

НУБІП України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

УДК 004.9:657

«ПОГОДЖЕНО»

«ДОПУСКАЄТЬСЯ ДО ЗАХИСТУ»

Декан
факультету інформаційних технологій

Завідувач
кафедри комп'ютерних наук

д.пед.н., професор Глазунова О.Г.

к.т.н., доцент Голуб Б.Л.

« » 20 р « » 20 р

МАГІСТЕРСЬКА РОБОТА

на тему: Програмне забезпечення системи обліку технічного
обслуговування автотранспорту в агрофірмі

Спеціальність 121 Інженерія програмного забезпечення

Освітня програма “Програмне забезпечення інформаційних систем”

Орієнтація освітньої програми освітньо-професійна

Орієнтація освітньої програми освітньо-професійна

Гарант освітньої програми

К.Т.Н., доцент
(науковий ступінь та вчене звання)

(підпис)

Голуб Б.Л.
(ПІБ)

Керівник магістерської кваліфікаційної роботи

ст. викладач
(науковий ступінь та вчене звання)

(підпис)

/ Бородкін Г.О. /
(ПІБ)

Науковий консультант магістерської кваліфікаційної роботи

К.Т.Н., доцент
(науковий ступінь та вчене звання)

(підпис)

/ Бородкіна І.Л. /
(ПІБ)

Виконав

(підпис)

/ Жежера В.Р. /
(ПІБ студента)

КИЇВ – 2021

НУБІП України

АНОТАЦІЯ

Мета проекту: метою даної дипломної роботи є розробка інформаційної системи автосервісу, яка дозволить підвищити ефективність управління за рахунок швидкого доступу до інформації про клієнтів та їх автомобілів.

Під час виконання дипломного проекту були досліджені теоретичні та практичні аспекти побудови та функціонування системи обліку технічного обслуговування автотранспорту в агрофірмі.

Також в ході дослідження:

- Розглянуто організацію технологічного процесу на станції технічного обслуговування.
- Розраховано економічну ефективність станції технічного обслуговування.
- Розглянуто безпеку та екологічність усього проекту, екологію та захист навколишнього середовища.

ABSTRACT

The purpose of the project: the purpose of this thesis is to develop an information system for car service, which will increase the efficiency of management through quick access to information about customers and their cars.

In the course of the diploma project the theoretical and practical aspects of construction and functioning of the system of accounting for maintenance of vehicles in the agricultural firm were studied.

Also during the study:

- The organization of the technological process at the service station is considered.
- The economic efficiency of the service station is calculated.

– Safety and environmental friendliness of the whole project, ecology and environmental protection are considered.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Завідувач кафедри
комп'ютерних наук

доц., к.т.н.
(вчене звання і ступінь)

_____ (підпис)

Б. Л. Голуб
(ініціали і прізвище)

« _____ » _____ 20__ р.

**ЗАВДАННЯ
ДО ВИКОНАННЯ МАГІСТЕРСЬКОЇ РОБОТИ СТУДЕНТУ**

Жежері Володимиру Романовичу

(прізвище, ім'я, по батькові)

Спеціальність 121 «Інженерія Програмного Забезпечення»
Освітня програма «Програмне забезпечення інформаційних систем»
Орієнтація освітньої програми освітньо-професійна

1. Тема магістерської роботи: Програмне забезпечення системи обліку

технічного обслуговування автотранспорту в агрофірмі затверджена наказом ректора НУБІП від «29» жовтня 2020 р. № 1636 «С»

2. Термін подання завершеної роботи на кафедру 2020 рік 11 місяць 30 число

3. Вихідні дані до магістерської роботи: дані отримано з загальнодоступних сайтів систем обліку технічного обслуговування.

4. Перелік питань, що підлягають дослідженню:

- 1) Дослідження предметної області, постановка задачі
- 2) Огляд існуючих рішень та літературних джерел.
- 3) Вибір методів дослідження.
- 4) Проектування системи, її випробування (апробація).
- 5) Вибір інструментарію та імплементація системи.

Дата видачі завдання «29» жовтня 2020 р.

Керівник магістерської роботи

_____ (підпис)

Бородкін Г.О.

(прізвище та ініціали)

Завдання прийняв до виконання

_____ (підпис)

Жежера В.Р.

(прізвище та ініціали студента)

	ЗМІСТ	
ВСТУП		3
1. АНАЛІЗ ВИМОГ ДО ЦЗ		5
1.1		5
1.2		8
1.3		11
2. МОДЕЛІ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПРОГРАМНОЇ СИСТЕМИ		17
2.1 Функціональне моделювання		18
2.2 Об'єктне моделювання		23
2.3 Абстракції предметної області		26
3.		28
3.1 Логічна модель даних		28
3.2 Проектування загальної архітектури програмної системи		29
3.3 Організаційна структура програмної системи		33
3.4 Архітектура програмного забезпечення		36
3.4.1.		36
3.4.2.		40
4. ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ		45
4.1		45
4.2 Аналіз даних та результати дослідження		55
4.3 Перевірка якості програмного продукту		55
ВИСНОВКИ		56
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ		58

ВСТУП

В даний момент досить швидко розробляються та покращуються засоби для зберігання великої кількості інформації та даних. Для доступу до цих даних використовуються різні інформаційні системи, які потрібні для зберігання інформації, а також її зміни та передачі.

Основною частиною інформаційної системи є база даних, яка є структурою інформаційної моделі. Створення бази даних є важливим пунктом для подальшого проектування та розробки інформаційної системи.

Бази даних є необхідними практично всім сучасним організаціям. Вони допомагають вести облік клієнтів та товарів. Також бази даних потрібні для автоматизації роботи автосервісу.

Електронні бази даних мають величезну кількість переваг перед паперовим обліком інформації, який застарів і став зовсім незручним, а пов'язано це насамперед із простотою та швидкістю знаходження потрібної інформації. Наприклад, щоб побачити, які марки автомобілів є в асортименті, досить просто відкрити потрібний запит. автосервіс інтерфейс облік???

Ця тема є особливо актуальною, тому що у компаній, що займаються продажем автомобілів, є досить велика клієнтська база, а також досить велика кількість проданих і куплених автомобілів, облік яких з точки зору регламентного технічного обслуговування необхідно відстежувати.

Реалізувати облік такого великого обсягу в електронній базі даних набагато легше, ніж на папері.

Метою даного дипломного проєкту є розробка інформаційної системи заданої предметної області, а саме автосервісу. Для досягнення заданої мети необхідно на першому етапі докладно дослідити предметну область із залученням для цього різноманітних джерел та засобів отримання інформації.

Наступним кроком дослідження має бути порівняльний аналіз аналогів.

Далі з отриманої інформації слід виявити основні функції та завдання, організаційну структуру та типові сценарії роботи, у зручній та зрозумілій формі подати документообіг автосервісу.

Об'єктом дослідження цього дипломного проекту є діяльність автосервісу.

Автосервіс у свою чергу займається безпосередньо технічним обслуговуванням та ремонтом автомобілів.

Предметом дослідження програмні системи, які створені задля автоматизації управлінських процесів в автосервісі агрофірми.

Структура роботи

Магістерська робота складається з пояснювальної записки обсягом у 62 сторінки та містить вступ, 4 розділи, висновок, список використаних джерел.

У першому розділі дипломного проекту наведено опис предметної області, опис вимог до ПЗ, встановлені атрибути якості створюваного ПЗ, розглянуті існуючі рішення та сформульована постановка завдання.

В другому розділі наведено функціональне та об'єктне моделювання та визначені абстракції предметної області.

Третій розділ присвячений проектуванню системи – в ньому розглядається логічна модель даних, загальна архітектуру ПЗ, організаційна структура, а також опис системи у вигляді сайту.

Четвертий розділ присвячений впровадженню системи та аналізу отриманих результатів – в ньому розглянуто інтерфейс програми, проаналізовані дані та результати досліджень та виконана оцінка якості програмного продукту.

Пояснювальна записка містить 38 рисунків та 3 таблиці.

1. АНАЛІЗ ВИМОГ ДО ПЗ

НУБІП України

1.1 Опис предметної області

Автосервіс - це комплекс, який поєднує в собі як ремонт кузова та автоскла, так і такі сучасні технології комп'ютерної діагностики двигуна та стендової перевірки і налаштування розвалу.

НУБІП України

Технічне обслуговування (ТО) - це комплекс операцій або операція щодо підтримки справного стану колісного транспортного засобу (складових частин, систем колісного транспортного засобу) відповідно до інструкцій його

НУБІП України

виробника.

Наявність таких складних та дорогих технологій пов'язана з популяризацією автомобілів та постійним ускладненням їх технічної конструкції та електрообладнання. Також у сучасних сервісах працюють професійні автомеханіки, що дозволяє скоротити час процесу ремонту та покращити якість профілактичних робіт.

НУБІП України

Обслуговування клієнтів автосервісу здійснюється різними шляхами:

1. Огляд

1.1. Огляд автомобіля щодо несправностей;

НУБІП України

1.2. Огляд на прохання клієнта окремої частини машини

2. Діагностика

2.1. Комп'ютерна діагностика двигуна;

2.2. Діагностика на стенді «Схід-розвал».

НУБІП України

3. Кузовні роботи

3.1. Усунення вм'ятин, подряпин, тріщини;

3.2. Антикоровізна обробка;

3.3. Фарбування окремої деталі чи всього кузова.

4. Ремонт

НУБІП України

4.1. Ремонт основних вузлів;

4.2. Ремонт електрообладнання.

5. Шиномонтаж

5.1. Балансування шин;

5.2. Усунення пошкоджень покришок та камери.

Спочатку, приїжджаючи у сервіс, клієнт звертається до адміністратора і може отримати всю необхідну довідкову інформацію про вартість наявних послуг, професійну підготовку механіків та техніків, обрати необхідні процедури та віддати ключі від машини в обмін на талон, що підтверджує факт здачі автомобіля в сервіс.

Потім він може або поспілкуватися безпосередньо з механіком, пояснити йому неполадки та передбачувану причину їх появи, або пройти в кімнату очікування, якщо ремонт займає нетривалий час, або звернеться до адміністратора з проханням викликати таксі.

Під час огляду автомобіля фіксуються такі дані:

1. Дата;
2. Час;
3. Реєстраційні номери автомобіля;
4. Прізвище механіка, який проводить огляд;
5. Знайдені несправності;
6. Підсумкова вартість огляду.

За необхідності, оформляється квитанція з переліком знайдених несправностей, у якій також фіксуються дата, час та перелік несправностей, які мають бути усунені.

Після виконання роботи реєструються також дані механіка:

1. Дата;
2. Години;
3. Прізвище механіка;
4. Обсяги та перелік виконаних робіт.

Також до документів прикладається копія лістингу роботи програми діагностики (логи), бо оригінал надається клієнту.

При виконанні кузовних робіт фіксуються дані про повний опис проблеми

автомобіля та механіка, що виконує ремонт:

НУБІП України

1. Дата;
2. Години;
3. Прізвище механіка;

НУБІП України

4. Ймовірна дата та час завершення робіт;
5. Опис виконаних робіт;
6. Вартість виконаних робіт.

При необхідності друкується звіт про виконану роботу.

Під час виконання операції «Ремонт» фіксуються:

НУБІП України

1. Дата;
2. Години;
3. Прізвище механіка;

4. Номери транспортного засобу та каталожні номери деталей, що

НУБІП України

- замінюються;
5. Назва несправності;
6. Використані для ремонту матеріали та запасні частини.

7. Вартість виконаних робіт.

НУБІП України

НУБІП України

НУБІП України

1.2 Опис вимог до програмного забезпечення

1.2.1 Опис функціональних вимог до ПЗ.

Даний підрозділ описує високо рівневі вимоги, якими повинен володіти програмний продукт, щоб мати можливість добре виконувати покладені на нього функції. Зазначені умови, яких повинен дотримуватись користувач для роботи з веб-сайтом та вказані основні переваги веб-сайту. Приведені можливості веб-сайту, обмеження та показники якості.

Загальний опис та можливості веб-сайту:

- Відображення інформації про актуальні послуги автосервісу;
- Надання контактної інформації;
- Можливість замовлення сервісів та товарів;
- Реєстрація та авторизація користувачів;
- Додавання, редагування, видалення сервісів та товарів;
- Перегляд, редагування, видалення замовлень;
- Формування та експорт звітів.

Вимоги користувачів:

- зовнішній користувач – відвідувач:
 - Перегляд інформації на сайті;
 - Замовлення сервісів та товарів;
 - Реєстрація та авторизація.
- внутрішній користувач – адміністратор:
 - Дії над сервісами та товарами (додавання, редагування, видалення);
 - Дії над замовленнями (перегляд, редагування, видалення);
 - Формування та експорт звітів.

Функціональні вимоги:

- Можливість реєстрації та авторизації користувачів;
- Валідація введених даних;
- Пошук, фільтрація та сортування даних;
- Відображення результатів виконаної процедури;
- Можливість збереження інформації;

Системні вимоги:

Вимоги до середовища виконання:

- Доступ до мережі Інтернет;
- Операційна система Windows 7 або вище;
- Процесор з тактовою частотою не нижче 2.0 ГГц;
- Оперативна пам'ять 2 Гб або вище;

- Веб-браузери:

- Opera не нижче 36.0;
- Google Chrome не нижче 50.0;
- Mozilla Firefox не нижче 41.0.

Вимоги до інтерфейсу:

Інтерфейс веб-сайту повинен бути зрозумілим та зручним, не перевантаженим графічними елементами. Навігаційні елементи та компоненти інтерактивної взаємодії повинні бути виконані у зручній для користувача формі.

На сайті має бути розташоване меню з розділами, а на кожній сторінці повинна бути можливість повернутися на головну.

Меню повинно бути зручним, помітним, бути зверху сторінки, чи збоку, щоб користувачу не довелося його шукати. Якщо розділів сайту багато, і вони мають ще й підрозділи, потрібно зробити випадаюче меню.

До інтерфейсів користувача висувуються наступні вимоги:

- мінімально можливий час відгуку;
- уніфікація розміщення керуючих елементів;

- підтвердження виконання критичних дій;
- мінімізація кількості полів, що заповнюються.

Вимоги до дизайну:

Дизайн має бути адаптивним і відповідати усім сучасним нормам. Він повинен бути виконаний в стилі мінімалізму (нічого зайвого, відволікаючого увагу) в спокійній кольоровій гамі. Важливо також виділяти на сторінках головне (кольором, шрифтом, зображенням), а другорядне – «приховувати», щоб воно не заважало користувачу, але він міг при бажанні дізнатися більше.

Графічні матеріали (фотографії та малюнки) створюються у форматах JPEG (фотографії) з обов'язковою оптимізацією для відображення за розміром і кількістю кольорів.

В текстах потрібно використовувати посилання. Всі посилання повинні бути виділені одним шрифтом, відмінним від шрифту тексту, щоб користувач розумів, що це.

1.2.2 Опис нефункціональних вимог до ПЗ.

Нефункціональні вимоги - тип вимоги, що описує атрибути продуктивності чи якості системи, які мають бути задоволені. Нефункціональні вимоги, як правило, є вимірними та діють як обмеження на проектування рішення.

Фіксують умови, які безпосередньо не належать до функціональності рішення, а скоріше описують умови навколишнього (зовнішнього) середовища, в умовах якого рішення має зберігати ефективність та якість, якими система має мати. Вони також відомі як вимоги до якості або додаткові вимоги.

Дані вимоги описують як повинна працювати система або програмний продукт, і які властивості або характеристики вона повинна мати.

- Не використовує інформацію, що є недійсною або може зашкодити психіці користувачів;
- Не містить реклами, принаймні тієї, що не стосується тематики сайту.
- Не використовує інформацію, заборонену нормами моралі та законодавством;

Нефункціональні вимоги оформляються окремо і повинні враховувати екстремальні особливості, які повинна витримувати система. Одним із найоптимальніших способів виконання нефункціональних вимог на даний момент є використання веб-кластера, що вирішує завдання відмовостійкості (дублює кілька вузлів), масштабованості (можна додавати сервери додатків та бази даних), надійності (дані розподілені по кількох машинах).

До нефункціональних вимог належать такі властивості системи, як обмеження середовища проживання і реалізації, продуктивність, залежність від платформи, розширюваність, надійність тощо. Під надійністю розуміються такі характеристики, як придатність, точність, середнє напруження на відмову, кількість помилок на тисячу рядків програми, кількість помилок на клас.

Вимоги щодо продуктивності – це швидкість, пропускна здатність, час відгуку, використовувана пам'ять. Багато вимог, пов'язані з продуктивністю, повинні бути описані в конкретних варіантах використання, а не в розділі, що відноситься до всієї системи.

Також можна відзначити, що часто нефункціональні вимоги не можуть бути прив'язані до конкретного варіанта використання та повинні бути винесені до окремого списку додаткових вимог до системи.

1.3 Постановка завдання

Веб-сайт повинен бути побудований на клієнт-серверній архітектурі з використанням веб-технологій, що не вимагають додаткового ліцензування.

Серверна частина веб-сайт має базуватися на веб-сервері Apache2 або Nginx та системі керування базами даних MySQL.

Базовою мовою програмування пропонується обрати PHP. Клієнтська частина повинна бути побудована на стандартних веб-технологіях, що не вимагають встановлення додаткових компонентів на комп'ютер.

Сторінки інтерфейсу генеруються сервером за стандартом HTML5, оформлення здійснюється за допомогою таблиць стилів CSS, а інтерактивність сторінок забезпечується скриптами Javascript та асинхронними запитами через AJAX. Інтерфейс повинен коректно працювати у сучасних веб-браузерах Mozilla Firefox та Google Chrome.

Етапи розробки веб-сайту:

Етапи розробки веб-сайту описано у таблиці 1.1.

