

Міністерство освіти і науки України
Одеський національний технологічний університет
Кафедра комп'ютерної інженерії



**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО КВАЛІФІКАЦІЙНОЇ РОБОТИ**

на тему Розробка автоматизованої інформаційної
(назва кваліфікаційної роботи згідно наказу ОНТУ)
системи станції технічного обслуговування

Здобувача Шубенкова А.Д.
(прізвище, ініціали)

4 курсу 541a групи

Керівник: ст.викл. Бондаренко В.Г.
(посада, прізвище та ініціали)

Консультанти: д.е.н., проф. Басюркіна Н.Й.
(посада, прізвище та ініціали)

доцент, ст. викл. Сахарова С.В.
(посада, прізвище та ініціали)

Кваліфікаційна робота допускається до захисту

Рішення кафедри від 10.06 2023 р., протокол № 8

Завідувач кафедри комп. інженерії Сергій АРТЕМЕНКО
(назва кафедри) (підпис) (Ім'я ПРІЗВИЩЕ)

Одеса - 2023 рік

ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет комп'ютерної інженерії, програмування та кіберзахисту
Кафедра комп'ютерної інженерії
Ступінь вищої освіти бакалавр
Спеціальність 123 «Комп'ютерна інженерія»
Освітня програма Мережеві технології та інтернет речей

ЗАТВЕРДЖУЮ

Зав. кафедри комп'ютерної інженерії
Сергій АРТЕМЕНКО
« 10 » квітня 2023 року

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ ЗДОБУВАЧА

Шубенкову Акіму Дмитровичу

1. Тема роботи Розробка автоматизованої інформаційної системи станції технічного обслуговування

Затверджена наказом університету від « 10 » квітня 2023 р., наказ № 146-03

2. Термін здачі здобувачем закінченої роботи 5 червня 2023 р.

3. Вихідні дані роботи

Розробити дизайн і навігацію сайту, структуру реляційної бази, програмне забезпечення клієнтської і серверної частин системи, використовувати комп'ютеризовані системи обробки та пошуку інформації

4. Перелік питань, які потрібно розробити

1. Порівняльний аналіз існуючих подібних інформаційно-довідкових систем

2. Аналіз і вибір засобів підтримки проектування та розробки інформаційно-довідк. системи СТО

3. Розробка архітектури інформаційно-довідкової системи СТО

4. Розробка структури бази даних інформаційно-довідкової системи СТО

5. Розробка інтерфейсу користувача інформаційно-довідкової системи СТО

6. Розробка програмного забезпечення серверної і клієнтської частин проекту

7. Техніко-економічний розрахунок проекту 8. Охорона праці 9. Загальні висновки

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

Слайд 1. Мета, предмет, об'єкт. Слайд 2. Існуючі системи. Слайд 3. Порівняння

моделей. Слайд 4. Трирівнева модель. Слайд 5. Препроцесор РНР. Слайд 6. Типи зв'язку

Слайд 6. Структура бази даних. Слайд 7. Загальний алгоритм. Слайд 8. Економіка

6. Консультанти по роботі, із зазначенням розділів роботи, що стосуються їх

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
<i>Економіка</i>	<i>Басюркіна Н.Й., д.е.н., проф.</i>		
<i>Охорона праці</i>			
<i>Нормоконтроль</i>			

7. Дата видачі завдання 17.04.2023

Керівники _____ *Валерій БОНДАРЕНКО*
 _____ *Світлана САХАРОВА*
 Завдання прийняв до виконання _____ *АКІМ ШУБЕНКОВ*

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1.	<i>Порівняльний аналіз існуючих подібних інформаційно-довідкових систем</i>	<i>13.03.2023</i>	
2.	<i>Аналіз і вибір існуючих засобів підтримки проектування та розробки інформаційно-довідкової системи СТО</i>	<i>27.03.2023</i>	
3.	<i>Проектування загальної архітектури інформаційно-довідкової системи СТО</i>	<i>03.04.2023</i>	
4.	<i>Проектування бази даних інформаційно-довідкової системи СТО</i>	<i>17.04.2023</i>	
5.	<i>Проектування інтерфейсу користувача інформаційно-довідкової системи СТО</i>	<i>01.05.2023</i>	
6.	<i>Проектування програмного забезпечення користувача інформаційно-довідкової системи</i>	<i>08.05.2023</i>	
7.	<i>Оформлення розрахунково-пояснювальної записки</i>	<i>15.05.2023</i>	
8.	<i>Підготовка графічного матеріалу</i>	<i>01.06.2023</i>	

Керівники роботи _____ *Валерій БОНДАРЕНКО*
 _____ *Світлана САХАРОВА*

Несу відповідальність за ідентичність електронного та друкованого варіантів кваліфікаційної роботи, даю згоду на обробку персональних даних та не заперечую проти розміщення кваліфікаційної роботи на офіційних web-ресурсах ОНТУ.

Підтверджую, що в кваліфікаційній роботі відсутні порушення норм академічної доброчесності.

Здобувач - дипломник _____ *Акім ШУБЕНКОВ*

АНОТАЦІЯ

Метою розробки інформаційно-довідкової системи автосалону є підвищення ефективності обліку автосалону і якісне поліпшення обслуговування (скорочення часу обробки оперативних даних, зменшення кількості помилок при обробці інформації).

Об'єктом проектування можна вважати процес обслуговування автосалону, а предметом проектування - програмний засіб, що дозволяє автоматизувати процес обслуговування автосалону.

Виконана робота складається з декомпозиції задачі, аналізу існуючих і розробки власних методів вирішення виділених підзадач, виділення окремих компонентів системи у вигляді функціонально закінчених програмних модулів.

Розроблена система значно підвищує ефективність обліку і якісно покращує обслуговування (скорочення часу обробки оперативних даних, зменшення кількості помилок при обробці інформації).

ABSTRACT

The goal of developing a car dealership information and reference system is to increase the efficiency of car dealership accounting and to improve service quality (reducing the processing time of operational data, reducing the number of errors in processing information).

The object of design can be considered the process of servicing a car dealership, and the subject of design is a software tool that automates the process of servicing a car dealership.

The work done consists of decomposing the task, analyzing existing ones and developing their own methods for solving the selected subtasks, highlighting the individual components of the system in the form of functionally complete software modules.

The developed system significantly increases the efficiency of accounting and qualitatively improves service (reducing the processing time of operational data, reducing the number of errors in processing information).

ЗМІСТ

ВСТУП.....	7
2 ПРОЕКТУВАННЯ ЗАГАЛЬНОЇ АРХІТЕКТУРИ.....	28
3 АНАЛІЗ І ВИБІР ЗАСОБІВ ПІДТРИМКИ ПРОЕКТУВАННЯ ТА РОЗРОБКИ СИСТЕМИ.....	36
3.1 ВИБІР СЕРВЕРНОЇ МОВИ СЦЕНАРІЇВ І ЗАСОБУ ПРОЕКТУВАННЯ <i>WEB</i> –САЙТУ ..	36
3.2 ВИБІР ВЕБ–СЕРВЕРА	42
3.3 ВСТАНОВЛЕННЯ <i>RHP</i> І НАСТРОЙКА <i>ARACHE</i> ДЛЯ РОБОТИ З <i>RHP</i>	45
4 ПРОЕКТУВАННЯ БАЗИ ДАНИХ.....	48
4.1 ОСНОВИ РОЗРОБКИ РЕЛЯЦІЙНОЇ БАЗИ ДАНИХ.....	48
4.2 ВИЗНАЧЕННЯ ТАБЛИЦЬ. СПЕЦИФІКАЦІЇ.....	56
5 ПРОЕКТУВАННЯ ПРОГРАМНИХ КОМПОНЕНТІВ.....	60
5.1 ЗАГАЛЬНИЙ АЛГОРИТМ.....	60
5.2 ДОПОМІЖНІ ВІКНА	67
6 ЕКОНОМІЧНА ЧАСТИНА	75
6.1 ОРГАНІЗАЦІЙНО-ЕКОНОМІЧНЕ І МАРКЕТИНГОВЕ ОБҐРУНТУВАННЯ ПРОЕКТУ .	75
6.2 ЕКОНОМІЧНІ РОЗРАХУНКИ ПРОЕКТУ.....	82
7 ОХОРОНА ПРАЦІ НА ПІДПРИЄМСТВІ	92
7.1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ОХОРОНУ ПРАЦІ НА ПІДПРИЄМСТВАХ.....	92
7.2 ОСНОВНІ ЗАВДАННЯ І ФУНКЦІЇ СЛУЖБИ ОХОРОНИ ПРАЦІ	94
7.3 ПОВНОВАЖЕННЯ СЛУЖБИ ОХОРОНИ ПРАЦІ.....	96
ВИСНОВКИ	98
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	99

					<i>КРБ.КІ. 1.146-03.2.2</i>			
Змн.	Арк.	№ докум.	Підпис	Дата				
Розроб.		Шубенков А.Д.			<i>Розробка автоматизованої інформаційної системи станції технічного обслуговування</i>	Літ.	Арк.	Аркуші
Перевір.		Бондаренко ВГ					6	110
Нормоконтр						<i>ОНТУ, зр.541а</i>		
Рецензент								
Затверд.		Артеменко СВ						

ВСТУП

Практично всі системи в тій чи іншій мірі пов'язані з функціями довготривалого зберігання і обробки інформації. Збільшилися інформаційні потоки і підвищилися вимоги до швидкості обробки даних, і тепер уже більшість операцій не може бути виконано вручну, вони вимагають застосування найбільш перспективних комп'ютерних технологій. Будь-які адміністративні рішення вимагають чіткої і точної оцінки поточної ситуації та можливих перспектив її зміни. І, якщо раніше в оцінці ситуації брало участь кілька десятків чинників, які могли бути обчислені в ручну, то тепер таких факторів сотні і сотні тисяч, і ситуація змінюється не протягом року, а через кілька хвилин, а обґрунтованість прийнятих рішень потрібна велика, тому що і реакція на неправильні рішення більш серйозна, більш швидка і більш потужна, ніж раніше. І, звичайно, обійтися без інформаційної моделі виробництва, що зберігається в базі даних, в цьому випадку неможливо.

Як правило, бази даних створюються зазвичай не для вирішення якої-небудь однієї задачі для одного користувача, а для багатоцільового використання. Бази даних відображають певну частину реального світу. Ця інформація повинна по можливості фіксуватися в базі даних одноразово, і всі користувачі, яким ця інформація потрібна, повинні мати можливість працювати з нею. Відсутність централізованих методів управління доступом до інформації послужило причиною розробки систем управління базою даних, а самі сховища інформації, які працювали під управлінням даних систем, називалися базами або банками даних.

Процес автоматизації починається з середини 70-80 рр. і пов'язаний з появою першої обчислювальної техніки, яка була не досконалою в сучасному розумінні і вимагала спеціального програмного забезпечення і обслуговування - так процес автоматизації прийшов в Україну.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

В даний час процес автоматизації торкнулися не тільки виробничу, технічну і технологічну сфери діяльності людства, а й інформаційний простір; зокрема видавництва, бібліотеки, музеї, інформаційні центри. Вже складно уявити життя суспільства без таких понять як інтернет, засоби телекомунікації і т.д. Автоматизація найбільше торкнулася інформаційну сферу, так як саме інформація потребувала і потребує швидкого пошуку, відборі та зберіганні, а так само в забезпеченні доступу до неї.

Основна перевага автоматизації - це скорочення надмірності збережених даних, а отже, економія обсягу використовуваної пам'яті, зменшення витрат на багаторазові операції оновлення надлишкових копій і усунення можливості виникнення протиріч через зберігання в різних місцях відомостей про одне й те ж об'єкті, збільшення ступеня достовірності інформації і збільшення швидкості обробки інформації; зайва кількість внутрішніх проміжних документів, різних журналів, папок, заявок і т.д., повторне внесення однієї і тієї ж інформації в різні проміжні документи. Також значно скорочує час автоматичний пошук інформації, який проводиться зі спеціальних екранних форм, в яких вказуються параметри пошуку об'єкта.

Необхідність розробки програмного засобу комп'ютерної автоматизації обумовлена тенденцією зростання об'єму і звороту інформації.

24 травня 2012 року Верховна Рада України прийняла в другому читанні Закони про розвиток індустрії програмної продукції. Мова йде про законопроект [6], якій регулює проведення так званого економічного експерименту про створення сприятливих умов роботи ІТ-індустрії. Головні завдання держпідтримки – створення умов для інноваційного розвитку ІТ-індустрії в Україні, збільшення числа робочих місць у галузі, залучення інвестицій і т.д.

При створенні інформаційної системи треба керуватися принципами,

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

визначеними РД 50–680–88 «АС Основні положення» [11]: системності, розвитку (відкритості), сумісності, стандартизації (уніфікації) та ефективності.

Принцип системності: при декомпозиції мають бути встановлені такі зв'язки між структурними елементами системи, які забезпечують цілісність інформаційної системи та її взаємодію з іншими системами.

Принцип розвитку (відкритості): виходячи із перспектив розвитку об'єкта автоматизації інформаційну систему треба створювати з урахуванням можливості поповнення та оновлення функцій і складу інформаційної системи, не порушуючи її функціонування.

Принцип сумісності: при створенні систем мають бути реалізовані інформаційні інтерфейси, завдяки яким вона може взаємодіяти з іншими системами за встановленими правилами.

Принцип ефективності: досягнення раціонального співвідношення між затратами і цільовими ефектами, включаючи кінцеві результати, отримані завдяки автоматизації.

Принцип стандартизації (уніфікації): при створенні систем мають бути раціонально використані типові, уніфіковані й стандартизовані елементи, проектні рішення, пакети прикладних програм, комплекси, компоненти.

При створенні інформаційної системи треба керуватися існуючими стандартами. При створенні систем мають бути раціонально використані типові, уніфіковані й стандартизовані елементи, проектні рішення, пакети прикладних програм, комплекси, компоненти. До засадничих стандартів по розробці і функціонуванню баз банних відносяться стандарти [17-19], які визначають концепцію і термінологію для схем баз, еталонну модель управління даними, структуру систем словників інформаційних ресурсів. Стандарти [18,19] є одночасно міждержавними.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

Стандарт [17] встановлює рівень даних, засоби і інтерфейси інформаційної системи, словників інформаційних ресурсів, використовуваних для управління і документування інформаційних ресурсів наочних областей (НО). Словник-довідник інформаційних ресурсів як інфраструктура СУБД містить інформацію для одного або декількох застосувань.

Основні поняття і терміни концептуальних схем і інформаційних баз, що охоплюють розробку, опис і вживання концептуальних схем і інформаційних баз, маніпулювання даними, а також опис і реалізація обробки даних описані в [18]. У стандарті концептуальна схема розглядається як несуперечлива сукупність припущень, що виражають вислови, що відносяться до НО. Положення стандарту можна використовувати для оцінки СУБД.

Стандарт [19] встановлює еталонну модель управління даними. Еталонна модель визначає загальну термінологію і поняття, що відносяться до даних інформаційних систем. Такі поняття використовуються визначення послуг, СУБД, що надаються, або системами словників даних.

Засадничі стандарти [20-24] встановили перелік і порядок процесів життєвого циклу (ЖЦ) систем і програмного забезпечення, принципи категоризації програм визначили вимоги до супроводу програмного продукту. У [21] встановлена загальна схема ЖЦ програмних систем і вимоги до процесів і стандартів життєвого циклу систем. Стандарт [24] є керівництвом по використанню [20].

Процеси ЖЦ програмного забезпечення розділені на три групи: основні, організаційні і підтримка ЖЦ [20]. До основних віднесені: замовлення, постачання, розробка, експлуатація і супровід програмного засобу; до організаційних – управління, створення і супровід інфраструктури, навчання; до підтримки ЖЦ – документування, конфігурація, забезпечення

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

якості, верифікація і валидація, загальний передивляється, аудит.

У стандарті [22] описана схема класифікації програмних засобів по функціях програмного засобу, сфері застосування, режимі роботи, масштабі програмного засобу (малий, середній, великий), критичності, класі користувачів, доступності програмного продукту, використанні даних, необхідній продуктивності, вимогах до безпеки і надійності, комп'ютерної платформи, середовища і споживаних ресурсів.

У стандарті [23] конкретизовані положення [20], що стосуються процесу супроводу. У цьому стандарті введені різні типи супроводу – зміни, що коректують, застережливе, адаптивне і поліпшуюче.

Об'єктом проектування можна вважати процес обслуговування автосалону, а предметом проектування – програмний засіб, що дозволяє автоматизувати процес обслуговування автосалону.

Метою даного дипломного проектування є підвищення ефективності обліку автосалону і якісне поліпшення обслуговування (скорочення часу обробки оперативних даних, зменшення кількості помилок при обробці інформації). Проектування високорівневих методів інформатики і програмування працівника автосалону здійснюється з використанням реляційної бази даних.

Перспективність та актуальність тематики, пов'язаною з розробкою програмного засобу, підтверджується тим, що в Україні в 2008 році була створена Асоціація розробників електронного бізнесу [56], задача якої об'єднати всіх виробників апаратних та програмних засобів інформаційних технологій (ІТ) технологій для прогресивного розвитку галузі, а також Законами України «Про Концепцію Національної програми інформатизації» [7], «Про Національну програму інформатизації» [8], «Про основні засади розвитку інформаційного суспільства в Україні на 2007–2015 рік» [9].

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1 ПОРІВНЯЛЬНИЙ АНАЛІЗ ІСНУЮЧИХ ІНФОРМАЦІЙНО- ДОВІДКОВИХ СИСТЕМ АВТОСАЛОНІВ

На сьогоднішній день існує досить автоматизованих систем обліку автомобілів в автосалоні, але в більшості випадків АІС спроектована під конкретний автосалон, має обмежені можливості або входить до складу великого комплексу програм.

Ауді Центр Запоріжжя (<http://audizp.com/>)

В автоцентрі Ауді Центр Запоріжжя можна вибрати новий Audi, а для клієнтів, які в усьому дотримуються власного стилю, пропонуються автомобілі на замовлення в індивідуальній комплектації.

КАРТА САЙТУ:

1. Модельний ряд
2. Покупцям
3. Власникам
4. Про компанію
5. Онлайн
6. Контакти

Меню другого рівня:

1. Модельний ряд

- 1.1. Всі моделі: A4; A5; A6; A7; A8; Q2; Q3; Q4; Q5; Q6; Q7; Q8; S; RS; e-tron

(При виборі моделі можна дізнатися таку інформацію: середня витрата пального: л/100км; середні викиди CO₂: г/км)

1.2. Конфігуратор

Конфігуратор наразі доступний лише в якості інформаційного джерела. Інформація, зображення, колір та технічні характеристики автомобілів відображені на даному веб-сайті виключно у загальних цілях для зображення автомобілів та аксесуарів, які пропонуються.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Веб-сайт не може відтворити фактичний колір та дизайн. Незважаючи на усі зусилля, докладені задля забезпечення правильності та актуальності усієї інформації, не гарантується, що ця інформація є достовірною, повною, точною або виключає помилки.

1.3. Автомобілі в наявності

Характеристики моделей (тип кузова, привід, колір, лінії и т.д.) не стосуються конкретного транспортного засобу та не є частиною пропозиції, але служать лише для порівняння між різними типами автомобілів. Показники викидів CO_2 та споживання палива - значення, обчислені як частина попередньої конфігурації транспортного засобу. Не виключено, що в процесі виробництва трапляються відхилення цих значень CO_2 та споживання палива, закладених на етапі конструювання, тому значення CO_2 та споживання палива, обчислені в контексті попередньої конфігурації даних, не відповідають величинам CO_2 та витратам палива автомобіля, що купується. Витрата палива та викиди CO_2 транспортного засобу залежать не тільки від ефективного використання пального автомобіля, а й від поведінки за кермом, відстані руху та інших нетехнічних факторів. Значення показників автомобіля, що є предметом купівлі, може відхилятися від пропозиції вгору або вниз. Зокрема, деякі опції та аксесуари (наприклад, ширші шини, кондиціонер, багажник на даху, чинники аеродинаміки тощо) можуть призвести до змін витрат палива та CO_2 .

2. Покупцям

2.1. Спеціальні пропозиції

Audi Центр Запоріжжя робить виняткова пропозиція на обмежену кількість автомобілів наступних моделей:

Audi A7 Sportback; Audi Q2;

Audi Q5; Audi Q7

2.2. Кредит

Audi A7 Sportback; Audi Q2;

Audi Q5; Audi Q7

					КРБ.КІ.1.146-03.2.2	Арк.
						13
Змн.	Арк.	№ докум.	Підпис	Дата		

Кредит Нашим партнером по оформлению сделок выступает компания *Porsche Finance Group*, которая является 100% собственностью «Порше Банк», который в свою очередь является частью австрийского «Порше Холдинга», который представлен в 16 странах Европы.

Кредит от *Porsche Finance Group Ukraine* это:

- Все с одних рук: Вы общаетесь только с менеджером отдела продаж *Audi*.
- Мы работаем для всех: как для физических, так и для юридических лиц.
- Выгодные и прозрачные условия финансирования.
- Быстро: все документы Вы подаете Вашему продавцу и решение о финансировании принимается от 1-го до 3-х дней.
- Никаких дополнительных ежемесячных комиссий.
- Возможность досрочного погашения кредита без ограничений и дополнительных расходов.
- Неограниченные финансовые активы, которые можем использовать для финансирования.

