Модуль роботи з довідником водіїв реалізує функції пошуку, додавання, редагування та видалення записів про водіїв.
- Модуль роботи з довідником автомобілів забезпечує пошук, додавання, редагування та видалення інформації про транспортні засоби.
- Модуль роботи з довідником причепів надає можливість додавання, редагування та видалення записів про причепи.
- Модуль роботи з довідником видів пального реалізує функції додавання, редагування та видалення записів про види пального.

2.5.2 Функції частин програмного забезпечення

За розробленими в технічному проекті мініспеціфікаціям процесів були закодовані програмні модулі. Тексти основних модулів розміщені в додатку Е.

2.5.4 Операційна система

Під час розроблення концепції інформаційної системи обліку паливних матеріалів для підприємства «Південь-Транс» як серверну операційну систему було обрано MS Windows 11 Pro. Усе програмне забезпечення, описане в попередніх розділах, функціонуватиме під керуванням цієї операційної системи.

MS Windows 11 Pro є багатозадачною мережевою операційною системою корпоративного рівня. Вона характеризується високою масштабованістю, продуктивністю, стабільністю та розвиненими механізмами безпеки, а також підтримує широкий спектр прикладного програмного забезпечення. Основні технічні характеристики операційної системи наведено в таблиці. 2.2.

Таблиця 2.2 – Характеристики MS Windows 10 Professional

Характеристика	Значення
Процесор	Core i5 і вище
Максимальна ємкість пам'яті, Гбайт	128
Мінімальна ємкість пам'яті, Мбайт	2048
Ємкість пам'яті, що рекомендується,	8 Гб – 32 Гб
Жорсткий диск	80 Гбайт і більше
Сумісність з WSD	Немає
Кластерна служба	Немає
Сумісність з NLB	Немає
Управління об'єктом завдання	API
Joint Support Queue	Немає
Сертифікація програм	Сертифікат Windows 11
Продавець	Microsoft
Апаратна сумісність	Стандартний Windows 11 HCL

Висновок

У розділі сформовано комплекс загальносистемних, організаційних, інформаційних, технічних та програмних рішень, що визначають структуру, функціонування та умови експлуатації проєктованої системи. Встановлено організаційну та функціональну структури, наведено узагальнений опис системи та визначено механізми її захисту від несанкціонованого доступу.

Розроблено інформаційне забезпечення, яке включає концептуальну, логічну та фізичну моделі даних, а також оцінку обсягу бази даних. Визначено вимоги до технічного забезпечення та обґрунтовано його склад. Сформовано структуру програмного забезпечення, визначено функції його компонентів та обрано операційну систему для забезпечення коректної роботи системи.

Отримані результати забезпечують цілісність проєктних рішень і створюють необхідну основу для подальшої реалізації та впровадження системи.

3 Розробка робочої документації

Вступ

Робоча документація повинна складатись із певного набора документів, перелік яких регламентується галузевими стандартами та корпоративними правилами написання документів.

3.1 Документи які розроблено

У технічному завданні (постановці задачі) на розробку системи визначені вимоги до документації (дивиться п. 1.5.9). Відповідно до цих вимог були розроблені наступні документи:

- Загальний опис системи (Додаток А).
- Керівництво користувача (Додаток Б).
- Інструкція з формування і ведення бази даних (Додаток В).
- Текст програми (Додаток Г).

Висновок

Для супроводу проєкта було створенокомплект документів, що відповідають галузевим стандартам

4. РОЗРАХУНОК ЕКОНОМІЧНОГО ЕФЕКТУ ВІД ВПРОВАДЖЕННЯ ІС

4.1 Вступ до проблеми

Інформаційні системи широко використовуються багатьма організаціями України різної форми власності. Ці системи розповсюджуються все більше і більше, змінюючись відповідно до вимог замовників та модернізуючись з плином часу.

Системи управління контентом в цілому не є принципово новою технологією, однак як парадигма створення інформаційних систем та надання послуг у сфері ІТ вони знаходяться лише на початку свого шляху. Безперечно, з розвитком цієї технології все більше і більше компаній використовуватимуть системи управління контентом для створення власної комп'ютерної інфраструктури, вирішення обчислювальних задач та ін.

Створення інформаційної системи вимагає одноразових витрат на її розробку та придбання необхідних технічних засобів, а також поточних витрат, пов'язаних з функціонуванням підсистеми. Економічний ефект від експлуатації системи визначається з урахуванням витрат на її експлуатацію. Відношення економічного ефекту до витрат на створення системи характеризує економічну ефективність капітальних вкладень. Економічні показники визначаються за діючими на момент розрахунку цінами, тарифами та ставками заробітної плати.

4.2 Розрахунок вартості створення інформаційної системи

Загальні витрати (на проектування, реалізацію та впровадження системи)

$$З = С_{п\text{-}р} + С_{тз} + С_{пз} + С_{впр},$$

де $С_{п\text{-}р}$ – витрати на проектування і реалізацію;

$С_{тз}$ – витрати на придбання або модернізацію обчислювальної техніки, периферійних пристроїв, засобів зв'язку, допоміжного устаткування та оргтехніки (з урахуванням витрат на транспортування, монтаж, налагодження і запуск), виробничо-господарського інвентарю;

$С_{пз}$ – витрати на придбання та інсталяцію програмного забезпечення (ПЗ), необхідного для розробки;

$С_{впр}$ – витрати на впровадження системи (навчання персоналу тощо).

Витрати на проектування та реалізацію розраховуються за формулою

$$С_{п\text{-}р} = С_з + С_{від} + С_м + С_н,$$

де $С_з$ – заробітна плата розробників;

$С_{від}$ – відрахування на соціальні заходи (відповідно до чинного законодавства (становить 47,5% від фонду оплати праці);

$С_м$ - експлуатація та амортизація, $С_м = З_{експл} + З_{амор}$;

$С_н$ – накладні витрати у розмірі 50-150% від витрат на оплату праці (охоплюють оплату праці управлінського персоналу з нарахуваннями; оплату службових відряджень; оплату консультаційно-інформаційних послуг; оплату ремонту і техобслуговування основних фондів, крім персональних комп'ютерів (ПК)).

1) Розробка підсистеми вимагає 3 місяці (експертна оцінка, яка базується на досвіді розробки систем такого класу). Заробітна платня розробників за місяць дорівнює 12000 грн (відповідно до сучасних розцінок та попиту на спеціалістів такого профілю). Кількість розробників – одна людина.

$$С_з = 1 * 12000 * 3 = 36000 \text{ грн.}$$

$$С_{від} = 36000 * 0{,}475 = 17100 \text{ грн.}$$

2) Витрати на амортизацію ЕОМ, на яких виконується розробка:

$$З_{амор} = (Ц_{еом}/Т_{сл}) * (Т_{разр}/12),$$

де $Ц_{еом} = 20000$ грн. – балансова вартість ЕОМ;

$Т_{сл} = 5$ років – термін служби ЕОМ;

$Т_{разр} = 0{,}5$ року – час, необхідний для розробки системи, виражений в місяцях, і тому ділиться на 12, щоб одержати число років.

$$З_{амор} = (20000/5) * 0{,}5 = 2000 \text{ (грн.);}$$

3) витрати на експлуатацію ЕОМ, на якій виконується розробка, полягають в оплаті споживаної нею електричної енергії:

$$З_{експл} = Р_{ЕОМ} * Т_{разр} * N_{рд} * N_{рч} * Е_{ст},$$

де $Р_{ЕОМ} = 0{,}4$ кВт/год – потужність ЕОМ;

$Т_{разр} = 3$ місяці – тривалість розробки;

$N_{рд} = 21$ день – число робочих днів у місяці;

$N_{рч} = 8$ годин – число годин у робочому дні;

$Е_{ст} = 0.9$ грн – вартість 1 кВт електроенергії

$$З_{експл} = 0{,}4 * 3 * 21 * 8 * 0{,}9 = 181{,}44 \text{ (грн.)}$$

$$С_м = 2000 + 181,44 = 2181,44 \text{ (грн.)}$$

$$С_н = 36000 * 0,5 = 18000 \text{ (грн.)}$$

$$С_{п\text{-}р} = 36000 + 17100 + 2181,44 + 18000 = 73281,44 \text{ (грн.)}$$

Витрати на програмне забезпечення не включаються до даного кошторису, оскільки у функціонуванні системи передбачається застосування безкоштовного програмного забезпечення з відкритим вихідним кодом. Витрати на апаратне забезпечення становлять 18594 грн, що охоплює придбання серверного обладнання, комутаційних пристроїв та необхідних матеріалів. До складу витрат, пов'язаних із впровадженням системи, належать також витрати на навчання персоналу, монтаж і налаштування мережі, придбання устаткування, а також витрати на програмне забезпечення.

Витрати на впровадження системи приведені в таблиці 4.1.

Таблиця 4.1 – Витрати на впровадження системи.

Найменування інвестицій	Сума, грн
Вартість устаткування	18594,00
Вартість програмного забезпечення	-
Вартість підготовки кадрів	500,00
Монтаж і настроювання мереж	300,00
Разом на впровадження системи	**19394,00**

Таким чином, загальні витрати на проектування, реалізацію та впровадження системи складають:

$$З = С_{п\text{-}р} + С_{тз} + С_{пз} + С_{впр} = 73281,44 + 19394 = 92675,44 \text{ (грн).}$$

Витрати на створення та експлуатацію системи в перший рік експлуатації

$$З_{се} = З + З_{ер},$$

де $З_{ер}$ – експлуатаційні витрати на технічне забезпечення за рік.

$$З_{ер} = А_{ОБ} + З_м + З_е,$$

де $А_{ОБ}$ – амортизаційні відрахування на устаткування;

$З_м$ – витрати на основні і допоміжні матеріали;

$З_е$ – витрати на електроенергію.

Амортизаційні відрахування на устаткування складають 60% балансової вартості в рік (за чинним законодавством)

$$А_{ОБ} = 19394 * 0,6 = 11636,4 \text{ (грн.)}$$

В масштабі підприємства річні витрати на основні та допоміжні матеріали (флеш-накопичувачі, папір тощо) визначаються в розмірі 3% вартості основного устаткування

$$З_М = 19394 * 0,03 = 581,82 \text{ (грн.)}$$

Річний обсяг робіт ПК у нормо-годинах

$$Ф_М = 253,3 * Т_С,$$

де $Т_С$ – середнє місячне завантаження устаткування (близько 5 годин); 253,3 – середня кількість робочих днів у році.

$$Ф_М = 253,3 * 5 = 1266,5 \text{ (годин.)}$$

Витрати на електроенергію під час 1266,5 годин роботи устаткування в рік при потужності ЕОМ 0,4 кВт/год та ціні 9 грн за 1 кВт електроенергії складуть

$$З_е = 1266,5 * 9 * 0,4 = 4559,4 \text{ (грн.)}$$

Експлуатаційні витрати на технічне забезпечення за рік

$$З_{ер} = 11636,4 + 581,82 + 4559,4 = 12674,16 \text{ (грн.)}$$

Отже, у перший рік витрати на створення та експлуатацію системи складуть

$$З_{се} = 92675,44 + 12674,16 = 105349,6 \text{ (грн.)}$$

4.3 Розрахунок економічної ефективності

Економічна ефективність використання інформаційної системи полягає у наступному:

- зменшення часу на обслуговування системи за рахунок її відмовостійкості та автоматичного масштабування;
- можливість надання користувачам інформації в оренду.

Це зменшення відбудеться за рахунок зменшення складності користування системою, збільшення надійності, стабільності та рівня безпеки. Визначимо, що це зменшення складе 50% від сучасного.

Однак, не піддається прямій грошовій оцінці зменшення кількості помилкових та необережних рішень, підвищення оперативності керування, поліпшення організації

роботи та своєчасне отримання необхідної інформації про пріоритетність варіантів рішень тощо.

Використання даної системи дозволяє вивільнити 0,1 робочого часу. Так як з системою працює біля 50 користувачів, то система умовно вивільнить 5 робітників.

Визначимо пряму економічну ефективність, ґрунтуючись на тому, що впровадження системи вивільнить 5 робітників. Річна платня за рахунок вивільнення робочого часу складе

$$\Delta C_n = 5000 * 5 * 12 - 12674,16 = 287325,84 \text{ (грн.)}$$

Річний економічний ефект розраховується за наступною формулою:

$$E_{год} = \Delta C_n - E_n * k,$$

де *En* – нормативний коефіцієнт економічної ефективності капітальних вкладень у галузь і для обчислювальної техніки приймається 0,5;

k – додаткові капітальні вкладення з урахуванням витрат на проектування, створення і функціонування системи.

$$E_{год} = 287325,84 – 0,5 * 105349,6 = 234651,04 \text{ (грн.)}$$

Строк окупності підсистеми

$$T = k / \Delta C_n$$

Тобто

$$T = 105349,6 / 287325,84 = 0,37 \text{ (року)}$$

Таким чином, витрати на створення розподіленої інформаціонної системи окупляться протягом 0,37 року.

Висновки до розділу 4

В результаті розрахунків витрати на створення системи обліку робочого часу (проектування, реалізація і впровадження) складуть 92675,44 грн; експлуатаційні витрати щорічно складатимуть 12674,16 грн. Загалом в перший рік витрати на модернізацію й експлуатацію складуть 105349,6 грн.

Вивільнюється 5 умовних фахівці.

Вивільнена річна зарплатня працівників складає 287325,84 грн.

Витрати на створення системи окупляться протягом 0,37 року.

За отриманими значеннями річного економічного ефекту та строку окупності можливо зробити висновок про економічну вигідність і обґрунтованість створення інформаційної системи. Слід зазначити, що підвищити економічну ефективність цієї системи можна за рахунок запропонування стороннім користувачам систему як продукт. Таким чином можна буде скоротити строк, за який окупляться витрати на створення системи, а також здобути прибуток у вигляді продажу продукту.

5. ОХОРОНА ПРАЦІ

Вступ

Охорона праці є невід'ємною складовою організації трудової діяльності в будь-якій сфері. Особливої актуальності питання забезпечення безпечних і здорових умов праці набувають у галузі інформаційних технологій, де фахівці значну частину робочого часу здійснюють професійну діяльність за персональними комп'ютерами. Розроблення інформаційних систем, як один із ключових напрямів діяльності ІТ-спеціалістів, передбачає тривалу роботу з комп'ютерним обладнанням, моніторами та іншими технічними засобами.

Порушення вимог охорони праці під час роботи з комп'ютерною технікою може спричинити розвиток низки професійно зумовлених захворювань, зокрема синдрому зап'ясткового каналу, комп'ютерного зорового синдрому, захворювань опорно-рухового апарату тощо. Крім того, неналежна організація робочого місця, недостатній рівень освітлення, невідповідні параметри мікроклімату у виробничих приміщеннях здатні негативно позначатися на самопочутті та працездатності розробників ІС.

У зв'язку з цим дотримання вимог охорони праці та створення належних умов для збереження здоров'я і безпеки фахівців, задіяних у процесі розроблення інформаційних систем, є одним із пріоритетних завдань керівників підприємств та організацій. Забезпечення безпечних умов праці не лише сприяє поліпшенню фізичного та психологічного стану працівників і зниженню ризиків виникнення професійних захворювань, але й позитивно впливає на продуктивність праці та якість кінцевого програмного продукту.