Таблиця 1.1

Етапи розробки веб-сайту

Етап	Зміст робіт	Результат
1	Створення UML-моделі веб-сайту	Готова UML-модель веб-сайту та згенеровані частини коду
2	Проектування інтерфейсів та розробка графічного дизайну всіх модулів	Графічний дизайн у вигляді макетів формату PSD

3	HTML-верстка інтерфейсів	Шаблони веб-сторінок у вигляді набору файлів HTML, CSS, Javascript
4	Проектування та створення бази даних	Схема даних, таблиці для збереження даних та зв'язки між ними у СУБД MySQL
5	Розробка програмної частини модулів, інтеграція шаблонів, налаштування інтерфейсів	Готовий програмний продукт на сервері розробника

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

1.4 Огляд існуючих рішень

Найкращими з альтернативних систем є:

- 1С: Підприємство. Автосервіс (<https://solutions.1c.ru/catalog/autoservice/features>);
- iDirector (<https://auto.a25.ru>);
- «АвтоДилер» з модулем «Сервіс» (<https://autodealer.ru/soft/service>);
- Bitrix24;
- amoCRM.

Розглянемо ці системи детальніше.

1С: Підприємство. Автосервіс

"1С. Автосервіс 8" - це галузеве спеціалізоване рішення, призначене для автоматизації управління та обліку в автосервісах, станціях технічного обслуговування і автомийках.

Це програмне забезпечення ідеально підходить практично для будь-якої сфери бізнесу. Її різні версії впроваджуються, враховуючи особливості кожної індивідуальної компанії і налаштовуються під її потреби. Впровадження 1С:Підприємство відбувається максимально швидко і під повним контролем замовника.

Зображення інтерфейсу можна побачити на рис. 1.1

Дане рішення є традиційною системою «Управління невеликою фірмою» системи програм "1С: Підприємство" зі збереженням усіх можливостей і механізмів типового рішення, система додатково враховує специфіку ремонтних і обслуговуючих заходів авто бізнесу.

Переваги:

- облік руху грошових коштів в касі і на банківських рахунках;
- аналіз клієнтської бази;
- статистика, звіти та інші показники;
- надійне збереження даних;

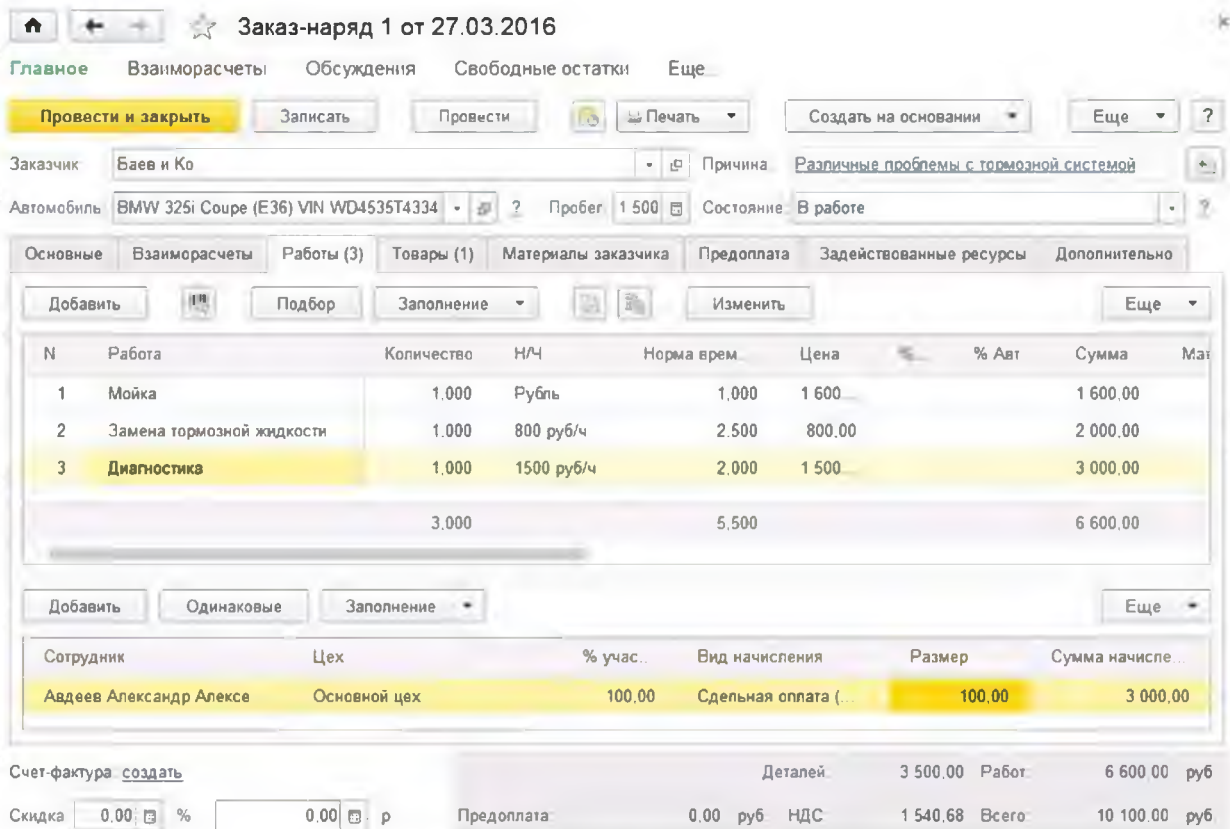


Рис. 1.1 Интерфейс програми IC: Автосервіс

Недоліки:

- важке і специфічне налаштування системи;
- не має доступ з будь-яких пристроїв через інтернет;
- дана система є зареєстрованою на території України;

iDirector

iDirector є сучасним вузькоспеціалізованим рішенням для просунутих автосервісів. Поєднує в собі легкий зрозумілий інтерфейс і потужні модулі. Також має хмарну версію, доступну навіть з мобільних пристроїв.

Клієнт отримує можливість викачувати необхідні документи (рахунки, акти, замовлення-наряди), отримує доступ до історії ремонтів та іншої необхідної для роботи інформації.

За допомогою особистого кабінету клієнт також отримує можливість отримувати консультації ваших майстрів і планувати свої візити в ваш сервіс

здалегідь. Підключіть додаткові можливості, і клієнт зможе спостерігати за процесом ремонту, або переглядати докладні дані діагностики свого автомобіля в режимі реального часу.

Зображення інтерфейсу можна побачити на рис. 1.2

Крім стандартних можливостей, таких як: управління клієнтською базою, введення замовлень, облік складу управління персоналом, є і ряд додаткових можливостей:

Інтеграція з сайтом - дозволяє встановити форму для оформлення замовлень на сайт автосервісу, а також додати онлайн-консультанта.

Таймлайн - порівняння зображення з камер спостереження і зіставлення з фактичним занесенням замовлення в систему, запобігання «свівака» автослюсаря.

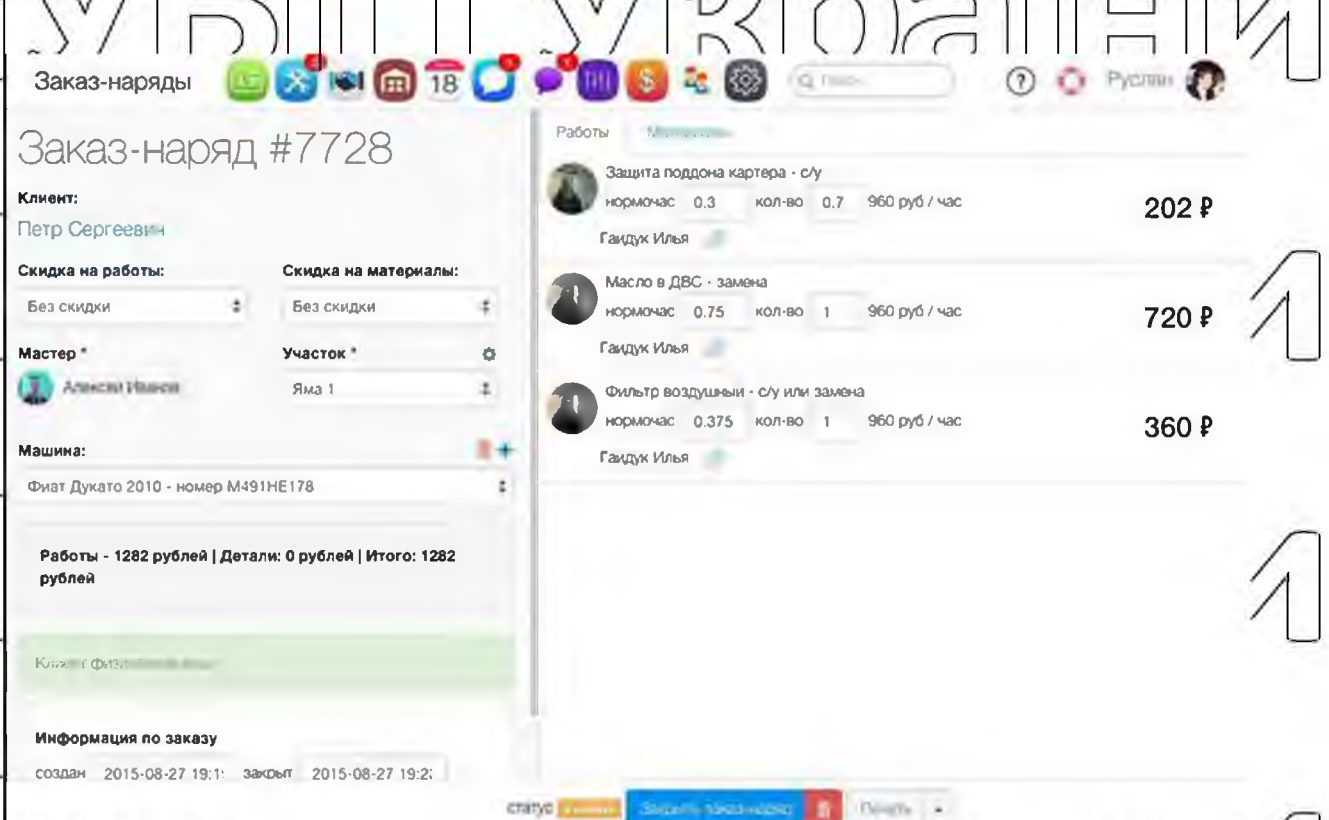


Рис. 1.2. Інтерфейс системи iDigestor

Резервування - автоматичний регулярний «бекап» всіх даних, і можливість відновлення системи до певної дати.

Переваги:

- збір і накопичує дані про клієнтів

- буде звіти про прибутки і збитки

Недоліки.

- висока ціна;

«АвтоДилер» з модулем «Сервіс»

Система «АвтоДилер» - це спеціалізоване програмне забезпечення для автобізнесу.

Система призначена для автоматизації обліку, планування та аналізу роботи будь-яких підприємств: великих і дрібних автомайстерень, автосалонів, магазинів автозапчастин, автомийок, шиномонтажних майстерень і станцій заміни масла, автостраховиків.

Модуль «Сервіс» призначений для автоматизації обліку робіт в автосервісах і на станціях технічного обслуговування автомобілів. Система дозволяє значно скоротити час на оформлення документів. При повторному зверненні клієнта, у користувача є вся історія взаємин з ним.

У програмному комплексі існує можливість оформлення необхідних документів як для клієнта, так і для внутрішніх операцій.

Зображення інтерфейсу можна побачити на рис. 1.3.

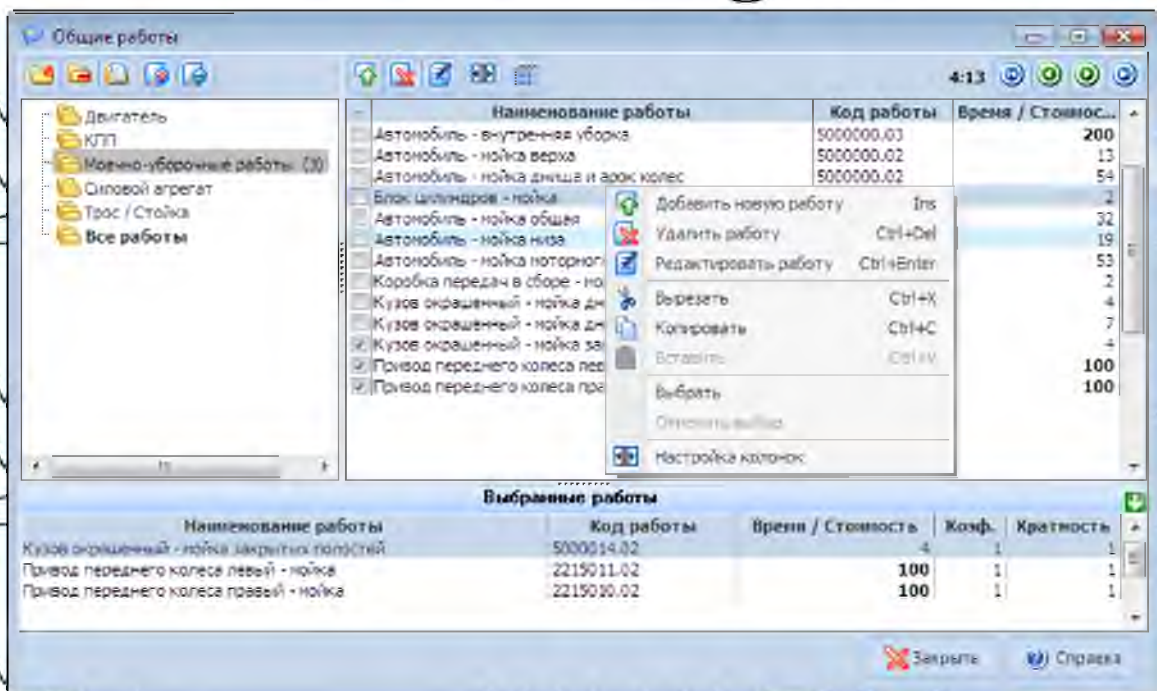


Рис. 1.3 Интерфейс системы «АвтоДилер»

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

НУБІП України

2. МОДЕЛІ ПРЕДМЕТНОЇ ОБЛАСТІ ТА

ПРОГРАМНОЇ СИСТЕМИ

Для моделювання предметної області було обрано мову UML. UML

(уніфікована мова моделювання) – це об'єктно-орієнтована графічна мова для

візуалізації, конструювання та документування систем, де велика роль відводиться опису бізнес-процесів в інформаційних системах.

UML є не першою мовою моделювання. У момент її появи нараховувалися десятки інших мов, які відрізнялися системними позначеннями, ступенем універсальності, способами застосування. У літературі можна зустріти опис ери "до UML" як "війни методів" моделювання, ні один з яких не дотягнув до рівня індустріального стандарту.

На початку 90-х років минулого століття три найбільших спеціаліста в цій області, автори найбільш популярних методів, вирішили об'єднати зусилля саме з метою об'єднання своїх (і не тільки своїх) розробок відповідно до соціальним замовленням.

Автори UML при підтримці та діяльності всієї міжнародної програмної громадськості змогли вести єдино (уніфікувати) велику частину того, що було відомо і до них. В результаті уніфікації вийшла теоретично вичерпна і практично корисна річ – UML.

Модель UML - це набір діаграм.

Діаграма - це графічне представлення більшості елементів. Зазвичай виявляється у вигляді графа з вершинами (сутностями) і ребрами (відношеннями). Прикладів діаграм безліч: блок-схема, схема монтажу обладнання, дерево файлів.

При проектуванні ПО за допомогою діаграми можна візуалізувати систему з різними точками зору.

Одна з діаграм, наприклад, може описувати взаємодію користувача із системою, інша - зміна наявної системи в процесі її роботи, третя - взаємодія між собою елементів системи.

Складну систему можна і потрібно представити у вигляді набору невеликих і майже незалежних моделей-діаграм, тому ні одна з них не є достатньою для опису системи та отримання повного представлення про неї, оскільки вона з них зосереджується на тому, яким є певний аспект функціонування та виражає систему різними рівнями абстракції. Іншими словами, кожна модель відповідає якійсь визначеній, на точці зору проєктованої системи. Ні одна роздільна діаграма не є моделлю.

Діаграми - це лише засіб візуалізації моделі, і ці два поняття слід розрізняти.

Лише набір діаграм становить модель системи і найбільш повне її описує, а не одна діаграма, вирвана з контексту.

2.1 Функціональне моделювання

2.1.1. Діаграма прецедентів.

Діаграма прецедентів дозволяє створити перелік операцій, які виконує система. Часто цей вид діаграм називають діаграмою функцій, тому що на основі набору таких діаграм створює список вимог до системи та визначає безліч виконуваних системою функцій.

Прецедент — опис типової взаємодії користувача із системою, яка виконує наступне:

- описує видиму функцію;
- представляє різні рівні деталізації функцій;
- забезпечує досягнення конкретної мети.

Кожна така діаграма або, як її зазвичай називають, кожен Use case - це опис сценарію поведінки, якому слідує дійові особи.

Прецедент зображується у вигляді овалу, зв'язаного з типовими користувачами, так званими акторами (actors). Актором є будь яка сутність, що взаємодіє із системою зовні, наприклад людина, обладнання, інша система. Прецедент визначає, що система надає актору.

Таким чином визначається набір закінчених дій (транзакцій), що виконуються актором при взаємодії із системою. Діаграма прецедентів

використання має високий рівень абстракції і дозволяє визначити функціональні вимоги до інформаційної системи, відображаючи об'єкти як самої системи, так і предметної області, а також завдання, що виконуються.

Даний тип діаграм використовується при описі бізнес процесів предметної області, що автоматизує, визначення вимог до майбутньої програмної системи.

Дана діаграма подана на рис. 2.1.