2.3. Лізинг

Існує багато можливостей для того, щоб профінансувати покупку автомобіля. Найпоширеніші способи - придбати авто за готівку або взяти банківський кредит. Ще одна альтернативна можливість придбати автомобіль – це лізинг.

Значними перевагами лізингу є те, що купуючи автомобіль Ви зберігаєте значні вільні кошти і кредитні лінії. В кінці терміну дії договору лізингу право власності або повністю переходить до лізингоотримувача (фінансовий лізинг), або автомобіль просто повертається лізинговій компанії (оперативний лізинг).

					КРБ.КІ.1.146-03.2.2	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

Нашим партнером по оформленню лізингових угод виступає компанія Porsche Finance Group, яка є 100% власністю «Порше Банк», який в свою чергу є частиною австрійського «Порше Холдингу», представленого в 16 країнах Європи.

2.4. Страховання

Застрахований автомобіль - запорука Вашого спокою на дорогах! Автострахування пропонує захист Вашого автомобіля від пошкоджень або втрати, а також Страхування відповідальності водія. Ми працюємо тільки з компаніями, яким довіряємо, оскільки мова йде про спокій наших Клієнтів. Ми пропонуємо своїм клієнтам страхування авто-КАСКО на спеціальних умовах.

2.5. Пропозиції для корпоративних клієнтів

Audi постійно і активно співпрацює з корпоративними клієнтами по всій Україні. Це і міжнародні компанії, і державні структури, і дипломатичні місії, і міжнародні організації.

Audi Центр Запоріжжя пропонує клієнтам спеціальні умови:

- індивідуальні пропозиції придбання автомобіля;
- лізингові програми від нашого партнера «*Porsche Finance Group Ukraine*».
- Система *Trade-in*, завдяки якій Ви можете вирішити проблему реалізації автомобілів Вашого автопарку.
- Технічне обслуговування Ваших корпоративних автомобілів на вищому рівні сервісу.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

3. Власникам

3.1. Audi-сервіс

- сучасне діагностичне обладнання;
- повний спеціальний інструментарій;
- прямий доступ до повного асортименту оригінальних запасних частин і аксесуарів *Audi*;
- впевненість у відмінному стані протягом всього терміну експлуатації автомобіля;
- гарантії високої якості обслуговування з дотриманням всіх вимог *AUDI AG*.

3.2. Пряме приймання

- пряме приймання дає Вам можливість особисто оцінити стан автомобіля разом з майстром приймальником сервісного центру;
- під час огляду Вашого *Audi*, Ви отримаєте від майстра приймальника відповіді на питання щодо технічного стану Вашого автомобіля і докладні рекомендації щодо необхідного ремонту;
- ви не будете сумніватися в правильності рекомендацій майстра приймальника;
- за результатами діагностики, яка проходила у Вашій присутності, Ви приймаєте самостійне рішення про ремонт виявлених несправностей.

3.3. Акції та спеціальні пропозиції

3.4. Оригінальні запасні частини

3.5. Оригінальні аксесуари *Audi*

3.6. Гарантія

3.7. Програма допомоги *Audi Assistance*. Цілодобово.

Категорії послуг за програмою *Audi Assistance*:

- інформаційні послуги;
- організаційні послуги;

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

- послуги «консьєрж»;
- послуги з евакуації автомобіля в разі ДТП або несправності авто;
- технічна допомога;
- послуги з бронювання та оплати проживання в готелі і замовлення зворотного квитка;
- альтернативна евакуація.

Компанія ДП «АВТОЦЕНТР ОДЕСА» (<http://skoda.odessa.ua/>)

Новий офіційний дилер «Шкода в Одесі» представляє повну лінійку модельного ряду шкода від компактних міських авто до моделей бізнес-класу. В автосалоні Skoda на даний момент модельний ряд представлений 15 моделями: сімейний і практичний *fabia* (Шкода Фабіа), *fabia combi*, універсальний *romster*, *praktik*, молодіжний *spaceback* (спейсбек), народний автомобіль *rapid* (рапід), елегантна *octavia*, флагман модельного ряду *superb* (суперб). Skoda24

КАРТА САЙТУ:

1. Програма Шкода24.
2. Новини та акції.
3. Сервіс.
4. Послуги.
5. Тест Драйв.
6. Акційні авто.
7. Бонуси.
8. Контакти.

1. Програма Шкода24.

- Технічна консультація
- Організація технічної допомоги на дорогах
- Евакуація

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		17

2. Новини та акції

3. Сервіс

- комп'ютерна діагностика;
- комп'ютерне регулювання кутів розвалу і сходження;
- ремонт ходової частини будь-якої складності;
- капітальний ремонт двигунів;
- обслуговування кондиціонерів;
- ремонт гальмівної системи;
- проточка гальмівних дисків;
- промивання інжекторів;
- заміна технічних рідин (в тому числі вакуумна);
- обслуговування систем запалювання, перевірка, чищення свічок;
- замір компресії в циліндрах двигуна;
- шиномонтаж та балансування;
- регулювання напрямку світла фар;
- сигналізації (різної складності, з блокуванням двигуна, датчиками руху);
- додаткові датчики обсягу;
- установка електросклопід'ємників;
- центральні замки;
- датчики паркування;
- замки КПП (*MUL-T-LOCK, Bear - Lock, Construct*);
- установка радіоантен (звичайних і з підсилювачем сигналу);
- захист двигуна;
- антикорозійна обробка;
- підігрів сидінь;
- акустичні системи та автомагнітоли;
- автомобільні відео системи;

					КРБ.КІ.1.146-03.2.2	Арк.
						18
Змн.	Арк.	№ докум.	Підпис	Дата		

- тонування скла (будь-яка прозорість);
- декоративний металевий обвіс (дуги, підніжки і т.д)

4. Послуги

– автомобільний лізинг - один з найбільш затребуваних видів придбання автомобілів. При покупці в лізинг автомобіль передається у тимчасове користування клієнту на умови оплати ним лізингових платежів з можливим викупом наприкінці строку;

– *trade-in* - це спосіб задовольнити відразу два бажання: стати власником новенького *Skoda* і позбутися від старого авто. Ви оплачуєте різницю між вартістю Вашої старої машини, яку оцінюють наші фахівці, і нового автомобіля, обраного з широкого модельного ряду *Skoda*;

– різні види страхування Вашого автомобіля від провідних страхових компаній за найвигіднішими умовами.

5. Тест-драйв

Всі ми дорожимо своїм часом і фінансами - покупки автомобіля це стосується в першу чергу. Тест-драйв допомагає уникнути довгих і болісних роздумів з приводу вибору відповідної моделі та комплектації і в режимі реального часу оцінити вподобану тільки на картинці і по опису машину за наступними пунктами:

– ступінь комфорту. Посидіти в автомобілі, що стоїть в салоні - не означає відчутти ступінь зручності. А ось налаштувавши кермо і крісла під себе, оцінивши розташування кнопок і ручок, відчувши легкість ходу руки при роботі з коробкою передач, можна зрозуміти, наскільки ця машина вам підходить;

– динамічність і керованість. У кожного свій стиль водіння і свої дороги - часом вибір між потужною екстравагантної *ŠKODA SUPERB SPORTLINE* і *YETI* у вседорожній версії *OUTDOOR* цілком очевидний. Але якщо Ви на роздоріжжі, то на тест-драйві неодмінно зрозумієте, який автомобіль

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		19

найбільш успішно справляється з поставленими завданнями;

– потрібні й непотрібні опції. Можливо, Вам підходить тільки максимально насичена комплектація, але не виключено, що Ви цілком можете обійтися без будь-яких функцій і при цьому заощадити при покупці. Найчастіше для цього достатньо просто покататися на сюжеті автомобілі.

Ще один привід замовити пробну поїздку - одним з перших оцінити гарячу новинку сезону. У 2017 році такий новинкою для ŠKODA став потужний, впевнений і стильний позашляховик KODIAQ. Не відмовляйте собі в задоволенні спробувати щось нове!

Записатися на тест-драйв можна у офіційного дилера ŠKODA, зателефонувавши по телефону або залишивши заявку на сайті із зазначенням своїх даних і зручного часу для поїздки.

6. Акційні авто

Характеристики:

Рік випуску, Витрата палива, Тип кузова, Колір кузова, Місткість, Об'єм двигуна, Тип палива, Тип КПП, Тип приводу, Потужність.

7. Service bonus

Ці правила встановлюють офіційні правила участі в бонусній програмі лояльності «Service Bonus» (надалі – Правила), які визначають умови участі в бонусній програмі лояльності «Service Bonus» (надалі - Програма), а також визначають умови, період та порядок участі в Програмі.

Правила розроблені у відповідності до Цивільного кодексу України, Закону України «Про інформацію», Закону України «Про захист персональних даних», інших нормативно – правових актів України, внутрішніх положень Товариства.

8. Контакт

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

АВТОЦЕНТР КІА» (<http://kia.odessa.ua/>)

Офіційний дилер ТОВ «ФАЛЬКОН-АВТО», по автомобілям КІА в Одеській області. Концептуальний центр успішно працює з 2007 року і здійснює свою роботу в трьох напрямках: продаж автомобілів, їх гарантійне та післягарантійне обслуговування, а також продаж оригінальних запасних частин і аксесуарів.

КАРТА САЙТУ:

1. КАТАЛОГ МОДЕЛЕЙ
2. СЕРВІС
3. ІНФОРМАЦІЯ ДЛЯ КЛІЄНТІВ
4. ПРО ДИЛЕРА
5. ІНШЕ
6. НОВИНИ
7. СПЕЦІАЛЬНІ ПРОПОЗИЦІЇ
8. ЗВОРОТНІЙ ЗВ'ЯЗОК

1. КАТАЛОГ МОДЕЛЕЙ

– Легкові;

- Новий *Picanto*
- *Rio* Хетчбек
- *Rio Sedan*
- Новий *Ceed*
- Новий *Ceed SW*
- Новий *ProCeed*
- Новий *Sorento*
- *Stinger*

– Гібридні та електричні

- *Soul EV*

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

- Кросовери
 - *Stonic*
 - *Sportage FL*

2. СЕРВІС

- Записатися на сервіс
- Гарантія
- Вартість технічного обслуговування
- *KIA Assistance*

3. ІНФОРМАЦІЯ ДЛЯ КЛІЄНТІВ

- записатися на тест-драйв;
- завантажити брошуру;
- завантажити прайс;
- корпоративним клієнтам;
- *KIA* Аксесуари.

4. ПРО ДИЛЕРА

- контакти
- історія інновацій
- бренд *KIA*

5. ІНШЕ

6. НОВИНИ

7. СПЕЦІАЛЬНІ ПРОПОЗИЦІЇ

8. ЗВОРОТНІЙ ЗВ'ЯЗОК

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

«ПРЕСТИЖ-АВТО» (<http://www.prestige-auto.com.ua/>)

Товариство працює в автобізнесі більше десяти років та впевнено займає одне з провідних місць на ринку автопослуг. Компанія «Престиж-Авто» є офіційним дилером марки *Volkswagen* у Західному регіоні України та представляє на ринку України високоякісний продукт концерну *Volkswagen AG*. «Престиж-Авто» перший в Хмельницькому концептуальний автоцентр, який відповідає європейським стандартам та вимогам, як щодо будівлі автосалону, так і щодо рівня технічного оснащення. «Престиж-Авто» пропонує до вашої уваги широкий модельний ряд легкових та комерційних автомобілів *Volkswagen*. Кожен клієнт має можливість самостійно обрати модель та комплектацію за своїм бажанням, можливістю та потребам. З 2008 року партнером компанії стала найбільша фінансова група Західної Європи — це компанія Порше Фінанс Груп. Це дало можливість пропонувати клієнтам нові та більш досконалі фінансові продукти: фінансовий та оперативний лізинг, за найнижчою відсотковою ставкою та надійний кредит. Налагоджена співпраця з найбільшими та стабільними банками міста також полегшує можливість придбання бажаного *Volkswagen*. Автосалон «Престиж-Авто» це не лише шоу-руми з новенькими автомобілями, це і великий сервіс, обладнаний найновішою комп'ютерною діагностикою та всіма необхідними сучасними підйомниками, рихтувально-малярний цех та мийка. Всі клієнти мають можливість одержати кваліфікований гарантійний та післягарантійний сервіс для своїх автомобілів, компанія гарантує якість послуг, адже персонал компанії пройшов багаторівневу підготовку не лише в Україні, а і за кордоном. Фірмовий сервіс VW — це висококваліфіковані працівники та можливість отримати повний перелік послуг по обслуговуванню та діагностиці автомобілів *Volkswagen Group*. Найширший склад запасних частин та аксесуарів, дає змогу ремонтувати автомобіль без затримок та додаткових затрат з боку клієнта. «Престиж-Авто» — це не лише європейський

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

рівень обслуговування, професіональний інструментарій, кваліфіковані працівники, але перш за все — це турбота та увага до клієнта.

КАРТА САЙТУ

1. НОВИНИ
2. НОВІ АВТО
3. АВТО З ПРОБІГОМ
4. СЕРВІС АКСЕСУАРИ
5. КОМАНДА
6. ВАКАНСІЇ
7. КОНТАКТИ

«АЛЬФА-АВТО: АВТОСАЛОН + АВТОСЕРВІС + АВТОЗАПЧАСТИНИ»

Опис: Програмний продукт «Альфа-Авто: Автосалон + Автосервіс + Автозапчастини» призначений для комплексної автоматизації обліку на підприємствах автобізнесу (магазинів оптового та роздрібного продажу запчастин, автосервісів і станцій технічного обслуговування автомобілів, автосалонів і дилерських центрів).

За допомогою «Альфа-Авто: Автосалон + Автосервіс + Автозапчастини» можливо:

- оформляти замовлення та продажу автомобілів;
- виконувати передпродажну підготовку автомобілів;
- надавати послуги з ремонту та обслуговування автомобілів;
- вести оптову та роздрібну торгівлю запасними частинами;
- організувати роботу складу;
- враховувати оплати і відслідковувати стан взаєморозрахунків з покупцями і постачальниками;

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

Конфігурація дозволяє:

1. Запасні частини:

- оптовий продаж запчастин;
- роздрібний продаж запчастин;
- робота по замовленнях запчастин;
- закупівлі запчастин;
- організація внутрішньофірмового руху товару;

2. Автосервіс:

- планування ресурсів;
- оформлення ремонтів;
- автоматизоване робоче місце співробітника автосервісу;

3. Автосалон:

- клієнтські і складські замовлення;
- покупка і продаж автомобілів;
- передпродажна підготовка автомобілів;
- угоди «*Trade-in*»;
- відповідальне зберігання автомобілів;

4. Фінансовий блок:

- оплати покупців і постачальникам;
- ведення взаєморозрахунків з підзвітними особами;
- бюджетування;

5. Обмін даними:

- обмін даними з бухгалтерськими системами;
- обмін даними з каталогами виробників;

Користувачі програми мають можливість швидко формувати необхідні документи. Керівництво може оперативно отримувати і використовувати дані про різні аспекти діяльності компанії. Система надає інформацію, необхідну для прийняття управлінських рішень.

					КРБ.КІ.1.146-03.2.2	Арк.
						25
Змн.	Арк.	№ докум.	Підпис	Дата		

«Автосалон v.3»

Опис: Програма автосалону призначена для автоматизації діяльності автосалонів, які торгують автомобілями, в частині, що стосується ведення обліку, продажів автомобілів і формування договірних документів. Також передбачено формування і друк документів, специфічних для даного виду діяльності (довідка-рахунок, ПТС, транзитний номер).

Третя версія програми є продовженням серії однойменних програмних продуктів, створених фахівцями нашої компанії, і написана з урахуванням побажань і рекомендацій користувачів.

Можливості системи:

1. Друк документів на бланках суворої звітності (довідка-рахунок, ПТС, транзитний номер).
2. Необмежена кількість робочих місць в рамках однієї локальної мережі.
3. Скільки завгодно багато підприємств (торгуючих організацій).
4. Повна мультивалютність.
5. Ведення різних довідників (клієнти, автомобілі, комплектації і т.д.).
6. Ведення Складу і Журналу продажів.
7. Формування різних звітів і статистики.

Обґрунтування розробки автоматизованої інформаційної системи «інформаційно - довідкової системи автосалону»

У існуючих аналогів були виявлені такі недоліки:

- немає гнучких механізмів розмежування доступу до даних;
- недостатня функціональність;
- немає на багато користувачів доступу;
- висока ціна.

Створення власної автоматизованої системи дозволить врахувати всі особливості обліку автомобілів в автосалоні.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Очікуваний ефект від впровадження:

1. Актуальність - можливість оперативно відображати зміни.
2. Повнота інформації, тобто надання максимально повної інформації про замовлення, про оплату, про автомобілі.
3. Зручність - зрозумілий інтерфейс.
4. Несуперечність і достовірність даних.

Аналіз кожної з розглянутих систем показує надлишкову складність для їх використання, через що доведеться платити багато грошей за установку і супровід тих функцій, які є зайвими для невеликої організації. Потрібно розробити простий, зручний, недорогий у використанні програмний бібліотечний інформаційний засіб, в той же час.

Створення власної автоматизованої системи дозволить врахувати всі особливості об'єкту, для якого розробляється лише те, що потрібно, і як потрібно.

Таким чином, метою даної роботи є створення максимально простого і зручного в рамках конкретного підприємства, програмного засобу, що спрощує облік авто.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

2 ПРОЕКТУВАННЯ ЗАГАЛЬНОЇ АРХІТЕКТУРИ

Клієнт–серверна система характеризується наявністю двох взаємодіючих самостійних процесів – клієнта і сервера, які, в загальному випадку, можуть виконуватися на різних комп'ютерах, обмінюючись даними по мережі.

Процеси, що реалізують деяку службу, наприклад службу файлової системи або бази даних, називаються серверами (*servers*). Процеси, що запрошують служби в серверів шляхом посилки запиту і подальшого чекання відповіді від сервера, називаються клієнтами (*clients*).

За такою схемою можуть бути побудовані системи обробки даних на основі СУБД, поштові і інші системи. У даному проекті необхідно розробити архітектуру з присутністю баз даних і системах на їх основі. І тут зручніше буде не просто розглядати клієнт–серверну архітектуру, а порівняти її з іншою – файл-серверною.

У файл-серверній системі дані зберігаються на файловому сервері (наприклад, *Novell NetWare* або *Windows NT Server*), а їх обробка здійснюється на робочих станціях, на яких, як правило, функціонує одна з, так званих, «настільних СУБД» – *Access*, *FoxPro*, *Paradox* і тому подібне.

Застосування на робочій станції «відповідає за все» – за формування призначеного для користувача інтерфейсу, логічну обробку даних і за безпосереднє маніпулювання даними. Файловий сервер надає послуги лише найнижчого рівня – відкриття, закриття і модифікацію файлів. Поважно, що ці послуги стосуються – файлів, а не бази даних. Система управління базами даних розташована на робочій станції.

Таким чином, безпосереднім маніпулюванням даними займається декілька незалежних і неузгоджених між собою процесів. Крім того, для здійснення будь-якої обробки (пошук, модифікація, підсумовування і тому

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

подібне) всі дані необхідно передати по мережі з сервера на робочу станцію (див. рисунок 2.1 – Порівняння файл-серверної і клієнт-серверної моделей).

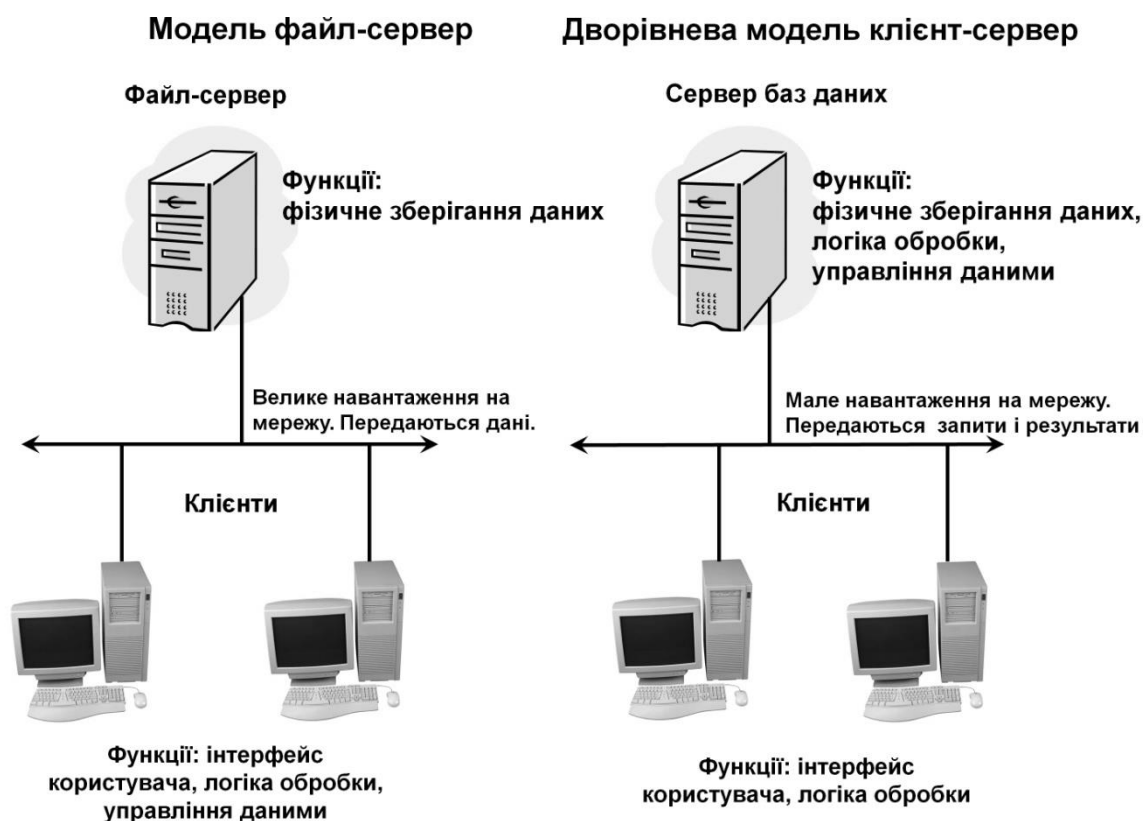


Рисунок 2.1 – Порівняння файл-серверної і клієнт-серверної моделей

У клієнт-серверній системі функціонують (як мінімум) два застосування – клієнт і сервер, що ділять між собою ті функції, які в серверній для файлу архітектурі цілком виконує застосування на робочій станції. Зберіганням і безпосереднім маніпулюванням даними займається сервер баз даних, як яке може виступати *Microsoft SQL Server, Oracle, Sybase MySQL* і тому подібне.