У цьому розділі дипломної роботи, присвяченої розробленню інформаційної системи підтримки волонтерської діяльності, буде розглянуто основні аспекти охорони праці в галузі інформаційних технологій, зокрема у контексті проєктування та розроблення ІС. Особливу увагу буде приділено аналізу нормативно-правових актів, що регламентують питання охорони праці, вимогам до організації робочого місця розробника, забезпеченню безпечних умов праці та мінімізації впливу шкідливих виробничих факторів, регламентації режиму праці та відпочинку, а також специфіці застосування заходів електробезпеки.

5.1. Нормативно-правові акти, що регулюють питання охорони праці при роботі з персональними комп'ютерами та іншими пристроями для розробки вебзастосунків

Питання охорони праці під час роботи з персональними комп’ютерами (ПК) та іншими пристроями, що застосовуються для розробки інформаційних систем, регулюються низкою нормативно-правових актів України. Основним документом, що визначає вимоги безпеки й захисту здоров’я працівників при роботі з екранними пристроями, є Наказ Міністерства соціальної політики України «Про затвердження Вимог щодо безпеки та захисту здоров’я працівників під час роботи з екранними пристроями» від 14.02.2018 № 207. Дія цього наказу поширюється на всіх суб’єктів господарювання незалежно від форми власності та організаційно-правової структури, які здійснюють діяльність, пов’язану із застосуванням екранних пристроїв.

Відповідно до зазначеного документа, роботодавець зобов’язаний забезпечити відповідність робочих місць працівників ергономічним, організаційно-технічним і санітарно-гігієнічним вимогам. Зокрема, роботодавець має інформувати працівників про умови праці та наявність небезпечних або шкідливих виробничих чинників, які залишаються невирішеними, а також забезпечити проведення інструктажів з питань охорони праці та пожежної безпеки під час роботи з екранними пристроями.

Окрім згаданого нормативного акта, питання охорони праці при роботі з ПК регламентуються й іншими нормативними документами, наприклад, ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», який містить санітарно-гігієнічні вимоги до умов праці, організації робочих місць з ВДТ ЕОМ, параметрів виробничого середовища, освітлення, рівнів шуму тощо.

Електробезпека при експлуатації ПК і периферійних пристроїв забезпечується виконанням положень Правил улаштування електроустановок (ПУЕ), Правил технічної експлуатації електроустановок споживачів (ПТЕЕС), Правил безпечної експлуатації електроустановок споживачів (ПБЕЕС). Згідно з пунктом 1.7.110 ПУЕ, конструкція та розміщення комп’ютерного обладнання повинні відповідати встановленим вимогам електробезпеки і забезпечувати зручність його технічного обслуговування.

Строге дотримання вимог НПАОП 0.00-7.15-18 «Вимоги щодо безпеки та захисту здоров'я працівників під час роботи з екранними пристроями», НПАОП 40.1-1.21-98 «Правила безпечної експлуатації електроустановок споживачів», а також інших державних стандартів і санітарних норм, є обов'язковим для всіх роботодавців та працівників, що використовують ПК і суміжне обладнання для розробки вебзастосунків. Лише системний підхід і суворе виконання положень нормативно-правових актів щодо охорони праці дають змогу створити безпечні та комфортні умови для ІТ-фахівців, істотно знизити ризики виникнення професійних захворювань і травматизму.

5.2. Вимоги до організації робочого місця розробника інформаційної системи

Організація робочого місця розробника вебзастосунків відіграє ключову роль у забезпеченні безпечних та комфортних умов праці, а також у підвищенні продуктивності та ефективності роботи спеціаліста. При облаштуванні робочого простору необхідно враховувати комплекс ергономічних, санітарно-гігієнічних та організаційно-технічних вимог, що регламентуються відповідними нормативно-правовими документами.

Одним із основних нормативних документів, що визначає вимоги до робочого місця з персональним комп'ютером, є ДСТУ ISO 9241-5:2004 «Ергономічні вимоги до роботи з відеотерміналами в офісі. Частина 5. Вимоги до компонування робочого місця та до робочої пози». Згідно з цим стандартом, робоче місце розробника вебзастосунків має бути достатньо просторим для зміни робочого положення та забезпечувати зручний доступ до всіх необхідних елементів обладнання.

Робочий стіл повинен мати площу, достатню для розміщення комп'ютера, периферійних пристроїв, документації та інших необхідних матеріалів. Рекомендовані розміри робочої поверхні становлять: ширина – не менше 1200 мм, глибина – не менше 800 мм. Висота робочої поверхні має забезпечувати правильне положення рук користувача на клавіатурі та відповідати антропометричним даним працівника. Оптимальна висота столу варіюється в межах 680–800 мм.

Робоче крісло розробника повинно бути підйомно-поворотним, з регулюванням висоти та кута нахилу спинки, що забезпечує надійну підтримку тіла у зручному положенні. Сидіння та спинка крісла мають бути м'якими, з заокругленими краями та

виготовленими з повітропроникного матеріалу. Для зменшення статичного навантаження на м’язи верхніх кінцівок рекомендується використовувати стаціонарні або регульовані підлокітники.

Монітор слід розташовувати на відстані 600–700 мм від очей користувача, причому верхня точка екрану має знаходитися на рівні очей або трохи нижче. Кут нахилу екрана повинен становити 10–15° при перпендикулярному розташуванні площини екрану до лінії зору користувача. Клавіатуру слід розміщувати на робочій поверхні або на спеціальній полиці на рівні ліктів користувача з кутом нахилу до горизонтальної площини 15–20°.

Для запобігання перенавантаженню зорового аналізатора та розвитку комп’ютерного зорового синдрому необхідно забезпечити належний рівень освітленості робочої зони. Рекомендована освітленість робочого столу становить 300–500 лк, а освітленість екрана монітора не повинна перевищувати 300 лк. Слід уникати прямого світла в очі та відблисків на екрані монітора.

Мікроклімат приміщення, у якому працює розробник вебзастосунків, має відповідати санітарним нормам. Оптимальна температура повітря складає 22–24°C у холодний період року та 23–25°C – у теплий, відносна вологість – 40–60%, швидкість руху повітря – не більше 0,1 м/с. Для підтримки комфортних параметрів мікроклімату рекомендується застосовувати системи кондиціонування або вентиляції.

Дотримання ергономічних та санітарно-гігієнічних вимог до організації робочого місця розробника вебзастосунків створює оптимальні умови для праці, знижує ризик розвитку професійних захворювань опорно-рухового апарату та органів зору, а також підвищує працездатність і ефективність роботи фахівця.

5.3. Забезпечення безпечних умов праці та мінімізація впливу шкідливих факторів при роботі з комп'ютерним обладнанням

Забезпечення безпечних умов праці та мінімізація впливу шкідливих факторів під час роботи з комп’ютерним обладнанням є одним із ключових завдань охорони праці в сфері інформаційних технологій. Незважаючи на те, що професійна діяльність розробника вебзастосунків не належить до категорії робіт з підвищеною небезпекою, вона супроводжується низкою потенційних ризиків для здоров’я працівника.

До основних шкідливих і небезпечних факторів, які можуть негативно впливати на стан здоров’я розробника під час роботи з комп’ютерною технікою, належать:

підвищений рівень електромагнітного випромінювання, статичне електричне поле, неіонізуюче електромагнітне випромінювання оптичного діапазону, підвищений рівень шуму та вібрації, незадовільні параметри мікроклімату, недостатнє або нераціональне освітлення робочої зони, а також психофізіологічні фактори, такі як розумове перенавантаження, перевантаження зорового та опорно-рухового апарату [2].

Для мінімізації впливу зазначених факторів та створення безпечних умов праці необхідно дотримуватися комплексу вимог та рекомендацій. Зокрема, відповідно до ДСанПіН 3.3.2-007-98 «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин», рівень електромагнітного випромінювання на відстані 50 см навколо відеотермінала за електричною складовою не повинен перевищувати 2,5 В/м у діапазоні частот 5 Гц – 2 кГц та 0,25 В/м у діапазоні частот 2 – 400 кГц. Напруженість електростатичного поля на робочому місці з ВДТ не повинна перевищувати гранично допустиме значення – 20 кВ/м [2].

Для зниження впливу електромагнітного випромінювання рекомендується використовувати монітори, що відповідають сучасним стандартам безпеки, зокрема MPR II або TCO. Також слід дотримуватися регламентованих перерв при роботі з комп’ютером, під час яких доцільно виконувати комплекс вправ для очей, шиї, плечей, спини та кистей рук.

Важливим аспектом забезпечення безпечних умов праці є підтримка оптимальних параметрів мікроклімату у виробничому приміщенні. Згідно з ДСанПіН 3.3.2-007-98, на робочих місцях з ВДТ необхідно забезпечувати температуру повітря 22–24°C у холодний період року та 23–25°C у теплий, відносну вологість 40–60% і швидкість руху повітря не більше 0,1 м/с. У приміщеннях з комп’ютерним обладнанням слід щоденно проводити вологе прибирання та регулярно провітрювати приміщення.

Для запобігання перенавантаженню зорового аналізатора під час роботи з комп’ютером необхідно забезпечити раціональне освітлення робочої зони. Освітленість робочого столу у зоні розташування документів має становити 300–500 лк. Яскравість світильників загального освітлення в зоні кутів випромінювання 50°–90° відносно вертикалі в подовжній та поперечній площинах не повинна перевищувати 200 кд/м², а яскравість відблисків на екрані ВДТ – 40 кд/м² [2].

Для зниження ризику розвитку синдрому зап’ястного каналу та інших захворювань опорно-рухового апарату рекомендується використовувати ергономічні клавіатури та маніпулятори типу «миша». Крім того, необхідно дотримуватися правильної робочої пози: сидіти прямо, не сутулячись; лікті повинні бути зігнуті під кутом 70–90°, стегна – розташовані горизонтально, а стопи – на підлозі або на спеціальній підставці.

Не менш важливим фактором забезпечення безпеки праці є дотримання правил електробезпеки. Комп’ютерне обладнання та периферійні пристрої повинні підключатися до електромережі через справні штепсельні роз’єми. Забороняється використовувати саморобні подовжувачі, нестандартне опалювальне електрообладнання, а також залишати увімкнені прилади без нагляду [2].

Для запобігання ураженню електричним струмом у разі випадкового контакту зі струмопровідними частинами обладнання необхідно забезпечити наявність заземлення або занулення. Крім того, рекомендовано періодично перевіряти стан ізоляції електропроводів та ефективність заземлюючих пристроїв.

Висновок

Дотримання наведених вище вимог і рекомендацій щодо забезпечення безпечних умов праці та мінімізації впливу шкідливих факторів під час роботи з комп’ютерним обладнанням дає змогу зберегти здоров’я і працездатність розробника вебзастосунків, а також підвищити ефективність і якість його професійної діяльності.

6. ОХОРОНА НАВКОЛИШНЬОГО СЕРЕДОВИЩА

6.1 Вступ до проблеми

Охорона навколишнього середовища здійснюється через різні засоби, зокрема й правові механізми. У сфері права захист охоплює майже всі складові природного середовища. Основними сучасними нормативно-правовими актами, які визначають основи організації охорони природи в Україні, є Закони «Про охорону навколишнього природного середовища» (25 червня 1991 р.), «Про охорону атмосферного повітря» (16 жовтня 1992 р.), «Про природно-заповідний фонд України» (16 червня 1992 р.), «Про тваринний світ» (3 березня 1993 р.), «Про карантин рослин» (30 червня 1993 р.) тощо. Крім того, питання використання і захисту природи регулюють також кодекси – земельний, водний, лісовий, про надра, а також закони «Про плату за землю» (3 липня 1992 р.), «Про ветеринарну медицину» (25 червня 1992 р.). Важливим є й затверджений Верховною Радою Порядок щодо обмеження, тимчасової заборони чи припинення діяльності підприємств, установ, організацій та об’єктів у разі порушень законодавства про охорону довкілля.

Види права природокористування включають землекористування, водокористування, лісокористування, користування надрами, тваринним світом та природно-заповідним фондом. Природокористування — це раціональне застосування природних ресурсів людиною для задоволення своїх потреб. Ключові принципи цього права: цільове використання, плановість, тривалість, ліцензування, а також врахування його особливої ролі для суспільства. Виокремлюють загальне й спеціальне природокористування стосовно землі, вод, лісу, надр, тваринного світу тощо. Згідно із Законом «Про охорону навколишнього природного середовища», усі громадяни мають право на загальне користування природними ресурсами безкоштовно та без спеціальних дозволів. Це стосується відвідування парків, скверів, водойм, лісу, збору дикорослих ягід, грибів, горіхів тощо. Стаття 13 Конституції України гарантує це право кожному. Спеціальне природокористування виникає, коли частину ресурсів держава передає у визначених законом випадках індивідам або організаціям на певний термін і за відповідну плату; передача закріплюється спеціальним дозволом.

Права природокористувачів супроводжуються екологічними правами та обов’язками, що інтенсивно розвиваються в юридичній науці. За Конституцією України,

кожен має право на безпечне довкілля і відшкодування шкоди, завданої порушенням цього права. Також гарантується доступ до інформації про стан довкілля та її поширення, що закріплено і в Законі «Про охорону навколишнього природного середовища». Над цим правом лежить обов'язок держави впроваджувати санітарно-гігієнічні заходи для покращення стану довкілля. Екологічні права громадян захищаються у судовому порядку. Разом із цим Закон передбачає певні обов'язки, незалежно від статусу природокористувача: усі зобов'язані берегти природу, використовувати ресурси раціонально та не завдавати їм шкоди. Для суб'єктів спеціального використання природних ресурсів встановлюються додаткові обов'язки, включно зі сплатою за використання та забруднення довкілля згідно з нормативами і лімітами. Контроль у цій сфері здійснюють як державні органи, так і громадські організації шляхом перевірок, нагляду, обстежень та експертиз. Державний контроль здійснюють Ради народних депутатів, державні адміністрації та Міністерство охорони навколишнього природного середовища разом із його місцевими структурами.

Основні пріоритети охорони довкілля та раціонального використання природних ресурсів включають:

- забезпечення екологічної безпеки ядерних об'єктів і захист населення й довкілля від радіації, а також мінімізацію негативних наслідків аварії на Чорнобильській АЕС;
- покращення екологічного стану річкових басейнів України та якості питної води;
- стабілізацію й поліпшення екологічної ситуації у містах і промислових центрах Донецько-Придніпровського регіону;
- будівництво та модернізацію міських очисних каналізаційних споруд;
- боротьбу із забрудненням Чорного й Азовського морів та покращення їхнього стану;
- формування збалансованої системи природокористування, структурну перебудову економіки, екологізацію процесів у промисловості, енергетиці, будівництві, сільському господарстві та транспорті;
- збереження біологічного й ландшафтного різноманіття, розвиток заповідної справи.