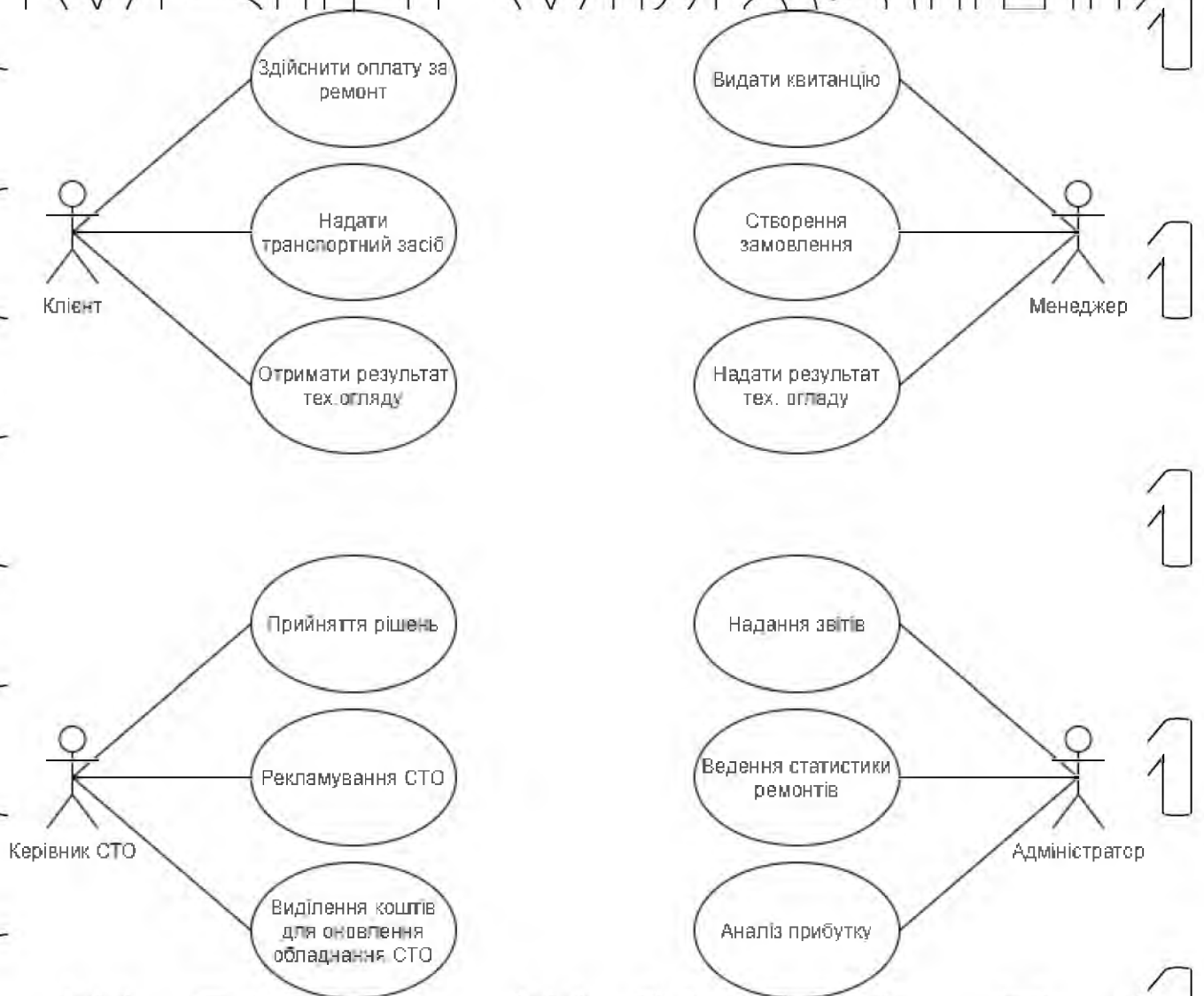


Рис. 2.1 Діаграма прецедентів

2.1.2. Діаграма розгортання або топологія системи.

Діаграма розгортання призначена для представлення загальної конфігурації або топології розподіленої програмної системи і містить зображення розміщення різних артефактів на окремих вузлах системи.

Діаграма розгортання призначена для візуалізації елементів і компонентів програми, які існують лише на етапі її виконання (runtime). При цьому представлені лише виконувані компоненти - екземпляри програми - файли або динамічні бібліотеки. Ті компоненти, які не використовуються на етапі виконання не показані на схемі розгортання.

Діаграма розгортання містить графічні зображення процесорів, пристроїв, процесів та зв'язки між ними. На відміну від логічних схем переглядів, діаграма розгортання є єдиною для системи в цілому, оскільки вона має повною мірою відображати особливості його реалізації. Ця діаграма, по суті, завершує процес ООАР для конкретної програмної системи та її розробку як зазвичай останній крок у специфікації моделі.

При розробці схем розгортання переслідуються наступні цілі:

- вказати фізичні вузли, які необхідно розмістити на них виконувані компоненти програмної системи;

- показати фізичні зв'язки між вузлами реалізації системи на етапі її виконання;

- виявити вузькі місця в системі та змінити її топологію для досягнення необхідної продуктивності.

Діаграма розгортання зображена на рис. 2.2

2.1.3. Структура оперативної бази даних або модель даних.

Розглянемо структуру оперативної бази даних, що зображена на рис. 2.3.

Оперативні бази даних містять інформацію, необхідну для діяльності організацій. В основному це вся інформація про конкретну людину чи співробітника, департаменту чи предмета. Тому їх також називають предметною областю баз даних. Таким чином, ці бази даних засновані на функціональних організаціях.

Сховища даних дозволяють розвантажувати оперативні бази даних і тим самим дають можливість користувачам більш ефективно і швидко витягувати необхідну інформацію.

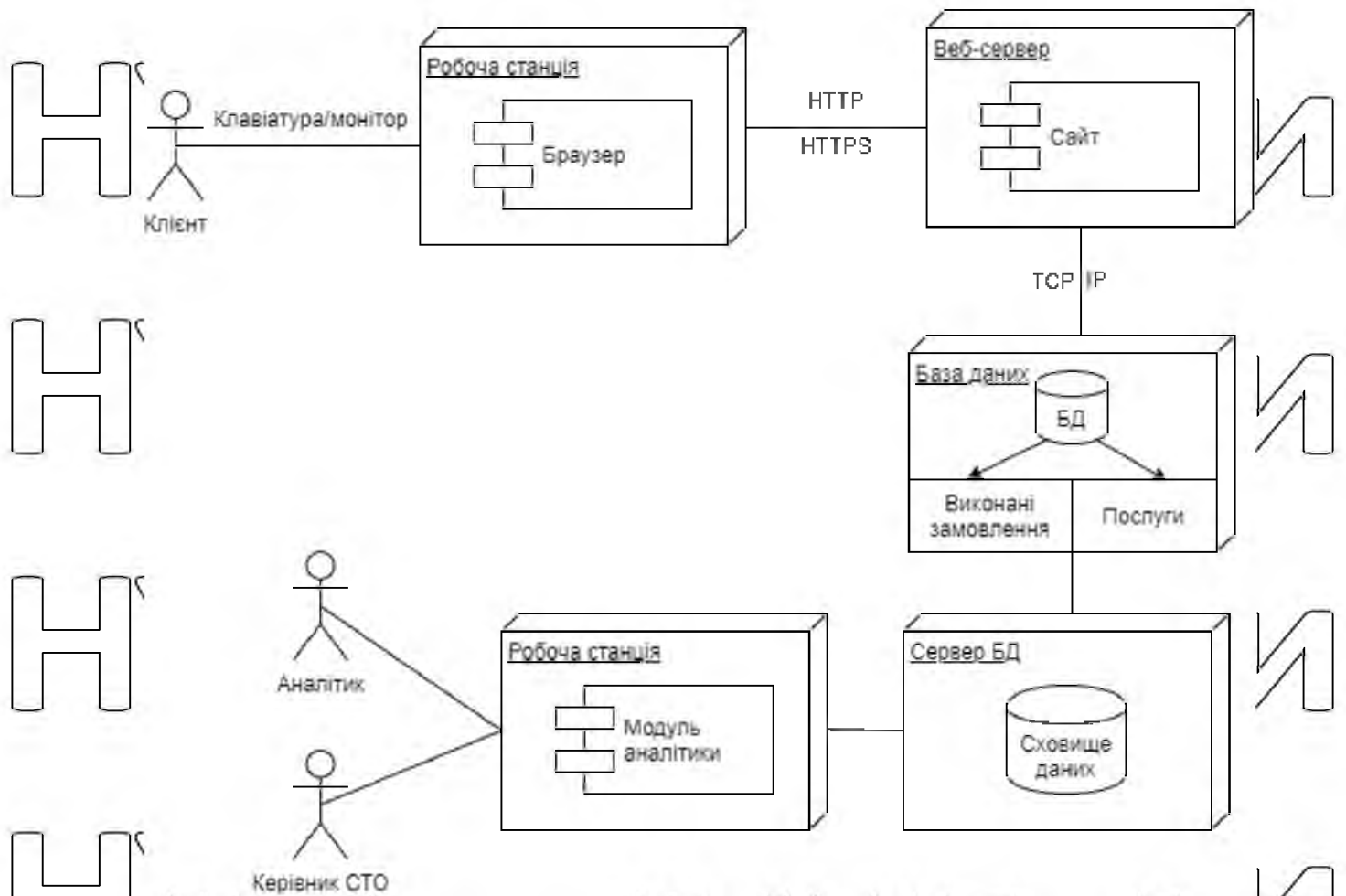


Рис. 2.2 Діаграма розгортання

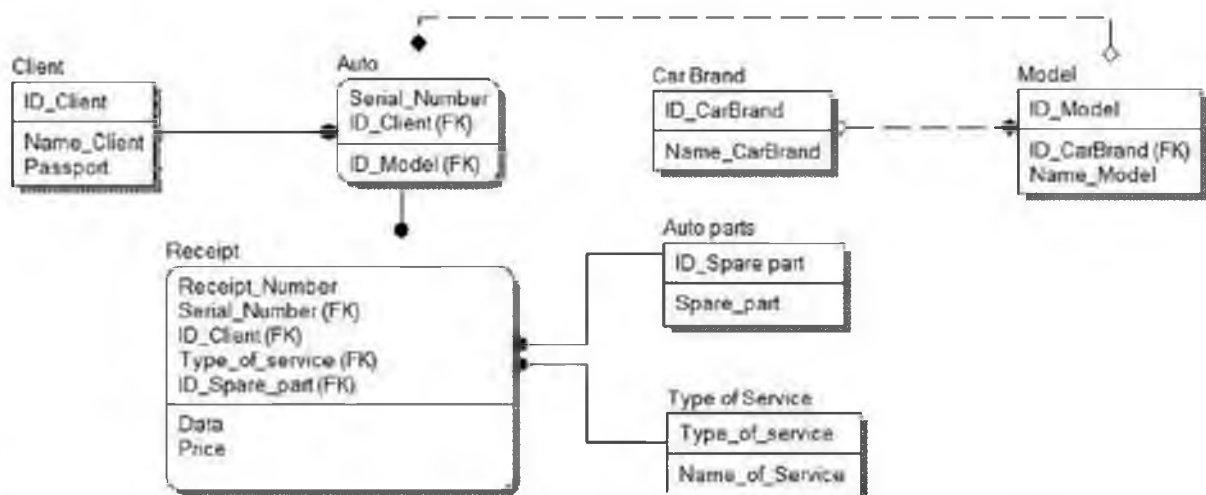


Рис. 2.3 Структура оперативної бази даних

Вони можуть бути включені в загальний корпоративний набір, який зберігається ваздалегідь визначеним розкладом, як правило, у період найменших завантажень мереж і серверів, копіюється інформація на день або на тиждень.

Оскільки дані змінюються не часто, то для зберігання даних не пред'являються жорсткі вимоги, які зазвичай пред'являються до звичайної бази даних, відсутність аномалій при виконанні операцій оновлень або видалення та збереженості інформації. За цією причиною може виникнути незвичне представлення, що проектувати сховище простіше, чим базу даних, призначені для оперативної обробки інформації. Насправді проектування сховища даних є складною задачею.

Оперативні бази даних зберігалися на магнітних дисках або барабанах, які забезпечували доступ до будь-якого елемента даних протягом секунди.

Ці пристрої та програмне забезпечення керування даними давали можливість програмам зчитувати кілька записів, змінювати їх та потім повертати нові значення оперативному користувачеві.

Спочатку системи забезпечували простий пошук даних або прямий пошук за номером запису, або асоціативний пошук за ключом.

При побудові даної моделі, були виділені такі ключові сутності:

- Клієнт(Client) – описує інформацію про клієнта.

Основні атрибути: ID Client, Name Client, Passport.

- Авто(Auto) – описує інформацію про автомобіль клієнта.

Основні атрибути: Serial number, ID Model.

- Марка Авто(CarBrand) – описує інформацію про марку авто.

Основні атрибути: ID Car Brand, Name Car Brand.

- Запчастини(AutoParts) – описує інформацію про запчастини.

Основні атрибути: Spare part.

- Тип послуги

Основні атрибути: Name of Service.

- Квитанція(Receipt) – описує інформацію у квитанції.

Основні атрибути: Receipt Number, Data, Price.

2.2 Об'єктне моделювання

Моделювання – це процес створення діаграм, діаграми відображають результати проектування. При проектуванні складної системи важливо розглянути її з різних точок зору – понятійної, логічної та фізичної структури, а також структурної та динамічної семантики.

2.2.1. Розглянемо діаграму класів.

Діаграма класів (Class diagram) – статична структурна діаграма, що описує структуру системи, вона демонструє класи системи, їх атрибути, методи і залежності між класами.

Діаграма класів описує типи об'єктів системи і різного роду статичні відносини, які існують між ними. На діаграмах класів відображаються також властивості класів, операції класів та обмеження, які накладаються на зв'язку між об'єктами. В UML термін функціональність (feature) застосовується в якості основного терміну, що описує і властивості, і операції класу.

Діаграма класів це певний граф, вершинами якого є елементи типу «класифікатор», які пов'язані різними типами структурних відносин. Слід зауважити, що діаграма класів може також містити інтерфейси, пакети, відносини і навіть окремі екземпляри, такі як об'єкти і зв'язки. Коли говорять про дану діаграму, мають на увазі статичну структурну модель спроектованої системи.

Тому діаграму класів прийнято вважати графічним представленням таких структурних взаємозв'язків логічної моделі системи, що не залежать або інваріантні від часу.

Під час побудови діаграми класів слід враховувати таке:

- ✓ Назва діаграми класів повинно мати сенс для опису аспекту системи.
- ✓ Кожен елемент і їх відносини повинні бути визначені заздалегідь.
- ✓ Відповідальність (атрибути і методи) кожного класу повинні бути чітко визначені
- ✓ Для кожного класу має бути зазначено мінімальну кількість властивостей, оскільки непотрібні властивості ускладнять діаграму.

✓ Використовуйте примітки щоразу, коли необхідно описати деякі аспекти діаграми. В кінці креслення це повинно бути зрозуміло розробнику / програмісту.

✓ Нарешті, перед тим, як зробити остаточний варіант, діаграма повинна бути намальована на звичайному папері і перероблена якомога більше разів, щоб зробити її правильною.

Діаграма класів проєктованої системи зображена на рис. 2.4.

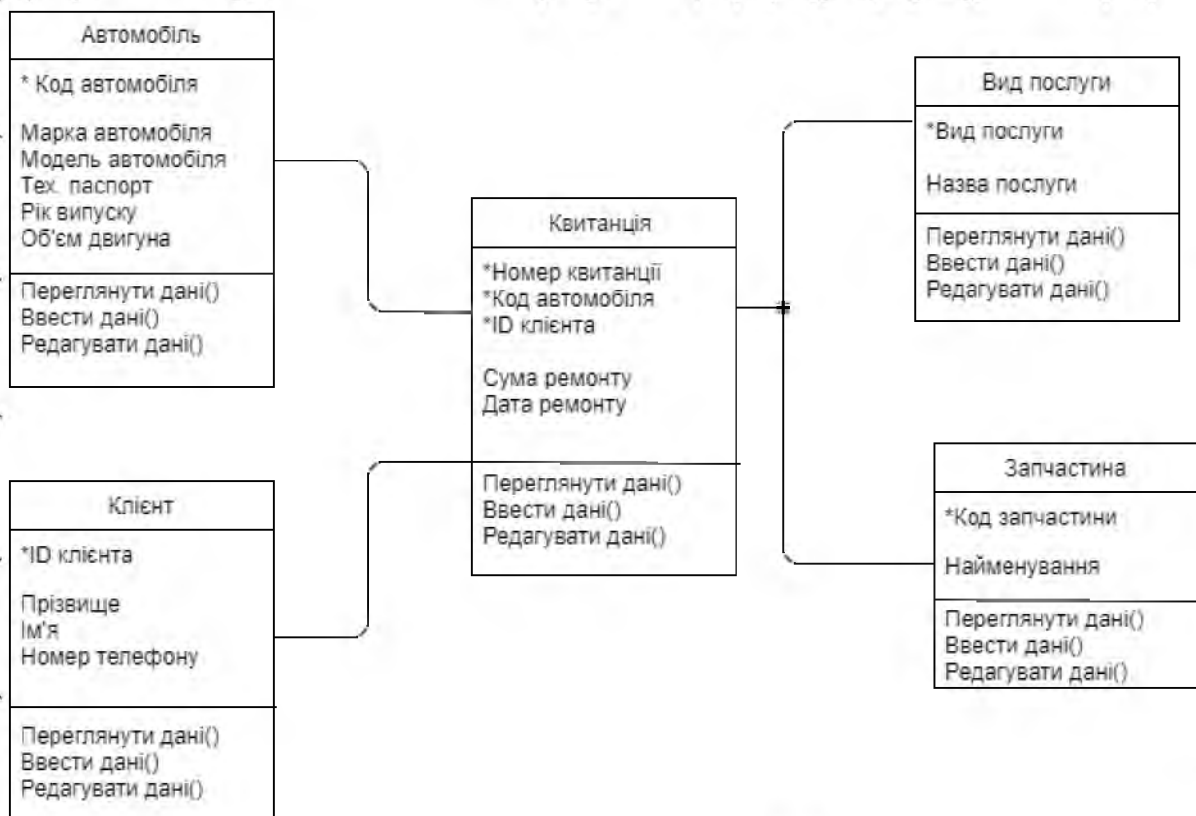


Рис. 2.4 Діаграма класів

2.2.2. Діаграма пакетів

Пакети надають механізм угрупування взаємозалежних елементів UML та обмеження видимості їхніх імен. Наприклад, всі елементи, пов'язані з 3D- моделюванням можна розмістити в пакеті з ім'ям 3DGraphics.