Формуванням призначеного для користувача інтерфейсу займається клієнт, для побудови якого можна використовувати цілий ряд спеціальних інструментів, а також більшість настільних СУБД. Логіка обробки даних може виконуватися як на клієнтові, так і на сервері. Клієнт посилає на сервер

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

запити, сформульовані, як правило, на мові *SQL*.

Сервер обробляє ці запити і передає клієнтові результат (зрозуміло, клієнтів може бути багато).

Таким чином, безпосереднім маніпулюванням даними займається один процес. При цьому, обробка даних відбувається там же, де дані зберігаються – на сервері, що виключає необхідність передачі великих об'ємів даних по мережі.

З точки зору потреб бізнесу, архітектура клієнт–сервер привносить в інформаційну систему надійність.

Сервер баз даних здійснює модифікацію даних на основі механізму транзакцій, який додає будь–якій сукупності операцій, оголошених як транзакція, наступні властивості:

- атомарність – за будь–яких обставин будуть або виконані всі операції транзакції, або не виконана жодна; цілісність даних при завершенні транзакції;
- незалежність – транзакції, ініційовані різними користувачами, не втручаються в справи один одного;
- стійкість до збоїв – після завершення транзакції, її результати вже не пропадуть.

Механізм транзакцій, підтримуваний сервером баз даних, набагато ефективніший, ніж аналогічний механізм в настільних СУБД, оскільки сервер централізований контролює роботу транзакцій. Крім того, в серверній для файлу системі збій на будь–якій з робочих станцій може привести до втрати даних і їх недоступності для інших робочих станцій, в той час, як в клієнт–серверній системі збій на клієнтові, практично, ніколи не позначається на цілісності даних і їх доступності для інших клієнтів.

Масштабованість – здатність системи адаптуватися до зростання кількості користувачів і об'єму бази даних при адекватному підвищенні

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

продуктивності апаратної платформи, без заміни програмного забезпечення.

Можливості настільних СУБД серйозно обмежені – це п'ять–сім користувачів і 30–50 Мб, відповідно. Цифри, зрозуміло, є деякими середніми значеннями, в конкретних випадках вони можуть відхилятися як в ту, так і в інший бік. Що найістотніше, ці бар'єри не можна здолати за рахунок нарощування можливостей апаратури.

Системи ж на основі серверів баз даних можуть підтримувати тисячі користувачів і сотні ГБ інформації – дайте їм лише відповідну апаратну платформу.

Сервер баз даних надає потужні засоби захисту даних від несанкціонованого доступу, неможливі в настільних СУБД. При цьому, права доступу адмініструються дуже гнучко – до рівня полів таблиць. Крім того, можна взагалі заборонити пряме звернення до таблиць, здійснюючи взаємодію користувача з даними через проміжні об'єкти – вистави і процедури, що зберігаються. Отже адміністратор може бути упевнений – жоден дуже розумний користувач не прочитає те, що йому читати не належить.

У застосуванні, що працює з даними, можна виділити три логічних шаруючи:

- призначеного для користувача інтерфейсу;
- правил логічної обробки (бізнес–правил);
- управління даними.

У серверній для файлу архітектури все три шару реалізуються в одному монолітному застосуванні, що функціонує на робочій станції. Тому зміни в будь–якому з шарів наводять однозначно до модифікації застосування і подальшого оновлення його версій на робочих станціях.

У дворівневому клієнт–серверному застосуванні, показаному на

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		31

рисунку вище, як правило, всі функції по формуванню призначеного для користувача інтерфейсу реалізуються на стороні клієнта, всі функції по управлінню даними – на сервері, а ось бізнес–правила можна реалізувати як на сервері використовуючи механізми програмування сервера (процедури, що зберігаються, тригери, вистави і тому подібне), так і на стороні клієнта.

У трирівневому застосуванні з'являється третій, проміжний рівень, що реалізовує бізнес–правила, які є найбільш часто змінними компонентами застосування (див. рисунок 2.2 – Трирівнева модель клієнт–серверного застосування).

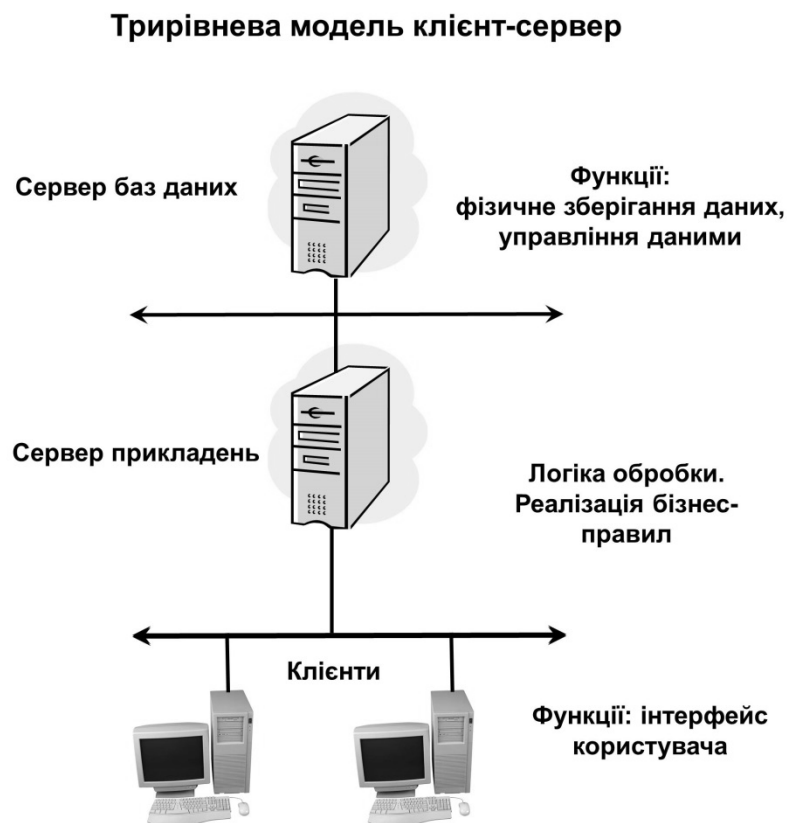


Рисунок 2.2 – Загальна архітектура проектного програмного засобу

Наявність не одного, а декількох рівнів дозволяє гнучко і з мінімальними витратами адаптувати додаток до вимог, що змінюються, бізнесу.

Якщо змінилися правила розрахунку оцінки учня і потрібно відновити

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		32

відповідне програмне забезпечення, то потрібно виконати наступні кроки:

- у серверній системі ми вносимо зміни до застосування і оновлюємо його версії на робочих станціях. Але це спричиняє за собою максимальні трудовитрати;

- у трьохрівневої клієнт–серверній системі, якщо алгоритм розрахунку зарплати реалізований на сервері у вигляді правила, його виконує сервер бізнес–правил, виконаний, наприклад, у вигляді *OLE*–сервера, і ми відновимо один з його об'єктів, нічого не міняючи ні в клієнтському застосуванні, ні на сервері баз даних.

Проектований комплекс планується в двох частинах: серверної і клієнтської. Система розробляється на основі вживання сучасного графічного інтерфейсу.

Інформаційна система (ІС), що розробляється, призначена для комплексної автоматизації обліку на підприємствах автобізнесу (магазинів оптового та роздрібного продажу запчастин, автосервісів і станцій технічного обслуговування автомобілів, автосалонів і дилерських центрів). Реалізація такої системи дозволить значно полегшити роботу співробітників автосалону.

В результаті впровадження ІС очікуються наступні переваги:

- підвищення оперативності виконання функцій;
- безперервний оперативний контроль за ходом реалізації автомобілів;
- зниження числа помилок менеджера при введенні даних.

ІС повинна забезпечувати виконання таких функцій:

1. Введення довідкових даних.
2. Введення відомостей про постачальників.
3. Введення відомостей про покупців.
4. Введення відомостей про автомобілі.
5. Введення відомостей про наявність замовлень.
6. Внесення змін відомостей про автомобілях.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		33

7. Внесення змін відомостей про постачальників.
8. Формування різних звітів: прайс-лист автомобілів, аналіз продажів, реєстр довідок рахунків, замовлення автомобілів, стан замовлень автомобілів.
9. Внесення відомостей про продані автомобілях.
10. Виписка рахунку на оплату.

Важливою частиною проектування процесу автоматизованої системи є математична постановка задачі. Вона включає в себе: вибір, обґрунтування і опис моделі об'єкта проектування; формальну постановку мети, вибір методів вирішення поставлених завдань і критеріїв досягнення цілей.

Розрахунок кінцевої вартості автомобіля виконується за наступною схемою:

$$S_a = S_z + S_d + S_n,$$

де S_a - вартість автомобіля,

S_z - закупівельна ціна,

S_d - вартість доставки,

S_n - податок.

ІС призначена для автоматизації діяльності автомобільного салону. Розробляється система повинна забезпечувати вирішення наступних завдань:

- автоматизувати процес оформлення продажу і поставки автомобілів;
- підвищити точність, збереження і захищеність даних;
- прискорити процес пошуку автомобілів;
- підвищити зручність роботи з базою даних;
- автоматизувати розрахунок вартості продажу автомобіля;
- здійснити контроль даних, що вводяться;
- систематизувати зберігання даних про клієнтів, співробітників, автомобілях;

					КРБ.КІ.1.146-03.2.2	Арк.
						34
Змн.	Арк.	№ докум.	Підпис	Дата		

- максимально перехопити системні помилки;
- спростити процедуру введення даних.

Функції системи:

- зберігання і зміна інформації про автомобілі (ідентифікатор, марка, модель, колір, країна виробник, вартість);
- зберігання і зміна інформації про співробітників (ПІБ, дата народження, адреса, дата найму); про клієнтів (ПІБ, адреса, телефон, дата народження, вік); про продажі (дата, вартість за оформлення, вартість доставки, вартість ПДВ, загальна вартість);
- пошук інформації про автомобілі, клієнтів, співробітників;
- формування чека, платіжного доручення, заявок на замовлення;
- розрахунок вартості покупки з урахуванням знижок;
- складання діаграми про роботу співробітників;
- зберігання даних в архіві про продані автомобілях.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		35

3 АНАЛІЗ І ВИБІР ЗАСОБІВ ПІДТРИМКИ ПРОЕКТУВАННЯ ТА РОЗРОБКИ СИСТЕМИ

Сучасні веб-сайти стають все більш складними і все більш перевантажуються логікою. Раніше продуктивність таких додатків визначалася, в основному, швидкістю роботи того чи іншого *SQL*-сервера і тим, чи існує для нього достатньо ефективна реалізація драйвера доступу до *SQL*-серверу для вибраної мови програмування. Це пояснюється тим, що перше покоління веб-сайтів просто читало і писало інформацію у бази даних. Користувачів при цьому було відносно небагато. Таким чином, час на відпрацювання *SQL*-запиту становив 70–90% від загального часу обробки *HTTP*-запиту.

З підвищенням вимог до масштабованості (збільшення кількості користувачів) і нарощуванням логіки додатка вимоги до мови програмування і середовища виконання істотно зростають. До цього слід також додати, що відносно недавно веб-сайти перейшли зі світу Інтернет у світ корпоративних додатків. Це знову підвищило вимоги до ефективності середовища виконання.

У теперішній час для створення веб-сайтів існує безліч різних мов програмування. Найпопулярніші з них – *PHP*, *Perl*, *C # (DOT.NET)*, *Java2*. Класична технологія *ASP* стає історією, оскільки *ASP DOT.NET* практично витіснило цю технологію на *Windows*-серверах в нових програмах.

Для розробки в роботі вибрана мова *PHP* у зв'язці з сервером баз даних *MySQL*. Для їх роботи потрібна установка веб-сервера *Apache*. В якості середовища розробки вибрано засіб створення *web*-вузлів *Macromedia Dreamweaver MX 2004*.

3.1 Вибір серверної мови сценаріїв і засобу проектування *WEB*-сайту

PHP – це інструмент об'єктно-орієнтованого програмування з

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

відкритим кодом (*open source*), мова серверних скриптів (*server scripting language*), що вбудовується в *HTML*, який інтерпретується і виконується на сервері.

PHP є препроцесором *HTML*. Його робота побудована за схемою, показаної на рисунку 3.1

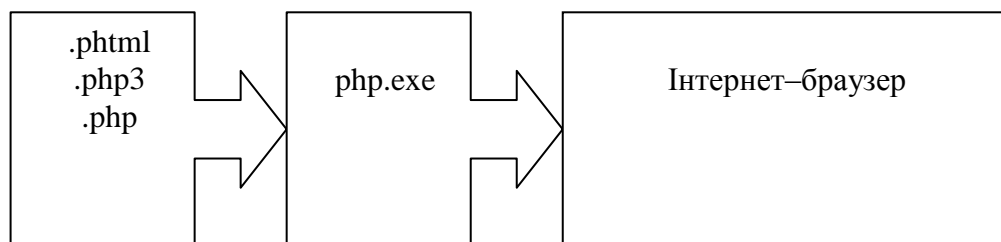


Рисунок 3.1 – Схема роботи препроцесора *PHP*

До того, як сервер «віддасть» файл браузеру, його переглядає препроцесор-інтерпретатор. Для того, щоб це відбувалося, файли, які піддаються обробці препроцесором, повинні мати певне розширення (зазвичай це *.html* або *.php*, але ці значення можна поміняти) і містити (хоча це не обов'язкова вимога) код для препроцесора. Перед відправкою сторінки *PHP*-код програється на сервері і браузеру видається результат у вигляді знову таки *HTML*-сторінки, яка може сильно відрізнятися від тієї, що зберігається на сервері. Звичайні ж сторінки, що мають розширення *.html/htm web*-сервер буде відправляти браузеру без будь-якої обробки.

Основна відмінність від *CGI*-скриптів, написаних на інших мовах, типу *Perl* або *C* – це те, що в *CGI*-програмах ви самі пишете виведений *HTML*-код, а, використовуючи *PHP* – ви вбудовуєте свою програму-скрипт в готову *HTML*-сторінку, використовуючи відкриваючий та закриваючий теги (у прикладі `<? php i?>`).

PHP називається мовою серверних скриптів на відміну від *JavaScript* / *Jscript* / *VBScript*, які є мовами клієнтських скриптів. Це означає, що *PHP*–

скрипт виконується на сервері, а клієнту передається результат його роботи, тоді як в *JavaScript*—код повністю передається на клієнтську машину і тільки там виконується браузером.

Користувачі *MS Internet Information Server* знайдуть, що *PHP* дуже схожий на *Active Server Pages (ASP)*, а користувачі *Java* скажуть, що *PHP* схожий на *Java Server Pages (JSP)*. Ще деякими аналогами *PHP* є мови *ColdFusion* і *embPerl*. Всі ці мови дозволяють розміщувати код, що виконується на *Web*—сервері, усередині *HTML*—сторінок. В результируючої *web*—сторінці немає і сліду *PHP*—коду.

На *PHP* можна зробити все, що можна зробити за допомогою *CGI*—програм. Наприклад: обробляти дані з форм, генерувати динамічні сторінки, одержувати і посилати куки (*cookies*).

Крім цього в *PHP* включена підтримка багатьох баз даних (*databases*), що робить написання *Web*—додатків з використанням БД до неможливості простим.

Ось неповний перелік підтримуваних БД: *Adabas D, InterBase, Solid, dBase, mSQL, Sybase, Empress, MySQL, Velocis, FilePro, Oracle, Unix dbm, Informix, PostgreSQL, ODBC*.

Додатково до всього *PHP* розуміє протоколи *IMAP, SNMP, NNTP, POP3* і навіть *HTTP*, а також має можливість працювати з сокетом (*sockets*) і спілкуватися по інших протоколах.

Розробникам *web*—додатків немає необхідності говорити, що *web*—сторінки – це не тільки текст і картинки. Гідний уваги сайт повинен підтримувати деякий рівень інтерактивності з користувачем: пошук інформації, продаж продуктів, конференції тощо До недавніх пір все це традиційно реалізувалося *CGI*—скриптами, написаними на *Perl*. Але виявилось, що *CGI*—скрипти дуже погано масштабуються. Кожен новий виклик *CGI*—скрипта, вимагає від ядра породження нового процесу, а це

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

займає процесорний час і витрачає оперативну пам'ять. *PHP* пропонує інший варіант – він працює як частина web-сервера, і цим самим схожий на *ASP* від *Microsoft* або *ColdFusion* від *Allaire*.

Синтаксис *PHP* дуже схожий на синтаксис *C* або *Perl*. Люди, знайомі з програмуванням, дуже швидко зможуть почати писати програми на *PHP*. У цій мові немає строгої типізації даних і немає необхідності в діях по виділенню / звільненню пам'яті.

Програми, написані на *PHP*, читаються досить легко. На відміну від *Perl*-програм *PHP*-код легко візуально прочитати і зрозуміти.

На додаток до своєї безкоштовності (хоча *MySQL* вимагає придбання ліцензії при використанні її в комерційних цілях) зв'язка *PHP-MySQL* є крос-платформеною. Це означає, що ви можете, працюючи в *Windows*, розробляти програми, призначені для роботи під *Unix*. Крім того, *PHP* може працювати як зовнішній *CGI*-процес, або як звичайний інтерпретатор скриптів, або як модуль, що підключається до web-серверу *Apache* або *IIS*.

І нарешті, так як даний продукт розробляється спільними зусиллями, існує величезна кількість документації та списків розсилки, до яких можна звернутися у разі виникнення будь-яких питань.

Знайдені помилки виправляються досить швидко, ваші пропозиції та зауваження завжди вислухають, розглянуть, і якщо вони виявляться цінними – реалізують в новій версії.

Вибір системи управління базами даних

MySQL – невеликий, компактний багатопоточний сервер баз даних. *MySQL* характеризується великою швидкістю, стійкістю і легкістю у використанні.

MySQL був розроблений компанією *TcX* для внутрішніх потреб, що полягали у швидкій обробці дуже великих баз даних. Компанія стверджує, що використовує *MySQL* з 1996 року на сервері з більш ніж 40 БД, що

					КРБ.КІ.1.146-03.2.2	Арк.
						39
Змн.	Арк.	№ докум.	Підпис	Дата		

містять 10,000 таблиць, з яких більш ніж 500 мають більш 7 мільйонів рядків. *MySQL* є ідеальним рішенням для малих і середніх додатків. Вихідні тексти сервера компілюються на безлічі платформ. Найбільш повно можливості сервера проявляються на *Unix*-серверах, де є підтримка багатопоточності, що дає значний приріст продуктивності. У варіанті під *Windows*, *MySQL* може запускатися як сервіс *Windows NT* або як звичайний процес на *Windows 95/98*.

MySQL-сервер є безкоштовним для некомерційного використання.

MySQL підтримує мову запитів *SQL* у стандарті *ANSI 92*, і крім цього має безліч розширень до цього стандарту, яких немає в жодній іншій СУБД.

Короткий перелік можливостей *MySQL*:

- підтримується необмежена кількість користувачів, що одночасно працюють з базою даних;
- кількість рядків у таблицях може досягати 50 млн;
- швидке виконання команд. Можливо *MySQL* найшвидший сервер з існуючих;
- проста і ефективна система безпеки.

MySQL дійсно дуже швидкий сервер, але для досягнення цього розроблювачам довелося пожертвувати деякими вимогами до реляційних СУБД. В *MySQL* відсутні:

- не реалізована підтримка транзакцій. Натомість пропонується використовувати *LOCK/UNLOCK TABLE*;
- немає підтримки тригерів і збережених процедур;
- немає підтримки зовнішніх ключів.

За словами творців саме ці пункти дали можливість досягти високої швидкодії. Їх реалізація істотно знижує швидкість серверу. Ці можливості не є критичними при створенні *web*-додатків, що в поєднанні з високою швидкодією і малою ціною дозволило серверу набути велику популярність.

Вибір засобу проектування *WEB*-сайту.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		40

На сьогоднішній момент існує два потужних засоба створення веб-сайтів – це останні версії *FrontPage* 2014 компанії *Microsoft* і *Dreamweaver MX* 2012 компанії *Macromedia*.

Програмний пакет *FrontPage* довгий час розвивався і вдосконалювався фірмою *Microsoft*. Його версія *FrontPage* 2000 відрізняється від усіх попередніх тим, що вона вперше інтегрована в структуру пакету програм *Microsoft Office*. Ймовірно, це пояснюється тією обставиною, що проектування веб-сайтів і мультимедійних електронних видань з завдання для професіоналів перейшли в сферу офісної діяльності. *FrontPage* 2002 знову поставляється окремо від *Microsoft Office XP*.