Для досягнення цих цілей передбачено такі завдання:

- зменшення рівня радіаційного забруднення до мінімуму;

- захист повітря від забруднення, особливо у великих містах і промцентрах;
- захист і збереження земель від виснаження, забруднення та нераціонального використання;
- збереження та розширення природних ландшафтів, посилення охорони заповідних і рекреаційних територій;
- підвищення стійкості лісів і їхніх екологічних функцій;
- знешкодження, утилізація та захоронення побутових і промислових відходів;
- запобігання забрудненню вод, скорочення й припинення скиду забруднених стоків у водойми, захист підземних вод;
- збереження й відродження малих річок, управління водними ресурсами за басейновим принципом;
- завершення формування державної системи моніторингу довкілля;
- створення систем прогнозування, запобігання й оперативного реагування на надзвичайні ситуації природного та техногенного характеру;
- екологічний супровід конверсії військово-промислового комплексу;
- контроль за екологічною діяльністю Збройних Сил України;
- розробка ефективних механізмів природокористування;
- впровадження економічних важелів впливу на систему природокористування;
- формування системи екологічної освіти, виховання й інформування.

6.2 Забруднення навколишнього середовища підприємством з розробки програмного забезпечення

Основним джерелом забруднення навколишнього середовища підприємством є використання автотранспорту. Підприємство впливає на стан атмосферного повітря, однак не спричиняє забруднення водного чи ґрунтового середовища.

У сучасних умовах автомобільний транспорт є ключовим фактором погіршення якості атмосферного повітря, особливо в межах великих міст. Атмосфера урбанізованих територій характеризується постійним надходженням забруднюючих речовин, що істотно відрізняє її параметри від природного стану. У міських поселеннях спостерігаються найвищі рівні антропогенних навантажень, унаслідок чого змінюються мікрокліматичні фактори та фізико-хімічні властивості повітряного басейну. Згідно з

метеорологічними дослідженнями, температура повітря у містах переважно вища, ніж у передмісті, а також частіше фіксуються атмосферні опади, тумани та смоги – густі утворення, що містять дим, кіптяву і вихлопні гази. Прозорість міської атмосфери є нижчою порівняно із сільською місцевістю, а запиленість та тумани суттєво зменшують проникнення сонячної радіації.

Якість атмосферного повітря у містах дедалі більше погіршується через викиди автотранспортних засобів. Збільшення кількості автомобілів призводить до зростання об’ємів забруднювачів, зокрема оксидів вуглецю, вуглеводнів, оксидів азоту, сажі, а також свинцевих і сірчаних сполук, концентрація яких значно підвищилась. Міська інфраструктура не розрахована на існуючу кількість автотранспорту, що спричиняє короткі ділянки безперервного руху (400–600 м) та низьку середню швидкість (12–20 км/год), збільшуючи витрати пального у 3–4 рази, відповідно посилюючи викиди шкідливих речовин.

Автотранспорт також є джерелом специфічних видів забруднення, зокрема пилу від стирання шин, який у значних обсягах потрапляє у повітря. Окрім продуктів згоряння пального, автомобільний транспорт сприяє накопиченню свинцю в довкіллі. В Україні продовжують використовувати бензин із вмістом свинцю 0,36 г/л, тоді як у Великій Британії, Німеччині та США цей показник становить 0,013–0,15 г/л.

6.3 Розробка заходів щодо зменшення забруднення

Природоохоронна діяльність охоплює всі заходи, спрямовані на підтримку якості довкілля на рівні, що гарантує стійкість біосфери. Вона включає масштабні державні програми зі збереження унікальних природних територій та різноманіття видів, організацію наукових досліджень, підготовку екологічних фахівців і просвітницьку роботу серед населення. Крім того, підприємства здійснюють локальні дії: очищають стічні води та відхідні гази, знижують використання природних ресурсів тощо. Здебільшого такі заходи реалізуються інженерними методами.

Виділяють два основні напрями природоохоронної діяльності підприємств. Перший — очищення шкідливих викидів, який сам по собі є малоефективним, оскільки не завжди дозволяє повністю припинити надходження забруднюючих речовин у біосферу. Зменшення забруднення одного компонента часто призводить до посилення забруднення іншого. Наприклад, впровадження вологих фільтрів у системах

газоочистки зменшує забруднення повітря, але збільшує забруднення води. Уловлені речовини можуть отруювати значні площі землі. Очищувальні споруди скорочують рівень забруднення, проте цілком проблему не вирішують, адже самі генерують нові, хоча й менш об'ємні, але концентровані відходи. Діяльність більшості очисних установок потребує значних енергетичних ресурсів, створюючи додатковий екологічний ризик.

Значна частина забруднювачів, які витрачають кошти на їхнє знешкодження, є речовинами, що вже мають виробничу цінність і можуть бути використані у господарстві. Для інтеграції еколого-економічних рішень важливо поєднати процес очищення шкідливих викидів із утилізацією уловлених речовин.

Другий напрям — усунення причин забруднення шляхом розробки маловідходних або безвідходних технологій, які дають змогу раціонально використовувати сировину і максимально утилізувати шкідливі речовини. Однак зараз не для всіх галузей знайдено економічно прийнятні рішення щодо зменшення відходів та їх утилізації, тому необхідно діяти комплексно у двох напрямках.

Слід пам’ятати, що жодні очисні споруди чи безвідходні технології не здатні відновити стійкість біосфери, якщо буде перевищено граничні показники перетворення природних систем, згідно із законом незамінності біосфери. Критичними є використання понад 1% енергії біосфери й трансформація понад 10% природних територій (правило одного і десяти відсотків). Тому технічний прогрес не замінює необхідності перегляду суспільних пріоритетів, стабілізації населення, створення заповідників та інших заходів.

Основні напрями захисту атмосфери від автотранспортних викидів:

- розвиток і виробництво автомобілів із високоефективними, малотоксичними двигунами, включаючи дизелізацію транспорту;
- впровадження дієвих систем нейтралізації вихлопних газів;
- зниження токсичності моторних палив;
- оптимізація організації транспортного руху в містах та вдосконалення дорожнього будівництва для забезпечення безперервного руху на автомагістралях.

Висновок

Отже, забезпечення ефективного захисту навколишнього середовища вимагає інтегрованого підходу, який поєднує впровадження технологічних інновацій із цілеспрямованою трансформацією суспільних цінностей. Необхідно не лише застосовувати сучасні методи очищення та нейтралізації забруднень, а й оптимізувати використання природних ресурсів і сприяти збереженню екосистем. Лише всебічне поєднання технологічних рішень і свідомої екологічної відповідальності дозволить підтримати стабільність біосфери для майбутніх поколінь.

Висновки

У межах даної роботи здійснено розроблення інформаційної системи обліку паливних матеріалів для підприємства «Південь-Транс». У процесі виконання проєкту були розв'язані такі завдання:

- проведено аналіз сучасних інформаційних технологій, що застосовуються під час створення інформаційних систем;
- сформульовано вимоги до системи;
- розроблено концепцію функціонування системи;
- виконано техніко-економічне обґрунтування доцільності її розроблення;
- визначено загальносистемні рішення, а також рішення щодо інформаційного, організаційного, технічного та програмного забезпечення;
- створено необхідне програмне забезпечення;
- підготовлено комплект супровідної документації.

Під час аналізу сучасних інформаційних технологій було здійснено огляд літературних джерел та матеріалів мережі Internet, у результаті чого визначено технологічні засоби, необхідні для реалізації системи.

Формування вимог до системи ґрунтувалося на дослідженні чинної неавтоматизованої системи обліку, виявленні її недоліків та обґрунтуванні потреби в автоматизації. Було визначено цілі, критерії, обмеження, функції та завдання розроблюваної інформаційної підсистеми.

Під час розроблення концепції системи було сформовано кілька варіантів її реалізації, кожен з яких відповідав установленим вимогам. Для вибору оптимального рішення проведено техніко-економічне обґрунтування із застосуванням методів функціонально-вартісного аналізу. Обраний варіант передбачає використання Web-сервера, сервера доступу до даних та сервера баз даних. Як засоби розроблення застосовано Delphi XE5 компанії Embarcadero Inc.

У межах розроблення інформаційного забезпечення створено концептуальну, логічну та фізичну моделі даних, а також структури файлів аутентифікації та записної книжки.

У частині технічного забезпечення спроєктовано локальну комп'ютерну мережу підприємства «Південь-Транс», необхідну для функціонування системи.

Відповідно до обраної концепції реалізовано програмне забезпечення модульної структури, що забезпечує виконання всіх передбачених функцій.

Підготовлена документація включає: загальний опис системи, керівництво користувача, інструкцію з формування та ведення бази даних, керівництво адміністратора щодо встановлення програмного забезпечення на серверах, а також програму і методику випробувань.

Розроблена інформаційна система обліку паливних матеріалів повністю відповідає вимогам технічного завдання.

Список використаних джерел

1. Chizh, I. M. and Rusanov, S. N. Life Safety: Textbook. Kyiv: Medkniga, 2020.
2. NASA. Handbook for Industrial Noise Control. NASA SP-5108. Washington, 1982.
3. Zinovieva, O. Використання CASE-засобів для проектування інформаційних систем. 2023.
4. Groff, J. R. and Weinberg, P. N. SQL: The Complete Reference. 3rd ed. New York: McGraw-Hill, 1999.
5. Vaskevitch, D. Client/Server Strategies: A Survival Guide for Corporate Reengineers. New York: Wiley, 1996.
6. Robertson, J. and Robertson, S. Complete Systems Analysis: The Workbook, the Textbook, the Answers. New York: Dorset House, 1998.
7. Kalmykov, Y. Delphi Programming for Beginners. LearnDelphi.org, 2020.
8. Konopka, R. Introducing Delphi 95. DelphiByDesign, 1995.
9. Swan, T. and Cogswell, J. Delphi 32-Bit Programming Secrets. Indianapolis: Wiley, 1997.
10. Hermanson, R. H., Edwards, J. D. and Maher, M. W. Accounting Principles: A Business Perspective. 2011.
11. Henderson, K. Database Developer’s Guide with Delphi 2. Indianapolis: Sams Publishing, 1996.

ДОДАТКИ

Додаток А

Загальний опис системи

Призначення системи

Призначенням системи є автоматизація функцій, які виконуються бухгалтером, а також створення єдиної бази даних, у якій зберігається вся інформація про шляхові листи, водіїв, автомобілів, видів палива, поїздки, а також інша інформація.

Інформаційна система володіє наступними функціональними характеристиками:

1)Аутентифікація користувача.

2)Ідентифікація користувача (надання користувачу тільки тих даних, які необхідні йому для виконання своєї роботи).

3)Конфіденційність передачі даних по комп'ютерній мережі.

4)Ведення обліку паливних матеріалів за шляховим листом:

- введення шляхового листа
- пошук шляхового листа за номером
- пошук, довведення, зміни та видалення водія з довідника
- пошук, довведення, зміни та видалення автомобіля з довідника
- довведення, зміни та видалення причепа з довідника
- довведення, зміни та видалення палива з довідника

5) Надання користувачу інформації про його обов’язки.

6)Надання зручного Web-інтерфейсу:

- однакове представлення інформації незалежно від платформи і програмного забезпечення машини користувача;
- сортування даних;
- проста навігація;
- можливості налаштування шрифтів інтерфейсу й інших параметрів інтерфейсу;
- можливість збереження параметрів інтерфейсу між сеансами роботи;
- можливість автоматичного заповнення полів форм.

7)Ведення записної книжки користувача.

- додавання запису;
- зміна запису;
- вилучення запису.

8) Можливість віддаленої роботи з інформаційною системою як у локальній мережі, так і в Internet.

Опис системи

Користувачами системи є директор та головний бухгалтер. Також головний бухгалтер повинен виконувати функції адміністратора – підтримувати роботу системи..

Інформаційне забезпечення системи складається з:

-Бази даних, у якій зберігається вся інформація про ведення обліку паливних матеріалів, шляхових листів.

-Файлів аутентифікації, у яких зберігаються логіни та паролі (у зашифрованому виді) користувачів системи. З цими файлами безпосередньо працює тільки адміністратор системи.

Технічне забезпечення системи складається з одного чи декількох серверів і робочих станцій, встановлених в окремій кімнаті. Для максимальної продуктивності та масштабованості системи необхідно використовувати 2 сервери (хоча для коректної роботи системи досить і одного сервера, на який потрібно установити все програмне забезпечення):

- Web-сервер.
- Сервер доступу до даних чи сервер COM-об'єктів.
- Сервер баз даних (БД).

Програмне забезпечення системи складається з декількох пакетів програм:

- Пакет програм для забезпечення функціональності, необхідної працівникам (описана вище).
- Пакет програм для адміністратора системи.

Кожен пакет програм складається з певної кількості програм (модулів), що реалізують окремі функції системи.

Бізнес-процеси реалізуються CGI-сценаріями, що встановлюються на Web-сервері.

Доступ до бази даних здійснюється за допомогою COM-об'єктів, наданих COM-серверами, і технології OLE DB. COM-сервера та вхідні в них COM-об'єкти встановлюються на сервер доступу до даних.

База даних є реляційною і може працювати під управлінням реляційної СУБД, що підтримує стандарт SQL-92 та має OLE DB і/або ODBC драйвера.

Операційна система, установлена на Web-сервері і сервері доступу до даних, повинна мати реалізацію DCOM/COM+ (операційна система, що рекомендується – MS Windows 2000 Server). Сервер баз даних не має ніяких обмежень, крім тих, котрі накладає СУБД, яка використовується.

На Web-сервері системи повинен бути встановлений HTTP-сервер з підтримкою HTTP 1.1, CGI і можливістю аутентифікації користувача.

Опис взаємозв'язків системи з іншими системами

База даних системи обліку паливних матеріалів на підприємстві «Південь-Транс» може використовуватися іншими інформаційними системами даного підприємства.

Додаток Б

Керівництво користувача

Вступ

Інформаційна система обліку паливних матеріалів розроблена для автотранспортного підприємства «Південь-Транс».

Система автоматизує наступні функції:

- введення шляхового листа
- пошук шляхового листа за номером
- пошук, довведення, зміни та видалення водія з довідника
- пошук, довведення, зміни та видалення автомобіля з довідника
- довведення, зміни та видалення причепа з довідника
- довведення, зміни та видалення палива з довідника

Користувачами системи є бухгалтер та директор підприємства.

Початок роботи

Робота починається з авторизації користувача в системі. Для цього з'являються відразу після запуску програми вікні потрібно ввести ім'я, пароль та натиснути кнопку "Так". Для скасування авторизації і негайного виходу з підсистеми потрібно натиснути кнопку "Скасувати". Після успішної процедури ідентифікації з'являється головне вікно програми з меню керування її модулями.

Робота з довідником видів пального

Для роботи з довідником видів пального необхідно вибрати пункт меню «Довідники | Довідник пального».

Вибір пального здійснюється за допомогою миші або клавіш управління курсором. Потрібно вибрати потрібний тип пального та натиснути кнопку "Вибрати". У разі відсутності потрібного типу пального потрібно створити його. Для цього треба натиснути кнопку "Додати" Для збереження слід натиснути "Зберегти" для скасування змін клавішу "Назад".

Щоб змінити або видалити пального служать кнопки "Змінити" і "Видалити" відповідно.

Примітка: якщо є шляхові листи, в яких присутній видаляємий вид пального, то його видалення неможливо.

Робота з довідником причепів

Для роботи з довідником причепів необхідно вибрати пункт меню «Довідники | Довідник причепів».