Діаграми пакетів забезпечують відмінну можливість візуального подання залежностей між частинами вашої системи та часто використовуються для діагностики чи визначення порядку компіляції.

У пакетах можуть групуватися практично будь-які елементи UML (у тому числі самі пакети). Кожен пакет має ім'я, що уточнює область видимості кожного елемента пакета.

Наприклад, якщо клас з ім'ям Timer належить пакет Utilities, повне ім'я класу має вигляд Utilities: :Timer. Елементи, що входять в один пакет можуть звертатися один до одного без уточнення імен.

Дану діаграму подано на рис. 2.5.

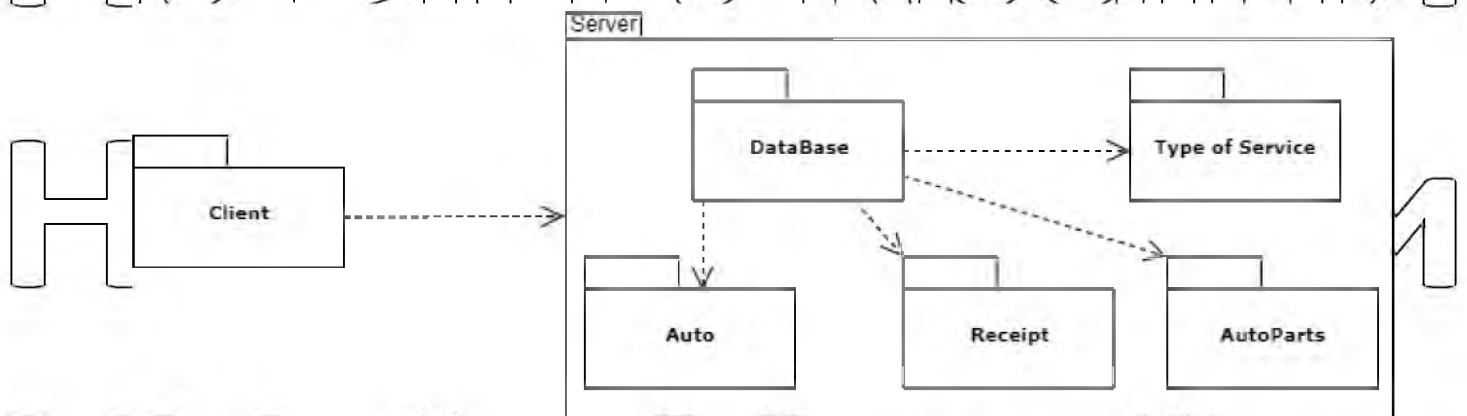


Рис. 2.5 Діаграма пакетів

2.3 Абстракції предметної області

Абстракція — це надання об'єкту характеристик, які відрізняють його від інших об'єктів, які чітко визначають його концептуальні межі.

Основна ідея полягає в тому, щоб відокремити використання складених об'єктів даних від деталей їх реалізації у вигляді простіших об'єктів, так само як функціональна абстракція відокремлює використання функції та деталі її реалізації в термінах більш примітивних функцій, таким чином дані обробляються функцією високого рівня за допомогою використання низькорівневих функцій.

Таблиця 2.1
Абстракції предметної області

Абстракція	Властивості	Обов'язки
Адміністратор	Логін Пароль	<ul style="list-style-type: none"> • Додавання нових клієнтів • Додавання даних у квитанцію
Клієнт	Логін Пароль ПІБ Номер телефону Електронна пошта	<ul style="list-style-type: none"> • Редагування персональних даних • Відображення даних про ремонт автомобіля
Квитанція	Номер квитанції Клієнт Автомобіль Дата Сума Тип сервісу Найменування запчастини	<ul style="list-style-type: none"> • Представлення інформації про ремонт • Надсилання даних до системи
Склад запчастин	Серійний номер запчастини Назва запчастини Ціна	<ul style="list-style-type: none"> • Зберігання даних про запчастини • Надсилання даних до системи
Автомобіль	Марка Модель Рік Об'єм двигуна	<ul style="list-style-type: none"> • Виведення даних до таблиць

3. ПРОЕКТУВАННЯ СИСТЕМИ

3.1 Логічна модель даних

Логічне моделювання бази даних необхідне для складання бізнес-вимог і подання вимог як моделі. В основному це пов'язано зі збиранням бізнес-потреб, а не з проектуванням бази даних. Інформація, яку необхідно зібрати, стосується організаційних підрозділів, бізнес-структур та бізнес-процесів.

Після збору інформації складаються звіти та діаграми, що включають такі:

- ERD–Діаграма відносин сутностей показує відносини між різними категоріями даних і показує різні категорії даних, необхідні для розробки бази даних.
- Діаграма бізнес-процесів-вона показує діяльність окремих осіб усередині компанії. Показує, як дані переміщуються всередині організації, на основі якого може бути розроблений додатковий інтерфейс.
- Документація щодо відгуків користувачів.

Логічні моделі баз даних переважно визначають, чи були зібрані всі вимоги бізнесу. Він перевіряється розробниками, керівництвом та, нарешті, кінцевими користувачами, щоб дізнатися, чи необхідно зібрати додаткову інформацію до початку фізичного моделювання.

Логічна модель даних є початковим прикладом майбутньої бази даних.

Логічна модель будується в інформаційних одиницях, але без посилання на конкретну базу даних. Більше того, логічна модель даних не обов'язково повинна бути виражена за допомогою реляційної моделі даних.

Основним засобом розробки логічної моделі даних зараз є різні версії ER-діаграм.

Для побудови логічної моделі використовуються такі її основні елементи, як: сутність, зв'язок між ними та їх властивості (атрибути).

Сутність за формою є лише деяким реальним описом об'єкта, точніше набір описів його значущих атрибутів. Конкретний набір значень атрибутів об'єкта буде називатися екземпляром сутності. Сутності є основними типами

інформації, що зберігається в базі даних (у реляційній базі даних кожній сутності присвоюється таблиця).

Структура бази даних зображено на рис. 3.1

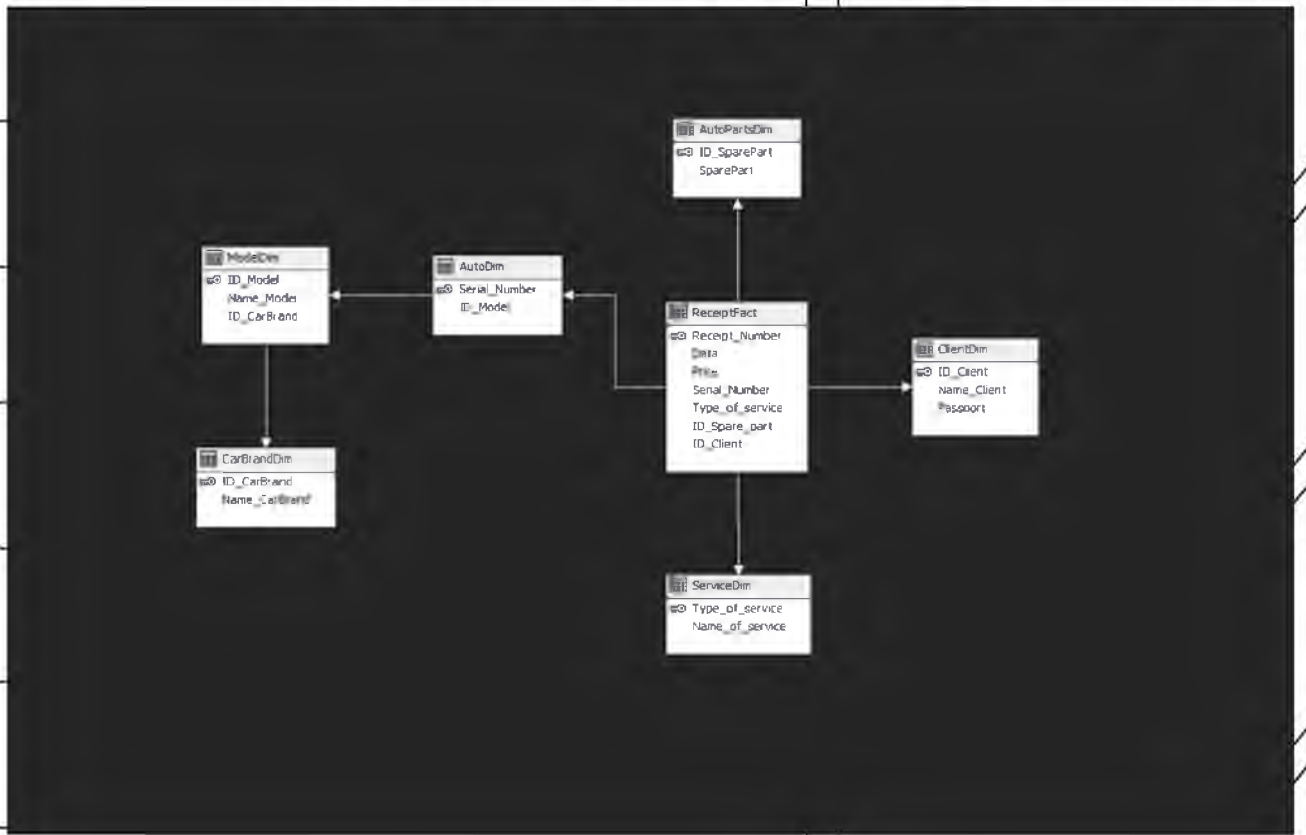


Рис. 3.1 Структура Базы Даних

3.2 Проектування загальної архітектури програмної системи

3.2.1. Визначення СУБД

Для розробки архітектури необхідно вибрати систему управління базами даних (СУБД). За способом доступу до бази даних (БД) розрізняють наступні СУБД:

- клієнт-серверні;
- файл-серверні;
- вбудовані

Клієнт-серверна система характеризується наявністю двох взаємодіючих самостійних процесів - клієнта і сервера, які можуть виконуватися на різних

комп'ютерах, обмінюючись даними по мережі. За такою схемою можуть бути побудовані системи обробки даних на основі СУБД, поштових та інших систем.

У системі файлів-серверів зберігаються файли сервера (наприклад, Novell NetWare або Windows NT Server), а їх обробка здійснюється на робочих станціях, на яких, як правило, функціонує один із, так званих, "настільних СУБД" - Access, FoxPro, Парадокс і тому подібні.

Додаток на робочій станції "відповідає за все" - для формування користувальницького інтерфейсу, логічної обробки даних і для безпосереднього маніпулювання даними. Файловий сервер надає послуги лише самого низького рівня - відкриття, закриття та модифікацію файлів, підкреслю - файли, а не базу даних. База даних існує тільки в "мозку" робочої станції.

Непосереднім маніпулюванням даними займається кілька незалежних і неузгоджених між собою процесів. Крім того, для здійснення будь-якої обробки всі дані необхідно передати в мережу із сервером на робочій станції.

У клієнт-серверній системі функціонують (як мінімум) два додатки - клієнт і сервер, що діють між собою, ці функції, які у файл-серверній архітектурі цілі виконують додаток на робочій станції. Зберіганням і непосреднім маніпулюванням даними задаються базою даних сервера, в якості якого може виступати Microsoft SQL Server, Oracle, Sybase, Firebird, Interbase, IBM DB2, Informix, PostgreSQL, MySQL, Cache, ЛИНТЕР.

Вибір системи управління баз даних (СУБД) являє собою складну задачу і є одним із важливих етапів при розробці додатків бази даних. Вибраний програмний продукт повинен удосконалювати як поточний, так і майбутнім потребам підприємств, при цьому слід враховувати фінансові витрати. Інформаційна система повинна відображати найбільш загальні технічні вимоги.

3.2.2. Архітектура системи

Для розробки системи була обрана Клієнт-серверна архітектура.

Клієнт-сервер (Client-server) - обчислювальна чи мережева архітектура, у якій завдання чи мережне навантаження розподілені між постачальниками послуг (сервісів), званими серверами, і замовниками послуг, званими клієнтами.

Нерідко клієнти та сервери взаємодіють через комп'ютерну мережу та можуть бути як різними фізичними пристроями, так і програмним забезпеченням.

Сама технологія дуже проста. Наприклад, користувач робить запит (у google шукає інформацію), а сервер видає відповідь (у вигляді списку сайтів за цією інформацією). Це є найпростіший приклад цієї технології. Графічно зображена на рисунку 3.2

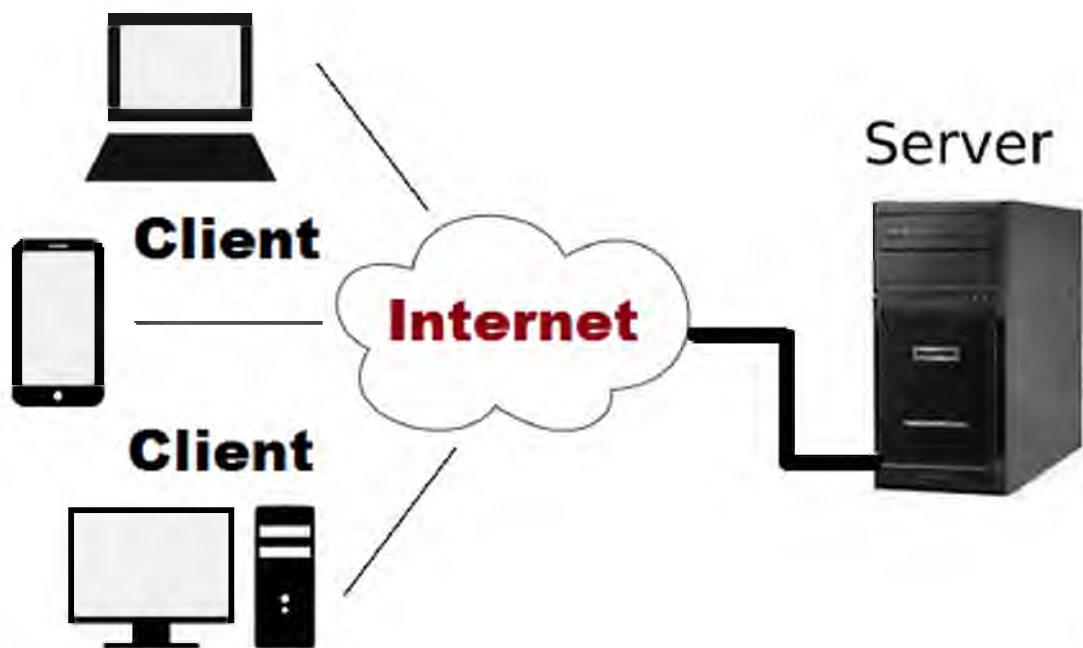


Рис. 3.2 Архітектура клієнт-сервер

Модель клієнт-сервер використовується при побудові системи обробки інформації на основі СУБД та поштової системи. Існує ще так звана файл-серверна архітектура, яка суттєво відрізняється від клієнт-серверної.

Дані у файл-серверній системі зберігаються на файловому сервері (Novell NetWare або WindowsNT Server), а обробляються вони на робочих станціях за допомогою функціонування "настільних СУБД", таких як Access, Paradox, FoxPro і т.п.

Неважко помітити, що більшість недоліків класичної або 2-х шарової архітектури клієнт-сервер походить від використання клієнтської станції як виконавець бізнес-логіки ІС. Тому очевидним кроком подальшої еволюції архітектури ІС стала ідея "тонкого клієнта", тобто розбиття алгоритмів обробки даних на частини пов'язані з виконанням бізнес-функцій та пов'язані з відображенням інформації у зручному для людини поданні.

При цьому на клієнтській машині залишають лише другу частину, яка пов'язана з первинною перевіркою та відображенням інформації, та розміщують всю реальну функціональність системи на серверну частину.

3.2.3. Переваги та недоліки клієнт-серверної архітектури.

Як і будь яка інша технологія, клієнт серверна архітектура має свої переваги та недоліки. Розглянемо їх

Переваги:

- Виконує більшу частину роботи потужною серверною частиною при мінімумі навантаження на клієнта.
- Більшість даних зберігаються на сервері. У цьому, зазвичай, краще захищений від різноманітних загроз, ніж звичайний клієнтський ПК.
- Можливість чіткішого розмежування повноважень доступу до різних рівнів інформаційної системи. Кожному клієнту – свій рівень доступу.
- Кросплатформненість. Простіше кажучи, будь-який клієнт може працювати з ресурсами сервера незалежно від операційної системи, що використовується.
- Зменшення навантаження на мережу через те, що клієнт переважно передає серверу команди, а той вже їх виконує.

Недоліки:

- Вихід з ладу сервера може призвести до непрацездатності всієї системи, яка його використовує.
- Висока вартість серверного обладнання та його обслуговування (зокрема, може знадобитися окремий спеціаліст для обслуговування)

- Високе навантаження на серверне обладнання та канал зв'язку до нього.

3.3 Організаційна структура програмної системи

Організаційна структура інтерфейсної частини програмної системи

наведена у табл. 3.1.

Таблиця 3.1

Структура інтерфейсної частини додатку

Контролер	Назва	Опис
Create_service_table	Додавання типу сервісу	Сторінка додавання нового типу сервісу, зображена на рис. 3.3
Create_category_table	Додавання категорії сервісу	Сторінка додавання нової категорії сервісу, зображена на рис. 3.4
Create_order_table	Створення запису на сервіс	Сторінка створення нового запису на сервіс, зображена на рис. 3.5
Create_client_table	Ресстрація нового клієнта	Сторінка ресстрації нового клієнта у базі, зображена на рис. 3.6
Create_product_table	Додавання нової запчастини	Сторінка додавання нової запчастини, зображена на рис. 3.7
State_table	Сторінка статистики	Перегляд сторінки статистики, зображено на рис. 3.8
Create_user_table	Ресстрація нового користувача	Сторінка ресстрації нового користувача у базі, зображена на рис. 3.9
Service_table	Сторінка перегляду статусу робіт	Сторінка перегляду статусу робіт, зображена на рис. 3.10

Нижче наведено приклади у вигляді скріншотів.