FrontPage спеціалізований на підготовці електронних видань, зокрема – на розробці *web*-вузлів цілком. Він підтримує фреймову структуру, орієнтований на простоту і зручність створення гіперпосилань, забезпечує єдиний стиль оформлення різних *web*-сторінок, оптимізує швидкість їх завантаження. Таким чином, пакет відрізняється продуманістю реалізації як естетичних, так і технічних аспектів створення електронних видань і дозволяє, в якійсь мірі, автоматизувати процес проектування. Незручністю в роботі є автоматична вставка не завжди потрібних тегів.

Програмний пакет *Dreamweaver*, як і *FrontPage*, призначений для проектування різних сайтів, у тому числі і сайтів видавництва, одним із завдань яких є поширення електронних видань. Він придатний як для створення невеликих, так і дуже великих сайтів. У першому випадку розробкою займається єдиний фахівець – *web*-майстер. В останньому випадку над проектом працює цілий колектив, який включає адміністратора проекту, *web*-дизайнера, *HTML*-верстальника, *web*-програміста, коректора, менеджера розсилок. Багато фахівців стверджують, що професійні *web*-дизайнери в своїй роботі найчастіше використовують *Adobe Photoshop* для підготовки графіки і *Macromedia Dreamweaver* – для дизайну окремих *web*-

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		41

сторінок. Особливістю даного програмного засобу є хороша підтримка мови *PHP*, який використовується в проекті для програмування і автоматична або ручна синхронізація змінених даних з віддаленим сервером. Ці особливості, а також простота і швидкість роботи програми багато в чому зумовлюють вибір саме цього програмного продукту в якості засобу проектування розробляється *web*-додаток.

3.2 Вибір веб-сервера

Функції веб-сервера виконує програма *Apache*. Коли браузер користувача підключається до веб-сервера і посилає заголовок *GET* (це прохання передати файл), його запит обробляє *Apache*. *Apache* перевіряє, чи є зазначений у заголовку *GET* файл, і, якщо є, відправляє його разом з заголовками браузеру.

Apache – це стандарт веб-сервера в Інтернеті. Його основним конкурентом є *IIS (Internet Information Server)* від *Microsoft*, який працює під *Windows*. *Apache*, хоча й існують його версії під *Windows*, переважно встановлюється на *Unix*-подібних операційних системах – *Linux* і *FreeBSD*. Потрібно також зауважити, що більшість хостерів використовують саме *Apache*, а не *IIS*. *IIS* встановлюється в основному на корпоративних серверах, що працюють під *Windows*.

Сервер *Apache* просто необхідний – без нього ми не зможемо протестувати веб сценарії *PHP*. Звичайно, можна тестувати і без сервера, але тоді потрібно буде використовувати сервер хостера. У цьому випадку витрати на Інтернет сильно зростуть.

Для установки *Apache* для *Windows* потрібно відвідати сайт <http://www.apache.org> і завантажити версію *Apache* для *Windows*. Встановлення сервера проблем не викликає: потрібно просто запустити програму установки і вибрати тип установки *Typical*. Ще потрібно вказати

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		42

каталог, в який буде проводитися установка. За замовчуванням буде встановлений в каталог "C:\Program Files\Apache Group\Apache".

По закінченні установки *Apache* необхідно налаштувати. За замовчуванням кореневим каталогом для *HTML*–документів буде *C:\Program Files\Apache Group\Apache\htdocs*. Таке розташування не дуже зручно, тому створимо каталог *f:\home\localhost*, в якому будуть знаходитися файли. У цьому ж каталозі створимо два підкаталоги – *cgi* і *html*. У першому будуть знаходитися *CGI*–додатки, а в другому – *HTML*–файли.

Потім потрібно відкрити в будь–якому текстовому редакторі файл *C:\Program Files\Apache Group\Apache\conf\httpd.conf* і відредагувати наступні директиви:

ServerName – встановити ім'я сервера. При цьому, директива *ServerName* за замовчуванням закоментований, тобто ігнорується сервером. Тому необхідно розкоментувати цю директиву, прибравши перед нею значок #.

DocumentRoot – встановимо значення *C:/www/html*:

DocumentRoot c:/www/html

Таким чином, зазначимо каталог, в якому розміщуватимуться *HTML*–файли.

Встановимо директиви *ScriptAlias* так:

ScriptAlias /cgi / "c: / www / cgi /"

ScriptAlias /cgi-bin / "c: / www / cgi /"

Завдяки цьому створимо два псевдоніма для каталогу *c:/www/cgi*. Таким чином, якщо буде вказано шлях *http://localhost/cgi* або *http://localhost/cgi-bin*, то насправді звернення буде до каталогу *c:/www/cgi*. Секцію *<Directory "C: / Program Files / Apache Group / Apache / htdocs">* замінимо на наступну:

<Directory "c: / www">

					КРБ.КІ.1.146-03.2.2	Арк.
						43
Змн.	Арк.	№ докум.	Підпис	Дата		

Options Indexes Includes AllowOverride All Allow from all </ Directory>

Таким чином, задамо ряд налаштувань за замовчуванням для каталогу *c:\www* і всіх його підкаталогів. А саме: автоматичну генерацію списку вмісту каталогу при перегляді його в браузері, підтримку *SSI*, а також можливість використовувати файли. *access* для індивідуальної настройки каталогів.

Переконаємося, що є директива *AddHandler*:

AddHandler cgi-script.bat.exe.cgi

Ця директива вказує *Apache* на те, що файл з розширеннями *.bat .exe* і *.cgi* слід сприймати як *CGI*-модулі.

Встановимо параметри *SSI*:

Addtype text/html.shtml

AddHandler server-parsed .shtml .html .htm Завдяки цьому файли із зазначеними розширеннями будуть оброблятися *SSI*.

Необхідно зберегти відредагований файл.

Тепер протестуємо роботу *Apache*. Запуск:

C:\Program Files\Apache Group\Apache\apache.exe

При успішному запуску з'явиться напис

Apache / 2.0.35 (Win32) running ...

Тепер можна перевірити коректність роботи сервера. Запустимо Internet Explorer або будь-який інший браузер і введемо: *http://localhost*. Ви повинні побачити стартову сторінку *Apache*. Тепер перевіримо *CGI* і *SSI*. Для цього створимо файл *test.bat* і помістимо його в каталог *c:\www\cgi*:

echo off

echo Content-type: text/html

echo.

echo.

echo Hello!

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		44

У вікні браузера введіть *http: // localhost / cgi / test. bat* – увідіте слово *Hello!* Для перевірки *SSI* потрібно створити файл *test.html* і помістити його в каталог *c: \ www \ html:*

*SSI Test
*

<! –tinclue virtual = "/ index. html" –>

SSI (Server Side Include) – включення з боку сервера, дозволяє включати в *Web*–сторінку код *HTML*, що знаходиться в інших файлах.

У вікні браузера при виклику файлу *test. html* ви побачите *SSI Test*, а потім – стартову сторінку *Apache*.

3.3 Встановлення *PHP* і настройка *Apache* для роботи з *PHP*

Після установки *Apache* встановимо *PHP* 4. Версію *PHP* 4 для *Windows* можна завантажити за адресою *http://www.php.net* (остання версія *PHP* доступна за адресою *http://www.php.net/downloads.php*).

Запустимо програму установки, яка за замовчуванням встановить *PHP* 4 в каталог *C:\Program Files\PHP 4*. Проте в процесі установки можна вибрати інший каталог.

Відразу після установки необхідно відкрити файл *php.ini* (він буде в каталозі, куди ви встановили *PHP*) і відредагувати наступні параметри:

[Mail function]; For Win32 only. SMTP = localhost

sendmail_from = me@localhost.com

Параметр *SMTP* задає ім'я *SMTP*–сервера. За замовчуванням як *SMTP*–сервера використовується локальний комп'ютер – це добре для *Linux*, але в *Windows* вам потрібно вказати *SMTP*–сервер провайдера, тому що навряд чи у вас буде встановлений поштовик під *Windows*. Якщо ви не зробите цього, функція *mail ()*, що відправляє повідомлення, працювати не буде.

Після цього відкриємо файл *httpd.conf* (файл конфігурації *Apache*) і знайдемо в ньому рядок:

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		45

AddType application / x-httpd-php php

Переконаємося, що дана рядок не закоментований, тобто перед нею не стоїть значок *#*. Якщо він стоїть – приберемо його. Таким чином, всім файлам з розширенням *.php* буде поставлений у відповідність *MIME*-тип *application/x-httpd-php*.

Типи *MIME* використовуються для ідентифікації типу вмісту. Завдяки цьому додатку зможуть визначати, якого виду дані надіслані і в якому відповідно мережі проводити їх обробку. Перелік всіх типів *MIME*, підтримуваних системою, знаходиться в файлі *C:\Program Files\Apache Group\Apache\conf*. Тип *MIME* вказується у форматі: тип / підтип. Ось основні типи *MIME*:

application – додаток (програма або документ, пов'язаний з програмою).

audio – аудіофайл (*audio / mpeg*).

image – графічний файл (*audio / png*).

text – текст (*text / html* – у форматі *HTML*, *text / plain* – простий).

MIME-тип пересилаються по мережі даних вказується в заголовку *Content-type*.

Відразу ж після цього рядка додамо рядки:

ScriptAlias / _php _ / "c: / Program Files / PHP5 /" Action application / x-httpdhp "/ _php_ / php. Exe"

У цих двох рядках ми спочатку створюємо псевдонім *_php_* для каталогу *c: / Program Files / PHP 4 /*, а потім пов'язуємо всі файли типу *application / x-httpd-php* з обробником *php.exe*.

Тепер можна протестувати *PHP*. Для цього створимо файл *test.php*:

<?

phpinfo () ;? >

Даний файл потрібно помістити в каталог *c:\www\html*. У браузері введіть *http: // localhost / test. php*. У підсумку ви повинні побачити результат

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

роботи функції *phpinfo ()*.

Встановлення MySQL

Сервер *MySQL* можна завантажити за адресою *http: // www. mysql. com*.

Програма установки встановить сервер в каталог *C: \ Program Files \ MySQL *.

Для запуску *MySQL* створимо файл *server. bat*:

```
echo off
```

```
"C: \ Program Files \ MySQL \ bin \ mysqld"
```

```
start /m "C: \ Program Files \ Apache Group \ Apache \ apache"
```

Якщо ви працюєте в *Windows NT* або *2000*, *server.bat* повинен виглядати так, як це показано в наступному лістингу:

```
echo off
```

```
start C: \ Progra ~ l \ MySQL \ bin \ mysqld-nt - standalone
```

```
C: \ Progra ~ l \ Apache ~ l \ Apache \ apache - k start
```

Тепер для запуску *Apache* і *MySQL* потрібно ввести команду *server*.

Створимо файл *shutdown.bat*, який буде зупиняти обидва сервера:

```
echo off
```

```
C: \ Progra ~ l \ Apache ~ l \ Apache \ apache - k shutdown
```

```
C: \ Progra ~ l \ MySQL \ bin \ mysqladmin - u root shutdown
```

Також необхідно додати в кінець файлу *my.ini* наступні рядки (конкретно для даного проекту), що задають логін і пароль доступу до сервера:

```
user = root
```

```
password = geijr
```

Файл налаштування сервера баз даних *MySQL* знаходиться за замовчуванням в кореневому каталозі *Windows*. Переглянути всі бази даних і таблиці, створені на сервері *MySQL* можна утилітою *WinMySQLAdmin 1.4*

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		47

4 ПРОЕКТУВАННЯ БАЗИ ДАНИХ

4.1 Основи розробки реляційної бази даних

Метою розробки БД є визначення її логічної структури. В результаті проектування має бути визначений склад реляційних таблиць, для кожної таблиці – склад її атрибутів (стовпців) і логічні зв'язки між таблицями. Для кожного атрибуту мають бути задані тип даного, його розмір і обмеження цілісності. Для кожної таблиці – первинний ключ, потенційні ключі і зовнішні ключі. При цьому отримувана логічна модель оцінюється по досягненню наступних цілей проектування:

1. Можливості зберігання всіх необхідних даних в БД.
2. Виключення надлишкових даних.
3. Відомості числа стосунків, що зберігаються, в БД до мінімуму.
4. Нормалізації стосунків для спрощення вирішення проблем, пов'язаних з оновленням, додаванням і видаленням даних.

Перший крок процесу проектування полягає у визначенні як всіх атрибутів, наявність яких в БД чекає користувач, так і зв'язків між атрибутами. Цей крок виконується на основі аналізу документів з даними, запитів користувачів і інших відомостей про об'єкти і процеси, що характеризують наочну область. Після визначення складу даних, належних зберігання в БД, має бути проведений їх аналіз і структуризація.

Розробку логічної моделі можна здійснювати різними методами. Найбільш формалізованим і простим для розуміння є метод «сутність-зв'язок», або *ER*-метод. Суть методу полягає в побудові *ER*-діаграмм, що відображують в графічній формі основні об'єкти ПО і зв'язки між ними, і у визначенні характеристик цих зв'язків. Потім по чітких правилах робиться перехід від *ER*-діаграмм до таблиць БД, здійснюється наповнення таблиць атрибутами і перевірка їх на виконання умов нормалізації (НФБК).

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Визначаються ключові атрибути таблиць і зв'язку між таблицями. Результатом проектування є схема даних БД.

Сутність – це об'єкт, інформація про яке має бути представлена в БД (зазвичай відповідає іменнику). Екземпляр сутності – це інформація про конкретного представника об'єкту. Наприклад, для сутності Відповідальний Виконавець екземпляром є Іванов С.С., а для сутності Клієнт – екземпляром є Фірма «Одеські бублики».

Зв'язок – з'єднання між двома і більш сутностями. Екземпляр зв'язку – це конкретний зв'язок між конкретними представниками об'єктів. Наприклад, для зв'язку Відповідальний Виконавець працює у клієнта, екземпляром є Іванов С.С. працює у клієнта фірми «Одеські бублики».

Атрибут – властивість сутності або зв'язку. Наприклад, Прізвище, Ім'я, По батькові є атрибути сутності Особа, а слова Іванов, Володимир, Васильович є екземплярами цих атрибутів.

Атрибут або набір атрибутів, використовуваний для однозначної ідентифікації екземпляра сутності, називається ключем сутності. Кожен екземпляр зв'язку визначається набором ключів еств, що сполучаються цим зв'язком.

У *ER*-діаграмі для відображення сутностей використовуються прямокутники, а для відображення зв'язків – ромби. Розрізняють *ER*-діаграми для екземплярів сутностей і *ER*-діаграми для класів сутностей.

Зв'язок між сутностями має дві характеристики: міра зв'язку і клас приналежності сутності до зв'язку. Значення цих характеристик можуть бути визначені з аналізу зв'язків между екземплярами сутностей. Міра зв'язку показує, скільки екземплярів однієї сутності можуть бути пов'язані з кожним екземпляром іншої сутності. Міра зв'язку оцінюється з кожною із сторін. Міра зв'язку може мати одне з трьох значень:

– один до одного (1:1);

	+				КРБ.КІ.1.146-03.2.2	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

- один до багатьох (1: M);
- багато до багатьох (M:N).

Зв'язок 1:1 означає, що кожен екземпляр першої сутності може бути пов'язаний лише з одним екземпляром іншої сутності і навпаки (рисунок 4.1).

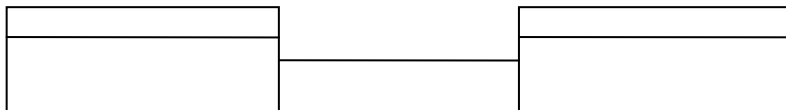


Рисунок 4.1 - Тип зв'язку «один-до-одного»

Зв'язок 1:M означає, що кожен екземпляр першої сутності може бути пов'язаний з декількома екземплярами іншої сутності, а кожен екземпляр іншої сутності може бути пов'язаний лише з одним екземпляром першої сутності (рисунок 4.2).

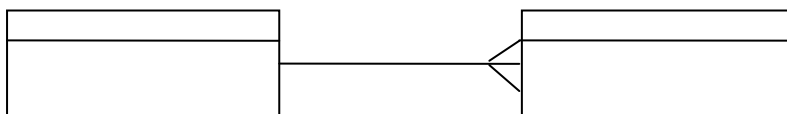


Рисунок 4.2 - Тип зв'язку «один-до-багатьох»

Зв'язок M:N означає, що кожен екземпляр першої сутності може бути пов'язаний з декількома екземплярами іншої сутності і навпаки.

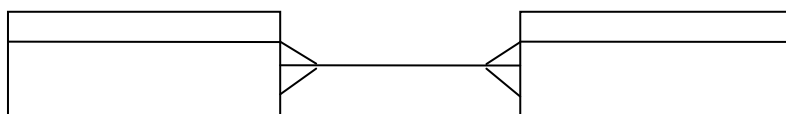


Рисунок 4.3 - Зв'язок «багато-до-багатьох» пов'язаний з декількома екземплярами іншої сутності і навпаки.

Клас приналежності сутності до зв'язку може бути обов'язковим і необов'язковим. При обов'язковому класі приналежності кожен екземпляр сутності обов'язково має бути пов'язаний з іншою сутністю, при необов'язковому класі приналежності не вимагається, аби кожен екземпляр сутності був пов'язаний з яким-небудь екземпляром іншої сутності. Іншими словами, в БД зберігається інформація про всі екземпляри сутностей, але лише деякі з них пов'язані з екземплярами іншої сутності.

Для відображення характеристик зв'язку на діаграмі може використовуватися, наприклад, символ «крапка». Якщо крапка усередині прямокутника, то відповідна прямокутнику с має обов'язковий клас приналежності. Якщо поза прямокутником, то необов'язковий клас приналежності. Цифри або букви поряд з крапками вказують на міру зв'язку. Поєднання трьох типів зв'язків з двома класами приналежності дають можливість опису безлічі різних варіантів зв'язків в наочній області.

На наступному етапі проектування ми будуватимемо даталогическую модель. Даталогическая модель бази даних є моделлю логічного рівня і є відображенням логічних зв'язків між елементами даних. Суть етапу даталогического проектування полягає у відображенні инфологической моделі наочної області бази даних на структурні і процедурні можливості. Вони надають: тип структури даних, наприклад, реляційна модель, вибраний тип СУБД, яка цю модель даних підтримує, технологію і засоби прикладного програмування.

Для кожної локальної вистави вказується:

- визначення типів сутностей;
- визначення типів зв'язків;
- визначення атрибутів і скріплення їх з типами сутностей і типами зв'язків;
- визначення доменів атрибутів;

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

- визначення атрибутів, що є первинними, альтернативними або зовнішніми ключами;
- визначення обмежень цілісності.

На основі уявлення про наочної області, для виявлення інформаційних об'єктів необхідно здійснити аналіз її вхідної інформації.

Щоб спростити собі роботу з базами даних *MySQL* використовуємо утиліту *phpMyAdmin*. *phpMyAdmin* представляє інтуїтивний веб-інтерфейс для управління базами даних *MySQL* (рисунок 4.4). Використовуючи цей інструмент, набагато легше працювати з базами даних, ніж керувати *MySQL* через консоль. Надалі, використовуючи конструктор бази даних, створюємо проекти таблиць, вказуємо типів даних і властивості полів, визначаємо індекси і встановлюємо зв'язки між таблицями, призначаємо методи контролю цілісності.

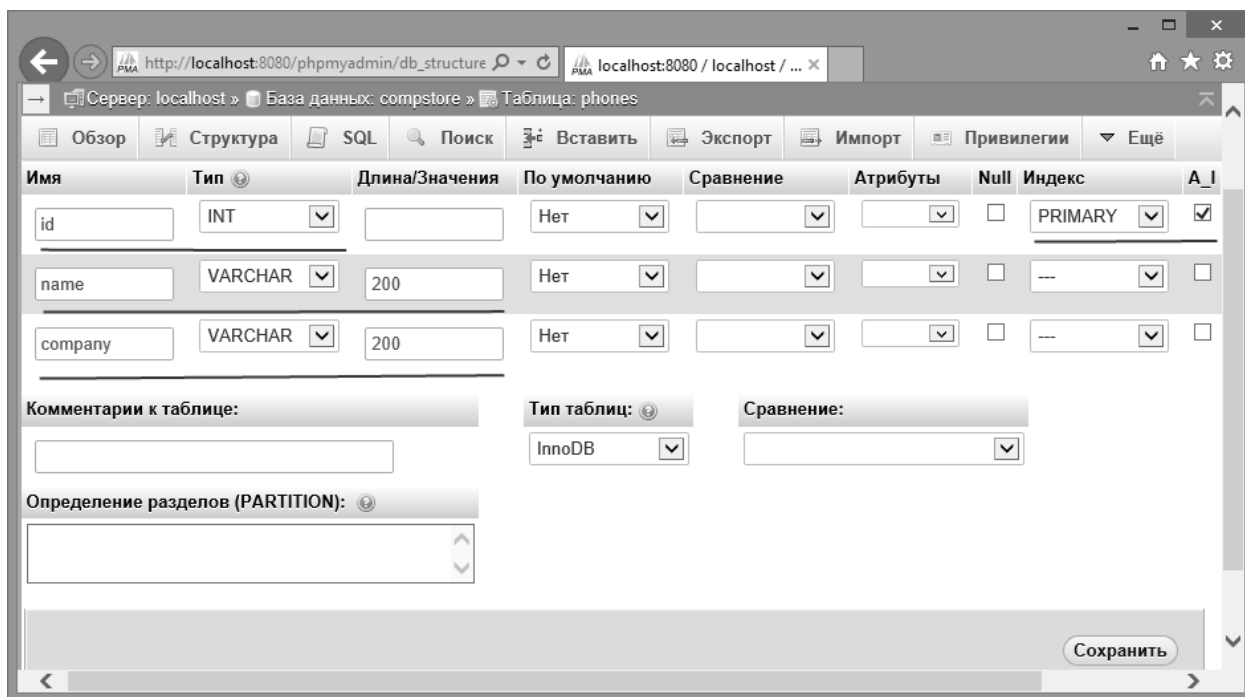


Рисунок 4.4 - Вікно конструктора таблиць

Для оптимізації роботи з таблицями існує декілька можливостей:

- проектувати таблиці, в яких не міститься надлишкові дані;
- вибирати найбільш відповідного типа даних для поля – це приведе до

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		52

– зменшення розміру таблиці і збільшить швидкість виконання операцій. При описі таблиці слід задавати для нього типа даних найменшого розміру, що дозволяє зберігати потрібні дані. База даних з точки зору *MySQL* (і деяких інших СУБД) - це звичайний каталог, що містить виконавчі файли певного формату - таблиці. Таблиці складаються із записів, а записи, в свою чергу, складаються з полів. Поле має два атрибути – ім'я і тип.