Вибір причепа здійснюється за допомогою миші або клавіш управління курсором.

Потрібно вибрати потрібний причіп і натиснути кнопку "Вибрати". У разі відсутності потрібного причепа потрібно створити його. Для цього треба натиснути кнопку "Додати" Для збереження слід натиснути "Зберегти" для скасування змін клавішу "Назад".

Щоб змінити або видалити причепа служать кнопки "Змінити" і "Видалити" відповідно.

Примітка: якщо є шляхові листи, в яких присутній видаляємий причіп, то його видалення неможливо.

Робота з довідником автомобілів

Для того, щоб почати роботу з довідником автомобілів необхідно вибрати пункт меню «Довідники | Довідник автомобілів».

Вибір автомобіля здійснюється за допомогою миші або клавіш управління курсором. Потрібно вибрати потрібний автомобіль і натиснути кнопку "Вибрати".

Якщо потрібно переглянути автомобілі по виду палива (бензин або дизель), потрібно натиснути клавішу F4 або F5 відповідно.

Для пошуку автомобіля за його марку або номером відповідно потрібно натиснути клавішу F6 або F7.

У разі відсутності потрібного автомобіля потрібно створити його. Для цього треба натиснути кнопку "Додати". У з'являються після цього вікні, необхідно ввести реквізити автомобіля: код, державний номер, інвентарний номер, марку вид палива, витрата пального, норми витрати масла (моторне, спеціальне, трансмісійне, пластичний). Для збереження слід натиснути "Зберегти" для скасування змін клавішу "Назад".

Щоб змінити або видалити автомобіля служать кнопки "Змінити" і "Видалити" відповідно.

Примітка: якщо є шляхові листи, в яких присутній видаляємий автомобіль, то його видалення неможливо.

Робота з довідником водіїв.

Для того, щоб почати працювати з довідником водіїв необхідно вибрати пункт меню «Довідники | Довідник водіїв». Для пошуку необхідного водія за його прізвища, у вікні, напортів слова «Прізвище» необхідно ввести початкові літери прізвища, у міру введення символів відбувається відображення табельного номера та повних ініціалів у вікні перегляду.

Якщо необхідно дізнатися повні ініціали водія за відомим табельний номер, то необхідно натиснути клавішу F7.

Щоб додати до довідник нового водія на навігаційній панелі потрібно натиснути кнопку «Додати».

Щоб змінити або видалити водія служать кнопки "Змінити" і "Видалити" відповідно.

Примітка: якщо є шляхові листи, в яких присутній видаляємий водій, то його видалення неможливо.

Вибір водія здійснюється за допомогою миші або клавіш управління курсором. Потрібно вибрати потрібного водія і натиснути кнопку "Вибрати".

Введення і корегування шляхового листа

Для введення нового шляхового листа необхідно вибрати пункт меню «Документи | Введення шляхового листа».

У вікні, що з'явилося, необхідно занести значення реквізитів шляхового листа, згідно з пояснювальні написами, розташованим поруч з вводяться реквізитом. Такі реквізити як номер автомобіля, табельний номер водія, номер причепа можна вибрати з довідника. Для цього необхідно на відповідному реквізиті натиснути клавішу F7 або праву кнопку миші.

Після введення шляхового листа можна продовжити введення іншого шляхом натискання кнопки «Додати» або покинути вікно, натиснувши кнопку «Назад».

Щоб змінити або видалити шляховий лист служать кнопки "Змінити" і "Видалити" відповідно.

Перегляд шляхових листів

Для перегляду шляхових листів необхідно вибрати пункт меню «Документи | Перегляд». У вікні, що з'явилося, необхідно задати умови перегляду листів, тобто проміжок дат виписки аркушів, номер автомобіля, номер пачки, номер водія. Можна залишити всі поля не заповненими, тоді для перегляду будуть представлені всі шляхові листи за поточний звітний період.

Для відмови від перегляду потрібно натиснути кнопку «Скасування», для продовження - «Продовжити».

Після натискання кнопки «Продовжити», з'являється вікно корегування шляхових листів, робота з якими описана вище.

Формування та друк звітів.

Для формування звітів служать відповідні пункти меню в підменю "Звіти" головного меню програми. Після вибору пункту та введення параметрів звіту (початкова дата, кінцева дата, код автомобіля і т.д.) відбувається відображення звіту на екрані в тому вигляді, як він буде надрукований. Для друку звіту на принтері необхідно натиснути кнопку «Друк».

Вихід з програми.

Для завершення роботи з програмою необхідно вибрати пункт меню "Вихід".

Додаток В

Інструкція з формування і ведення бази даних

Правила підготовки даних

Перш, ніж інформаційна система обліку паливних матеріалів зможе коректно працювати, необхідно підготувати дані і заповнити базу даних.

Більшість необхідних даних вже існує у відповідних документах і відомостях. Так, існують документи, що містять інформацію про:

- водія.
- автомобіль
- причеп
- норми витрат палива

Кожному пункту інформації, що представлена в списку, повинні відповідати спеціальні форми для введення даних у базу даних.

Порядок і засоби заповнення бази даних

Базу даних можна заповнювати декількома способами:

- За допомогою спеціально розробленого для цього програмного забезпечення для адміністратора бази даних.
- За допомогою програмного забезпечення для бухгалтера або директора.
- За допомогою команд СУБД, що використовується.

Програмне забезпечення для адміністратора бази даних повинне використовуватися адміністратором для заповнення бази даних. Перш, ніж із системою зможуть працювати користувачі, адміністратор повинен подбати про занесення інформації в базу даних.

Користувачі системи заповнюють деякі таблиці бази даних, інформацію в який не має права змінювати ніхто інший. Використання програмного забезпечення для працівника докладно описано в керівництві користувача.

Для додавання, вилучення і зміни даних можна також використовувати команди СУБД (у даному випадку це MS SQL Server 2008): INSERT, DELETE, UPDATE і BULK INSERT. Докладніше про ці команди читайте в документації СУБД. Використовувати цей метод не рекомендується, тому що можна порушити цілісність даних.

Процедури зміни і контролю бази даних

Після того, як адміністратор ввів усю необхідну інформацію, він може додавати, вилучати або змінювати дані при необхідності (наприклад, з'явився новий водій, якого

немає в базі даних). Та інформація, що вводить адміністратор перед введенням системи в експлуатацію, є практично статичною.

Адміністратору не потрібно вести контроль таблиць, дані в які вводить бухгалтер. Весь необхідний контроль здійснює програмне забезпечення. Адміністратор повинен втрутитися лише в тому випадку, якщо програмне забезпечення видає повідомлення про помилки.

Після закінчення терміну в один рік адміністратор повинен зробити копію бази даних і покласти в архів. Після цього потрібно знову підготувати базу даних для наступного періоду, тому що могли змінитися деякі дані.

Порядок і засоби відновлення бази даних

Відповідно до проведеного розрахунку, розмір бази даних на один рік не перевищить 10 Мбайт. Таким чином, база даних має маленький розмір і найбільш доцільно використовувати повне резервне копіювання.

Раз у тиждень адміністратор повинен створювати повну резервну копію за допомогою засобів СУБД чи копіюванням файлів бази даних. У СУБД MS SQL Server 2008 для цих цілей можна використовувати Enterprise Manager або майстер Create Database Backup Wizard. Резервне копіювання потрібно робити на знімний жорсткий диск ємкістю не менше 10 Мбайт або на інші доступні носії інформації.

При виникненні збою в роботі сервера баз даних пов'язаного з інформацією, яка зберігається в базі даних, потрібно скористатися останньою повною резервною копією бази даних для відновлення інформації.

Додаток Г

Програма і методика випробувань

Об'єкт випробувань

Об'єктом випробувань є інформаційна система обліку паливних матеріалів, призначена для автоматизації роботи бухгалтера та директора на підприємстві «Південь-Транс»

Цілі випробувань

Цілі випробувань: перевірка придатності системи до використання по призначенню, її відповідність заданим вимогам і документації; перевірка працездатності і правильності роботи системи.

Вимоги до системи

Повний список вимог до системи приведений у пункті 1.5 пояснювальної записки. Нижче приводяться функціональні вимоги до системи.

Система повинна задовольняти наступним функціональним характеристикам:

1. Аутентифікація користувача.
2. Ідентифікація користувача (надання користувачу тільки тих даних, які необхідні йому для виконання своєї роботи).
3. Конфіденційність передачі даних по комп'ютерній мережі.
4. Ведення обліку паливних матеріалів за шляховим листом:
   - введення шляхового листа
   - пошук шляхового листа за номером
   - пошук, довведення, зміни та видалення водія з довідника
   - пошук, довведення, зміни та видалення автомобіля з довідника
   - довведення, зміни та видалення причепа з довідника
   - довведення, зміни та видалення палива з довідника

5. Надання користувачу інформації про його обов’язки.
6. Надання зручного Web-інтерфейсу:
   - однакове представлення інформації незалежно від платформи і програмного забезпечення машини користувача;
   - сортування даних;
   - проста навігація;

- можливості налаштування шрифтів інтерфейсу й інших параметрів інтерфейсу;
- можливість збереження параметрів інтерфейсу між сеансами роботи;
- можливість автоматичного заповнення полів форм.

7. Ведення записної книжки користувача.
   - додавання запису;
   - зміна запису;
   - вилучення запису.
8. Можливість віддаленої роботи з інформаційною системою як у локальній мережі, так і в Internet.

Вимоги до програмної документації

У комплект програмної документації повинні входити наступні документи:

- Загальний опис системи.
- Керівництво користувача.
- Інструкція з формування і ведення бази даних.
- Керівництво адміністратора з установки програмного забезпечення на серверах.
- Програма і методика випробувань.
- Текст програми.

Засоби і порядок випробувань

Вимоги до складу і параметрів технічних і програмних засобів вказані в пункті 1.5 пояснювальної записки. Для правильної установки і настройки програмного забезпечення системи необхідно скористатися інструкцією з формування і ведення бази даних. Сервера повинні бути об'єднані в комп'ютерну мережу з декількома клієнтами. Як мережне середовище можна використовувати кабелі типу «вита пара» категорії 3, 5 і вище. Мережне обладнання повинно підтримувати швидкість не менше 10 Мбіт/сек.

Випробування системи будуть проводитися протягом двох тижнів. Замовник і/або його представники самостійно проводять тестування усіх функцій системи в присутності розробника. При наявності відповідної можливості, у процесі тестування можуть додатково брати участь програмісти, тестувальники і системотехніки, які не брали участь у розробці системи.

Перед початком (не пізніше 3 днів) випробувань уточнюється кінцевий склад групи, що буде проводити випробування. Кожний член групи отримує необхідну документацію до системи (технічне завдання, загальний опис системи, керівництво

користувача, інструкція з формування і ведення бази даних, керівництво адміністратора з установки програмного забезпечення на серверах, програма і методика випробувань).

Порядок проведення випробувань:

- Виконуються контрольні тести в довільному порядку. Результати заносяться у відповідний журнал.
- Робиться аналіз отриманих результатів і встановлюється відповідність програми заданим вимогам і системним документам.

При виявленні помилок у роботі системи складається їхній перелік і обговорюється термін їхнього виправлення розробником. Після цього замовник проводить повторне тестування в повному обсязі (можливо використання нових чи додаткових тестів).

Методика випробувань

Процес тестування являє собою експлуатацію системи в контрольованих умовах і вивчення отриманих результатів. При цьому перевіряється робота системи з нормальними і помилковими даними і подіями. Тестування розподілених систем сильно ускладнено великою кількістю взаємозв'язків. Кожна залежність, Web-сторінка, компонент, база даних, а також код графічного інтерфейсу, програмне забезпечення середнього рівня і мережна інфраструктура повинні тестуватися не тільки на коректність роботи, але і на сумісність один з одним у різних конфігураціях.

Найкраще для тестування розподілених систем застосовувати метод «знизу-нагору». Спочатку розробник повинен перевірити кожен компонент (елемент системи) окремо; потім – протестувати взаємодію компонента з іншими компонентами системи на автономному комп'ютері; і тільки потім – перевірити роботу всієї системи в розподіленому середовищі. Усі ці види тестування виконуються розробником системи без участі замовника. Розробник не зобов'язаний надавати тести і протоколи результатів тестування кому-небудь.

При прийманні системи проводяться випробування системи в розподіленому середовищі. Замовник і/або його представники самостійно проводять тестування усіх функцій системи методом «чорного ящика» у присутності розробника.

Програма випробувань

Тести складаються замовником і/або його представниками. При цьому перевіряється функціонування системи з нормальними і помилковими даними і подіями.

Наприклад:

1) Аутентифікація користувача з правильними даними.
2) Аутентифікація користувача з неправильними даними.

Під неправильними чи помилковими даними в контексті даної системи потрібно розуміти:

- Логін і/або пароль користувача, яких немає у файлах аутентифікації (для перших двох тестів).
- Неіснуюча дата (для всіх інших тестів).

Програмне забезпечення розроблено таким чином, що користувач не може ввести які-небудь інші неправильні дані в систему через наданий йому інтерфейс. Таким чином, будь-які дані, що не є неправильними, потрібно вважати правильними.

- Для визначення переліку вхідних даних для кожного тесту можна скористатися формами для введення відповідних даних. У формах зазначені необхідні поля.
- Вихідні дані чи результат при правильних даних повинні відповідати виконанню функції, для тестування якої створений тест; при неправильних даних – відмовлення в доступі до системи (тест 2) чи повідомлення про помилку.

Додаток Д

Текст програми

Вступ

Програмне забезпечення інформаційної системи обліку паливних матеріалів на підприємстві «Південь-Транс» складається з модулів. Розмір первісного програмного коду всіх модулів наближається до 10000 рядків, тому наводити його повністю недоцільно. Для прикладу, наведемо текст декількох модулів.