Додати сервіс

Категорія

Назва

Ціна

Додати

Головна / Категорії сервісів / Додати сервіс

Рис. 3.3 Додавання типу сервісу

Додати категорію сервісів

Назва

Додати

Головна / Категорії сервісів / Додати категорію сервісів

Рис. 3.4 Додавання категорії сервісу

Додати запис на сервіс

Категорія сервісу

Сервіс

Залчастини

select

ПІБ клієнта

E-mail клієнта

Номер телефону клієнта

Марка автомобіля

Модель автомобіля

Номер автомобіля

Додати

Головна / Записи на сервіс / Додати запис на сервіс

Рис. 3.5 Створення запису на сервіс

Формуляція для додавання нового клієнта. Поля для введення:

- ПІБ
- Номер телефону
- Пароль
- Е-mail
- Адреса/адреси
- Статус: Активний

Кнопка: **Зберегти**

Рис. 3.6 Реєстрація нового клієнта

Формуляція для додавання нової запчастини. Поля для введення:

- Назва
- Id Service:
- Артикул
- Ціна
- Статус: Активний

Кнопка: **Зберегти**

Рис. 3.7 Додавання нової запчастини

Дашборд з чотирма картами:

- 1** Нові записи (Детальніше)
- 133** Сервіси (Детальніше)
- 8** Користувачі (Детальніше)
- 6** Зареєстровані клієнти (Детальніше)

Заявки на сервіс

Статуси

Дохід по сервісам

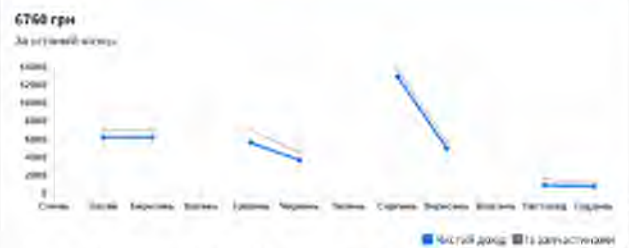
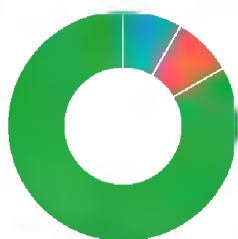


Рис. 3.8 Сторінка статистики

Додати користувача

Логін

ПІБ

Номер телефону

E-mail

Пароль

Зображення

Роль

Майстер

Статус

Активний

Рис. 3.9 Реєстрація нового користувача

Показані 1-6 із 6 записів.

#	Дата створення заявки	ПІБ клієнта	Номер телефону клієнта	Статус
1	11.11.2021 13:32	Костюк Олександр Володимирович	+38 (098) 883-31-11	Активний
2	11.11.2021 13:30	Белов Борис Владиславович	+38 (032) 939-93-33	Активний
3	11.11.2021 13:25	Мельничук Тарас Степанович	+38 (223) 331-11-11	Активний
4	11.11.2021 13:17	Терещенко Григорій Володимирович	+38 (032) 113-34-44	Активний
5	11.11.2021 11:19	Шевчук Вячеслав Денисович	+38 (213) 341-11-22	Активний
6	11.11.2021 12:42	Ткачук Богдан Матвійович	+38 (033) 111-22-31	Активний

Рис. 3.10 Сторінка перегляду статусу робіт

3.4 Архітектура програмного забезпечення

3.4.1. Вибір системи управління базою даних.

phpMyAdmin — веб додаток з відкритим кодом, написаний мовою PHP і є

веб- інтерфейсом для адміністрування СУБД MySQL. PHPMyAdmin дозволяє

через браузер здійснювати адміністрування сервера MySQL, запускати команди SQL та переглядати вміст таблиць та баз даних.

Додаток користується великою популярністю у веб розробників, оскільки дозволяє управляти СУБД MySQL без безпосереднього введення команд SQL, надаючи дружній інтерфейс.

Програма керування базами даних відкривається у вікні браузера. Її інтерфейс лаконічний та зрозумілий без додаткового інструктажу. Існує і русифікована версія. Інтерфейс phpMyAdmin складається з вікна, поділеного на дві частини.

Зліва знаходиться вузький стовпець, в якому представлені списки баз даних та деревоподібна структура кожної з них. Справа – робоче вікно, в якому відображаються вміст осередків бази, меню команд та службові кнопки.

База даних представлена на рисунку 3.11.

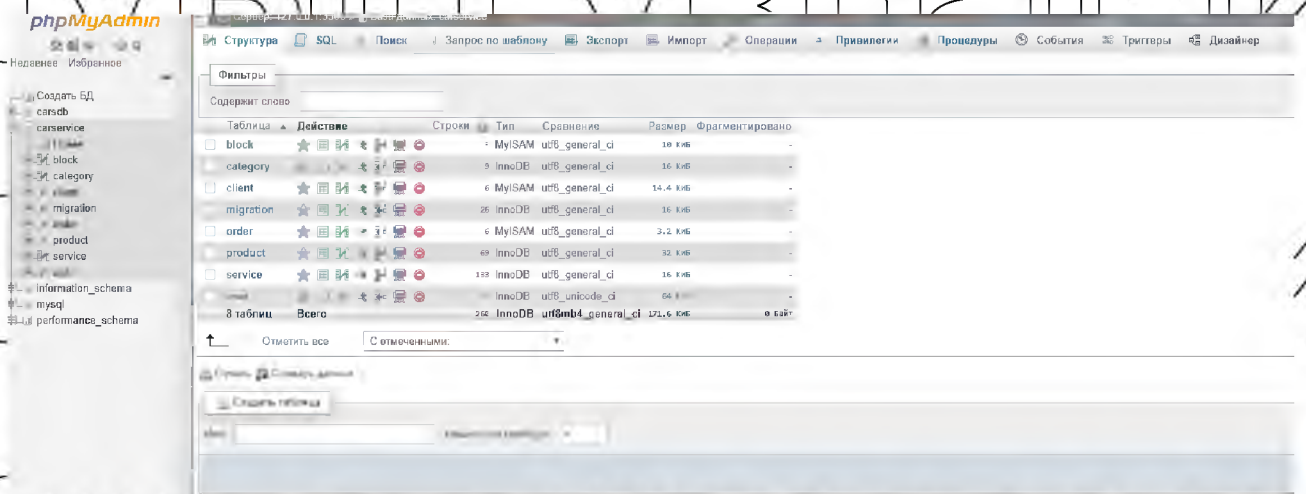


Рис. 3.11 Інтерфейс phpMyAdmin

Розглянемо базу даних більш детально. На рисунку 3.12 показані види сервісу.

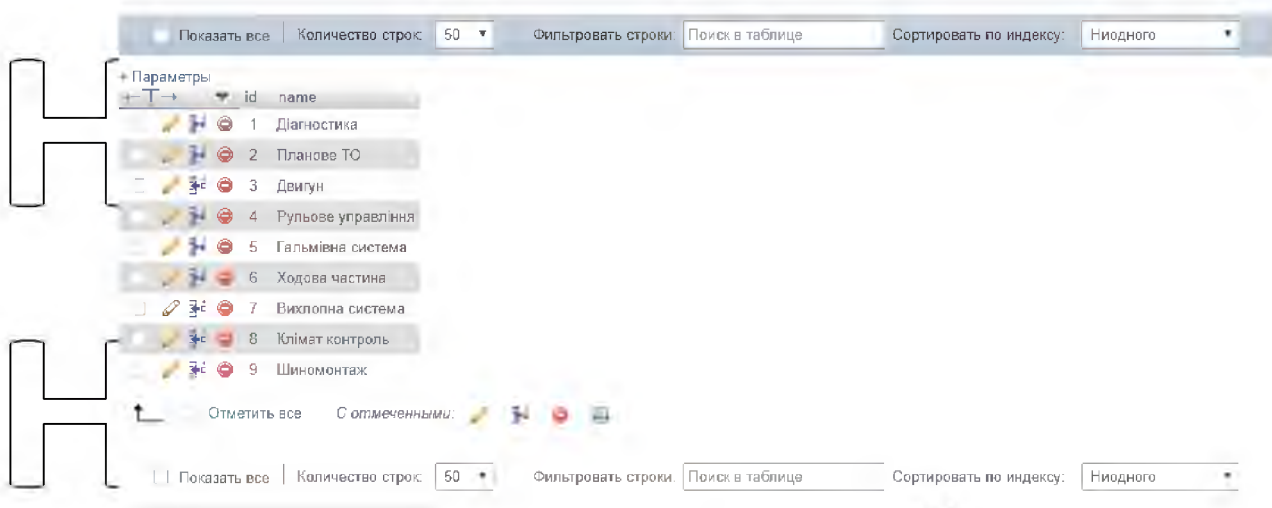


Рис. 3.12 Види сервісів

В залежності від вибору виду сервісу, будуть генеруватися списки типів сервісів. Типи сервісів представлено на рисунку 3.13

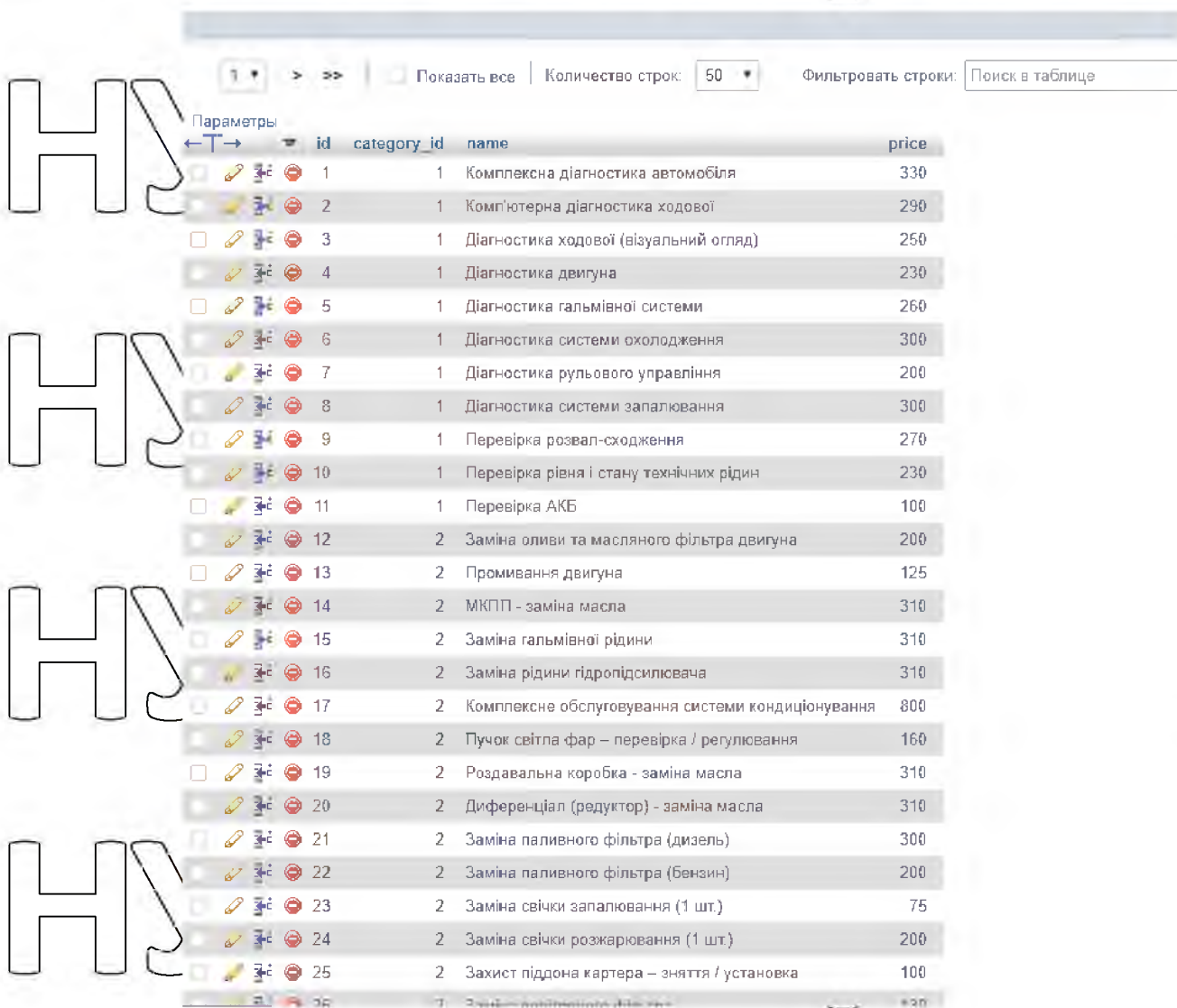


Рис. 3.13 Типи сервісів

НУБІП України

Також, після обраного типу сервісу, буде виведений список запчастин для ремонту.

На рисунку 3.14 представлений каталог запчастин.

+ Параметри						
←T→	id ID	status Status	name Name	price	article	id_service
	18	1	Амортизатор підвіски передній (543020008R)	2108	543020008R	1
	19	1	Амортизатор задній (562108593R)	1950	562108593R	1
	20	1	Опора (подушка) переднього амортизатора (540340002R)	840	540340002R	1
	21	1	Підшипник опори амортизатора (543256557R)	435	543256557R	11
	22	1	Пильник відбійник амортизатора (540500006R)	337	540500006R	11
	23	1	Ремінь ГРМ, комплект (7701477028)	1350	7701477028	4
	24	1	Зірка валу розподільного (7701478037)	726	7701478037	3
	25	1	Шестерня валу колінчастого (8200371496)	917	8200371496	3
	26	1	Фільтр повітряний (165467751R)	381	165467751R	2
	27	1	Фільтр масляний (8200768927)	165	8200768927	2
	28	1	Фільтр паливний (164009384R)	1036	164009384R	1
	29	1	Фільтр салону (272778970R)	460	272778970R	1
	30	1	Комплект щіток склоочисника безкаркасних 600/400 (...)	873	288905450R	1
	31	1	Колодки гальмівні, комплект (410600379R)	1134	410600379R	1
	32	1	Колодки гальмівні дискові, комплект (440605713R)	744	440605713R	1
	33	1	Насос охолоджуючої рідини (помпа) (210107477R)	8527	210107477R	1
	35	1	Фільтр повітряний (165462604R)	957	165462604R	1
	36	1	Фільтр паливний (164038899R)	998	164038899R	1
	37	1	Комплект щіток склоочисника безкаркасних 650/650 (...)	620	288906796R	1
	38	1	Корпус масляного фільтру (8201005241)	2906	8201005241	1
	42	1	Олива моторна Honda HG Ultimate 5W-30, 1 л (08798-...)	267	08798-9039	0
	44	1	Олива трансмісійна Honda CVT Hcf-2, 1 л (08200HCF2)	394	08200HCF2	0
	45	1	Олива трансмісійна Honda DPS-F, 1 л (082009007)	334	082009007	0
	46	1	Олива трансмісійна Honda MTF, 1 л (087989031)	264	087989031	0
	49	1	Диски Savannah R19 (5N0071499, 5N0601025G)	20500	5N0071499	0
	50	1	X-FLOW TYPE V 5W-30 1L	335	XFV1L	0
	51	1	X-FLOW TYPE V 5W-30 5L	1592	XFV5L	1
	54	1	X-FLOW TYPE Z 5W-30 1L	244	XFZ1L	0
	55	1	X-FLOW TYPE Z 5W-30 5L	1167	XFZ5L	0

Рис. 3.14 Каталог запчастин

НУБІП України

3.4.2. Обирання засобів конструювання системи

PHP (framework yii2)

PHP - одна з найпоширеніших мов web-розробки. Він застосовується для створення сайтів та web-додатків будь-якої складності. від лендингів та блогів до інтернет-магазинів та браузерних ігор.

Його переваги у широких можливостях та захищеності завдяки закритому вихідному коду. До мінусів можна віднести те, що без сервера з інтерпретатором код не працюватиме (виправлено в нових версіях мови).

У PHP підтримуються застосування функцій, математичні обчислення, робота зі змінними (як у прикладі вище) та масивами, прописування умов «якщо то», створення об'єктів та багато іншого. Найчастіше це використовується:

- для відправлення форм;
- роботи з базами даних;
- створення динамічних сторінок;
- використання сесій та cookies;
- завантаження та обробки файлів;
- створення зображень;
- парсингу.

Yii2 - об'єктно-орієнтований компонентний фреймворк для PHP, що реалізує парадигму MVC (Model-View-Controller). Yii є акронімом від "Yes It is", українською пишеться і читається як "юїї". Yii2 є другою версією фреймворку Yii.

Yii2 популярний додаток для початківців у справі проектування WEB-додатків. Щоб успішно написати свій маленький сайт, тут потрібно освоїти:

- ✓ PHP на рівні новачка. Достатньо знати трохи про класи, наслідування, трохи про методи. Наприклад, я спочатку познайомився з Yii1, а потім з PHP. Так, Yii дуже простий на початку освоєння.

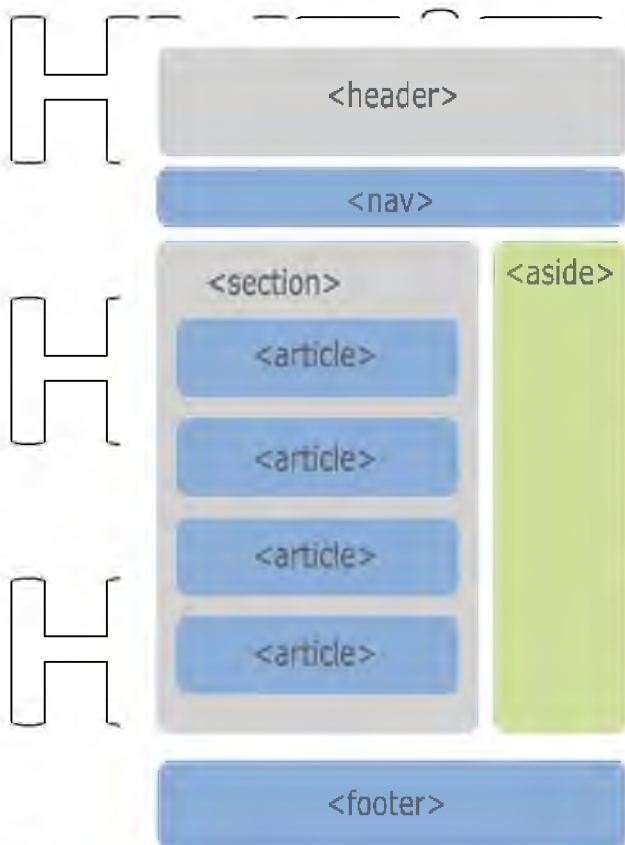
✓ jquery – базові основи. Ні, це не помилка – саме jquery, з урахуванням того, що yii2 побачив світ у жовтні 2014 року, тоді jquery був дуже популярним. jquery вбудований у yii2.