Таблиця 4.1 – Поля і їх типи в *MySQL*

Тип данных	Использование	Диапазоны
<i>TINYINT</i>	Дуже маленьке ціле число	Діапазон числа зі знаком від -128 до 127. Діапазон числа без знака (unsigned) від 0 до 255.
<i>SMALLINT</i>	Маленьке ціле число	Діапазон числа зі знаком від -32768 до 32767. Діапазон числа без знака (unsigned) від 0 до 65535.
<i>MEDIUMINT</i>	Середнє ціле число	Діапазон числа зі знаком від -8388608 до 8388607. Діапазон числа без знака (unsigned) від 0 до 16777215.
<i>INT</i> або <i>INTEGER</i>	Ціле число	Діапазон числа зі знаком від -2147483648 до 2147483647. Діапазон числа без знака (unsigned) від 0 до 4294967295.
<i>BIGINT</i>	Велике ціле число	Діапазон числа зі знаком від -9223372036854775808 до 9223372036854775807. Діапазон числа без знака (unsigned) від 0 до +18446744073709551615.
<i>FLOAT</i>	Мале (одинарної точності) число з плаваючою комою. Не може бути числом без знака	Діапазони від -3.402823466E + 38 до -1.175494351E-38, 0 і 1.175494351E-38 до 3.402823466E + 38. Якщо кількість знаків після коми не встановлено або <= 24 це число з плаваючою комою одинарної точності.
<i>DOUBLE</i> , <i>DOUBLE PRECISION</i> , <i>REAL</i>	Нормальне (подвійний точності) число з плаваючою комою. Не може бути числом без знака	Діапазони від -1.7976931348623157E + 308 до -2.2250738585072014E-308, 0 і 2.2250738585072014E-308 до 1.7976931348623157E + 308. Якщо кількість знаків після коми не встановлено або 25 <= кількість знаків <= 53 означає число з плаваючою комою подвійної точності.
<i>DECIMAL</i> , <i>NUMERIC</i>	Розпаковане число з плаваючою комою	Працює подібно типу даних <i>CHAR</i> : «розпакований» означає, що число зберігається у вигляді рядка, використовуючи один символ для кожної цифри-значення. Символ десяткової коми і символ негативного числа "-" не

								Арк.
								53
Змн.	Арк.	№ докум.	Підпис	Дата	КРБ.КІ.1.146-03.2.2			

		<p>враховується в довжину. Якщо десяткове значення дорівнює 0, значення не матиме десяткової коми або дробової частини. Максимальний розмір для DECIMAL значення таке ж, як і для DOUBLE, але фактичний діапазон для даного стовпця DECIMAL може бути обмежений у виборі довжини і десяткові дробі.</p>
--	--	---

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Продовження таблиці 4.1

<i>DATE</i>	Дата	Дата в діапазоні від «1000-01-01» до «9999-12-31». <i>MySQL</i> зберігає поле типу <i>DATE</i> у вигляді «YYYY-MM-DD» (PPPP-MM-ДД).
<i>DATETIME</i>	Дата та час	Допустимі діапазони від «1000-01-01 00:00:00» до «9999-12-31 23:59:59». <i>MySQL</i> зберігає поле типу <i>DATETIME</i> у вигляді «YYYY-MM-DD HH:MM:SS» (PPPP-MM-ДД ГГ-ХХ-СС).
<i>TIMESTAMP</i>	Дата та час	Діапазон від «1970-01-01 00:00:00» до, приблизно, 2037 року. <i>MySQL</i> може зберігати поле типу <i>TIMESTAMP</i> в видах «YYYYMMDDHHMMSS» (<i>TIMESTAMP</i> (14)), «YYMMDDHHMMSS» (<i>TIMESTAMP</i> (12)), «YYYYMMDD» (<i>TIMESTAMP</i> (8)) і ін.
<i>TIME</i>	Час	Діапазон від «-838: 59: 59» до «838: 59: 59». <i>MySQL</i> зберігає поле <i>TIME</i> у вигляді «HH: MM: SS», але дозволяє присвоювати значення стовпцями <i>TIME</i> з використанням або рядки або числа.
<i>YEAR</i>	Рік в 2- або 4 хціфровом вигляді (4 цифри по-замовчуванню)	Якщо ви використовуєте 4 цифри, то допустимі значення 1901-2155, і 0000. Якщо 2 цифри, то 1970-2069 (70-69). <i>MySQL</i> зберігає значення поля <i>YEAR</i> в форматі «YYYY».
<i>CHAR</i>	Рядок фіксованої довжини, яка справа доповнюється пробілами до зазначеної довжини, при зберіганні	Діапазон довжини від 1 до 255 символів. Завершальні прогалини видаляються, коли значення витягується. Значення <i>CHAR</i> упорядковано і порівнюються без урахування регістру в залежності від кодування за замовчуванням, якщо не встановлено прапор <i>BINARY</i> .
<i>VARCHAR</i>	Рядок змінної довжини. Примітка: кінцеві прогалини видаляються при збереженні (в отличие від специфікації <i>ANSI SQL</i>).	Діапазон довжини від 1 до 255 символів. Значення <i>VARCHAR</i> упорядковано і порівнюються без урахування регістру, якщо не встановлено прапор <i>BINARY</i> .
<i>TINYBLOB</i> , <i>TINYTEXT</i>		<i>BLOB</i> або <i>TEXT</i> з максимальною довжиною 255 (2 ⁸ - 1) символів.
<i>BLOB</i> , <i>TEXT</i>		<i>BLOB</i> або <i>TEXT</i> з максимальною довжиною 65535 (2 ¹⁶ - 1) символів.
<i>MEDIUMBLOB</i> , <i>MEDIUMTEXT</i>		<i>BLOB</i> або <i>TEXT</i> з максимальною довжиною 16777215 (2 ²⁴ - 1) символів.
<i>LOBLOB</i> , <i>LOBTEXT</i>		<i>BLOB</i> або <i>TEXT</i> з максимальною довжиною 4294967295 (2 ³² - 1) символів.

Продовження таблиці 4.1

<i>ENUM</i>	Перерахування	Рядок - об'єкт, який може приймати тільки одне значення, вибирається зі списку значень «значення 1», «значення 2» або <i>NULL</i> . <i>ENUM</i> максимум може мати 65535 різних значень.
<i>SET</i>	Набір	Рядок - об'єкт, який може приймати нуль і більш значний, каждох з яких має бути вибрано зі списку значень «значення 1», «значення 2», ... Поле <i>SET</i> може мати максимум 64 варіанти значень.

4.2 Визначення таблиць. Специфікації

Одним з найбільш складних етапів проектування, є розробка таблиць бази даних для зберігання інформації, так як результати які повинна видавати система не завжди дають повне уявлення про структуру таблиць.

При розробці, краще керуватися такими основними принципами:

- інформація в таблицях не повинна дублюватися. Коли певна інформація зберігається лише в одному місці, то немає необхідності в синхронізації цих даних, і забезпечить ефективність, і виключить можливість не збіги;

- кожна таблиця повинна містити інформацію лише на одну тему, в цьому випадку дані набагато легше обробляти, якщо вони міститися в різних таблицях.

Даний програмний продукт має сім таблиць БД.

Таблиця 4.2 – Власники - *vladelec.dbf*

Найменування поля	Тип	Розмір	Призначення
* <i>Kod_vlad</i>	<i>Number</i>	5	Код власника
<i>Fam</i>	<i>Character</i>	20	Прізвище
<i>Name</i>	<i>Character</i>	20	Ім'я
<i>Oth</i>	<i>Character</i>	20	По батькові
<i>Adres</i>	<i>Character</i>	20	Адреса
<i>Mail</i>	<i>Character</i>	30	Адреса електронної пошти
<i>Tel</i>	<i>Number</i>	20	Номер телефон
<i>Sot</i>	<i>Number</i>	20	Номер мобільного телефону

					КРБ.КІ. 1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

Таблиця 4.3 – Менеджери - *sotrud.dbf*

Найменування поля	Тип	Розмір	Призначення
* <i>Kod_sot</i>	<i>Number</i>	3	Код співробітника
<i>Fam</i>	<i>Character</i>	20	Прізвище
<i>Name</i>	<i>Character</i>	20	Ім'я
<i>Oth</i>	<i>Character</i>	20	По батькові

Таблиця 4.4 – Автомобілі - *avto.dbf*

Найменування поля	Тип	Розмір	Призначення
* <i>Kod_avto</i>	<i>Number</i>	6	Код автомобіля
<i>Kod_vlad</i>	<i>Number</i>	5	Код власника
<i>Kod_Marka</i>	<i>Number</i>	6	Код марки
<i>Model</i>	<i>Character</i>	20	Модель автомобіля
<i>V</i>	<i>Character</i>	5	Об'єм двигуна
<i>Gos_nom</i>	<i>Character</i>	8	Державний номер
<i>Cvet</i>	<i>Character</i>	20	Колір
<i>Tip_kuz</i>	<i>Character</i>	20	Тип кузова
<i>Foto_1</i>	<i>Character</i>	20	Фотографія 1
<i>Foto_2</i>	<i>Character</i>	20	Фотографія 2
<i>Cena</i>	<i>Number</i>	10	Ціна

Таблиця 4.5 – Реєстрація - *avto_in.dbf*

Найменування поля	Тип	Розмір	Призначення
* <i>Kod_in</i>	<i>Number</i>	6	Реєстраційний номер
<i>Akt_nom</i>	<i>Number</i>	6	Номер акта
<i>Kod_avto</i>	<i>Number</i>	6	Код автомобіля
<i>Kod_vlad</i>	<i>Number</i>	6	Код власника
<i>Kod_sot</i>	<i>Number</i>	6	Код менеджера
<i>Data</i>	<i>Date</i>		Дата реєстрації

Таблиця 4.6 – Комплектність - *komplect.dbf*

Найменування поля	Тип	Розмір	Призначення
<i>Kod_avto</i>	<i>Number</i>	6	Код автомобіля
<i>Koleso</i>	<i>Character</i>	15	Оформлення коліс
<i>Zapaska</i>	<i>Logical</i>		Наявність запасного колеса
<i>Instrument</i>	<i>Logical</i>		Наявність інструменту
<i>Music</i>	<i>Logical</i>		Наявність магнітоли
<i>Medic</i>	<i>Logical</i>		Наявність аптечки
<i>Elect_pak</i>	<i>Logical</i>		Наявність електропакета
<i>ABS</i>	<i>Logical</i>		Наявність АБС
<i>GUR</i>	<i>Logical</i>		Наявність гідропідсилювача рульового колеса
<i>Kondic</i>	<i>Logical</i>		Наявність кондиціонера
<i>Korobka</i>	<i>Character</i>	15	Тип КПП
<i>Privod</i>	<i>Character</i>	15	Вид приводу
<i>Info</i>	<i>Memo</i>		Додаткова інформація
<i>Defekt</i>	<i>Memo</i>		Опис дефектів

Таблиця 4.7 – Продаж - *avto_out.dbf*

Найменування поля	Тип	Розмір	Призначення
* <i>Kod_out</i>	<i>Number</i>	6	Код продажу
<i>Data</i>	<i>Date</i>		Дата продажу
<i>Marka_out</i>	<i>Number</i>	6	Код марки
<i>Model_out</i>	<i>Character</i>	20	Модель автомобіля
<i>V_out</i>	<i>Character</i>	5	Об'єм двигуна
<i>Cvet_out</i>	<i>Character</i>	20	Колір
<i>Tipkuz_out</i>	<i>Character</i>	20	Тип кузова
<i>Cena</i>	<i>Number</i>	10	Ціна

Таблиця 4.8 – Марки автомобілів - *marka.dbf*

Найменування поля	Тип	Розмір	Призначення
* <i>Kod_marka</i>	<i>Number</i>	6	Код марки
<i>Marka</i>	<i>Character</i>	20	Марка автомобіля

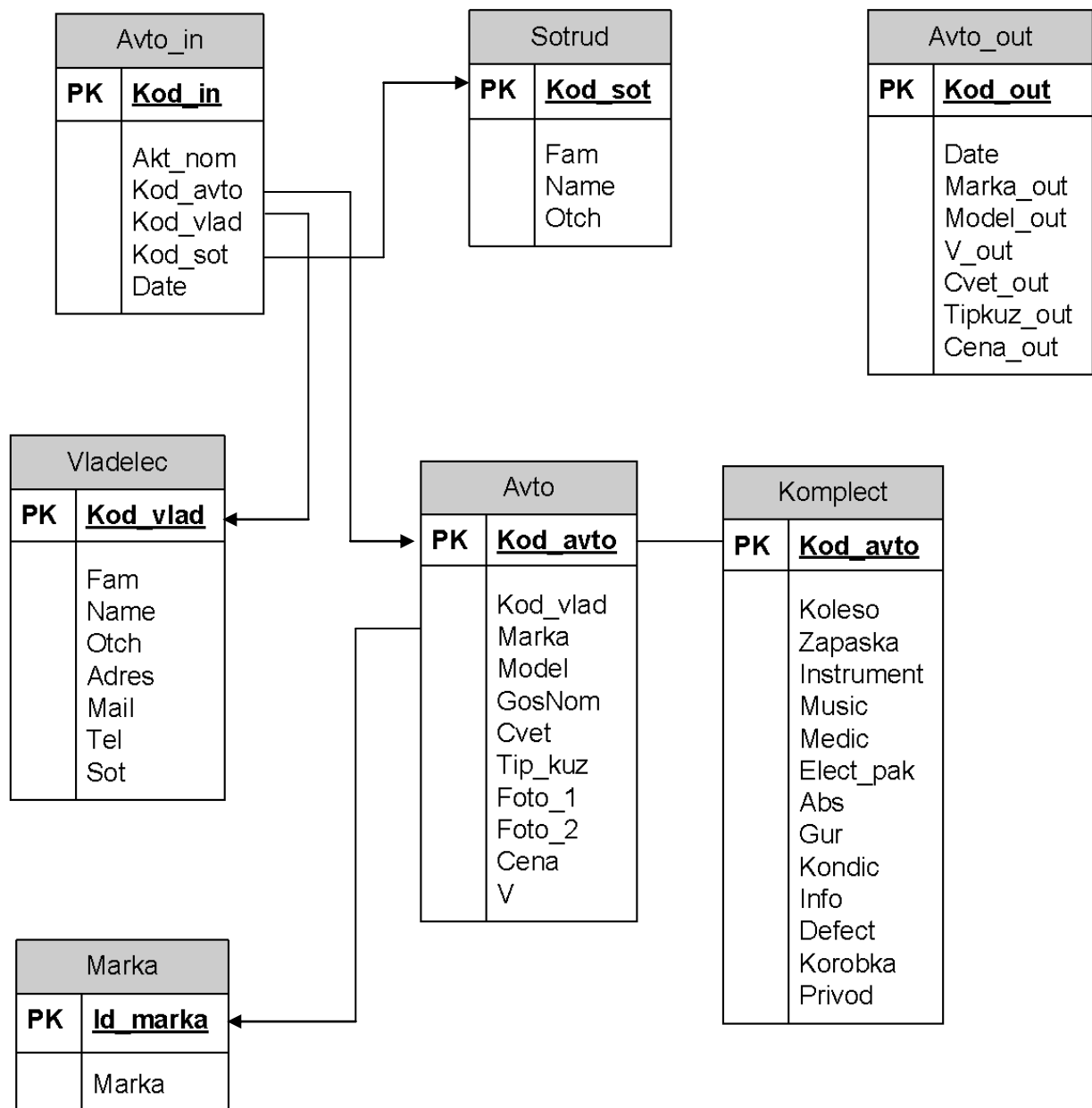


Рисунок 4.5 – Структура бази даних

5 ПРОЕКТУВАННЯ ПРОГРАМНИХ КОМПОНЕНТІВ

5.1 Загальний алгоритм

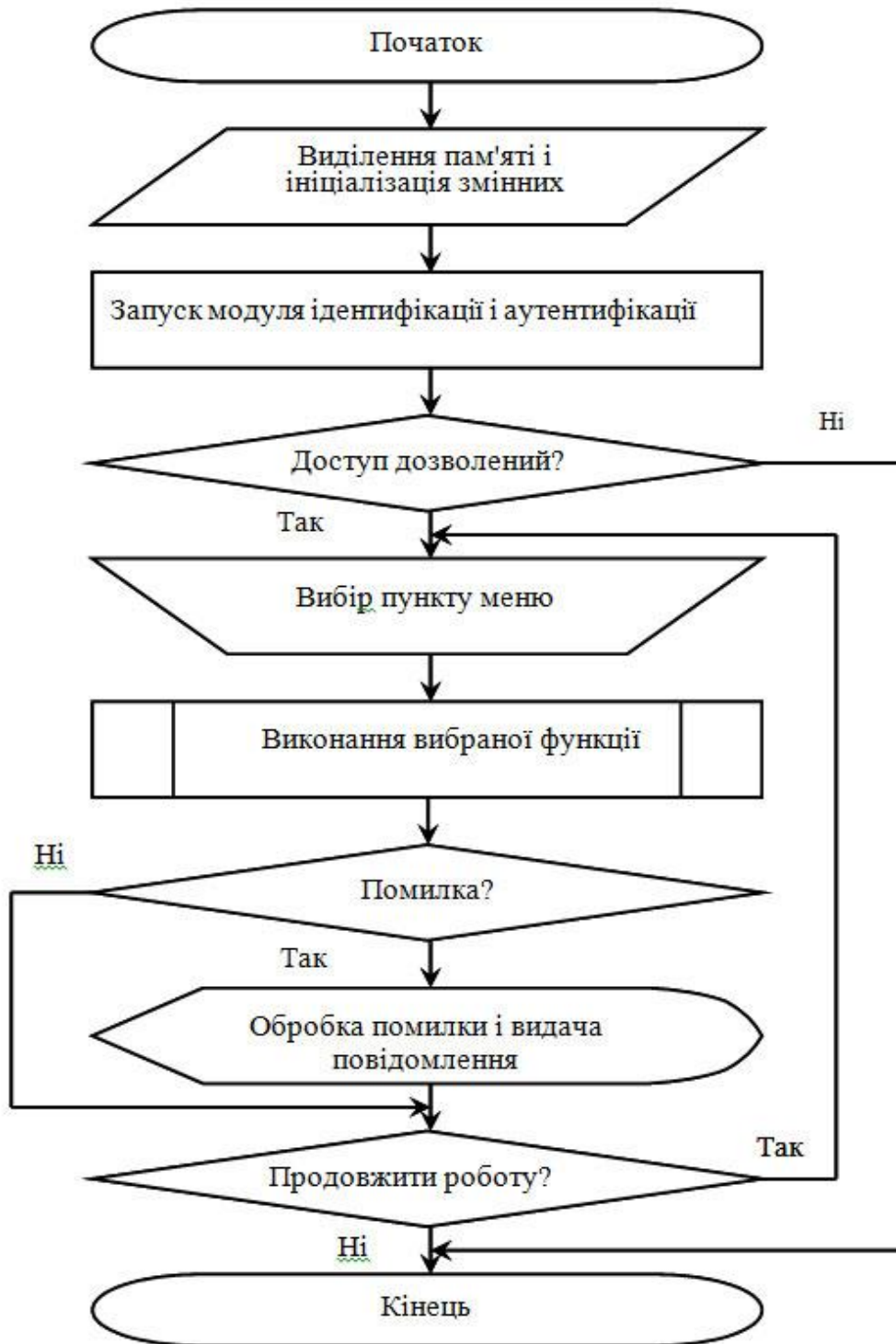


Рисунок 5.1 – Загальний алгоритм

При доступі в програму необхідно проводити ідентифікацію користувача і його прав для доступу до функцій програми. Це диктується тим, що доступ до програми матимуть не лише обслуговуючий персонал інформаційно-довідкової системи, але і будь-які інші користувачі. Форма демонструє цей процес показана на рисунку 5.3.

Рисунок 5.2 - Форма ідентифікації і аутентифікації

Після введення логіна і пароля користувача виробляється перевірка прав доступу.