Головний модуль

```
unit Main;
interface
uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, SoStatus, Menus, RxMenus, SoTools, SoMenuEd, SoCmplFm, Placemnt,
  SoCtrls, DBTables, StdCtrls, Buttons, RXShell, PicClip, Mask, ToolEdit,
  SpeedBar, ComCtrls, ToolWin, RxGrdCpt;
type
  TMainForm = class(TForm)
    MainMenu: TRxMainMenu;
    ReferenceItem: TMenuItem;
    DocumentsItem: TMenuItem;
    ReportItem: TMenuItem;
    ExitItem: TMenuItem;
    EmployeeList: TMenuItem;
    AutoList: TMenuItem;
    GasTypesList: TMenuItem;
    ItemNewDoc: TMenuItem;
    N9: TMenuItem;
    N10: TMenuItem;
    N12: TMenuItem;
    N13: TMenuItem;
    N14: TMenuItem;
    N15: TMenuItem;
    FormStorage: TFormStorage;
    Status: TsohoStatus;
    TrailerList: TMenuItem;
    N2: TMenuItem;
    N3: TMenuItem;
    N4: TMenuItem;
    ItemCorrectDoc: TMenuItem;
    ItemViewDoc: TMenuItem;
    sohoPanel1: TsohoPanel;
    SpeedBar1: TSpeedBar;
    SpeedbarSection1: TSpeedbarSection;
    SpeedbarSection2: TSpeedbarSection;
    btnAutomobile: TSpeedItem;
    btnEmployes: TSpeedItem;
    btnTrailers: TSpeedItem;
    btnNew: TSpeedItem;
    btnView: TSpeedItem;
    btnCorrect: TSpeedItem;
    btnFuel: TSpeedItem;
    SpeedbarSection3: TSpeedbarSection;
    btnExit: TSpeedItem;
    RxGradientCaption1: TRxGradientCaption;
```

```
    procedure EmployeeListClick(Sender: TObject);
    procedure ExitItemClick(Sender: TObject);
    procedure AutoListClick(Sender: TObject);
    procedure GasTypesListClick(Sender: TObject);
    procedure TrailerListClick(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure N13Click(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure N3Click(Sender: TObject);
    procedure ShowCaption;
    procedure N9Click(Sender: TObject);
    procedure N10Click(Sender: TObject);
    procedure N4Click(Sender: TObject);
    procedure N15Click(Sender: TObject);
    procedure N12Click(Sender: TObject);
    procedure N14Click(Sender: TObject);
    procedure ItemNewDocClick(Sender: TObject);
    procedure ItemCorrectDocClick(Sender: TObject);
    procedure ItemViewDocClick(Sender: TObject);
    procedure MainMenuGetItemParams(Sender: TMenu; Item: TMenuItem;
      State: TMenuOwnerDrawState; AFont: TFont; var Color: TColor;
      var Graphic: TGraphic; var NumGlyphs: Integer);
private
      { Private declarations }
  public
    { Public declarations }
  end;

var
  MainForm: TMainForm;

implementation
{$R *.DFM}
uses Employes, CommUtil, AutoList, GasTypes, PutList, Trailer, GetPDate,
     BSPrint, MyPpView, Data, Config, WayBillRep, Vault, Splash, StoreGas,
     Kard, Oil, TrailerReport;

procedure TMainForm.ShowCaption;
begin
 Status.Caption:= 'Учет работы автотранспорта с ' + FormatDateTime('d.mm.yyyy',
dmAutomobile.qryGetConfigDataSTARTDATE.AsDateTime) +
                  ' по ' + FormatDateTime('d.mm.yyyy',
dmAutomobile.qryGetConfigDataENDDATE.AsDateTime);
end;

procedure TMainForm.EmployeeListClick(Sender: TObject);
begin
{Employes}
 EmployeesShow(False);
end;

procedure TMainForm.ExitItemClick(Sender: TObject);
begin
 Application.Terminate;
end;

procedure TMainForm.AutoListClick(Sender: TObject);
var
  N: integer;
begin
 {AutoList}
  ShowAutomobile(False, N);
end;
```

```
procedure TMainForm.GasTypesListClick(Sender: TObject);
begin
 {GasTypes}
 ShowGasType;
end;

procedure TMainForm.TrailerListClick(Sender: TObject);
var n: integer;
begin
 {Trailer}
  ShowTrailer(False, n);
end;
procedure TMainForm.FormActivate(Sender: TObject);
begin
 {Data}
 with dmAutomobile.qryGetConfigData do
 begin
  if Active then Close;
  Open;
 end;
 ShowCaption;
end;

procedure TMainForm.N13Click(Sender: TObject);
begin
 {BSPrint}
 PrintOnBS;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
 with frmSplash do
 begin
  Status.Caption:= 'Откртытие базы...';
  dmAutomobile.dbWayBill.Open;
  Status.Caption:= 'Загрузка параметров...';
//  dmAutomobile.qryGetConfigData.Open;
  Status.Caption:= 'Запуск программы...';
 end;
  SetPPPreview(True);
end;

procedure TMainForm.N3Click(Sender: TObject);
begin
 {Config}
 if ShowConfig then ShowCaption;;
end;

procedure TMainForm.N9Click(Sender: TObject);
begin
 {WayBillRep}
 PrintWayBillRep;
end;

procedure TMainForm.N10Click(Sender: TObject);
begin
 {Vault}
 PrintVault;
end;

procedure TMainForm.N4Click(Sender: TObject);
begin
```

```
 {StoreGas}
 PrintStoreGas;
end;

procedure TMainForm.N15Click(Sender: TObject);
begin
 {Kard}
 PrintKard;
end;

procedure TMainForm.N12Click(Sender: TObject);
begin
 {Oil}
 PrintOilReport;
end;

procedure TMainForm.N14Click(Sender: TObject);
begin
 {TrailerReport}
 PrintTrailerReport;
end;

procedure TMainForm.ItemNewDocClick(Sender: TObject);
begin
 {PutList}
 ShowPutList(dmAutomobile.qryGetConfigDataSTARTDATE.AsDateTime,
dmAutomobile.qryGetConfigDataENDDATE.AsDateTime, atNew);
end;

procedure TMainForm.ItemCorrectDocClick(Sender: TObject);
begin
 {PutList}
 ShowPutList(dmAutomobile.qryGetConfigDataSTARTDATE.AsDateTime,
dmAutomobile.qryGetConfigDataENDDATE.AsDateTime, atCorrect);
end;

procedure TMainForm.ItemViewDocClick(Sender: TObject);
begin
 {PutList}
 ShowPutList(dmAutomobile.qryGetConfigDataSTARTDATE.AsDateTime,
dmAutomobile.qryGetConfigDataENDDATE.AsDateTime, atView);
end;

procedure TMainForm.MainMenuGetItemParams(Sender: TMenu; Item: TMenuItem;
  State: TMenuOwnerDrawState; AFont: TFont; var Color: TColor;
  var Graphic: TGraphic; var NumGlyphs: Integer);
begin
  if Item = AutoList then Graphic := btnAutomobile.Glyph;
  if Item = EmployeeList then Graphic:= btnEmployes.Glyph;
  if Item = TrailerList then Graphic:= btnTrailers.Glyph;
  if Item = GasTypesList then Graphic:= btnFuel.Glyph;
  if Item = ItemNewDoc then Graphic:= btnNew.Glyph;
  if Item = ItemViewDoc then Graphic:= btnView.Glyph;
  if Item = ItemCorrectDoc then Graphic:= btnCorrect.Glyph;
//  if Item = ExitItem then Graphic:= btnExit.Glyph;
end;

end.
```

Довідник водіїв

```
unit Employes;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, StdCtrls, DBCtrls, Grids, DBGrids, RXDBCtrl, DBTables, ExtCtrls,
  Placemnt, Menus, Clipbrd, ComCtrls, SohoBtns;

type
  TfrmEmployes = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    Panel3: TPanel;
    qryEmployees: TQuery;
    dsrcEmployers: TDataSource;
    dbgEmployers: TRxDBGrid;
    dbnEmployers: TDBNavigator;
    lblName: TLabel;
    edName: TEdit;
    qryEmployeesTABN: TIntegerField;
    qryEmployeesFIO: TStringField;
    FormStorage: TFormStorage;
    PopupMenu: TPopupMenu;
    miTabNSearch: TMenuItem;
    N4: TMenuItem;
    miShowAll: TMenuItem;
    N6: TMenuItem;
    miSelect: TMenuItem;
    miClose: TMenuItem;
    qryEmployeesSALDO_BS_71: TCurrencyField;
    btnSelect: TsohoBitBtn;
    btnCancel: TsohoBitBtn;
    procedure btnSelectClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
    procedure FormDestroy(Sender: TObject);
    procedure miFIOSearchClick(Sender: TObject);
    procedure miShowAllClick(Sender: TObject);
    procedure miTabNSearchClick(Sender: TObject);
    procedure miCloseClick(Sender: TObject);
    procedure edNameChange(Sender: TObject);
    procedure edNameKeyDown(Sender: TObject; var Key: Word;
      Shift: TShiftState);
    procedure N3Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmEmployes: TfrmEmployes;

function EmployersShow(NeedPaste: Boolean): integer;
function EmployeesShow(NeedPaste: boolean): Boolean;
function EmployeesSearch(PasteControl : TCustomEdit) : Boolean;
function EmployeesSearchInt(var EmpNo : integer) : Boolean;

implementation
uses DBUtils, StrUtils;
{$R *.DFM}
```

```
function EmployeesSearch(PasteControl : TCustomEdit) : Boolean;
var
  NeedPaste : Boolean;
  S : Integer;
begin
  NeedPaste := PasteControl <> nil;
  with TfrmEmployes.Create(Nil) do
    try
      btnSelect.Enabled := NeedPaste;
      miSelect.Enabled :=  NeedPaste;

      if DelSpace(PasteControl.Text) = ''
        then S := 0
        else S := StrToInt(DelSpace(PasteControl.Text));
      qryEmployees.Locate('TABN',Variant(S),[loPartialKey]);
      Result := ShowModal = mrOk;
      if Result and NeedPaste
        then PasteControl.Text := qryEmployeesTabn.AsString;
    finally
      Free;
    end;
end;

function EmployeesSearchInt(var EmpNo : integer) : Boolean;
begin
  with TfrmEmployes.Create(Nil) do
    try
      btnSelect.Enabled := True;
      miSelect.Enabled := True;

      qryEmployees.Locate('TABN',EmpNo,[loPartialKey]);
      Result := ShowModal = mrOk;
      if Result then EmpNo := qryEmployeesTabn.AsInteger;
    finally
      Free;
    end;
end;

function EmployeesShow(NeedPaste: boolean): Boolean;
begin
  Result := EmployersShow(NeedPaste) = mrOk;
end;

function EmployersShow(NeedPaste: boolean): integer;
begin
  with TfrmEmployes.Create(Nil) do
    try
      btnSelect.Enabled := NeedPaste;
      miSelect.Enabled := NeedPaste;

      Result := ShowModal;
    finally
      Free;
    end;
end;

procedure TfrmEmployes.btnSelectClick(Sender: TObject);
begin
  with TClipboard.Create do
    try
      AsText := qryEmployeesTabN.AsString;
    finally
      Free;
```

```
    end;
  ModalResult := mrOk;
end;

procedure TfrmEmployes.FormCreate(Sender: TObject);
begin
  if qryEmployees.Active then qryEmployees.Close;
  qryEmployees.Open;
end;

procedure TfrmEmployes.FormDestroy(Sender: TObject);
begin
 if qryEmployees.Active then qryEmployees.Close;
end;

procedure TfrmEmployes.miFIOSearchClick(Sender: TObject);
const
  S: string = '';
begin
  if InputQuery('Ввод фамилии', 'Фамилия', S) then
    with qryEmployees do
      begin
        DisableControls; Close;
        SQL.Clear;
        SQL.Add('SELECT TABN, FIO, SALDO_BS_71 FROM EMPLOYES '+
                'WHERE (DATUV IS NULL) AND (FIO LIKE "%'+S+'%") ORDER BY FIO');
        Open; EnableControls;
      end;
end;

procedure TfrmEmployes.miShowAllClick(Sender: TObject);
begin
  with qryEmployees do
    begin
      DisableControls; Close;
      SQL.Clear;
      SQL.Add('SELECT TABN, FIO, SALDO_BS_71 FROM EMPLOYES '+
              'WHERE DATUV IS NULL ORDER BY FIO');
      Open; EnableControls;
    end;
end;

procedure TfrmEmployes.miTabNSearchClick(Sender: TObject);
const
  S: string = '';
begin
  if InputQuery('Ввод табельного №', 'Табельный №', S) then
    with qryEmployees do
      begin
        DisableControls; Close;
        SQL.Clear;
        SQL.Add('SELECT TABN, FIO, SALDO_BS_71 FROM EMPLOYES '+
                'WHERE (DATUV IS NULL) AND (TABN = '+S+') ORDER BY FIO');
        Open; EnableControls;
      end;
end;

procedure TfrmEmployes.miCloseClick(Sender: TObject);
begin
  ModalResult := mrCancel;
end;

procedure TfrmEmployes.edNameChange(Sender: TObject);
```

```
begin
  qryEmployees.Locate('FIO',Variant((Sender as TEdit).Text),
                      [loCaseInsensitive,loPartialKey])
end;

procedure TfrmEmployes.edNameKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
   case Key of
     vk_return : btnSelect.Click;
     vk_Prior, vk_up  : qryEmployees.Prior;
     vk_Next, vk_down   : qryEmployees.Next;
   end;

  if Key in [vk_next, vk_Prior, vk_up, vk_down]
    then begin
      (Sender as TEdit).Text := qryEmployeesFIO.AsString;
      (Sender as TEdit).SelectAll;
    end;

end;

procedure TfrmEmployes.N3Click(Sender: TObject);
begin
  edName.SelectAll;
  edName.SetFocus;
end;

end.
```