✓ HTML+CSS. Yii2 має на увазі написання веб-сайту, HTML + CSS - це те, що оформить зовнішній вигляд вашого сайту. У yii2 також вбудована підтримка фреймворку Bootstrap, тому з малюванням симпатичних елементів сторінки проблем не буде.

Важливо розуміти таке: після отримання запиту від клієнта, сервер видає відповідь назад і чекає на новий запит, знищуючи все, що зберігалось в оперативній пам'яті для формування відповіді.

yii2 створювався ніби для розробників із невеликим досвідом. Творці його подбали про те, щоб зрозуміти базові принципи роботи було максимально просто. У yii2 існує два заготовлені шаблони: Advanced і Basic.

HTML



HTML-це мова розмітки гіпертексту (від англ. HyperText Markup Language).

На рисунку 3.15 представлений приклад розмітки сторінки.

Ця мова використовується для створення веб-сторінок. Він інтерпретується (обробляється) браузером і відображається у вигляді документа у зручній для людини формі.

HTML – це невід’ємна складова та основа практично будь-якої веб-сторінки.

Мова HTML насамперед постає як засіб логічної розмітки сторінки.

Саме HTML дозволяє нам наділяти вміст сторінки певним змістом, а це реалізується за допомогою так званих тегів.

Теги – це спеціальні маркери, які певним чином інтерпретуються браузером.

Суть тегів у тому, що вміст сторінки, укладений у різні теги, по-різному обробляється браузером.

Скажімо, ми можемо укласти контент (вміст) сторінки в тег параграфа, і цей вміст вважатиметься браузером параграфом.

Ми можемо укласти вміст у тег списку, і тоді інформація всередині нього буде інтерпретуватися як список.

Ми можемо помістити контент у тег таблиці, і вміст у підсумковому документі відобразатиметься у вигляді таблиці і т.д.

Іншими словами, кожен із HTML-тегів, призначених для розмітки текстової інформації, надає цій інформації деякого сенсу.

CSS
CSS – це мова опису зовнішнього вигляду документа, написаного за допомогою мови розмітки. Назва походить від англ. Cascading Style Sheets – каскадні таблиці стилів.

Простіше кажучи, мова CSS призначена для того, щоб надавати необхідний зовнішній вигляд HTML-документам.

Надання зовнішнього вигляду документам HTML – це хоч і найпопулярніший, проте окремий випадок застосування мови CSS, т.к. з його допомогою можна надавати вигляду та документам інших типів: XHTML, SVG і XUL. Про них окремо говорити не будемо, т.к. це виходить за рамки питання, що розглядається. Приклад правила CSS представлений на рисунку 3.15.

селектор

властивість

значення

```
a { background-color: yellow; }
```

Рис. 3.15 Приклад правила CSS

Отже, метою створення CSS було відокремлення опису логічної структури веб-сторінки від її зовнішнього вигляду. Як ви вже знаєте, для опису структури використовується HTML, для опису того, як ця логічна структура буде виглядати, відповідає якраз CSS.

Роздільний опис логічної структури та подання документа дозволяє краще керувати зовнішнім виглядом документа і мінімізувати обсяг повторюваного коду, який неминуче виникає при використанні HTML для опису зовнішнього вигляду документа.

За допомогою CSS веб-розробник може задавати для сторінки та окремих її елементів різні гарнітури та розміри шрифту, кольори елементів, відступи елементів один від одного, розташування окремих блоків на сторінці тощо.

Зрозуміло, щоб використовувати CSS для надання зовнішнього вигляду HTML-документу, потрібно цей документ якось зв'язати зі стилями, тобто.

"повідомити" HTML-документу, що він буде оформлений за допомогою CSS.

Для цього існують різні способи підключення CSS до документа, які дають браузеру знати, що до сторінки в цілому, або до якихось її елементів має бути застосоване стильове оформлення.

Таблиці стилів можуть розташовуватися як безпосередньо всередині того документа, до яких вони будуть застосовуватися, так і знаходитися в окремому файлі, що має розширення .css.

Важливо розуміти, що файл CSS – це звичайний текстовий файл. У ньому пишуться спеціальні інструкції, що описують зовнішній вигляд елемента та його позиціонування на сторінці та коментарі (довільні пояснення щодо написаних інструкцій).

jQuery

jQuery - це популярна бібліотека JavaScript. Вона була створена Джоном Резігом у 2006 році з метою полегшити розробникам використання JavaScript на веб-сайтах. Це не окрема мова програмування та працює у поєднанні з JavaScript.

Написання коду може стати стомлюючим, якщо в нього включено багато рядків. jQuery стискає кілька рядків коду в одну функцію, тому вам не потрібно

переписувати цілі блоки коду для виконання одного завдання.

Особливістю того, чому jQuery стала настільки успішною та популярною, це, ймовірно, кросплатформлені можливості. Вона автоматично виправляє помилки і працює таким же чином у браузерях, що найчастіше використовуються, таких як Chrome, Firefox, Safari, MS Edge, IE, Android та iOS.

jQuery також робить Ajax набагато простіше. Ajax працює асинхронно з рештою коду. Це означає, що код, написаний на Ajax, може взаємодіяти з сервером та оновлювати його вміст без необхідності перезавантаження сторінки.

Однак це пов'язано із проблемами. Різні браузери виконують Ajax API по-різному. Таким чином, код має відповідати всім браузерам. Вручну, це важка та трудомістка робота. На щастя, jQuery виконує всю важку роботу та адаптує код для всіх веб-браузерів.

Потім є маніпулювання DOM (Document Object Model) (англ), де є кілька методів, як це зробити. Простіше кажучи, він дозволяє вставляти та/або видаляти елементи DOM на HTML-сторінці, а також спрощує перенесення рядків.

Створення анімації також спрощено jQuery. Як і в наведеному вище фрагменті коду про анімацію, він покритий кількома рядками коду, все, що вам потрібно зробити, це вставити змінні.

Обхід документів HTML, а також виконання ефектів та обробка подій також покращено за допомогою jQuery.

4. ВПРОВАДЖЕННЯ ТА ЕКСПЛУАТАЦІЯ СИСТЕМИ

НУБІП України

4.1 Інтерфейс програми

Розпочнемо тестування системи.

При вході в систему, клієнт може увійти в Особистий кабінет (для зареєстрованих клієнтів) або пройти процес реєстрації нового клієнта.

Для авторизації, клієнту потрібно ввести номер телефону та пароль. Потім натиснути кнопку увійти.

Для реєстрації, клієнт вводить такі дані:

- Прізвище, ім'я та по-батькові
- Номер телефону
- Пароль
- E-mail

Приклади приведено на рисунках 4.1 та 4.2.

НУБІП України

НУБІП України

НУБІП України

НУБІП України

Н

АВТОРИЗАЦІЯ

Будь ласка, авторизуйтесь

Номер телефону

Необхідно заповнити "Номер телефону".

Пароль

Необхідно заповнити "Пароль".

Запам'ятати мене [Зареєструватися](#)

УВІЙТИ

України

Н

РЕЄСТРАЦІЯ

Будь ласка, введіть необхідні дані

Прізвище, ім'я та по батькові

Необхідно заповнити "Прізвище, ім'я та по батькові".

Номер телефону

Пароль

Необхідно заповнити "Пароль".

E-mail

Необхідно заповнити "E-mail".

ЗАРЕЄСТРУВАТИСЯ

України

України

України

України

України

НУБІП

України

Рис. 4.1 Авторизація.

Рис. 4.2 Реєстрація

Після авторизації, клієнт може виконати запис на сервіс, заповнивши такі

поля:

НУБІП України

1. Інформація про авто

- Марка автомобіля
- Модель автомобіля

НУБІП України

- Рік випуску
- Пробіг автомобіля
- Об'єм двигуна

- Реєстраційний номер

НУБІП України

2. Інформація про сервіс

- Дата запису
- Вибір послуг

НУБІП України

3. Інформація про клієнта (заповнюється автоматично, якщо клієнт зареєстрований)

- Прізвище, ім'я та по-батькові
- Номер телефону
- Пароль

НУБІП України

- E-mail

Приклади форм наведені на рис. 4.3 та рис.4.4.

НУБІП України

НУБІП України

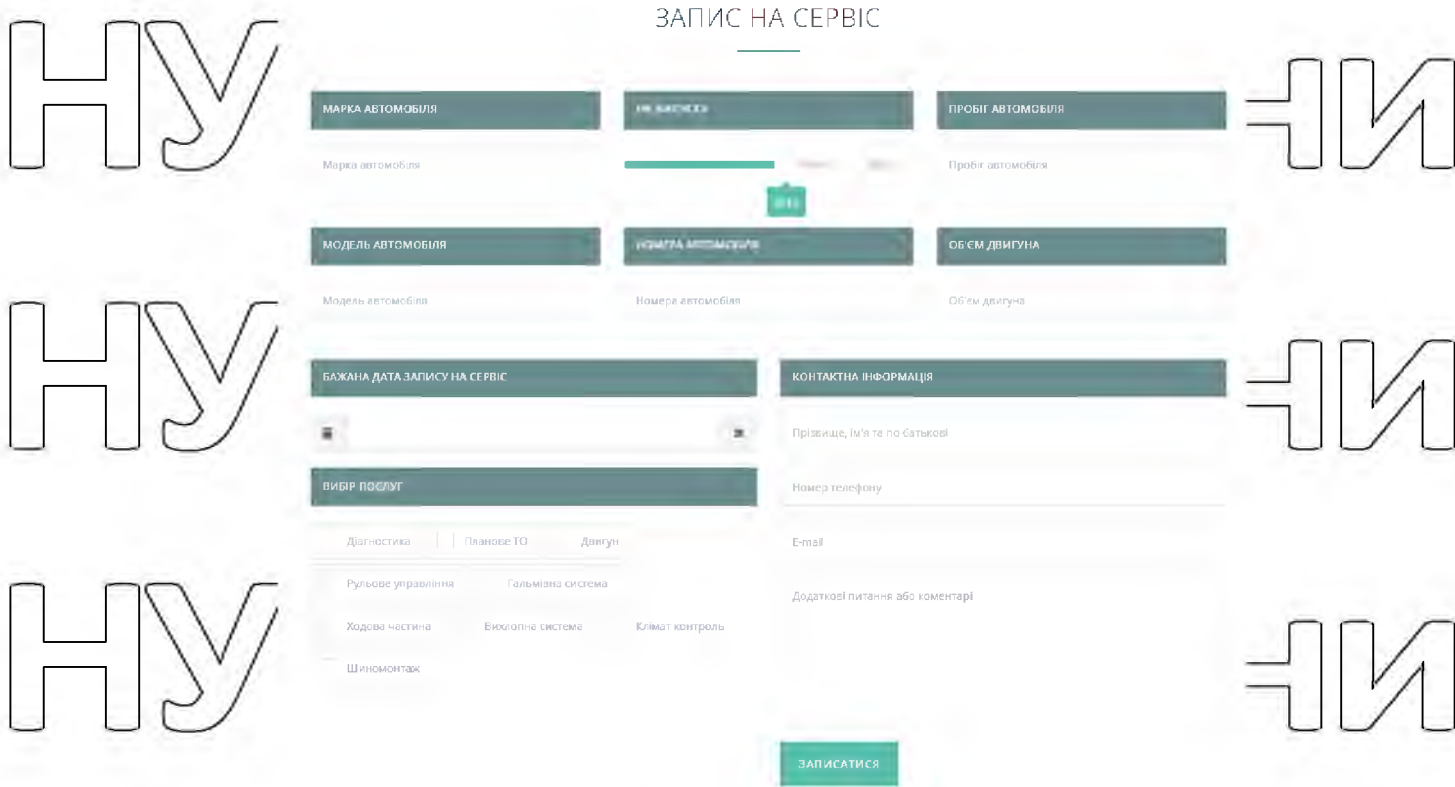


Рис. 4.3 Сторінка запису на сервіс

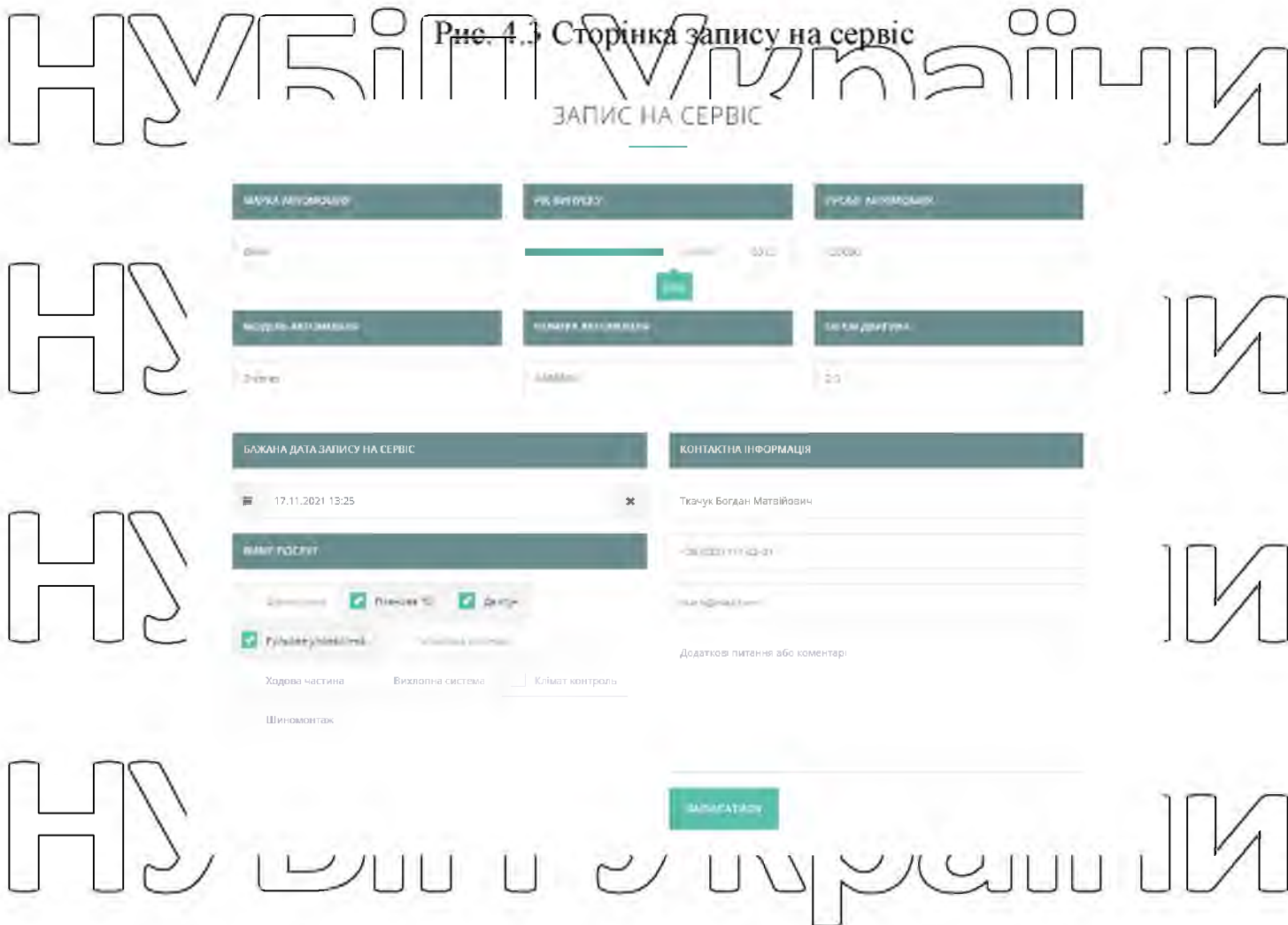


Рис. 4.4 Заповнення даних

Після запису на сервіс, клієнт у особистому кабінеті може перевіряти статус роботи. Також редагувати свої дані.

На рисунку 4.5 показаний інтерфейс особистого кабінету.