За наявності прав доступу лише на читання деякі функції будуть не доступні. Створення нових користувачів і призначення прав проводиться в модулі адміністратора.

Таблиця користувачів, яка має такі поля:

Id – ідентифікаційний номер користувача, унікальний номер за яким можна знайти повну інформацію про користувача, тому це ключове поле;

Login – ім'я за яким користувач проходить авторизацію на сайті;

Password – пароль користувача;

Email – адреса електронної пошти користувача;

Prava – права користувача, тобто його можливості на сайті.

Для авторизації користувача потрібна форма, в якій він буде вводити свою інформацію. Але для перевірки цієї інформації спочатку потрібно користувача зареєструвати. Для цього була створена спеціальна форма, яка буде використовуватися як для авторизації так і для реєстрації:

```

<?php
if(isset($_REQUEST['mess']))
// якщо є повідомлення від сервера
echo('<h1><b>'.$mess.'</b>!</h1>');
// показати повідомлення
?>
<h1> Ласкаво просимо, <b><?php echo($login) ?></b>!</h1>
// Ім'я користувача
<form id="log_form" name="log_form" action="autentic.php"
method="post">
// форма для реєстрації/авторизації метод передачі даних
post, через
// заголовок
<label>Логін</label><input id="login" name="логін"
type="text" />
// поле для ім'я користувача
<label>Пароль</label><input id="password" name="пароль"
type="password" /> // поле для паролю користувача
<input id="reg" type="submit" value="Ввід" />
//кнопка підтвердження даних
<a href="#" onclick="return AddRegForm();">Переєстрація</a>
// кнопка реєстрації, що додає необхідні поля і змінює тип
//форми
<?php
if($id_user!=-1)
// якщо користувач авторизувався показати
{
echo('<a href="index.php?event=1"> Вихід </a>');
// можливість виходу
echo('<a href="index.php?event=2&id='.$id_user.'"
onclick="return confirm(\' Ви впевнені?\');">Удалить
профиль</a>');
//можливість удалити свій профіль з сайту
}
?>
</form>

```

Для реєстрації користувача потрібні ще поля: поле для підтвердження паролю, поле для адреси електронної пошти. За допомогою можливостей мови *JavaScript*, до вже існуючої форми додаються необхідні поля і змінюється її значення. Це значно економить ресурси сервера (не створюється додаткова сторінка для реєстрації, код додавання полів виконується на стороні клієнта).

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		62

```

var isEdit=true;    // змінна для виконання функції один раз
function AddRegForm()
    // функція додавання полів і зміни значення форми
{
    if(isEdit)      // якщо функція визивається один раз
    {

        form=document.getElementById("log_form");
        // змінна, що відповідає формі
        a=document.getElementById("reg");
        // змінна, що відповідає кнопці «zareestruvatisia»
        newnode=document.createElement("span");
        // змінна нового елемента на формі
        newnode.innerHTML="<label>Повтор</label><input
id=\"password2\"      name=\"password2\"      type=\"password\"
/><label>Email</label><input      id=\"email\"      name=\"email\"
type=\"text\" />";
        // додавання до нового елемента необхідних полів
        form.insertBefore(newnode,a);
        // додавання до форми нових полів перед кнопкою
        // «zareestruvatisia»
        form.action='autentic.php?event=2';
        // змінюється значення форми з авторизації на реєстрацію
    }
    isEdit=false; // якщо функція виконана, зупиняє її наступне
    // виконання(до оновлення сторінки)
    return true;
}

```

Отже, цю форму можна використовувати як для реєстрації, так і для авторизації, в залежності від параметру «*action*» у формі. Якщо значення цього параметру «*autentic.php*», виконується код на авторизацію, а якщо додана змінна таким чином – «*autentic.php?event=2*», виконуються код на реєстрацію:

```

<?php
include("user_fns.php");
// додається файл з необхідними функціями
if(!isset($_REQUEST['event']))
// якщо форма використовується для авторизації
{
    session_start(); session_unset();
    // розпочато нову сесію і її обнуління
    if      (                isset($_REQUEST['login'])                &&
    isset($_REQUEST['password'])) )
    {

```

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		63

```

//якщо ім'я і пароль користувача задані
$login=$_REQUEST['login']; //змінна для ім'я
$password=$_REQUEST['password']; //змінна для паролю
if (check_password($login, $password, $id, $prava))
{
// якщо так ім'я і пароль існують в БД передаємо інформацію
// про користувача до сесії
$_SESSION["id_user"]=$id;
$_SESSION["prava"]=$prava;

$_SESSION["login"]=$login;
}
}
}
else if($_REQUEST['event']==2)
{
// якщо форма використовується для реєстрації
AddUser($_REQUEST['login'],$_REQUEST['password'],
&$id,$_REQUEST['email'],&$mess);
// додаємо користувача до БД
echo('
<script language="JavaScript" type="text/javascript">
<!--
parent.location.href=\'index.php?mess=\'.$mess.\';
//-->
</script>
');
// перехід на головну сторінку з певним повідомленням від
// серверу
}
?>
<script language="JavaScript" type="text/javascript">
<!--
parent.location.href='index.php';
//-->
</script> //перехід на головну сторінку при авторизації

// Функції, які потрібні для авторизації/реєстрації:
<?php
include_once ("db_fns.php"); //додаємо функції для роботи з
// БД
function check_password($login, $password, &$id, &$prava)
{
// функція для перевірки ім'я і паролю користувача
$privileges=0; //змінна прав користувача на сайті
if (!db_connect()) return false; //підключення до БД
$result = mysql_query("select * from `users` where
`login`='$login' and `password` = password('$password')");

```

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		64


```

// запит з БД строки з даним ім'ям і паролем
if (!$result) return false; // перевірка результату
if (mysql_num_rows($result)>0)
{
// якщо така строка існує повертаємо ідентифікаційний номер
// і права користувача для запису до сесії
$row = mysql_fetch_array($result);
$id=$row["id"]; $prava=$row["prava"]; return true;
}
else return false;
}

function AddUser($login,$password,&$id,$email,&$mess)
{ //функція додавання користувача до БД при реєстрації

if (!db_connect()) return false; //підключення до БД
$check_log=mysql_query("SELECT `id` FROM `users` WHERE
`login`='".$login."'");
// запит чи існують вже такі імена користувача
if(mysql_num_rows($check_log)>0)
{

// якщо існує, повідомити про це користувача і зупинити
// реєстрацію
$mess="Òàêîé ìîâèí óæå çàðåâñèòòðèîîâàí!";
return false;
}
else
{
// якщо такого імені не існує в БД
$result = mysql_query("INSERT INTO `users` VALUES (NULL,
'".$login."', PASSWORD('".$password."'), ' ".$email." ',
0)"); // запит на вставку даних в БД
$mess="Ok!"; // повідомлення про успіх
}
if (!$result) return false; // перевірка результату
$id=mysql_insert_id();
// ідентифікаційний номер користувача
return true;
}

function DelUser($id){ // функція видалення користувача
if (!db_connect()) return false; // підключення до БД
if (!mysql_query("DELETE FROM `users` WHERE `id`='".$id.'"
LIMIT 1 "))
return false; else return true;
// перевірка запиту на видалення
} ?>

```

Після завантаження програми на екрані з'являється основне робоче вікно. В даному вікні відображаються основні дані про наявні автомобілях, і

					КРБ.КІ.1.146-03.2.2	Арк.
						65
Змн.	Арк.	№ докум.	Підпис	Дата		

їх характеристики. З цього вікна можна перейти до будь-якого модулю програми після натискання на відповідну кнопку на панелі кнопок.

Меню, спроектоване на головній формі можна представити таким чином:

1. Довідники
 - Менеджери
 - Власники
 - Марки
 - Продані автомобілі
2. Операції
 - Зареєструвати автомобіль
 - Редагувати дані
 - Продати автомобіль
 - Пошук
3. Вид
 - Розмір шрифту
 - Великий
 - Середній
 - Маленький
 - Колір шрифту
4. Допомога
 - Довідка
5. Про програму
6. Вихід

Довідники	Операції	Вид	Допомога	Довідка	Про програму
-----------	----------	-----	----------	---------	--------------

Рисунок 5.3 – Кнопки навігації

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		66

Наступна частина коду виводить кнопки навігації (рисунок 5.3):

```
<? do_buttons()? >
```

```
function do_buttons() // Кнопки
```

```
{
```

```
? >
```

```
<center><table width="100%" cellpadding=6>
```

```
<td class="bor" align="center" width="25%">
```

```
<a href="salon.php? mode=enter">
```

```
<font size="4" color="#0000FF">Довідники</font></a></td>
```

```
<td class="bor" align="center" width="25%">
```

```
<a href="salon.php? mode=op">
```

```
<font size="4" color="#0000FF">Операції</font></a></td>
```

```
<td class="bor" align="center" width="25%">
```

```
<a href="salon.php? mode=vid">
```

```
<font size="4" color="#0000FF">Від</font></a></td>
```

```
<td class="bor" align="center" width="25%">
```

```
<a href="salon.php? mode=dop">
```

```
<font size="4" color="#0000FF">Допомога</font></a></td>
```

```
<td class="bor" align="center" width="25%">
```

```
<a href="salon.php? mode=dov">
```

```
<font size="4" color="#0000FF">Довідка</font></a></td>
```

```
</table></center>
```

```
<td class="bor" align="center" width="25%">
```

```
<a href="salon.php? mode=pro">
```

```
<font size="4" color="#0000FF">Про програму</font></a></td>
```

```
</table></center>
```

```
<?>
```

5.2 Допоміжні вікна

При натисканні на кнопку «Довідники \ Менеджери» - відкривається довідник менеджерів. У цьому вікні відображені дані про персонал (прізвище, ім'я, по батькові).

					КРБ.КІ.1.146-03.2.2	Арк.
						67
Змн.	Арк.	№ докум.	Підпис	Дата		

Менеджери

Перегляд	Пошук	Сортування	Фільтр	Новини	Редагувати	Додавання	Видалити	Друк
Код менеджера	Ім'я	Прізвище	Посада	Домашній телефон	Робочий телефон			
1	Глеб	Колосков	Головний бухгалтер	10-20-33	20-34-53			
2	Маша	Кузькина	Диспетчер	11-11-11	22-22-22			
3	Олег	Васечкин	Продавець	33-33-33	44-44-44			
4	Ольга	Іванова	Продавець	55-55-55	66-66-66			
5	Андрей	Зайцев	Продавець	77-77-77	88-88-88			
6	Іван	Козлов	Продавець	99-99-99	12-34-56			

Рисунок 5.4 – Менеджери

При натисканні на кнопку «Пошук» - Відбувається пошук по БД. Попередньо в поле необхідно ввести параметр пошуку (шукане значення), потім вибрати де шукати (ім'я, прізвище) і натиснути на кнопку "Пошук". Якщо в БД міститься запис з таким значенням курсор таблиці переміститься на неї, якщо ж ні, то програма видасть повідомлення про те що немає такого запису.

При натисканні на кнопку «Редагувати» - Внизу вікна з'являється панель, куди автоматично переносяться дані з виділеного запису. Після внесення необхідних коригувань потрібно натиснути на кнопку «Зберегти», в іншому випадку «Скасування». За натисканні на кнопку «Сортувати» - Відбувається сортування записів в таблиці в алфавітному порядку (для цифрових полів - по зростанню). При натисканні на кнопку «Додавання» - Внизу вікна з'являється панель для введення даних про нового співробітника. Після введення слід натиснути на кнопку «Зберегти», в іншому випадку «Скасування».

При натисканні на кнопку «Додавання» - Внизу вікна з'являється панель для введення даних про нового співробітника. Після введення слід натиснути на кнопку «Зберегти», в іншому випадку «Скасування».

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

По натисканню на кнопку «Видалити» - З'являється запит на підтвердження видалення. Якщо Ви вибрали «Yes» то виділена запис буде видалена.

При натисканні на кнопку «Довідники \ Власники» - відкривається довідник власників. У цьому вікні відображені дані про власників автомобілів (прізвище, ім'я, по-батькові, адреса, E-mail, телефон, стільниковий) які зареєстрованих в базі даних.

При натисканні на кнопку «Шукати» - Відбувається пошук по БД. Попередньо в поле необхідно ввести параметр пошуку (шукане значення), потім вибрати де шукати (прізвище, № телефону) і натиснути на кнопку "Шукати". Якщо в БД міститься запис з таким значенням курсор таблиці переміститься на неї, якщо ж ні, то програма видасть повідомлення про те що немає такого запису.

Власники

Перегляд	Пошук	Сортування	Фільтр	Новини	Редагувати	Додавання	Видалити	Друк
Код клієнта	Прізвище	Ім'я	По-батькові	Адреса	e-mail	Телефон		
1	Іванов	Іван	Іванович	Київ	10-20-33	20-34-53		
2	Сідоров	Сідор	Сідорович	Одеса	11-11-11	22-22-22		
3	Хохловський	Олег	Олегович	Херсон	33-33-33	44-44-44		
4	Руденко	Ольга	Петровна	Миколаїв	51-15-55	63-36-66		

Рисунок 5.5 – Власники

При натисканні на кнопку «Сортувати» - Відбувається сортування записів в таблиці БД в алфавітному порядку (для числових полів - в порядку зростання). Попередньо потрібно вказати поле по якому буде відбуватися сортування.

При натисканні на кнопку «Редагувати» - Внизу вікна з'являється панель, куди автоматично переносяться дані з виділеного запису. Після

внесення необхідних коригувань потрібно натиснути на кнопку «Зберегти», в іншому випадку «Скасування».

При натисканні на кнопку «Довідники \ Продані автомобілі» - відкривається вікно, що містить основну інформацію щодо проданих автомобілів (марка, модель, об'єм, світло, тип кузова, ціна, дата продажу), яких в даний момент фізично немає.

При натисканні на кнопку «Шукати» - Відбувається пошук по БД. Попередньо в поле необхідно ввести параметр пошуку (шукане значення), потім вибрати де шукати (марка, модель) і натиснути на кнопку "Шукати".

Якщо в БД міститься запис з таким значенням курсор таблиці переміститься на неї, якщо ж ні, то програма видасть повідомлення про те що немає такого запису.

Продані автомобілі

Перегляд	Пошук	Сортування	Фільтр	Новини	Редагувати	Додавання	Видалити	Друк
Код автомобіля	Марка	Модель	Комплектація	Ціна	Рік випуску	Об'єм двигуна	Потужність двигуна	Тип коробки
1	Mazda	6	Active AT	999000	2012	1999	150	Автомат
2	Mazda	3 MPS	2.3 Turbo 6MT	1266000	2012	2261	260	Механіка
3	Mazda	CX-9	3.7 AT6 4WD Sport	1919000	2012	3726	277	Автомат
4	Mitsubishi	ASX	1.8 Intense CVT (S10)	979000	2016	1798	140	Вариатор
5	Lada	2111	Standart	300000	2011	2000	200	Механіка
6	Mitsubishi	Lancer Evolution X	2.0 MR Ultimate (S03 RT)	2139000	2018	1988	295	Робот
7	Subaru	Forester	2.5XS MT (WV)	1341400	2012	2498	172	Механіка
8	Chevrolet	Camaro	6.2 AT 2SS	2635000	2013	6162	405	Автомат
9	Mitsubishi	Pajero Sport	3.0 Instyle AT (S07)	1599000	2012	2999	220	Автомат
10	Subaru	WRX STI Hatchback	E-5AT EA	2205600	2011	2457	300	Автомат
11	Toyota	Land Cruiser Prado	4.0 AT Lux 7-seater	2923000	2017	3956	282	Автомат
12	BMW	X6	X6 xDrive 35i	2999000	2018	2979	305	Механіка

Рисунок 5.6 – Продані автомобілі

При натисканні на кнопку «Сортувати» - Відбувається сортування записів в таблиці БД в алфавітному порядку (для числових полів - в порядку зростання). Попередньо потрібно вказати поле по якому буде відбуватися сортування.

При натисканні на кнопку «Друк» - Відбувається експорт даних з таблиці БД в MS Excel. Після закінчення створення звіту відкриється вікно MS Excel, в якому будуть поміщені дані.

При натисканні на кнопку «Назад» - Закривається довідник проданих автомобілів, відбувається повернення до основного робочого вікна.

При натисканні на кнопку «**Операції \ Зареєструвати автомобіль**» - відкривається вікно реєстрації автомобіля.

Це вікно призначене для додавання нового запису про автомобіль і його власника, комплектності та фотографій до відповідних баз даних.

Код автомобіля	<input type="text"/>
Марка автомобіля	<input type="text"/>
Серійний номер	<input type="text"/>
Рік випуску	<input type="text"/>
Об'єм двигуна	<input type="text"/>
Колір	<input type="text"/>
Нотатки	<input type="text"/>
Сума	<input type="text"/>
<input type="button" value="Записати"/> <input type="button" value="Скасування"/>	

Рисунок 5.7 – Реєстрація автомобілів

Після заповнення всіх полів слід натиснути на кнопку «Записати», після чого якщо всі поля заповнені коректно відбувається запис в БД.

При необхідності можна відчистити все поля разом, для цього необхідно натиснути на кнопку «Скасувати».

При натисканні на кнопку «**Операції \ Редагувати дані**» - відкривається вікно редагування запису.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		71

У вікні редагування відображається поточна виділена запис, для того щоб внести коригування в потрібний запис, слід спочатку виділити її.

При натисканні на кнопку «Операції \ Продати автомобіль»- система повідомить на підтвердження видалення. Якщо Ви вибрали «Yes», то в фоновому режимі відбувається копіювання основних даних в таблицю «Продані», після чого запис в основній таблиці видаляється.

Продаж

Код заказа	<input type="text"/>	Номер замовлення	<input type="text"/>	Марка автомобіля	<input type="text"/>
Код клієнта	<input type="text"/>			Марка автомобіля	<input type="text"/>
Прізвище	<input type="text"/>			Серійний номер	<input type="text"/>
Ім'я	<input type="text"/>			Рік випуску	<input type="text"/>
Телефон	<input type="text"/>			Об'єм двигуна	<input type="text"/>
Нотатки	<input type="text"/>			Колір	<input type="text"/>
Документ	<input type="text"/>	Номер замовлення	<input type="text"/>	Нотатки	<input type="text"/>
Номер документа	<input type="text"/>			Сума	<input type="text"/>
Адреса	<input type="text"/>			Дата	<input type="text"/>
				Код продажу	<input type="text"/>
				Опис продажу	<input type="text"/>

Відомості про машини

Рисунок 5.8 – Продаж автомобілів

Для правильного введення і наочності на формі є можливість переглянути дані про автомобілі, які проходили на замовлення, шляхом натискання кнопок "Всі автомобілі", "Автомобілі в наявності" і "продані автомобілі", розташованих внизу в поле "Відомості про машинах".

На рис. 5.9 представлений результат виконання запиту після натискання кнопки "Всі автомобілі".

Текст запиту на всі автомобілі:

```

SELECT Заказы.КодМашины, Машины.МаркаАвтомобиля,
Машины.СерийныйНомер, Таблица_описание_продажи.ОписаниеПродажи
FROM Таблица_описание_продажи INNER JOIN (Машины INNER JOIN
Заказы ON Машины.КодМашины = Заказы.КодМашины) ON
Таблица_описание_продажи.КодТипПродажи = Заказы.КодТипПродажи;

```

Текст запиту на популярність автомобілів:

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72


```

SELECT DISTINCTROW Машины.МаркаАвтомобил, First
(Таблица_описание_продажи.ОписаниеПродажи) AS [First -
ОписаниеПродажи], Count(*) AS [Count - Таблица_описание_продажи]
FROM Таблица_описание_продажи INNER JOIN (Машины INNER JOIN
Заказы ON Машины.КодМашины = Заказы.КодМашины) ON
Таблица_описание_продажи.КодТипПродажи = Заказы.КодТипПродажи
GROUP BY Машины.МаркаАвтомобил HAVING
(((First(Таблица_описание_продажи.ОписаниеПродажи))='продано'));

```

Код	Марка	Серійний номер	Стан
8	Toyota Avensis	1223F4534	Продано
9	Toyota Land Cruiser	S456345645	В наявності
8	Toyota Avensis	1223F4534	В наявності
6	Suzuki Boleno	22-342D-34	В наявності
6	Suzuki Boleno	22-342D-34	Повернення
7	Honda Accord	23353457	Продано
7	Honda Accord	23353457	Повернення
9	Toyota Land Cruiser	S456345645	Повернення
7	Honda Accord	23353457	В наявності

Рисунок 5.9 – Результат виконання запиту "Всі автомобілі"

На рис. 5.10 представлений результат виконання запиту на популярність.

Марка	Стан	Кількість
Toyota Avensis	Продано	3
Toyota Land Cruiser	Продано	2

Рисунок 5.10 – Результат виконання запиту на популярність

Текст запиту на кількість замовлених автомобілів за робочий день.

```

SELECT      Заказы.КодМашины,      Машины.МаркаАвтомобиля,
Машины.СерийныйНомер, Заказы.ДатаРазмещения

FROM Машины INNER JOIN Заказы ON Машины.КодМашины =
Заказы.КодМашины

WHERE (((Заказы.ДатаРазмещения)=Date()));

```

Код	Марка	Серійний номер	Дата розміщення
8	Toyota Avensis	1223F4534	19/05/2020

Рисунок 5.11 – Результат виконання запиту заказів за день

При натисканні на кнопку «**Операції \ Пошук**» відкривається вікно пошуку.