Довідник автомобілів

```
unit AutoList;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Db, DBTables, RxQuery, Placemnt, Grids, DBGrids, RXDBCtrl, DBCtrls,
  StdCtrls, SohoBtns, ExtCtrls, Menus, Myclass, SoDBGrid, SoCtrls;

type
  TfrmAutoList = class(TForm)
    Panel2: TPanel;
    Panel3: TPanel;
    dbnAuto: TDBNavigator;
    FormStorage: TFormStorage;
    qryAuto: TRxQuery;
    dsrcAuto: TDataSource;
    qryAutoAUTONUM: TIntegerField;
    qryAutoSTATENUM: TStringField;
    qryAutoBRAND: TStringField;
    PopupMenu: TPopupMenu;
    miShowAll: TMenuItem;
    miShowGas: TMenuItem;
    miShowDisel: TMenuItem;
    N4: TMenuItem;
    miBrandSearch: TMenuItem;
    N7: TMenuItem;
    miSelect: TMenuItem;
    RadioGroup: TRadioGroup;
    SearchPanel: TPanel;
    lblCaption: TLabel;
    edSearch: TETEdit;
    miNumberSearch: TMenuItem;
    dbgAuto: TsohoDBGrid;
    UpdateSQL: TUpdateSQL;
    qryAutoNAME: TStringField;
    qryAutoCONSUMPTION: TFloatField;
    qryAutoMOTOR_OIL: TFloatField;
    qryAutoTRANSMISION_OIL: TFloatField;
    qryAutoSPECIAL_OIL: TFloatField;
    qryAutoPLASTIC_OIL: TFloatField;
    Panel1: TPanel;
    Panel4: TPanel;
    Panel5: TPanel;
    btnSelect: TsohoBitBtn;
    btnCancel: TsohoBitBtn;
    sohoLabel1: TsohoLabel;
    sohoLabel2: TsohoLabel;
    qryAutoINVNO: TIntegerField;

    procedure OpenAutoQuery(mode: integer);
    procedure miShowAllClick(Sender: TObject);
    procedure edSearchExit(Sender: TObject);
    procedure dbnAutoBeforeAction(Sender: TObject; Button: TNavigateBtn);
    procedure dbgAutoKeyPress(Sender: TObject; var Key: Char);
    procedure miSelectClick(Sender: TObject);
    procedure edSearchKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  frmAutoList: TfrmAutoList;

  function ShowAutomobile(Selected: Boolean; var AutoNum: integer): Boolean;

implementation
uses Automobile, Search;
{$R *.DFM}

procedure TfrmAutoList.OpenAutoQuery(mode: integer);
{1 - показать все
 2 - бензин
 3 - диз. топливо
 4 - по марке
 5 - по номеру}
 var
  S: string;
begin
 S:='';
 with qryAuto do
 begin
  if Active then Close;
  case Mode of
   1: S:='0=0';
   2: S:='GAS <= 10';
   3: S:='GAS >10 AND GAS <=20';
   4: S:= 'BRAND LIKE ' + #39 + edSearch.Text + '%'+ #39;
   5: S:= 'AUTONUM = ' + edSearch.Text;
  end;
 MacroByName('options').AsString:= S;
 SearchPanel.Visible:= False;
 Open;
 end;
end;

function ShowAutomobile(Selected: Boolean; var AutoNum: integer): Boolean;
begin
 with TfrmAutoList.Create(nil) do
 try
//  btnSelect.Visible:= Selected;
  btnSelect.Enabled:= Selected;
//  miSelect.Enabled:= btnSelect.Visible;
  miSelect.Enabled:= btnSelect.Enabled;
  OpenAutoQuery(1);
  if ShowModal = mrOk then
  begin
   AutoNum:= qryAutoAUTONUM.AsInteger;
   Result := True;
  end
   else Result:= False;
 finally
  Free;
 end;
end;

procedure TfrmAutoList.miShowAllClick(Sender: TObject);
begin
 if Sender = RadioGroup then TComponent(Sender).Tag := RadioGroup.ItemIndex + 1;
 if Sender = PopupMenu then RadioGroup.ItemIndex:= TComponent(Sender).Tag - 1;
 case TComponent(Sender).Tag of
  1: OpenAutoQuery(1);
  2: OpenAutoQuery(2);
```

```
  3: OpenAutoQuery(3);
  4,5:begin
       SearchPanel.Visible:= True;
       if TComponent(Sender).Tag = 4 then lblCaption.Caption:= 'Марка:'
        else lblCaption.Caption:= 'Номер:';
       edSearch.SetFocus;
       RadioGroup.ItemIndex:= -1;
      end;
 end {case};
end;

procedure TfrmAutoList.edSearchExit(Sender: TObject);
begin
 if lblCaption.Caption= 'Марка:' then OpenAutoQuery(4)
 else OpenAutoQuery(5);
 dbgAuto.SetFocus;
end;

procedure TfrmAutoList.dbnAutoBeforeAction(Sender: TObject;
  Button: TNavigateBtn);
begin
 {Automobile}
 case Button of
  nbInsert: begin
             EditAutoList(qryAutoAUTONUM.AsInteger, atAdd);
             QryRefresh(qryAuto, 'AUTONUM');
             Abort;
            end;
  nbEdit:   begin
             EditAutoList(qryAutoAUTONUM.AsInteger, atEdit);
             QryRefresh(qryAuto, 'AUTONUM');
             Abort;
            end;
  nbDelete: begin
             if MessageDlg('Вы действительно хотите удалить запись?',
mtConfirmation, [mbYes, mbNo], 0) = mrYes
              then EditAutoList(qryAutoAUTONUM.AsInteger, atDelete);
             QryRefresh(qryAuto, 'AUTONUM');
             Abort;
            end;
 end {case};
end;

procedure TfrmAutoList.dbgAutoKeyPress(Sender: TObject; var Key: Char);
begin
 if (Key = #13) and (btnSelect.Visible = True)  then ModalResult := mrOk;
 if (Key = #27) then ModalResult:= mrCancel;
end;

procedure TfrmAutoList.miSelectClick(Sender: TObject);
begin
 Close;
end;

procedure TfrmAutoList.edSearchKeyPress(Sender: TObject; var Key: Char);
begin
 if key = #27 then
 begin
  miShowAllClick(RadioGroup);
  dbgAuto.SetFocus;
 end;
end;
```

```
end.
unit Automobile;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  StdCtrls, SohoBtns, ExtCtrls, Placemnt, SoCtrls, DBCtrls, Myclass, Mask,
  DBTables, Db, RxLookup;

type
  TActionType =(atAdd, atEdit, atDelete);

  TfrmAutomobile = class(TForm)
    Panel1: TPanel;
    btnCancel: TsohoBitBtn;
    Panel2: TPanel;
    FormStorage: TFormStorage;
    sohoLabel1: TsohoLabel;
    sohoLabel2: TsohoLabel;
    sohoLabel3: TsohoLabel;
    sohoLabel4: TsohoLabel;
    sohoLabel5: TsohoLabel;
    GroupBox: TGroupBox;
    sohoLabel6: TsohoLabel;
    sohoLabel7: TsohoLabel;
    sohoLabel8: TsohoLabel;
    sohoLabel9: TsohoLabel;
    sohoLabel10: TsohoLabel;
    btnOk: TsohoBitBtn;
    qryAuto: TQuery;
    dsrcAuto: TDataSource;
    UpdateSQL: TUpdateSQL;
    qryGasType: TQuery;
    qryAutoINVNO: TIntegerField;
    qryAutoGAS: TIntegerField;
    qryAutoAUTONUM: TIntegerField;
    qryAutoSTATENUM: TStringField;
    qryAutoBRAND: TStringField;
    qryAutoCONSUMPTION: TFloatField;
    qryAutoMOTOR_OIL: TFloatField;
    qryAutoTRANSMISION_OIL: TFloatField;
    qryAutoSPECIAL_OIL: TFloatField;
    qryAutoPLASTIC_OIL: TFloatField;
    dbeAutoNum: TETDBEdit;
    dbeStateNum: TETDBEdit;
    dbeInvNo: TETDBEdit;
    dbeMotor: TETDBEdit;
    dbeBrand: TETDBEdit;
    dbeTransm: TETDBEdit;
    dbeSpecial: TETDBEdit;
    dbePlastic: TETDBEdit;
    dbeConsamption: TETDBEdit;
    cmbGasType: TETRxDBLookupCombo;
    dsrcGasType: TDataSource;
    procedure btnOkClick(Sender: TObject);
    procedure FormCreate(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;
```

```
var
  frmAutomobile: TfrmAutomobile;

  procedure EditAutoList(AutoNum: Integer;Action: TActionType);
  procedure NewAuto(AutoNum: Integer);
  procedure EditAuto(AutoNum: Integer);
  procedure DeleteAuto(AutoNum: Integer);

implementation
 uses AutoList, Search;
{$R *.DFM}

procedure EditAutoList(AutoNum: Integer; Action: TActionType);
begin
 case Action of
  atAdd:    NewAuto(AutoNum);
  atEdit:   EditAuto(AutoNum);
  atDelete: DeleteAuto(AutoNum);
 end;
end;

procedure NewAuto(AutoNum: Integer);
begin
 with TfrmAutomobile.Create(nil) do
 try
  Caption:= 'Новая запись';

  if qryAuto.Active then qryAuto.Close;
  qryAuto.ParamByName('AUTONUM').AsInteger:= AutoNum;
  qryAuto.Open;

  qryAuto.Insert;
  ShowModal;
 finally
  Free;
 end;
end;


procedure EditAuto(AutoNum: Integer);
begin
 with TfrmAutomobile.Create(nil) do
 try
  Caption:= 'Редактирование ' + IntToStr(AutoNum)+' ...';

  if qryAuto.Active then qryAuto.Close;
  qryAuto.ParamByName('AUTONUM').AsInteger:= AutoNum;
  qryAuto.Open;

  ShowModal;
 finally
  Free;
 end;
end;

procedure DeleteAuto(AutoNum: Integer);
begin
 with TQuery.Create(nil) do
 try
  DatabaseName:= 'SQL';
  SQL.Clear;
  SQL.Append('DELETE FROM AUTOMOBILE WHERE AUTONUM = :AUTONUM');
  ParamByName('AUTONUM').AsInteger:= AutoNum;
```

```
  ExecSQL;
 finally
  Free;
 end;
end;

procedure TfrmAutomobile.btnOkClick(Sender: TObject);
begin
 qryAuto.ApplyUpdates;
end;

procedure TfrmAutomobile.FormCreate(Sender: TObject);
begin
 qryGasType.Open;
end;

end.
```

Довідник причепів

```
unit Trailer;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  DBCtrls, StdCtrls, SohoBtns, ExtCtrls, Db, DBTables, Placemnt, Grids,
  DBGrids, RXDBCtrl, SoDBGrid;

type
  TfrmTrailer = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    btnClose: TsohoBitBtn;
    btnSelect: TsohoBitBtn;
    dbnTrailer: TDBNavigator;
    FormStorage: TFormStorage;
    qryTrailers: TQuery;
    dsrcTrailer: TDataSource;
    UpdateTrailer: TUpdateSQL;
    qryTrailersNumber: TIntegerField;
    qryTrailersStateNo: TStringField;
    dbgTrailers: TsohoDBGrid;
    qryTrailersNote: TStringField;
    procedure qryTrailersAfterPost(DataSet: TDataSet);
    procedure dbgTrailerDblClick(Sender: TObject);
    procedure dbgTrailersKeyPress(Sender: TObject; var Key: Char);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  frmTrailer: TfrmTrailer;
  MySelect: Boolean;
  function ShowTrailer(selected: Boolean; var N: integer): Boolean;

implementation
{$R *.DFM}

function ShowTrailer(selected: Boolean; var N: integer): Boolean;
begin
 MySelect:= Selected;
```

```
 with TfrmTrailer.Create(NIL) do
 try
  btnSelect.Visible:= Selected;
  if qryTrailers.Active then qryTrailers.Close;
  qryTrailers.Open;
  if ShowModal = mrOk then
  begin
   N:= qryTrailersNumber.AsInteger;
   Result:= True;
  end
   else Result := False;
 finally
  Free;
 end;
end;

procedure TfrmTrailer.qryTrailersAfterPost(DataSet: TDataSet);
begin
 qryTrailers.ApplyUpdates;
end;

procedure TfrmTrailer.dbgTrailerDblClick(Sender: TObject);
begin
 ModalResult:= mrOk;
end;

procedure TfrmTrailer.dbgTrailersKeyPress(Sender: TObject; var Key: Char);
begin
 if (key = #13) and MySelect then ModalResult:= mrOk;
 if (Key = #27) then ModalResult:= mrCancel;
 if (Key = 'i') then Key:= 'i';
end;

end.
```

Довідник видів палива

```
unit GasTypes;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  ExtCtrls, Db, Placemnt, DBTables, StdCtrls, SohoBtns, Grids, DBGrids,
  RXDBCtrl, DBCtrls;

type
  TfrmGasTypes = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    dnbGasType: TDBNavigator;
    dbgGasType: TRxDBGrid;
    btnCancel: TsohoBitBtn;
    qryGasTypes: TQuery;
    dsrcGasTypes: TDataSource;
    UpdateSQL: TUpdateSQL;
    FormStorage: TFormStorage;
    qryGasTypesCODE: TIntegerField;
    Label1: TLabel;
    Label2: TLabel;
    qryGasTypesNAME: TStringField;
    procedure qryGasTypesAfterPost(DataSet: TDataSet);
  private
    { Private declarations }
```

```
  public
    { Public declarations }
  end;

var
  frmGasTypes: TfrmGasTypes;

  procedure ShowGasType;
implementation
{$R *.DFM}

procedure ShowGasType;
begin
 with TfrmGasTypes.Create(nil) do
 try
  if qryGasTypes.Active then qryGasTypes.Close;
  qryGasTypes.Open;
  ShowModal;
 finally
  Free;
 end;
end;
procedure TfrmGasTypes.qryGasTypesAfterPost(DataSet: TDataSet);
begin
 qryGasTypes.ApplyUpdates;
end;

end.
```