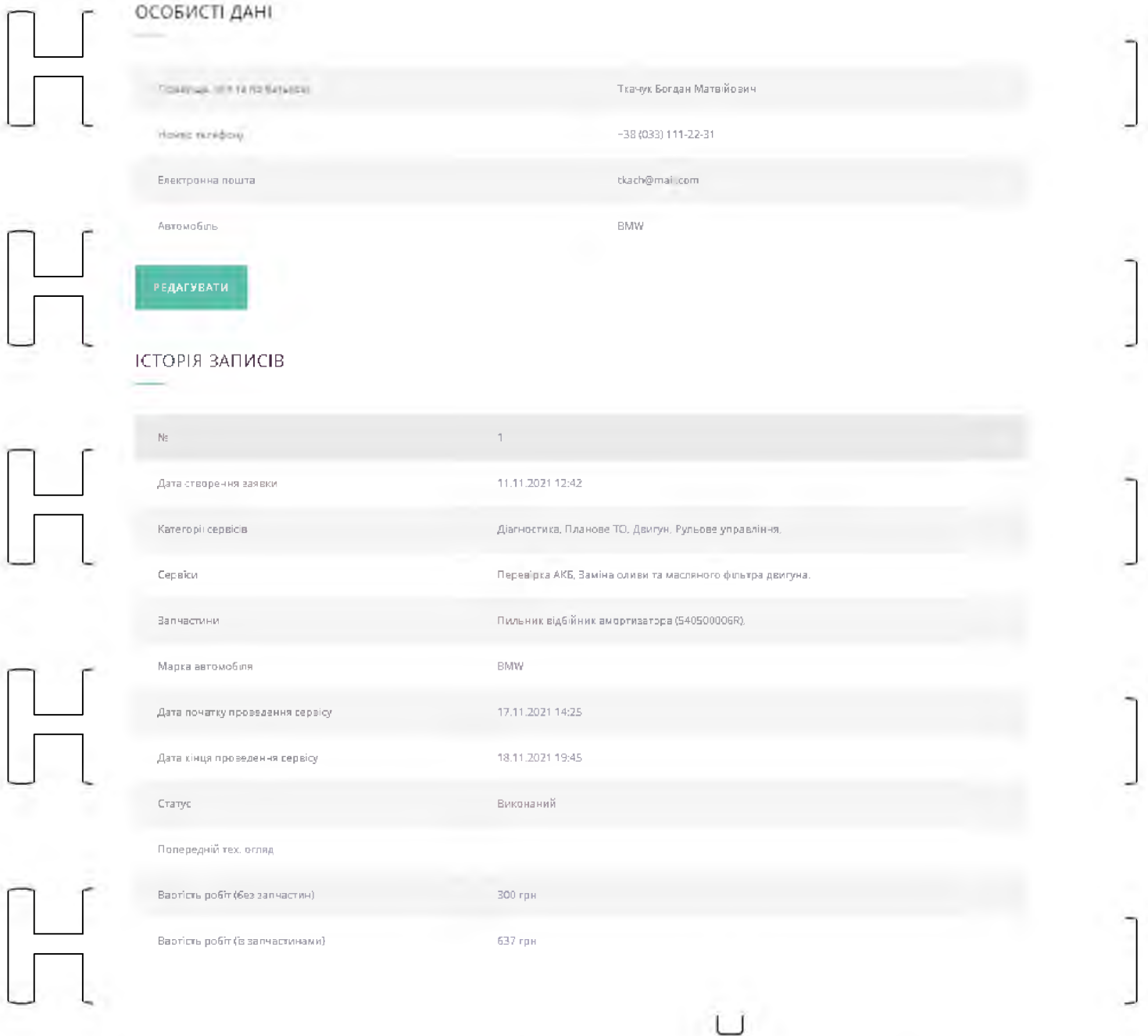


Рис. 4.5 Особистий кабінет клієнта

Після реєстрації клієнта на запис, у інтерфейсі менеджера з'являється новий запис.

Кожне замовлення, менеджер повинен обробляти. Як це виглядає, буде показано далі.

Інтерфейс запису на сервіс представлено на рисунку 4.6



Рис. 4.6 Запис на сервіс

Сторінка «Редагування замовлення» (Рис. 4.7).

На цій сторінці менеджер може:

- Змінювати статус замовлення
- Додавати види робіт
- Визначати дати проведення сервісу
- Додавати попередній технічний огляд
- Додавати відповідального майстра по ремонту

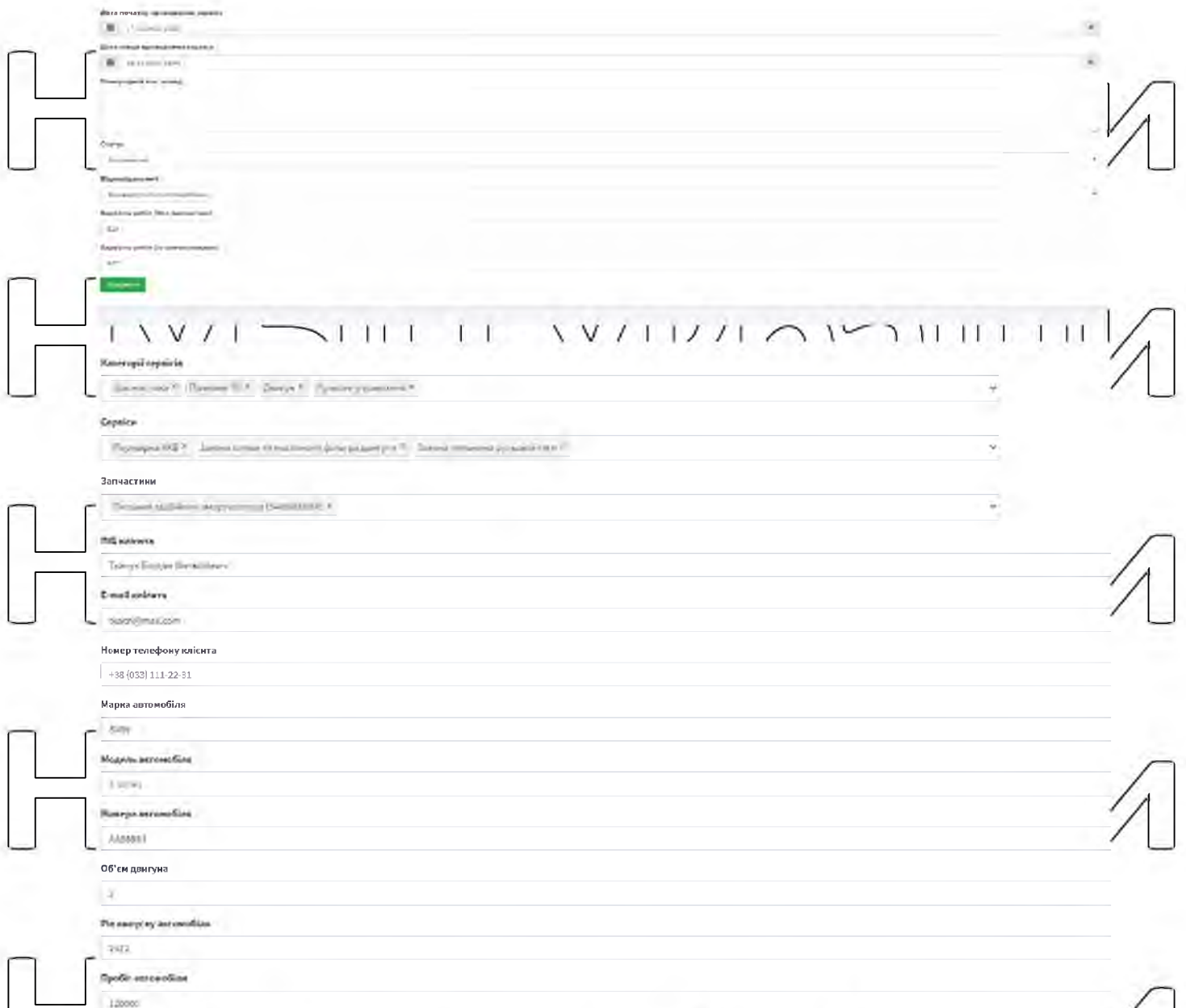


Рис. 4.7 Сторінка редагування замовлення

На цій сторінці менеджер може:

- Побачити нові записи
- Редагувати та додавати нові сервіси
- Редагувати та додавати нових користувачів
- Переглядати зареєстрованих клієнтів
- Аналізувати статистику автомайстерні

Сторінка представлена на рис. 4.8.

НУБІП України

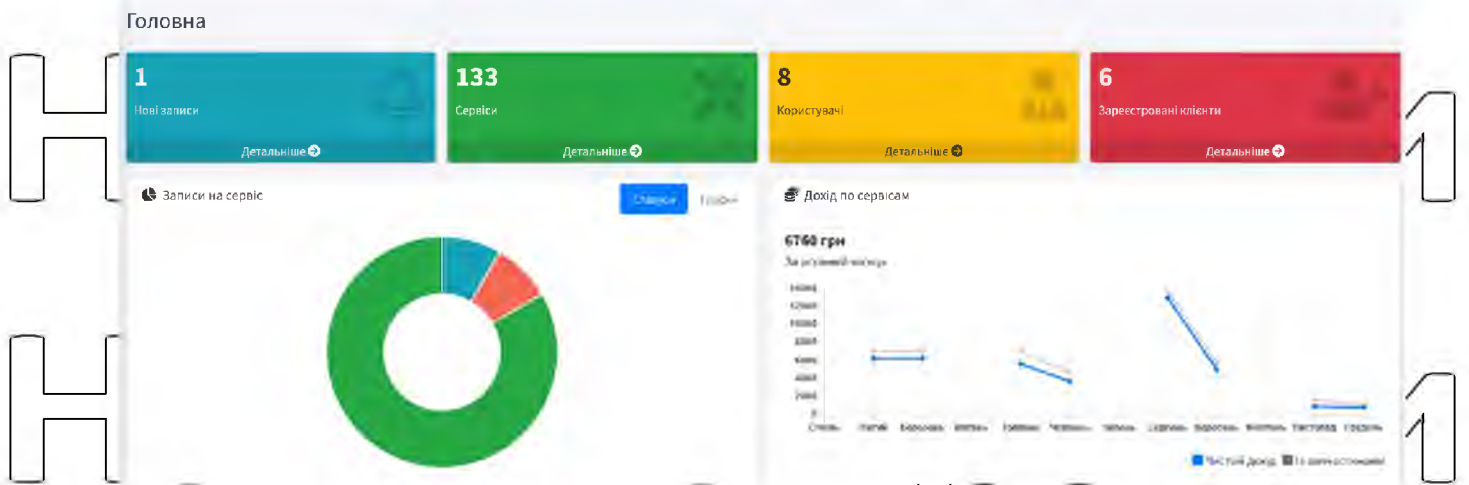


Рис. 4.8 Сторінка статистики

На цій сторінці (рисунок 4.9) представлено каталог товарів, доступних для встановлення на автомобіль клієнта.

Також каталог можливо редагувати, видаляти позиції та наповнювати новими запчастинами.

Товар

Показані 1-9 із 69 записів.

#	Назва	Статус
1	Амортизатор підвіски передній (543026008R)	Активний
2	Амортизатор задній (562108593R)	Активний
3	Опора (подушка) переднього амортизатора (540340002R)	Активний
4	Підшипник опори амортизатора (543256557R)	Активний
5	Пильник відбійник амортизатора (540503006R)	Активний
6	Ремінь ГРМ, комплект (7701477028)	Активний
7	Зірка валу розподільчого (7701478037)	Активний
8	Шестерня валу коліщастого (8200371495)	Активний
9	Фільтр повітряний (165467751R)	Активний

Рис. 4.9 Каталог товарів

На цій сторінці адміністратор може додавати, редагувати та видаляти дані щодо категорій сервісів. На даний час доступні такі категорії сервісів:

- Діагностика
- Планове ТО
- Інномонтаж

Ремонт пропонується за такими категоріям як:

- Двигун
- Рульове управління
- Гальмівна система
- Ходова частина

- Вихлопна система
- Клімат контроль

Сторінку обліку ремонтних ціл представлено на рисунку 4.10

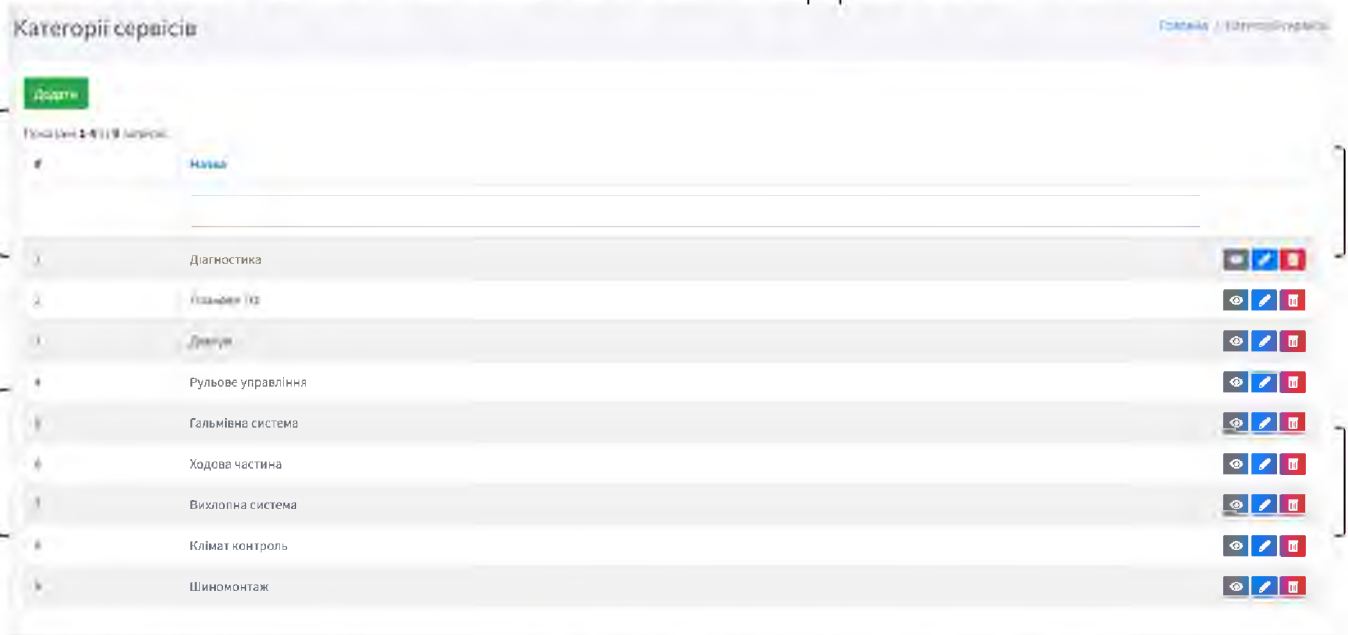


Рис. 4.10 Сторінка категорій сервісів

Менеджер перейшовши на сторінку типів сервісів може додавати, редагувати та видаляти дані позиції у представленому списку.

На рисунку 4.11 представлена «Сторінка типів сервісів».

Додати

Показані 1-20 із 133 записів.

#	Категорія	Назва	Ціна	
1	Діагностика	Комплексна діагностика автомобіля	330	
2	Діагностика	Комп'ютерна діагностика ходової	290	
3	Діагностика	Діагностика ходової (візуальний огляд)	250	
4	Діагностика	Діагностика двигуна	230	
5	Діагностика	Діагностика гальмівної системи	260	
6	Діагностика	Діагностика системи охолодження	300	
7	Діагностика	Діагностика рульового управління	200	
8	Діагностика	Діагностика системи запалювання	300	
9	Діагностика	Перевірка розвал-сходження	270	
10	Діагностика	Перевірка рівня і стану технічних рідин	230	
11	Діагностика	Перевірка АКБ	100	
12	Планове ТО	Заміна оливи та масляного фільтра двигуна	200	
13	Планове ТО	Промивання двигуна	125	
14	Планове ТО	МКПП - заміна масла	310	
15	Планове ТО	Заміна гальмівної рідини	310	
16	Планове ТО	Заміна рідини гідропідсилювача	310	
17	Планове ТО	Комплексне обслуговування системи кондиціонування	800	
18	Планове ТО	Пучок світла фар - перевірка / регулювання	160	
19	Планове ТО	Роздавальна коробка - заміна масла	310	
20	Планове ТО	Диференціал (редуктор) - заміна масла	310	

Попередня 1 2 3 4 5 6 7 Наступна

Рис. 4.11 Сторінка типів сервісів

На сторінці записів на сервіс (рисунки 4.12), менеджер може створювати записи на сервіс власноруч. Також в обов'язки менеджера входить зміна статусу замовлення, що відобразиться у кабінеті клієнта.

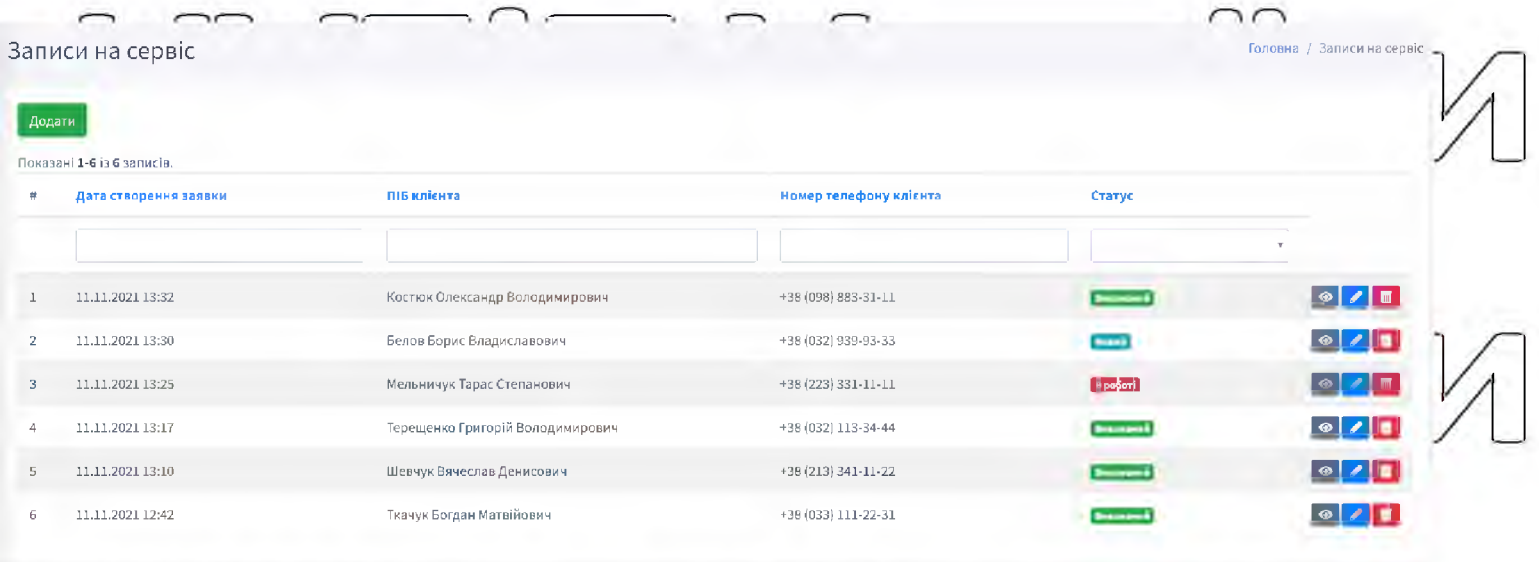


Рис. 4.12 Сторінка записів на сервіс

Сторінка користувачів представлена на рисунку 4.13.