Попередньо потрібно ввести параметр для пошуку і вибрати поле (марка, модель), по якому буде проводитись пошук. Якщо в БД міститься запис з таким значенням курсор таблиці переміститься на неї, якщо ж ні, то програма видасть повідомлення про те що немає такого запису.

За натисканні на кнопку «**Операції \ Довідка**» - Відкривається довідкова система по програмі.

За натисканні на кнопку «**Вихід**» - програма буде закрита.

За натисканні на кнопку «**Сформувати запит**» - Відкривається вікно формування запитів. Так само відкрити це вікно можна за допомогою меню: Файл \ Створити запит.

6 ЕКОНОМІЧНА ЧАСТИНА

6.1 Організаційно-економічне і маркетингове обґрунтування проекту

Порівняльний техніко-економічний аналіз проекту

В даній дипломній роботі була розроблена система для автоматизованого створення сайту. За допомогою цієї системи можна створити автосалон. Графічний інтерфейс програми зрозумілий, тому простий у використанні.

Для розробки цієї системи були обрані такі засоби реалізації:

- операційна система *Windows 2000/XP/Vista/Seven*;
- Програмне забезпечення *web-серверу Apache* та СУБД *MySQL*;
- інсталятор *ActualInstaller 3.8*.

Організаційне обґрунтування проекту

Метою даного дипломного проектування є підвищення ефективності обліку автосалону і якісне поліпшення обслуговування. Виходячи з сучасних вимог, що пред'являються до якості роботи фінансової ланки підприємства, слід зазначити, що його ефективна робота цілком залежить від рівня оснащення компанії інформаційними засобами на базі комп'ютерних програмних засобів обліку. Проектування високорівневих методів інформатики і програмування працівника автосалону здійснюється з використанням реляційної бази даних.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

Для досягнення позначеної мети були поставлені наступні задачі:

- а) опис предметної області;
- б) аналіз існуючих інформаційно-довідкових систем автосалонів;
- в) проектування структури системи:
 - 1) розробка загальної архітектури;
 - 2) розробка структури бази даних;
 - 3) розробка програмних компонентів.
- г) розробка інтуїтивно зрозумілого інтерфейсу;
- д) тестування системи.

Порівняно з аналогами даний проект значно дешевший і легкий у використанні.

Класифікаційна оцінка різновиду проекту

– клас - монопроект, оскільки проект орієнтований на певне середовище застосування і порівняно невеликий;

– тип - організаційно-технічний, оскільки характеризується показниками швидкості, продуктивності і якістю роботи, а так само забезпечує надійне зберігання інформації;

– вид - комбінований, оскільки містить інноваційні і ін. види;

– тривалість - короткостроковий, оскільки створюється за порівняно малі терміни;

– складність - простий;

– рівень - галузевий.

Етапи виконання проекту:

а) підготовчий етап (лютий - березень) - систематизація, поповнення технічних знань, практичних навиків по темі проекту, аналіз області, що розробляється, збір технічних і економічних необхідних для проекту даних;

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

б) основний (березень - квітень) - планування предметної області проекту, розробка звідного плану проекту, створення інформаційної системи в технічному варіанті, загальна перевірка системи;

в) завершальний (квітень - травень) - відладка програми і здача проекту в експлуатацію.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

Склад робіт проекту, їх тривалість

Розрахунок параметрів мережного графіка:

- тривалість робіт, t_{ij} ;
- ранній початок робіт, $T_{рн}$;
- раннє закінчення робіт, $T_{ро}$;
- пізнє початок робіт, $T_{пн}$;
- пізнє закінчення робіт, $T_{по}$;
- повний резерв часу роботи, $R_{п}$;

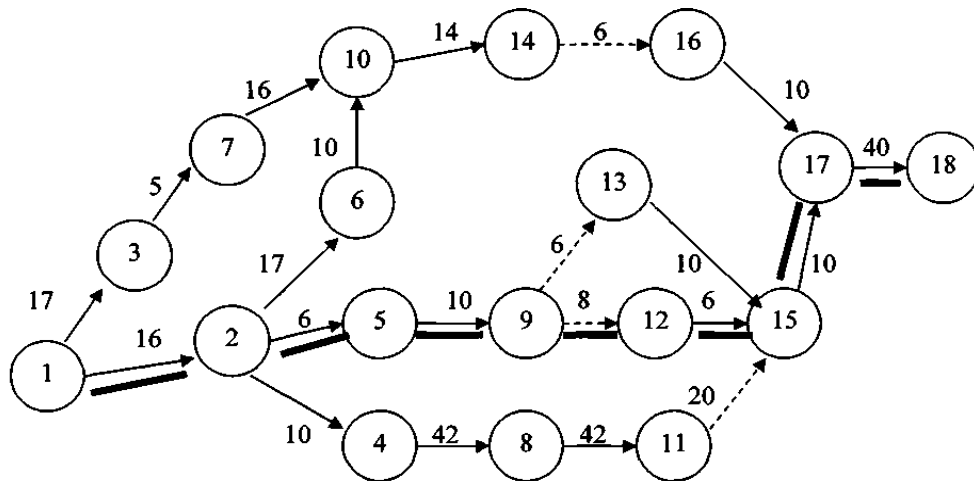


Рисунок 6.1 - Мережний графік робіт проекту

Таблиця 6.1 - Склад робіт і їх тривалість

Код роботи	Назва робіт	Тривалість, дні
1	2	3
1-2	Збір матеріалів для проекту	16
1-3	Аналіз вимог	17
2-4	Постановка завдання	10
2-5	Установка програмних засобів	6
2-6	Планування структури проекту	17
4-8	Планування інтерфейсу програми	42
5-9	Розробка ПО (предметна область)	10
6-10	Розробка моделі і програмного застосування	10
7-10	Розробка призначеного для користувача інтерфейсу	16
8-11	Створення схеми побудови графічного модуля	42
9-12	Фіктивна робота	8
9-13	Фіктивна робота	6
10-14	Реалізація графічного модуля	14
11-15	Фіктивна робота	20
12-15	Внесення необхідних даних	6
13-15	Створення супроводжуючої документації	10
14-16	Фіктивна робота	6
15-17	Тестування робочої програми	10
16-17	Відладка проекту і контрольне тестування	10
17-18	Підготовка документації і здача проекту	40

Таблиця 6.2 - Розрахунок параметрів мережного графіка

Кількість	Робота		Ранній термін звернення події, p, T_i	Тривалість роботи, t_{ij}	Ранній термін звернення події, p, T_j	Пізній термін звернення події, p, T_i	Тривалість роботи, t_{ij}	Пізній термін звернення події, p, T_j	Повний резерв, R_{ij}
	i	j							
1	2	3	4	5	6	7	8	9	10
-	1	2	0	16	16	0	16	16	0
-	1	3	0	17	17	72	17	89	72
1	2	4	16	10	26	16	10	26	0
1	2	5	16	6	22	110	6	106	84
1	2	6	16	17	33	83	17	100	67
1	3	7	17	5	22	89	5	94	72
1	4	8	26	42	68	26	42	68	0
1	5	9	22	10	32	106	10	116	84
1	6	10	33	10	43	100	10	110	67
1	7	10	22	16	38	94	16	110	72
1	8	11	68	42	110	68	42	110	0
1	9	12	32	8	40	116	8	124	84
1	9	13	32	6	38	114	6	120	82

Продовження табл. 6.2

1	2	3	4	5	6	7	8	9	10
2	10	14	38	14	52	110	14	124	72
1	11	15	110	20	130	110	20	130	0
1	12	15	40	6	46	124	6	130	84
1	13	15	38	10	48	120	10	130	82
1	14	16	52	6	58	124	6	130	72
3	15	17	130	10	140	130	10	140	0
1	16	17	58	10	68	130	10	140	72
2	17	18	140	40	180	140	40	180	0
Кількість днів без оптимізації						180			

В результаті розрахунку параметрів виявлено, що без оптимізації тривалість проекту складає 180 днів. Для того, щоб оптимізувати проект, менеджер проекту повинен використовувати шляхи з найбільшим повним резервом. Це шляхи: 2-5, 5-9, 9-12, 12-15, у яких $R_{ij} = 84$;

Маркетингове обґрунтування

Потенційним регіоном і ринком збуту даного програмного продукту є Україна, а так само територія СНД. Програмний продукт складається з власне самої програми і набору класів та бібліотек. Головними конкурентами проекту є більш розрекламовані і відомі аналоги - подібні програмні продукти, що знаходяться на ринку збуту не перший рік. У нашому випадку доцільно проводити рекламу продукту за допомогою спеціалізованих ЗМІ, наприклад, спеціалізованих періодичних видань, а так само розміщувати рекламу на Інтернет-сайтах.

					КРБ.КІ.1.146-03.2.2				Арк.
Змн.	Арк.	№ докум.	Підпис	Дата					81

6.2 Економічні розрахунки проекту

6.2.1 Визначення трудомісткості розробки ПП

ПП, що розробляється, має за аналог - ПЗ введення інформації $V_0 = 5750$ умовних машинних команд з трудомісткістю $T_{np} = 654$ чол/г. Трудомісткість ПП, що розробляється, визначається по кожному етапу окремо на підставі трудомісткості аналогу з урахуванням складності розробки, ступеня новизни і ступеня використання в розробці стандартних модулів на підставі формул:

$$T_{T3} = T_p - L_1 * K_n \quad (6.1)$$

$$T_{np} = T_p * L_2 * K_n \quad (6.2)$$

$$T_{pp} = T_p * L_3 * K_n * K_T \quad (6.3)$$

$$T_{nn} = T_p * L_4 * K_n \quad (6.4)$$

де T_p - укрупнена норма часу на розробку аналога ПЗ, чол/год, яка коректується поправочним коефіцієнтом, що враховує умови розробки ПЗ, тобто в умовах комп'ютера, $K_n = (0,7+0,8)$, тобто:

$$T_p = 654 * 0,8 = 523,20 \text{ чол/год}$$

Оскільки даний проект має аналоги, то його можна віднести до ступеня новизни: У L_j питома вага j-го етапу розробки (залежно від ступеня новизни і відповідних стадій):

$$L_1 = 0,15$$

$$L_2 = 0,16$$

$$L_3 = 0,55$$

$$L_4 = 0,14$$

де K_n - поправочний коефіцієнт, що враховує ступінь новизни, в даному випадку $K_n = 0,7$, K_T - поправочний коефіцієнт, що враховує ступінь використання в розробці типових програм $K_T = 0,8$. Тоді:

$$T_{T3} = 523,20 * 0,15 * 0,7 = 54,94 \text{ (чол./день)}$$

$$T_{np} = 523,20 * 0,16 * 0,7 = 58,59 \text{ (чол./день)}$$

$$T_{pp} = 523,20 * 0,55 * 0,7 * 0,8 = 161,14 \text{ (чол./день)}$$

					КРБ.КІ. 1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		82

$$T_{BH} = 523,20 * 0,14 * 0,7 = 51,27 \text{ (чол./день)}$$

$$T_{\text{общ}} = 54,94 + 58,59 + 161,14 + 51,27 = 325,94 \text{ (чол./день)}$$

$$T_{\text{пп}} = \sum \frac{T_{ij}}{8.0 * 0.73 * 360} \quad (6.5)$$

$$T_{\text{пп}} = (54,94 + 58,59 + 161,14 + 51,27) / (8.0 * 0.73) = 112 \text{ (днів)}$$

6.2.2 Визначення ціни ПП

Оскільки ПП розглядається і створюється як продукція виробничо-технічного призначення, що допускає багатократне тиражування і відчуження від безпосередніх розробників, значить:

$$Ц = K * C + П_r, \quad (6.6)$$

де C - витрати на розробку програмної продукції (кошторисна собівартість); K - коефіцієнт обліку витрат на виготовлення дослідного зразка ПП як продукції виробничо-технічного призначення;

$П_r$ - нормативний прибуток, що розраховується по формулі:

$$П_r = \frac{(C - C_m) * P_n}{100} \quad (6.7)$$

де P_n - норма рентабельності 25%; C_m - матеріальні затрати, грн.

Витрати на розробку програмної продукції можуть бути представлені у вигляді кошторису витрат, що включає наступні статті:

– матеріали.

Витрати на матеріали визначаються по формулі (6.8):

$$C_m = K_{\text{тр}} * \sum C_i * V_i \quad (6.8)$$

де $K_{\text{тр}}$ - коефіцієнт транспортно-заготовчих витрат; C_i - ціна одиниці i -го матеріалу, грн; V_i - придбана кількість i -го матеріалу.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		83

Таблиця 6.3 - Витрати на матеріали

Матеріал	Опис матеріалу	Кількість	Ціна/од.	Сума, Грн.
Бумага		1	45,00	45,00
SSD-диск	TeamGroup 512 Gb	1	999,00	999,00
Flash-накопичувач	Wansenda 32Gb (китайській бренд)	2	282,00	564,00
Всього:			1608,00	

$C_m = 1608 \text{ грн.}$

Спеціальне устаткування. Витрати, зв'язані з використанням обчислювальної техніки, визначаються по формулі (6.9):

$$C_{\text{ПК}} = t^{\text{ПК}} * K_{\text{и}}^{\text{ПК}} * C^{\text{ПК}} * K_3^{\text{ПК}} * K_{\text{бд}}^{\text{ПК}} \quad (6.9)$$

де $t^{\text{ПК}}$ - час використання ПК для розробки даного ПП, ч (290); $K_{\text{и}}^{\text{ПК}}$ - поправочний коефіцієнт обліку часу використання ПК (1,08); $C^{\text{ПК}}$ - ціна 1 години роботи ПК, грн (0,6); $K_3^{\text{ПК}}$ - 5,0. $K_{\text{бд}}^{\text{ПК}}$ - 1,1 (БД використовується)

Тоді $C_{\text{ЗВМ}} = 290 * 1,08 * 0,6 * 5 * 1,1 = 1033,56 \text{ грн.}$

Основна заробітна платня. У статтю включається основна заробітна платня трьох виконавців, безпосередньо зайнятих розробкою даного ПП (керівник, розробник), з урахуванням їх посадового окладу (2000, 890 грн відповідно) і часу участі в розробці. Розрахунок ведеться по формулі (5.10):

$$C_{30} = \sum \frac{Z_i * \hat{E}_o}{\alpha * \pi_i} \quad (6.10)$$

де Z_i - середньомісячний оклад i -го виконавця, грн; α - середня кількість робочих днів у місяці (22); π_i - трудомісткість робіт, виконуваних i -м виконавцем, чол-дні.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		84

Тоді:

$$C_{30} = (2000 * 0,1/22) * 20 = 181 \text{ грн. (керівник)}$$

$$C_{30} = (890/22) * 180 = 7282 \text{ грн. (розробник)}$$

$$C_{30} = 181 + 7282 = 7463 \text{ грн. (загалом)}$$

Додаткова заробітна плата. Розрахунок ведеться по формулі (6.11):

$$C_{3д} = C_{30} * \alpha_{д} \quad (6.11)$$

де $\alpha_{д}$ - коефіцієнт відрахувань на додаткову заробітну плату (0,1).

$$\text{Тоді } C_{3д} = 7463 * 0,1 = 746,3 \text{ грн. (загалом).}$$

Відрахування на єдиний соціальний внесок. У статті враховуються відрахування до фондів соціального страхування за встановленим законодавством тарифом від суми основної і додаткової заробітної плати, тобто

$$C_{сс} = \alpha_{сс} * (C_{30} + C_{3д}) \quad (6.12)$$

де $\alpha_{сс}$ - коефіцієнт відрахувань на соціальне страхування (38%).

$$\text{Тоді } C_{сс} = 0,22 * (7463 + 746,3) = 1806,1 \text{ грн. (загалом)}$$

Накладні витрати. У статті враховуються витрати на загальногосподарські витрати, непродуктивні витрати і витрати на управління. Накладні витрати визначають в процентному відношенні до основної заробітної плати, тобто:

$$C_{н} = \alpha_{н} * C_{30} \quad (6.13)$$

де $\alpha_{н}$ - коефіцієнт накладних витрат (50%).

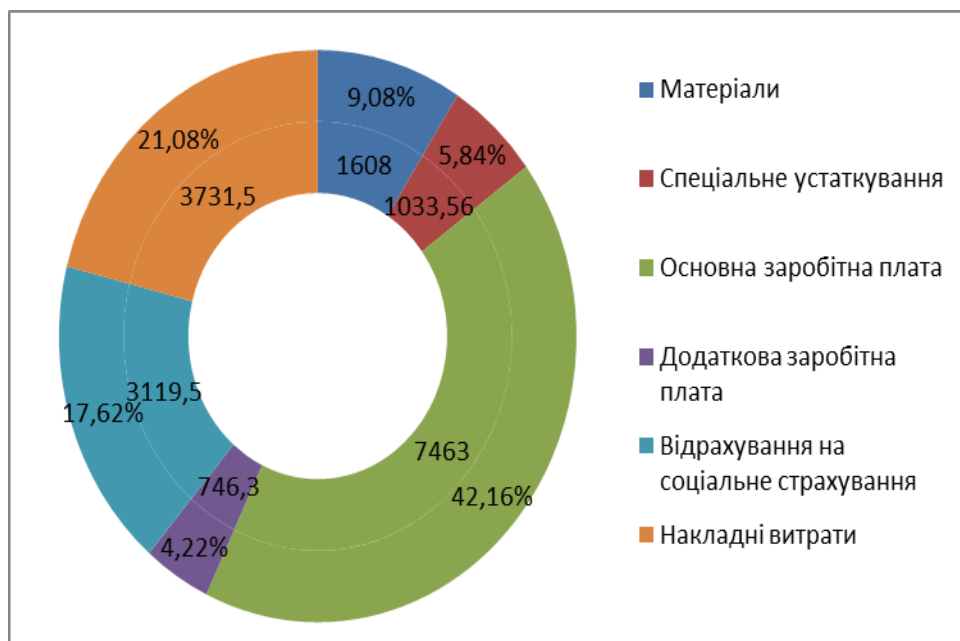
$$\text{Тоді } C_{н} = 0,50 * 7463 = 3731,5 \text{ грн. (загалом).}$$

Виконуємо розрахунок експлуатаційних витрат у користувача по основних статтях витрат: результати розрахунку кошторисної вартості ПП можна звести в таблицю 6.4

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		85

Таблиця 6.4 - Результати розрахунку кошторисної вартості

Найменування статті	собівартість, грн	Питома Вага, %
Матеріали	1608	9,08%
Спеціальне устаткування	1033,56	5,84%
Основна заробітна плата	7463	42,16%
Додаткова заробітна плата	746,3	4,22%
Відрахування на соціальне страхування	3119,5	17,62%
Накладні витрати	3731,5	21,08%



Таким чином,

$$C = 1608 + 1033,56 + 7463 + 746,3 + 1608,1 + 3731,5 = 16190,46 \text{ грн.}$$

$$P_p = (16190,46 - 1608) * 0.30 = 4374,74 \text{ грн.}$$

$$Ц = 1,1 * 17701,86 + 4374,74 = 23846,1 \text{ грн.}$$

6.2.3 Розрахунок капітальних витрат

Розрахунок капітальних витрат, пов'язаних з впровадженням ІС, здійснюється по формулі:

$$K = K_{\pi} + K_{\text{ко}} + K_{\text{во}} + K_{\text{с}} \quad (6.14)$$

де K_{π} - передвиробничі витрати; $K_{\text{ко}}$ - вартість комп'ютерного устаткування; $K_{\text{во}}$ - вартість допоміжного устаткування, необхідного для надійної роботи ІС; $K_{\text{с}}$ - вартість будівництва (реконструкції) у зв'язку з впровадженням ІС (0).

Передвиробничі витрати КП включають всі витрати, пов'язані з проектуванням, розробкою, відладкою і впровадженням ІС - передпроектні і проектні дослідження, постановка завдань і їх алгоритмізація, розробка, відладка і впровадження програмного забезпечення, навчання обслуговуючого персоналу і т.д. В середньому K_{π} складають до 100% загальної вартості впроваджуваних систем.

$$K_{\pi} = 24300,2 * 1,0 = 22660 \text{ грн.}$$

Витрати на комп'ютерне устаткування $K_{\text{ко}}$ визначаються на основі його ринкової вартості (1750 грн.) з використанням прайс-листів з урахуванням витрат на транспортування, монтаж і наладку (10%).

$$K_{\text{ко}} = 0,1 * 1750 + 1750 = 1925,00 \text{ грн.}$$

Витрати на допоміжне устаткування $K_{\text{во}}$ визначаються як 10% від $K_{\text{ко}}$.

$$\text{Тоді } K_{\text{во}} = 0,1 * 1925,00 = 192,50 \text{ грн.}$$

Таким чином, $K = K_{\pi} + K_{\text{ко}} + K_{\text{во}} = 23846,1 + 1925,00 + 192,50 = 25963,6$ грн.

					КРБ.КІ.1.146-03.2.2	Арк.
						87
Змн.	Арк.	№ докум.	Підпис	Дата		

6.2.4 Розрахунок поточних (експлуатаційних) витрат

само з збільшенням якості і легкості створення звіту, це завдання може виконувати одна людина за більш короткий час.