Перегляд, редагування шляхових листів та ведення обліку палива

```
unit PutList;

interface

uses
  Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,
  Grids, DBGrids, RXDBCtrl, DBCtrls, ExtCtrls, StdCtrls, Myclass, Mask,
  ToolEdit, RxLookup, Db, DBTables, RxQuery, SohoBtns, Menus, SoDBGrid, Clipbrd,
  Placemnt, SoScaleForm, RxMenus, SoCtrls;

type
  TActionType =(atNew, atCorrect, atView);

  TfrmPutList = class(TForm)
    Panel1: TPanel;
    Panel2: TPanel;
    dbnWayDent: TDBNavigator;
    dbgWayDent: TRxDBGrid;
    Panel8: TPanel;
    SwichPanel: TPanel;
    Panel4: TPanel;
    Label1: TLabel;
    Label7: TLabel;
    Label6: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    lblGosNumber: TLabel;
    lblMark: TLabel;
    edAutoNum: TETDBEdit;
    Panel5: TPanel;
    Label4: TLabel;
    Label19: TLabel;
    Label20: TLabel;
    lblGosTrailer: TLabel;
    edEmpNo: TETDBEdit;
    edEmpName: TETEdit;
    Panel6: TPanel;
    Label5: TLabel;
    Label8: TLabel;
    deOutDate: TETDBDateEdit;
    deInDate: TETDBDateEdit;
    Panel7: TPanel;
    GroupBox1: TGroupBox;
    Label9: TLabel;
    Label10: TLabel;
    Label11: TLabel;
    Label12: TLabel;
    Label13: TLabel;
    Label16: TLabel;
    Label17: TLabel;
    Label18: TLabel;
    GroupBox2: TGroupBox;
    Label14: TLabel;
    Label15: TLabel;
    edSpeedOut: TETDBEdit;
    edSpeedIn: TETDBEdit;
    RadioGroup: TRadioGroup;
    btnCancel: TsohoBitBtn;
    edTrailer: TETDBEdit;
    edPack: TETDBEdit;
    edListNum: TETMaskEdit;
    FormStorage: TFormStorage;
```

```
dbgWayKeyt: TRxDBGrid;
Label21: TLabel;
Label22: TLabel;
PopupMenu: TRxPopupMenu;
ViewItem: TMenuItem;
CorrectItem: TMenuItem;
N1: TMenuItem;
N2: TMenuItem;
cmbSroreMark: TETRxDBLookupCombo;
cmbCashMark: TETRxDBLookupCombo;
qryWayKeyt: TRxQuery;
dsrcWayKeyt: TDataSource;
UpdateWayKeyt: TUpdateSQL;
qryWayKeytDOCNO: TStringField;
qryWayKeytPACKNO: TIntegerField;
qryWayKeytAUTONUM: TIntegerField;
qryWayKeytTRAILER: TIntegerField;
qryWayKeytEMPNO: TStringField;
qryWayKeytDATE_OUT: TDateTimeField;
qryWayKeytDATE_IN: TDateTimeField;
qryWayKeytSPEED_OUT: TFloatField;
qryWayKeytSPEED_IN: TFloatField;
qryWayKeytCUSTOMER_GAS: TFloatField;
qryWayKeytREMAIN_GAS_OUT: TFloatField;
qryWayKeytREMAIN_GAS_IN: TFloatField;
qryWayKeytRETURN_GAS: TFloatField;
qryWayKeytRECIVED_GAS: TFloatField;
qryWayKeytRECIVED_TYPE: TIntegerField;
qryWayKeytCASH_GAS: TFloatField;
qryWayKeytCASH_TYPE: TIntegerField;
qryWayKeytGAS_NORM: TFloatField;
qryWayKeytGAS_FACT: TFloatField;
qryWayKeytDOCDATE: TDateTimeField;
qryWayDent: TQuery;
dsrcWayDent: TDataSource;
UpdateWayDent: TUpdateSQL;
qryWayDentDOCNO: TStringField;
qryWayDentB_ACCOUNT: TStringField;
qryWayDentRACE_TOTAL: TFloatField;
qryWayDentRACE_LOAD: TFloatField;
qryWayDentTRANSPORT_T: TFloatField;
qryWayDentTRANSPORT_TKM: TFloatField;
qryWayDentDOC_ID: TIntegerField;
qryWayDentCOUNT_RACE: TIntegerField;
qryExistDocNo: TQuery;
qryGetMaxID: TQuery;
qryGetMaxIDMAX_ID: TIntegerField;
qryGetFuelType: TQuery;
qryGetFuelTypeCODE: TIntegerField;
qryGetFuelTypeNAME: TStringField;
DSRCqryGetFuelType: TDataSource;
pmAutoSearch: TPopupMenu;
miAutoSearch: TMenuItem;
pmEmpNoSearch: TPopupMenu;
miEmpNoSearch: TMenuItem;
pmTrailerSearch: TPopupMenu;
miTrailerSearch: TMenuItem;
edRemainOut: TETDBEdit;
edStore: TETDBEdit;
edCash: TETDBEdit;
edClient: TETDBEdit;
edNorm: TETDBEdit;
edFact: TETDBEdit;
```

```
edRemainIn: TETDBEdit;
edReturn: TETDBEdit;
Panel3: TPanel;
dbnWayKeyt: TDBNavigator;
btnNewPack: TsohoBitBtn;
N3: TMenuItem;
miNewList: TMenuItem;
sohoLabel1: TsohoLabel;
qryWayKeytFIO: TStringField;
qryWayKeytSTATE_AUTONUM: TStringField;
qryWayKeytSTATE_TRAILER: TStringField;
qryWayKeytRECIVED_NAME: TStringField;
qryWayKeytCASH_NAME: TStringField;
qryWayKeytBRAND: TStringField;
miNewPack: TMenuItem;
lblProbeg: TsohoLabel;
scrDelete: TSQLScript;
procedure RadioGroupClick(Sender: TObject);
procedure ViewItemClick(Sender: TObject);
procedure PopupMenuPopup(Sender: TObject);
procedure qryWayKeytAfterOpen(DataSet: TDataSet);
procedure FormActivate(Sender: TObject);

procedure ClearString;
function ExistDocNo(DocNo: string):Boolean;

procedure qryWayKeytAfterPost(DataSet: TDataSet);
procedure qryWayKeytBeforePost(DataSet: TDataSet);
procedure qryWayDentAfterPost(DataSet: TDataSet);
procedure FormCreate(Sender: TObject);
procedure qryWayKeytAUTONUMValidate(Sender: TField);
procedure qryWayKeytEMPNOValidate(Sender: TField);
procedure qryWayKeytTRAILERValidate(Sender: TField);
procedure qryWayDentB_ACCOUNTValidate(Sender: TField);
procedure miAutoSearchClick(Sender: TObject);
procedure miEmpNoSearchClick(Sender: TObject);
procedure miTrailerSearchClick(Sender: TObject);
procedure dbnWayKeytBeforeAction(Sender: TObject;
  Button: TNavigateBtn);
procedure dbgWayDentKeyPress(Sender: TObject; var Key: Char);
procedure edSpeedInKeyPress(Sender: TObject; var Key: Char);
procedure qryWayKeytNewRecord(DataSet: TDataSet);
procedure qryWayKeytREMAIN_GAS_OUTChange(Sender: TField);
procedure N2Click(Sender: TObject);
procedure dbgWayDentEnter(Sender: TObject);
procedure edRemainOutKeyPress(Sender: TObject; var Key: Char);
procedure qryWayKeytEMPNOChange(Sender: TField);
procedure edPackExit(Sender: TObject);
procedure btnNewPackClick(Sender: TObject);
procedure qryWayKeytAfterInsert(DataSet: TDataSet);
procedure miNewListClick(Sender: TObject);
procedure qryWayKeytCalcFields(DataSet: TDataSet);
procedure cmbSroreMarkEnter(Sender: TObject);
procedure cmbCashMarkEnter(Sender: TObject);
procedure edStoreExit(Sender: TObject);
procedure edCashExit(Sender: TObject);
procedure dbgWayDentColExit(Sender: TObject);
procedure qryWayKeytSPEED_OUTValidate(Sender: TField);
procedure btnCancelClick(Sender: TObject);
procedure edNormExit(Sender: TObject);
procedure dbgWayDentExit(Sender: TObject);
procedure edListNumExit(Sender: TObject);
procedure edListNumKeyPress(Sender: TObject; var Key: Char);
```

```
    procedure qryWayDentBeforePost(DataSet: TDataSet);
    procedure qryWayDentRACE_LOADValidate(Sender: TField);
    procedure qryWayDentAfterInsert(DataSet: TDataSet);
  private
    { Private declarations }
  public
    { Public declarations }
     Cycle: Boolean;
     PackNo: integer;
  end;

var
  frmPutList: TfrmPutList;
  InsertDate: TDateTime;

  Push: Boolean;
  AType: TActionType;

  procedure OpenQryWayKeyt(Query: TRxQuery; Options: String);
  procedure OpenQryWayDent(Query: TQuery; DocNo: String);
  function GetQueryParams(Res: Variant; StartDate, EndDate: TDateTime): string;
{Возвращает строку параметров для qryWayKeyt}
  function InquiryParams(var Res: variant): Boolean;
  procedure ShowPutList(SDate, EDate: TDateTime; ActionType: TActionType);

implementation
uses GetPDate, StrUtils, CommUtil, AutoList, Employes, Trailer, MyCommonData,
Search, Data;
{$R *.DFM}

procedure TfrmPutList.ClearString;
begin
 edListNum.Text        := #32;
 lblGosNumber.Caption  := #32;
 lblMark.Caption       := #32;
 lblGosTrailer.Caption := #32;
 edEmpName.Text        := #32;
end;

function TfrmPutList.ExistDocNo(DocNo: string):Boolean;
begin
 with qryExistDocNo do
 begin
  if Active then Close;
  ParamByName('DOCNO').AsString:= DocNo;
  Open;
  if BOF = EOF then Result:= False
   else Result:= True;
  Close;
 end;
end;

function InquiryParams(var Res: variant): Boolean;
begin
if not
 GetAnyInfo('frmPutList',[eDate1, eDate2, eEdit1, eCodeEdit1, eCodeEdit2,
eCodeEdit3], ['Начальная дата', 'Конечная дата' + MakeStr(#32, 3),'Номер пачки',
'Автомобиль', 'Водитель' + MakeStr(#32, 5), 'Счет' + MakeStr(#32, 15)],
            ['select STATENUM from AUTOMOBILE (nolock) where AUTONUM = %AUTONUM ;
select STATENUM from AUTOMOBILE (nolock) order by STATENUM ;'+
             'select AUTONUM from AUTOMOBILE (nolock) where STATENUM = %STATENUM
;',
```

```
              'select FIO from EMPLOYES (nolock) where TABN <> 0 and tabn = %TABN ;
select FIO from EMPLOYES (nolock) where TABN <> 0 order by FIO;'+
              'select TABN from EMPLOYES (nolock) where FIO = %FIO ;',
              'select NAME from PLAN_ACCOUNT (nolock) where BS =
convert(varchar(5), %BS ) ; select NAME from PLAN_ACCOUNT (nolock) order by NAME
;'+
              'select BS from PLAN_ACCOUNT (nolock) where NAME = %NAME ;'], [true],
nil, Res)
 then  Result:= False
  else Result:= True;
end;


function GetQueryParams(Res: Variant; StartDate, EndDate: TDateTime): string;
var
 Options, TabN, AutoN, Account, PackN: string;
 SearchSDate, SearchEDate: TDateTime;
begin
 Options:='';
 SearchSDate:= GetEditValue(Res, eDate1);
 SearchEDate:= GetEditValue(Res, eDate2);
 PackN:= ExtractWord(1, GetEditValue(Res, eEdit1), [';']);
 AutoN:= ExtractWord(1, GetEditValue(Res, eCodeEdit1), [';']);
 TabN:= ExtractWord(1, GetEditValue(Res, eCodeEdit2), [';']);
 Account:= ExtractWord(1, GetEditValue(Res, eCodeEdit3), [';']);

 if (SearchSDate <> 0 ) and (SearchEDate <> 0) then  Options:= Options + ' DOCDATE
BETWEEN ' +
  #39+ FormatDateTime('yyyy',SearchSDate)+ FormatDateTime('mm',SearchSDate)+
FormatDateTime('dd',SearchSDate)+#39 +
  ' AND ' +
  #39+ FormatDateTime('yyyy',SearchEDate)+ FormatDateTime('mm',SearchEDate)+
FormatDateTime('dd',SearchEDate)+#39;
   if (Length(Options) > 0) and (Length(TabN) > 0 )then Options:= Options + '
AND';
 if Length(TabN) > 0 then Options:= Options + ' EMPNO = ' + #39 + MakeNils(TabN,
4) + #39;
  if (Length(Options) > 0) and (Length(Account) > 0 )then Options:= Options + '
AND';
 if Length(Account) > 0 then Options:= Options + ' DOCNO IN (SELECT DOCNO FROM
WAYDENT (nolock) '+
                                                 'WHERE B_ACCOUNT = ' + #39 +
Account + #39 + ')';
  if (Length(Options) > 0) and (Length(AutoN) > 0) then Options:= Options + '
AND';
 if Length(AutoN) > 0 then Options:= Options + ' AUTONUM = ' + AutoN;
  if (Length(Options) > 0) and (Length(PackN) > 0) then Options:= Options + '
AND';
 if Length(PackN) > 0 then Options:= Options + ' PACKNO = ' + PackN;
 if Length(Options) > 0 then  Result:= Options
  else Result:= ' DOCDATE BETWEEN ' + #39+ FormatDateTime('yyyy',StartDate)+
FormatDateTime('mm',StartDate)+ FormatDateTime('dd',StartDate)+#39 +
  ' AND ' +  #39+ FormatDateTime('yyyy',EndDate)+ FormatDateTime('mm',EndDate)+
FormatDateTime('dd',EndDate)+#39;
end;

procedure OpenQryWayKeyt(Query: TRxQuery; Options: String);
var
 MacroName: string[7];
begin
 MacroName:= 'OPTIONS';
 with Query do
 begin
```

```
  if Active then Close;
  MacroByName(MacroName).AsString:= Options;
  Open;
 end;
end;

procedure OpenQryWayDent(Query: TQuery; DocNo: String);
var
 Param: string[5];
begin
 Param:= 'DOCNO';
 with Query do
 begin
  if Active then Close;
  ParamByName(Param).AsString:= DocNo;
  Open;
 end;
end;

procedure ShowPutList(SDate, EDate: TDateTime; ActionType: TActionType);
var
 Res: Variant;
begin
 AType:= ActionType;
 InsertDate:= SDate;
 if AType = atView then
  if not InquiryParams(Res) then Exit;
 with TfrmPutList.Create(nil) do
 try
  case AType of
   atView:           begin
                      RadioGroup.ItemIndex:= 0;
                      OpenQryWayKeyt(qryWayKeyt, GetQueryParams(Res, SDate,
EDate));
                     end;
   atCorrect:        begin
                      RadioGroup.ItemIndex:= 1;
                      ClearString;
                     end;
   atNew:            begin
                      OpenQryWayKeyt(qryWayKeyt, ' 0 = 1');
                      qryWayKeyt.Insert;
                      ClearString;
                      RadioGroup.ItemIndex:= 1;
                     end;
  end; {case}
  RadioGroupClick(RadioGroup);

  ShowModal;
 finally
  Free;
 end;
end;

procedure TfrmPutList.RadioGroupClick(Sender: TObject);
begin
 PopupMenuPopup(PopupMenu);
 case RadioGroup.ItemIndex of
  0: SwichPanel.SendToBack;
  1: SwichPanel.BringToFront;
 end; {case}
end;
```

```
procedure TfrmPutList.ViewItemClick(Sender: TObject);
begin
 RadioGroup.ItemIndex := TMenuItem(Sender).Tag;
end;

procedure TfrmPutList.PopupMenuPopup(Sender: TObject);
begin
 ViewItem.Checked:= RadioGroup.ItemIndex = 0;
 CorrectItem.Checked:= RadioGroup.ItemIndex = 1;
end;

procedure TfrmPutList.qryWayKeytAfterOpen(DataSet: TDataSet);
begin
 edListNum.Text:= qryWayKeytDOCNO.AsString;
 OpenQryWayDent(qryWayDent, qryWayKeytDOCNO.AsString);

 qryWayKeytSPEED_OUTValidate(qryWayKeytSPEED_OUT);

 if qryWayKeytAUTONUM.AsInteger > 0 then lblGosNumber.Caption:=
GetStateAutoNum(qryWayKeytAUTONUM.AsInteger)
  else lblGosNumber.Caption:= #32;
 if qryWayKeytAUTONUM.AsInteger > 0 then lblMark.Caption:=
GetAutoMark(qryWayKeytAUTONUM.AsInteger)
  else lblMark.Caption:= #32;
 if qryWayKeytEMPNO.AsString <> '' then edEmpName.Text:=
GetEmployes(qryWayKeytEMPNO.AsInteger)
   else edEmpName.Text:= #32;
 if qryWayKeytTRAILER.AsInteger > 0 then lblGosTrailer.Caption:=
GetStateTrailerNum(qryWayKeytTRAILER.AsInteger)
   else lblGosTrailer.Caption:= #32;
end;

procedure TfrmPutList.FormActivate(Sender: TObject);
begin
  case AType of
   atView:    dbgWayKeyt.SetFocus;
   atCorrect: edListNum.SetFocus;
   atNew:     edPack.SetFocus;
  end; {case}
end;

procedure TfrmPutList.qryWayKeytAfterPost(DataSet: TDataSet);
begin
 if MakeNils(qryWayKeytDOCNO.AsString, 7) <> '0000000' then
  if PackNo = 0 then
   begin
    PackNo:= qryWayKeytPACKNO.AsInteger;
    btnNewPack.Enabled:= True;
   end;
  if MakeNils(qryWayKeytDOCNO.AsString, 7) <> '0000000' then
qryWayKeyt.ApplyUpdates
   else qryWayKeyt.Cancel;
end;

procedure TfrmPutList.qryWayKeytBeforePost(DataSet: TDataSet);
begin
 if AType = atNew then
 begin
  qryWayKeytDOCDATE.AsDateTime:= InsertDate;
  qryWayKeytDOCNO.AsString:= MakeNils(edListNum.Text, 7);
 end;
end;
```

```
procedure TfrmPutList.qryWayDentAfterPost(DataSet: TDataSet);
begin
 qryWayDent.ApplyUpdates;
 if qryWayDentB_ACCOUNT.IsNull and qryWayDentRACE_TOTAL.IsNull and
    qryWayDentRACE_LOAD.IsNull and qryWayDentTRANSPORT_T.IsNull and
    qryWayDentTRANSPORT_TKM.IsNull and qryWayDentCOUNT_RACE.IsNull then
    qryWayDent.Delete;
end;

procedure TfrmPutList.FormCreate(Sender: TObject);
begin
 Cycle:= False; PackNo:= 0; Push:= False;
 lblProbeg.Caption:= '';
 btnNewPack.Enabled:= False;
 if not qryGetFuelType.Active then qryGetFuelType.Open;
end;

procedure TfrmPutList.qryWayKeytAUTONUMValidate(Sender: TField);
begin
 if (not qryWayKeytAUTONUM.IsNull) and (qryWayKeytAUTONUM.AsInteger <> 0) then
 begin
  if not ExistAutoNo(qryWayKeytAUTONUM.AsInteger) then
   raise Exception.Create('Автомобиль с номером ' + qryWayKeytAUTONUM.AsString + '
не существует!')
  else
   begin
    lblGosNumber.Caption:= GetStateAutoNum(qryWayKeytAUTONUM.AsInteger);
    lblMark.Caption:= GetAutoMark(qryWayKeytAUTONUM.AsInteger);
   end;
 end;
end;

procedure TfrmPutList.qryWayKeytEMPNOValidate(Sender: TField);
begin
 if (not qryWayKeytEMPNO.IsNull) and (qryWayKeytEMPNO.AsString <> '') then
 begin
  if not ExistEmpNo(qryWayKeytEMPNO.AsInteger) then
   raise Exception.Create('Водитель с номером ' + qryWayKeytEMPNO.AsString + ' не
существует!')
  else edEmpName.Text:= GetEmployes(qryWayKeytEMPNO.AsInteger);
 end;
end;

procedure TfrmPutList.qryWayKeytTRAILERValidate(Sender: TField);
begin
 if not(qryWayKeytTRAILER.IsNull) then
 if not ExistTrailer(qryWayKeytTRAILER.AsInteger) then
  raise Exception.Create('Прицеп с номером ' + qryWayKeytEMPNO.AsString + ' не
существует!')
 else lblGosTrailer.Caption:= GetStateTrailerNum(qryWayKeytTRAILER.AsInteger);
end;

procedure TfrmPutList.qryWayDentB_ACCOUNTValidate(Sender: TField);
begin
 if not ExistAccount(qryWayDentB_ACCOUNT.AsString) then
  raise Exception.Create('Счет ' + qryWayDentB_ACCOUNT.AsString + ' не
существует!')
end;

procedure TfrmPutList.miAutoSearchClick(Sender: TObject);
var
 AutoNum: Integer;
begin
```

```
 if ShowAutomobile(True,AutoNum) then
 begin
  if not (qryWayKeyt.State in [dsInsert, dsEdit]) then qryWayKeyt.Edit;
  qryWayKeytAUTONUM.AsInteger:= AutoNum;
  edEmpNo.SetFocus;
 end;
end;

procedure TfrmPutList.miEmpNoSearchClick(Sender: TObject);
begin
 if EmployeesShow(True) then
 with TClipboard.Create do
  try
   if not (qryWayKeyt.State in [dsInsert, dsEdit]) then qryWayKeyt.Edit;
   qryWayKeytEMPNO.AsString:= MakeNils(AsText,4);
   edTrailer.SetFocus;
  finally
   Free;
  end;
end;

procedure TfrmPutList.miTrailerSearchClick(Sender: TObject);
var
 Trailer: integer;
begin
 if ShowTrailer(True, Trailer) then
 begin
  if not (qryWayKeyt.State in [dsInsert, dsEdit]) then qryWayKeyt.Edit;
  qryWayKeytTRAILER.AsInteger:= Trailer;
  deOutDate.SetFocus;
 end;
end;

procedure TfrmPutList.dbnWayKeytBeforeAction(Sender: TObject;
  Button: TNavigateBtn);
begin
 if edListNum.Text = '0000000' then edListNum.Text:= #32;

 if (Button = nbInsert) and (Sender = dbnWayKeyt) then  AType := atNew;

 if (Button = nbCancel) then AType:= atView;
 if Button in [nbEdit, nbInsert] then
  begin
  RadioGroup.ItemIndex:= 1;
  RadioGroupClick(RadioGroup);
  if (Sender = dbnWayKeyt) and (PackNo = 0) then edPack.SetFocus;
  if (Sender = dbnWayKeyt) and (PackNo <> 0) then
  begin
   edListNum.SetFocus;
  end;
  if Sender = dbnWayDent then dbgWayDent.SetFocus;
 end;

  if (Sender = dbnWayKeyt) and (Button = nbDelete) then
  try
  if MessageDlg('Вы действительно хотите удалить лист?', mtConfirmation, [mbYes,
mbNo], 0) = mrYes then
  begin
   scrDelete.ParamByName('DOCNO').AsString := qryWayKeytDOCNO.AsString;
   scrDelete.ExecSQL;
  end;
  except
   ShowMessage('Путевой лист не удален');
```

```
  end;
end;

procedure TfrmPutList.dbgWayDentKeyPress(Sender: TObject; var Key: Char);
begin
 if key = #13 then NextColumn(dbgWayDent, []);
 Key:= ChangeFracDiv(Key);
end;

procedure TfrmPutList.edSpeedInKeyPress(Sender: TObject; var Key: Char);
begin
 Key:= ChangeFracDiv(Key);
end;

procedure TfrmPutList.qryWayKeytNewRecord(DataSet: TDataSet);
begin
  AType:= atNew;
end;

procedure TfrmPutList.qryWayKeytREMAIN_GAS_OUTChange(Sender: TField);
begin
 if qryWayKeyt.State in [dsInsert, dsEdit] then
 qryWayKeytGAS_FACT.AsFloat:= qryWayKeytREMAIN_GAS_OUT.AsFloat +
                              qryWayKeytRECIVED_GAS.AsFloat +
                              qryWayKeytCASH_GAS.AsFloat +
                              qryWayKeytCUSTOMER_GAS.AsFloat -
                              qryWayKeytREMAIN_GAS_IN.AsFloat -
                              qryWayKeytRETURN_GAS.AsFloat;
end;

procedure TfrmPutList.N2Click(Sender: TObject);
var
 Res: Variant;
begin
 if qryWayKeyt.State in [dsEdit, dsInsert] then qryWayKeytAfterPost(qryWayKeyt);
 if qryWayDent.State in [dsEdit, dsInsert] then qryWayDentAfterPost(qryWayDent);
 AType := atView;
 if InquiryParams(Res) then
 begin
  OpenQryWayKeyt(qryWayKeyt, GetQueryParams(Res,
dmAutomobile.qryGetConfigDataSTARTDATE.AsDateTime,
dmAutomobile.qryGetConfigDataENDDATE.AsDateTime));
  SwichPanel.SendToBack;
 end;
end;

procedure TfrmPutList.dbgWayDentEnter(Sender: TObject);
begin
 if qryWayKeyt.State in [dsInsert, dsEdit] then
 begin
  qryWayKeyt.Post;
  AType:= atView;
 end;
end;

procedure TfrmPutList.edRemainOutKeyPress(Sender: TObject; var Key: Char);
begin
 Key:= ChangeFracDiv(Key);
end;

procedure TfrmPutList.qryWayKeytEMPNOChange(Sender: TField);
begin
  if (qryWayKeytEMPNO.AsString <> '') and (not qryWayKeytEMPNO.IsNull) then
```

```
  begin
   if cycle then Exit;
   try
    cycle:= True;
    qryWayKeytEMPNO.AsString:= MakeNils(qryWayKeytEMPNO.AsString, 4);
   finally
    cycle:= False;
   end;
  end;
end;

procedure TfrmPutList.edPackExit(Sender: TObject);
begin
// if (AType = atCorrect) then edAutoNum.SetFocus;
end;

procedure TfrmPutList.btnNewPackClick(Sender: TObject);
begin
 if qryWayKeyt.State in [dsInsert, dsEdit] then qryWayKeytAfterPost(qryWayKeyt);
 if qryWayDent.State in [dsInsert, dsEdit] then qryWayDentAfterPost(qryWayDent);
 if not (qryWayKeyt.State in [dsInsert, dsEdit]) then
 begin
  SwichPanel.BringToFront;
  qryWayKeyt.Insert;
  PackNo:= 0;
  edPack.SetFocus;
 end
 else MessageDlg('Сначала сохраните текущую запись.', mtError, [mbOk], 0);

end;

procedure TfrmPutList.qryWayKeytAfterInsert(DataSet: TDataSet);
begin
 if PackNo<> 0 then qryWayKeytPACKNO.AsInteger:= PackNo
  else qryWayKeytPACKNO.Clear;
end;

procedure TfrmPutList.miNewListClick(Sender: TObject);
begin
 if qryWayDent.State in [dsInsert, dsEdit] then qryWayDentAfterPost(qryWayDent);
 dbnWayKeyt.BtnClick(nbInsert);
end;

procedure TfrmPutList.qryWayKeytCalcFields(DataSet: TDataSet);
begin
  qryWayKeytFIO.AsString:= GetEmployes(StrToIntDef(qryWayKeytEMPNO.AsString,
999999));
  qryWayKeytSTATE_AUTONUM.AsString:= GetStateAutoNum(qryWayKeytAUTONUM.AsInteger);
  qryWayKeytBRAND.AsString:= GetAutoMark(qryWayKeytAUTONUM.AsInteger);
  qryWayKeytSTATE_TRAILER.AsString:=
GetStateTrailerNum(qryWayKeytTRAILER.AsInteger);

end;

procedure TfrmPutList.cmbSroreMarkEnter(Sender: TObject);
begin
 cmbSroreMark.DropDown;
end;

procedure TfrmPutList.cmbCashMarkEnter(Sender: TObject);
begin
 cmbCashMark.DropDown;
end;
```

```
procedure TfrmPutList.edStoreExit(Sender: TObject);
begin
 if qryWayKeytRECIVED_GAS.AsFloat = 0 then edCash.SetFocus;
end;

procedure TfrmPutList.edCashExit(Sender: TObject);
begin
 if qryWayKeytCASH_GAS.AsFloat = 0 then edClient.SetFocus;
end;

procedure TfrmPutList.dbgWayDentColExit(Sender: TObject);
begin
 if (dbgWayDent.Col = 1) and
    ((qryWayDentCOUNT_RACE.AsString <> '') or (qryWayDentCOUNT_RACE.AsInteger <>
1)) then
 begin
  if not (qryWayDent.State in [dsEdit, dsInsert])  then qryWayDent.Edit;
  qryWayDentRACE_TOTAL.AsFloat:= qryWayKeytSPEED_IN.AsFloat -
qryWayKeytSPEED_OUT.AsFloat;
 end;
{
 if ((qryWayDentCOUNT_RACE.AsString = '') or (qryWayDentCOUNT_RACE.AsInteger = 1))
and
    ((qryWayKeytSPEED_IN.AsFloat - qryWayKeytSPEED_OUT.AsFloat) > 0)then
  qryWayDentRACE_TOTAL.AsFloat:= qryWayKeytSPEED_IN.AsFloat -
qryWayKeytSPEED_OUT.AsFloat;
 }
end;

procedure TfrmPutList.qryWayKeytSPEED_OUTValidate(Sender: TField);
begin
 lblProbeg.Caption:= Format('Пробег:  %-10.2f км',[qryWayKeytSPEED_IN.AsFloat -
qryWayKeytSPEED_OUT.AsFloat]);
end;

procedure TfrmPutList.btnCancelClick(Sender: TObject);
begin
 if qryWayKeyt.State in [dsEdit, dsInsert] then qryWayKeyt.Post;
 if qryWayDent.State in [dsEdit, dsInsert] then qryWayDent.Post;
end;

procedure TfrmPutList.edNormExit(Sender: TObject);
begin
// qryWayDent.Insert;
// dbgWayDent.SetFocus;
   qryWayDentCOUNT_RACE.FocusControl;

end;

procedure TfrmPutList.dbgWayDentExit(Sender: TObject);
begin
 if qryWayDent.State in [dsInsert, dsEdit] then qryWayDent.ApplyUpdates;
end;

procedure TfrmPutList.edListNumExit(Sender: TObject);
begin
 edListNum.Text:= MakeNils(edListNum.Text ,7);
 if AType = atNew then
 begin
  if ExistDocNo(edListNum.Text) then
   begin
    OpenQryWayKeyt(qryWayKeyt, ' DOCNO = ' + #39 + edListNum.Text + #39);
```

```
    AType:= atView;
   end;
 end;
 if (AType = atCorrect) and (edListNum.Text <>'0000000' ) then
 begin
  if qryWayDent.Active then qryWayDent.Close;
  if qryWayKeyt.Active then qryWayKeyt.Close;
  OpenQryWayKeyt(qryWayKeyt, ' DOCNO = ' + #39 + edListNum.Text + #39);
  if qryWayKeyt.RecordCount = 0 then TWinControl(Sender).SetFocus
//   else begin push := True; TWinControl(Sender).SetFocus end;
 end;
 {if Push then
  begin
   TWinControl(Sender).SetFocus;
   Push:=False;
  end;}

end;

procedure TfrmPutList.edListNumKeyPress(Sender: TObject; var Key: Char);
begin
{ if (Key = #13) and edListNum.Modified then
  begin
   edListNum.Text:= MakeNils(edListNum.Text ,7);
   if AType = atNew then
   begin
    if ExistDocNo(edListNum.Text) then
     begin
      OpenQryWayKeyt(qryWayKeyt, ' DOCNO = ' + #39 + edListNum.Text + #39);
      AType:= atView;
     end;
   end;

   if (AType = atCorrect) and (edListNum.Text <>'0000000' ) then
   begin
    OpenQryWayKeyt(qryWayKeyt, ' DOCNO = ' + #39 + edListNum.Text + #39);
    if qryWayKeyt.RecordCount = 0 then TWinControl(Sender).SetFocus
   end;
  end
    else Push:=False;      }
end;

procedure TfrmPutList.qryWayDentBeforePost(DataSet: TDataSet);
begin
 if qryWayDentTRANSPORT_TKM.AsFloat <> 0 then
  qryWayDentRACE_LOADValidate(qryWayDentRACE_LOAD);
end;

procedure TfrmPutList.qryWayDentRACE_LOADValidate(Sender: TField);
 var
  CountRace: integer;
begin
 CountRace:= qryWayDentCOUNT_RACE.AsInteger;
 if CountRace = 0 then CountRace:= 1;
 if qryWayKeytSPEED_IN.IsNull and qryWayKeytSPEED_OUT.IsNull then
qryWayDentTRANSPORT_TKM.AsFloat:= 0
  else //qryWayDentTRANSPORT_TKM.AsFloat:=
RoundTo(qryWayDentTRANSPORT_T.AsFloat*(qryWayKeytSPEED_IN.AsFloat -
qryWayKeytSPEED_OUT.AsFloat)/CountRace, 2);
       qryWayDentTRANSPORT_TKM.AsFloat:=
RoundTo(qryWayDentTRANSPORT_T.AsFloat*qryWayDentRACE_LOAD.AsFloat/CountRace, 2);
end;
```

```
procedure TfrmPutList.qryWayDentAfterInsert(DataSet: TDataSet);
begin
 with qryGetMaxID do
 begin
  if Active then Close;
  Open; qryWayDentDOC_ID.AsInteger:= qryGetMaxIDMAX_ID.AsInteger + 1; Close;
 end;
 qryWayDentDOCNO.AsString:= qryWayKeytDOCNO.AsString;
 qryWayDentCOUNT_RACE.FocusControl;
end;

end.
```