Розрахунок поточних (експлуатаційних) витрат, пов'язаних з впровадженням ІС, здійснюється за формулою (6.15):

$$C = C_{\text{опл}} + C_A + C_{\text{зп}} + C_{\text{п}} + C_{\text{р}} + C_{\text{всп}}, \quad (6.15)$$

де $C_{\text{опл}}$ - річний фонд основної і додаткової оплати праці персоналу, що обслуговуючого ІС з нарахуваннями;

C_A - сума річних амортизаційних відрахувань від вартості основного й допоміжного устаткування; $C_{\text{зп}}$ - вартість витрат на електроенергію за рік;

$C_{\text{п}}$ - вартість річного утримання приміщень. Річний фонд основної заробітної плати персоналу, що обслуговує ІС: $Z_{\text{осн}} = \sum C_{ci} * Z_{ci} + \sum C_{pj} * t_{pj} * \Phi_{pj}$,

де C_{ci} , C_{pj} - чисельність, відповідно, фахівців і-тієї категорії й робочих j-го розряду, що обслуговують ІС;

Z_{ci} - річний фонд заробітної плати спеціаліста і-ої категорії;

$Z_{ci} = 80400$ (ставка виконавця * 12 місяців, $6700 * 12$) ; t_{cj} - годинна тарифна ставка робітника j-го розряду;

$$t_{cj} = 9,2;$$

Φ_{pj} - Річний фонд робочого часу робітника j-го розряду (рекомендується в межах 2016 год.).

Тоді після впровадження:

$$Z_{\text{осн}} = 1 * 6700 * 12 = 80400 \text{ грн.}$$

Фонд додаткової заробітної плати:

$$Z_{\text{доп}} = Z_{\text{осн}} * K_{\text{доп}}, \text{ де } K_{\text{доп}} - \text{коефіцієнт додаткової заробітної плати (0,1).}$$

Після впровадження:

$$Z_{\text{доп}} = 80400 * 0,1 = 8040 \text{ грн.}$$

Нарахування на заробітну плату:

$$Z_{\text{нач}} = (Z_{\text{осн}} + Z_{\text{доп}}) * K_{\text{отч}},$$

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		88

де $K_{отч}$ - коефіцієнт відрахувань на соціальні потреби (у фонди: пенсійний, зайнятості, соціального страхування), $K = 0,22$.

Тоді після впровадження: $Z_{нач} = (80400 + 8040) * 0,22 = 19456,8$ грн.

Таким чином, загальні витрати на оплату праці: $C_{опл} = Z_{осн} + Z_{доп} + Z_{нач}$,

Після впровадження: $C_{опл} = 80400 + 8040 + 19456,8 = 107896,8$ грн. Річна вартість споживаної електроенергії $C_{эл}$ визначається за формулою:

$$C_{эл} = M_y * T_{ко} * Ц_э * K_{и},$$

де M_y - встановлена сумарна потужність комп'ютерного обладнання, кВт (0,3); $T_{ко}$ - річний фонд роботи ПК з урахуванням часу на профілактичні огляди (складає 6320 год); $Ц_э$ - вартість 1 кВт-години електроенергії (1,68грн.);

$K_{и}$ - коефіцієнт інтенсивного використання потужності (0,9).

$$C_{эл} = 0,3 * 6320 * 1,68 * 0,9 = 2866,75 \text{ грн.}$$

Витрати на утримання виробничих приміщень у рік СП розраховуються виходячи з необхідної площі для розміщення всього обладнання ІВ і середньорічних витрат на утримання $1m^2$ виробничої площі (12-20 грн).

$$C_{п} = 17 * 20 * 12 = 4080 \text{ грн.}$$

Розрахунок амортизаційних відрахувань визначається за формулою:

$$C_A = K_{ко} * \alpha / 100,$$

де α - норма амортизаційних відрахувань (для комп'ютерного обладнання 20%).

$$C_A = K_{ко} * 0,2 = 1925 * 0,2 = 385 \text{ грн.}$$

СР складає 6% від вартості комп'ютерного обладнання.

$$C_p = 0,06 * 1750 = 105,00 \text{ грн.}$$

$C_{всп}$ становить 1,5% - 2% від вартості комп'ютерного обладнання

$$C_{всп} = 0,02 * 1750 = 35,00 \text{ грн.}$$

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		89

Таким чином, після впровадження: $C_2 = 27324 + 4080 + 2866,75 + 385 + 105 + 35 = 34795,1$ грн.

Базовим варіантом є діяльність без застосування даного ПП. Структуризація і сортування даних по БД і пошук інформації виконують двоє людей з зарплатою 6700 і 6700 грн. Для їх роботи необхідно буде встановити дві ПК, а також необхідні програмні засоби.

$$Z_{\text{осн}}^1 = (6700 + 6700) * 12 = 160800 \text{ грн.}$$

$$Z_{\text{доп}}^1 = 160800 * 0,1 = 16080 \text{ грн.}$$

$$Z_{\text{нлч}}^1 = (160800 + 160800) * 0,22 = 70752 \text{ грн.}$$

$$C_{\text{опл}}^1 = 160800 + 160800 + 70752 = 392352 \text{ грн.}$$

$$C_{\text{эл}} = 2 * 0,3 * 6320 * 1,68 * 0,9 = 2730,2 \text{ грн.}$$

$$C_{\text{п}} = 17 * 20 * 12 = 4080 \text{ грн.}$$

$$C_{\text{а}} = 2 * 1925 * 0,6 = 2310 \text{ грн.}$$

$$C_{\text{р}} = 2 * 0,06 * 1750 = 210,00 \text{ грн.}$$

$$C_{\text{всп}} = 2 * 0,02 * 1750 = 70,00 \text{ грн.}$$

Таким чином, до впровадження:

$$C_1 = 392352 + 2730,2 + 4080 + 2310 + 210,00 + 70,00 = 401752,2 \text{ грн.}$$

6.2.5 Розрахунок показників економічної ефективності проекту (у користувача)

Очікуваний економічний ефект визначається за формулою: $\mathcal{E}_o = \mathcal{E}_r - E_n * K_{\text{п}}$

де \mathcal{E}_r - річна економія на поточних витратах (грн.), $\mathcal{E}_r = (C^1 - C^2)$, де C^1 , C^2 - відповідні поточні витрати (до і після впровадження проекту);

$K_{\text{п}}$ - одноразові витрати за проектом (25963,6 грн.)

E_n - нормативний коефіцієнт ефективності одноразових витрат ($E_n = 0,3$)

$$\mathcal{E}_r = 64048,2 - 34064,1 = 29984,1 \text{ грн.}$$

$$\mathcal{E}_o = 29984,1 - 0,3 * 24777,5 = 22550,9 \text{ грн.}$$

					КРБ.КІ. 1.146-03.2.2	Арк.
						90
Змн.	Арк.	№ докум.	Підпис	Дата		

Коефіцієнт ефективності розраховується за формулою: $E = \Delta_T / K_{\Pi}$

$E = 29984,1 / 24777,5 = 1,21$ $E > E_n$, отже проект ефективний. Термін окупності розраховується за формулою: $T = 1 / E$

$$T = 1 / 1,21 = 0,83(\text{р.})$$

Таблиця 6.5 - Техніко-економічні показники проекту

Найменування показника		Одиниця виміру	Значення показника
Трудомісткість розробки ПП		дні	180
Ціна ПП		грн	23846,1
Капітальні витрати		грн	25963,6
Поточні витрати	-До впровадження проекту	грн/рік	401752,2
	-Після впровадження проекту		34795,1
Економічний ефект від реалізації продукту		грн/рік	22550,9
Термін окупності		міс.	12
рівень рентабельності			1,2

Зробивши економічні розрахунки, можна зробити висновок про те, що впровадження даного ПП є досить вигідним, про що свідчить рівень рентабельності = 1,2, строк його окупності 12 міс. Цей ПП дозволяє економити час користувача шляхом пришвидшення процесу отримання інформації. Робота самого додатку була оптимізована задля швидкого виконання складових функцій для того, щоб зробити її майже непомітною для користувача.

7 ОХОРОНА ПРАЦІ НА ПІДПРИЄМСТВІ

7.1 Загальні відомості про охорону праці на підприємствах

Охорона праці – це система заходів та засобів, спрямованих на збереження життя, здоров'я та працездатності людини під час трудової діяльності. Вимоги Кодексу законів про працю вимагають створення безпечних та нешкідливих умов праці на всіх підприємствах, установах та організаціях. Власник чи уповноважений ним орган несе відповідальність за забезпечення цих умов.

Ці умови охоплюють безпеку праці на робочому місці, технологічних процесів, машин, механізмів, обладнання та інших засобів виробництва. Також враховуються стан засобів колективного та індивідуального захисту, які використовуються працівниками, а також санітарно-побутові умови. Усі ці аспекти повинні відповідати вимогам нормативних актів, що регулюють охорону праці.

Власник або уповноважений ним орган має зобов'язання використовувати сучасні засоби техніки безпеки, які запобігають виробничому травматизму, і забезпечувати санітарно-гігієнічні умови, які запобігають професійним захворюванням працівників.

Власник або уповноважений ним орган не може вимагати від працівника виконання роботи, яка становить очевидну небезпеку для життя, а також в умовах, які не відповідають законодавству про охорону праці. Працівник має право відмовитися від виконання цієї роботи, якщо виробнича ситуація створює небезпеку для його життя, здоров'я або для людей, які його оточують, а також для навколишнього середовища.

Власник або уповноважений ним орган зобов'язаний систематично проводити інструктаж (навчання) працівників з питань охорони праці і протипожежної безпеки.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		92

Працівники, які працюють у важких та шкідливих умовах, мають право на додаткові пільги та компенсації, які надаються безкоштовно. Ці пільги включають лікувально-профілактичне харчування, молоко або еквівалентні харчові продукти, газовану солону воду, оплачувані перерви санітарно-оздоровчого призначення, скорочення тривалості робочого часу, додаткову оплату праці у підвищеному розмірі та інші пільги, встановлені законодавством.

Закон України "Про охорону праці" вимагає створення служби охорони праці на підприємствах будь-якої форми власності, якщо кількість найманих працівників становить 50 і більше осіб. Така служба має обов'язковий статус згідно з Типовим положенням, затвердженим Державним комітетом України з нагляду за охороною праці у наказі від 15.11.2004 № 255. Служба охорони праці безпосередньо підпорядковується роботодавцю. Враховуючи специфіку виробництва, видів діяльності, кількості працівників, умов праці та інших факторів, роботодавець розробляє Положення про службу охорони праці відповідного підприємства.

У організаціях з кількістю працюючих менше 20 осіб для виконання функцій служби охорони праці можуть залучатися спеціалісти на договірній основі. При цьому такий спеціаліст повинен мати стаж роботи не менше 3 років та пройти навчання з охорони праці.

На підприємствах, де працює менше 50 осіб, створення повноцінної служби охорони праці також не є обов'язковим. У такому випадку функції служби охорони праці можуть виконуватися особами, які мають відповідну підготовку та освіту, такими як фахівці або інженери з охорони праці, за умови сумісництва.

Керівники та фахівці служби охорони праці мають такий же статус посадових осіб, як керівники та спеціалісти основних виробничо-технічних служб, і отримують відповідну заробітну плату.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		93

7.2 Основні завдання і функції служби охорони праці

Отже, згідно зі стандартним положенням щодо служби охорони праці, на підприємстві передбачені такі завдання:

1. Розробка ефективної системи управління охороною праці на підприємстві і підтримка кожного структурного підрозділу та працівника у вдосконаленні діяльності в цьому напрямку.

2. Забезпечення професійної підтримки роботодавця у прийнятті рішень щодо охорони праці.

3. Організація проведення профілактичних заходів, спрямованих на усунення шкідливих та небезпечних виробничих факторів, запобігання нещасним випадкам на виробництві, професійним захворюванням та іншим загрозам життю або здоров'ю працівників.

4. Вивчення та підтримка впровадження досягнень науки та техніки, прогресивних та безпечних технологій, сучасних засобів колективного та індивідуального захисту працівників у виробництві.

5. Контроль за дотриманням працівниками вимог законів та інших нормативно-правових актів з охорони праці, положень галузевих угод (якщо такі є), розділу "Охорона праці" колективного договору та актів з охорони праці, які діють на підприємстві.

6. Інформування та надання роз'яснень працівникам підприємства з питань охорони праці.

Функції служби охорони праці здійснюються на підприємстві з метою виконання вимог законодавства і включають наступні дії:

1. Розробка спільно з іншими підрозділами комплексних заходів, планів роботи і програм з покращення умов праці, запобігання виробничому травматизму та професійним захворюванням.

					КРБ.КІ.1.146-03.2.2	Арк.
						94
Змн.	Арк.	№ докум.	Підпис	Дата		

2. Підготовка проектів наказів з питань охорони праці та їх подання на розгляд роботодавцю.

3. Перевірка дотримання працівниками нормативно-правових актів з охорони праці.

4. Складання звітності з охорони праці.

5. Проведення інструктажів з охорони праці для працівників.

6. Облік та аналіз причин виробничого травматизму.

7. Забезпечення належного оформлення та зберігання документації з охорони праці, а також її передачу до архіву для тривалого зберігання вчасно.

8. Складання переліків професій, посад та видів робіт, для яких необхідно розробити інструкції з охорони (безпеки) праці, залучення керівників підрозділів підприємства до цього процесу.

9. Інформування працівників про основні вимоги законів, нормативно-правових актів та актів з охорони праці, які діють на підприємстві.

10. Розгляд питань, пов'язаних з підтвердженням наявності небезпечної виробничої ситуації, що спричинила відмову працівника від виконання завдання, а також розгляд листів, заяв і скарг працівників щодо дотримання законодавства з охорони праці.

11. Забезпечення підрозділів підприємства нормативно-правовими актами з охорони праці, які діють в межах підприємства, посібниками та навчальними матеріалами з цих питань; організація роботи кабінету з охорони праці, проведення нарад, семінарів та інших заходів з охорони праці.

12. Участь у розслідуванні нещасних випадків, професійних захворювань та аварій на виробництві є однією з найважливіших функцій служби охорони праці. Також спеціалісти з охорони праці беруть участь у складанні санітарно-гігієнічних характеристик робочих місць працівників, що проходять обстеження на професійні захворювання; проведення внутрішнього аудиту з охорони праці та атестації робочих місць згідно з нормативно-правовими актами з охорони праці; складання списків професій та посад, для яких працівники повинні проходити обов'язкові попередні та

					КРБ.КІ.1.146-03.2.2	Арк.
						95
Змн.	Арк.	№ докум.	Підпис	Дата		

періодичні медичні огляди; організація навчання з питань охорони праці та роботи комісії з перевірки знань у цих питаннях.

7.3 Повноваження служби охорони праці

Служба охорони праці на підприємстві має повноваження перевіряти, чи дотримується роботодавець вимог законодавства з охорони праці. В зв'язку з цим, вона має право видачі обов'язкових приписів керівникам структурних підрозділів підприємства щодо усунення виявлених недоліків та отримання від них необхідної інформації, документів та пояснень з питань охорони праці. Припис, виданий спеціалістом з охорони праці, може бути скасований тільки роботодавцем. Припис складається у двох примірниках: один передається керівнику робіт, об'єкта або цеху, а другий залишається і реєструється в службі охорони праці протягом 5 років.

Якщо керівник структурного підрозділу підприємства відмовляється підписати отриманий припис, спеціаліст з охорони праці подає відповідну заяву на ім'я особи, якій адміністративно підпорядкований цей структурний підрозділ, або роботодавцю.

Додатково, служба охорони праці має право:

- зупиняти роботу виробництва, ділень, машин, механізмів та обладнання у випадках, коли порушення створюють загрозу життю або здоров'ю працівників;

- вимагати припинення праці осіб, які не пройшли обов'язковий медичний огляд, навчання, інструктаж, перевірку знань або не мають допуску до відповідних видів робіт або не дотримуються вимог нормативно-правових актів з охорони праці.

- надсилати роботодавцю подання щодо притягнення до відповідальності посадових осіб та працівників, які порушують вимоги щодо охорони праці.

					КРБ.КІ.1.146-03.2.2	Арк.
						96
Змн.	Арк.	№ докум.	Підпис	Дата		

- робити пропозиції щодо заохочення працівників за активну роботу з покращення стану безпеки праці.

- за погодженням з роботодавцем та керівниками підрозділів підприємства, залучати спеціалістів підприємства для проведення перевірок стану охорони праці.

Висновки розділу

У даній частині дипломного проекту були викладені загальні відомості про охорону праці на підприємствах, а також основні завдання, функції і повноваження служби охорони праці. Наведені відомості дозволяють грамотно організувати заходи щодо охорони праці в будь-якій організації, зокрема, на станції технічного обслуговування.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		97

ВИСНОВКИ

У ході виконання випускної кваліфікаційної роботи було отримано проект простого, зручного, недорогого у використанні програмного інформаційного засобу автосалону. З його допомогою користувачі зможуть отримувати необхідну інформацію про авто, а також отримувати різні звіти по нових поступленням.

При розробці були проаналізовані сучасні подібні засоби, і визначена необхідність розробки власного програмного засобу (ПЗ).

При розробці інформаційної підсистеми був пройдений повний цикл проектування програми від постановки завдання замовником до створення проекту.

Розроблений програмний засіб задовольняє всім вимогам, поставленим на етапі постановки завдання.

При розробці була використана система управління базами даних (СУБД) *MySQL*, в основні функції якої входять: введення, обробка, і керування даними.

Розроблений програмний засіб дозволяє досягти наступних ефектів:

- зменшення часу необхідного для пошуку необхідної інформації та складання звітів;
- автоматизації контролю за авторесурсами;
- більш ефективне використання авторесурсів.

Як подальше вдосконалення ПЗ представляється можливим доопрацювання інтерфейсу з метою подальшого підвищення його інформативності, привабливості і зручності.

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		98

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Волгин В.В. Автомобильный дилер: практическое пособие по маркетингу и менеджменту сервиса и запасных частей /В.В.Волгин. - М.: Осъ-89, 1997.

2. Закон України «Про авторське право і суміжні права» [Електронний ресурс]. – Режим доступу: <http://zakon3.rada.gov.ua>. Дата звернення: 10.03.2014.

3. Рекомендації НМК МОНУ з «Комп'ютерної інженерії» від 16.05.2006.

4. Гайворонська Г.С, Сахарова С.В. Вимоги та рекомендації до підготовки випускної кваліфікаційної роботи, Посібник. Одеса, ОДАХ, 2009.

5. *Uafoss* – Украинская Ассоциация Пользователей и Разработчиков Свободного и Открытого Программного Обеспечения [Електронний ресурс]. – Режим доступу: <http://uafoss.org.ua/> – Дата звернення: 03.03.14.

6. Закон України про державну підтримку розвитку індустрії програмної продукції [Електронний ресурс]. – Режим доступу: <http://zakon4.rada> – Дата звернення: 04.03.14.

7. Закон про Концепцію Національної програми інформатизації [Електронний ресурс]. – Режим доступу: <http://zakon4.rada> – Дата звернення: 04.03.14.

8. Закон про про Національну програму інформатизації [Електронний ресурс]. – Режим доступу: <http://zakon4.rada> – Дата звернення: 04.03.14.

9. Закон України «Про телекомунікації» від 18.11.2003 (із змінами від 02.02.2014).

10. Закон України «Про Концепцію Національної програми інформатизації» від 04.02.1998 (із змінами від 11.08.2013).

11. Бібліотека ГОСТов, стандартів і нормативів. Автоматизовані системи. Основні положення. РД 50–680–88 [Електронний ресурс]. – Режим

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		99

доступу: http://www.infosait.ru/norma_doc/10/10101/index.htm. Дата звернення: 10.03.2014.

12. Дублінське ядро ([англ. Dublin Core](#)) – Стандарт [метаданих](#), простий і ефективний набір для опису найширшого діапазону мережевих ресурсів [Електронний ресурс]. – Режим доступу: http://znaimo.com.ua/Дублінське_ядро. Дата звернення: 10.03.2014.

13. *Open-source Integrated Library System* [Електронний ресурс]. – Режим доступу: <http://www.koha.org>. Дата звернення: 10.03.2014.

14. *DSpace is the software of choice for academic, non-profit, and commercial organizations building open digital repositories* [Електронний ресурс]. – Режим доступу: <http://www.dspace.org>. Дата звернення: 10.03.2014.

15. *Flexible Repository Software* [Електронний ресурс]. – Режим доступу: <http://www.eprints.org>. Дата звернення: 10.03.2014.

16. АБІС «Славутич» [Електронний ресурс]. – Режим доступу: <http://slavutich.ed-sp.net>. Дата звернення: 10.03.2014.

17. ДСТУ 3302-96. Система стандартів баз даних. Структура системи словників інформаційних ресурсів.

18. ДСТУ 3329-96 (ГОСТ34.320-96). Система стандартів баз даних. Концепція та термінологія для концептуальної схеми й інформаційної бази.

19. ДСТУ 3330-96 (ГОСТ34.321-96). Система стандартів баз даних. Еталонна модель керування даними.

20. ДСТУ 3918-99 (ISO/IEC 12207-95). Інформаційні технології. Процеси життєвого циклу програмного забезпечення.

21. ДСТУ ISO /IEC 15288. Інженерія систем. Процеси життєвого циклу систем. ISO/IEC 15288:2002, IDT).

22. ДСТУ ISO /IEC TR 12182:2004. Інформаційні технології. Категоризація програмного забезпечення.

23. ДСТУ ISO /IEC 14764:2002. Інформаційні технології. Супровід програмного забезпечення.

					КРБ.КІ.1.146-03.2.2	Арк. 100
Змн.	Арк.	№ докум.	Підпис	Дата		

24. ДСТУ ISO /IEC TR 15271:2008. Інформаційні технології.
Настанови щодо застосування ДСТУ 3918 (процеси життєвого циклу
програмного забезпечення).

					КРБ.КІ.1.146-03.2.2	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		101