На цій сторінці адміністратор може додавати, редагувати та видаляти користувачів, задавати їм роль та змінювати їх статус (активний або неактивний).

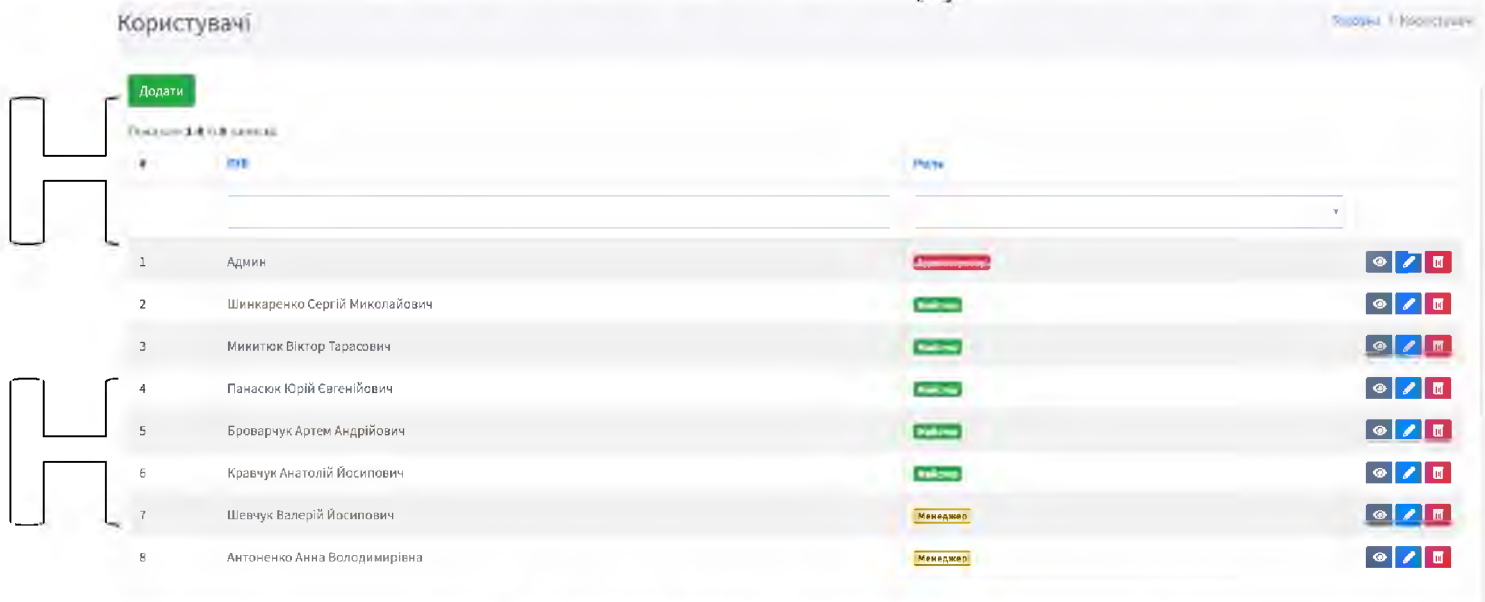


Рис. 4.13 Сторінка користувачів

На сторінці «Клієнти», менеджер може додати нового клієнта (за бажанням), переглядати зареєстрованих клієнтів та редагувати клієнтську базу.

Інтерфейс сторінки клієнтів представлений на рисунку 4.14

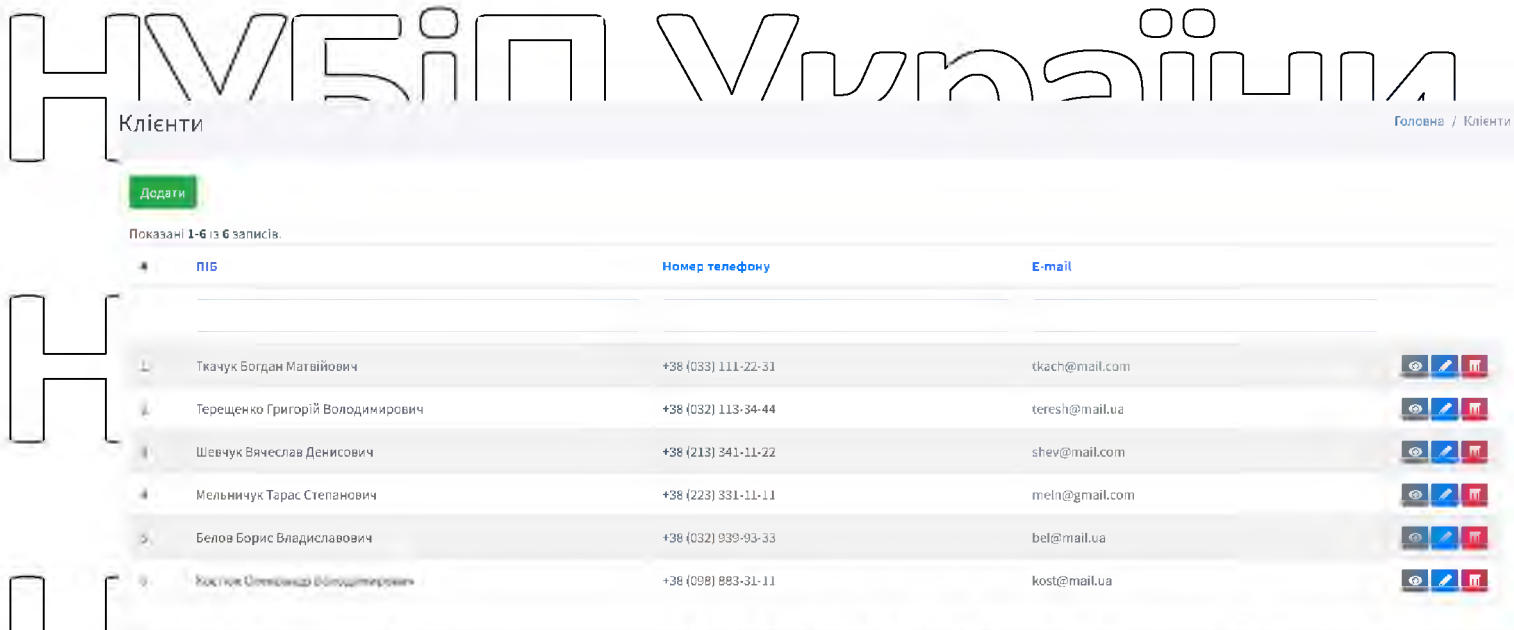


Рис. 4.14 Сторінка клієнтів

4.2 Аналіз даних та результати дослідження

Якщо переглянути статистику, яка зображена на рис. 4.8, то можна зробити висновки, що клієнтам було достатньо зручно здійснювати запис на СТО саме за допомогою створеної системи. Також керівництву СТО зручніше аналізувати дохід та потік клієнтів (зображено на рис. 4.14).

4.3 Перевірка якості програмного продукту

Необхідно перевірити відповідність продукту усім критеріям програмного продукту, отже:

- **Функціональність.** Система відповідає всім вимогам та виконує всі поставлені на неї задачі.
- **Стабільність.** Протестована апаратна частина, не було виявлено ніяких проблем та збоїв системи.
- **Практичність.** Система є зручною та інтуїтивною для кожного користувача. Має простий і лаконічний дизайн.

ВИСНОВКИ

Метою дипломного проекту була розробка програмного забезпечення системи обліку технічного обслуговування автотранспорту в агрофірмі.

У системі використовуються різні процедури та функції, призначені для обробки таблиць, пошів та записів бази даних та роботи з формами та вікнами. Ця система реалізує велику кількість різних методів роботи з інформацією, використовує безліч компонентів, що дозволяють швидко та зручно організувати зручну роботу.

Результати тестування показали, що система працює вірно у всіх сучасних браузерах.

Цілі та завдання дипломного проекту були досягнуті, розроблена та впроваджена система обліку технічного обслуговування, розглянуто багато питань, що сприяли розвитку навичок програмування у JetBrains PHP Storm, можливості самостійної роботи з літературою та програмним кодом.

В результаті виконаної роботи досягнуто наступних результатів.

- 1) проведено техніко-економічне обґрунтування даних для подальших розрахунків СТО;
- 2) проведено розрахунок параметрів СТО з розрахунком площі станції та техніко-економічною оцінкою проекту;
- 3) запропоновано конструкцію та проведено розрахунки знімачів;
- 4) здійснено організаційно-економічний розрахунок проекту;
- 5) запропоновано заходи з охорони праці, протипожежної та екологічної безпеки під час проведення робіт на СТО.

До основних відмінних рис системи «Agro-service» можна віднести таке:

- Невеликий розмір файлів з кодами Web-сторінок (їх листинг наведено в додатку), що забезпечує їх швидке завантаження з Мережі на клієнтській машині;

- Стислі формати графічних файлів, що також позитивно впливає на зменшення розміру Web-сторінок і часу їх завантаження по каналах мережі;

- Відсутність проблем сумісності з різними браузерами, наприклад широко поширеними, як Internet Explorer, Mozilla Firefox і Opera;

В результаті виконаної роботи було вдосконалено робоче місце менеджера за допомогою введення додаткового способу оформлення замовлень через веб-сайт.

Таким чином, матеріали дипломної роботи показують, що поставлені у ній завдання повністю вирішені.

За даною темою були проведені виступи на таких конференціях:

- 1) Міжнародна науково-практична конференція молодих вчених «Інформаційні Технології: економіка, техніка, освіта» 11 Листопада 2021

НУБІП України, Київ

- 2) Постерна конференція магістрів 2-го року навчання 15 Листопада 2021

НУБІП України, Київ

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Марков О.Д. Автосервіс: Ринок, автомобіль, клієнт. – М.: Транспорт, 1999. – 270 с.
2. Управление автосервисом: Учебное пособие для вузов /Под общ.ред. д.т.н., проф. Л.Б. Миротина. – М.: Издательство «Екзамен». 2004. – 320с. Автосервіс: станції технічного обслуговування автомобілів: Учебник /Грибут І.Э.,
3. Артюшенко В.М., Мазаева Н.П. и др / под ред. В.С.Шуплякова, Ю.П. Свириденко. – М.:Альфа-М: ИНФРА-М, 2008. – 480 с.: ил. – (Серія «Сервіс і туризм»).
4. Варнаков В.В. і др. Организация и технология технического сервиса машин/ В.В. Варнаков, В.В. Стрельцов, В.Н. Попов, В.Ф. Карпенков.- МП.: КолосС 2007. – 277с, ил. –(Учебники и учеб. пособия для студентов высш. учеб. заведений).
5. Технологічне проектування підприємств автосервісу. Навчальний посібник / За ед. І. П. Курнікова – К.: Видавництво «Іван Федоров», 2003. – 262с.
6. Автоматизована система проведення маркетингових досліджень у Білгородському філії МЕСІ [Електронний ресурс] // – Режим доступу: <http://ukrefs.com.ua/page,6,173673-Avtomatizirovannaya-sistema-provedeniya-marketingovyh-issledovaniiy-v-Belgorodskom-filiale-MESI.html>
7. Моделирование систем. Загальні уявлення про uml. [Електронний ресурс]. – Режим доступу : <https://studfile.net/preview/5110520/>
8. Лекція №4 Основні поняття об'єктно-орієнтованого програмування. інкапсуляція, успадкування та поліморфізм [Електронний ресурс]. – Режим доступу : <https://zavantag.com/docs/464/index-2037655.html>
9. Лабораторна робота №8 на тему Діаграма розгортання (Діаграма розгортання) [Електронний ресурс] Режим доступу: https://ua.kursovyks.com.ua/metodychni_ykazivky/article_post/1648-laboratorna-robita-no8-na-temu-diagrama-rozgortannya-deployment-diagram
10. Дизайн сайту [Електронний ресурс]. – Режим доступу: <https://websait.uz.ua/ru/dyzajrn-saitu/>
11. Уніфікована мова моделювання UML. Загальна характеристика. [Електронний ресурс]. Режим доступу :

<https://svitppt.com.ua/informatika/unifikovana-mova-modelyuvannya-uml-zagalna-harakteristika.html>.

12. Структурно-функціональний аналіз [Електронний ресурс] // сайт "Лексика - українські енциклопедії та словники". – Режим доступу:

https://leksika.com.ua/13211106/ure/strukturno-funktsionalniy_analiz

13. Функціональна декомпозиція (FDD – Functional Decomposition Diagram) [Електронний ресурс] // конспект лекцій з дисципліни "Системний аналіз і прийняття інноваційних рішень". – Режим доступу:

http://elkniga.info/book_26_glava_16_3.2._Metodi_ta_proceduri_sist.html

14. Принципи об'єктного підходу. Абстрагування. [Електронний ресурс] // сайт "Фрейми та об'єктно – орієнтоване програмування". – Режим доступу:

<http://shpora.me/mike/methody>

15. Тема 1. Місце та роль баз даних (БД) та бази знань (БЗ) у сучасних комп'ютерних інформаційних технологіях [Електронний ресурс] // – Режим доступу: <https://shpora.me/martik/bd5>

16. Діаграма класів (діаграма класів) [Електронний ресурс] // – Режим доступу: http://ni.biz.ua/6/6_3/6_33178_diagramma-klassov-class-diagram.html

17. Предметна область БД та її моделі [Електронний ресурс] // : Книга: Організація баз даних та знань. – Режим доступу до навч. посіб.

<https://www.bestreferat.ru/referat-377919.html>

18. Реферат. Проектування локальної мережі [Електронний ресурс] // – Режим доступу: <http://ukrefs.com.ua/169340-Proektirovanie-lokal-noiy-seti.html>

19. Проектування інформаційних систем [Електронний ресурс] // – Режим доступу: <http://um.co.ua/8/8-2/8-228764.html>

20. Робота з базами даних [Електронний ресурс] // – Режим доступу: <http://lib.mdpu.org.ua/e-book/web/lec6.htm>

21. Мова розмітки гіпертексту HTML 5 [Електронний ресурс] // – Режим доступу: <https://ppt-online.org/339940>

22. HTML, CSS, PHP, JavaScript, SQL – що і навіщо? [Електронний ресурс] // – Режим доступу: <https://kuryliak.pp.ua/reguser/1/P7/>

23. В.В. Волгин. Автосервис: структура и персонал. - 4 изд. М.: Издательско-

торговая кор-
порация «Дашков и К^о», 2008 – 692с

24. Положение о техническом обслуживании и ремон-

те дорожных втомобільно средств втомобільного транспорта. – Приказ

Минтранса Украи-

ны от 30.03.98 г. № 102.

25. Марков О.Д. Станции технического обслуживания автомобилей.

– К.: Кондор, 2008. – 536 с.10.

26. Котлер Ф., Армстронг Г., Сондерс Д., Вонг В. Основы мар-

кетинга // Пер. с англ.; 2-е европ. изд. – К.; М.; СПб.: Изд. дом «Вильямс», 1998.

1056 с.

27. ДСТУ ISO 9000 – 2001. Системи управління якістю. Основні положення

та словник.

28. ДСТУ ISO 9001 – 2001. Системи управління якістю. Вимоги.

29. ДСТУ ISO 9004 – 2001. Системи управління якістю. Настанови щодо

поліпшення

діяльності.14. Сертифікація в Україні. Нормативні акти та інші документи. Т.2.

Київ: Держстандарт України, 1998. – С. 143 – 154.15.

30. Правила обов'язкової сертифікації послуг з

ремонту та технічного обслуговування дорожніх транспортних засобів та їх

складових(затверджено наказом Держстандарту України від 28 серпня 1997 р.

№520).Вкн. Сертифікація в Україні. Нормативні акти та інші документи. Т.1. –

Київ. Держстандарт України, 1998. – С. 335 – 348

31. Головний сервісний центр МВС. URL: <http://texty.org.ua/cars> (дата

звернення: 19.03.2019).

32. Гончаров В., Ларін О. Аналіз існуючих підходів і методів оцінки

конкурентоспроможності промислового підприємства. Журнал «Схід». 2006.

№5(77). URL: [http://www.experts.in.ua/baza/analitc/](http://www.experts.in.ua/baza/analitc/index.php?ELEMENT_ID=11254)

[index.php?ELEMENT_ID=11254](http://www.experts.in.ua/baza/analitc/index.php?ELEMENT_ID=11254) (дата звернення: 19.03.2019).

33. Гончарова С.Ю., Отенко І.П. Стратегічне управління: навч. посібн. Харківський державний економічний університет. Харків: Вид-во ХДЕУ, 2004. 172 с.

34. Горбашко Е.А. Менеджмент качества и конкурентоспособности : учеб. пособ. СПб. : Изд-во ГУЭФ, 2010. – 207 с. 28- 181

35. Горбенко О.В. Логістика: Навчальний посібник. Київ: Знання, 2014. 315 с.

36. Городецький М. Я. “Management of international competitive capacity of an enterprise” (Управління міжнародною конкурентоспроможністю підприємства).

Mechanisms of interaction between competitiveness and innovation in modern international economic relations (Механізми взаємодії конкурентоспроможності та інновацій у сучасних міжнародних економічних відносинах): collective monograph / edited by M. Bezpartochnyi, in 4 Vol. ISMA University, Riga, Latvia.

37. Городецький М. Я. Алгоритм формування факторних дефініцій на конкурентоспроможність автосервісів. Politechnika Rzeszowska im. Ignacego Lukasiwicza. SYSTEMY I ŚRODKI TRANSPORTU SAMOCHODOWEGO. WYBRANE ZAGADNIENIA. SYSTEM AND MEANS OF MOTOR TRANSPORT. SELECTED PROBLEMS. - POLZA, M. RZESZOW – 2016. POD REDAKCJA NAUKOWA KAZIMIERZA LEJDY. MONOGRAFIA nr 7 SERIA: TRANSPORT.

38. Городецький М. Я. Бідняк М.Н. Методичні засади підвищення конкурентоспроможності підприємств автосервісу. Вісник наукових праць «Економічний аналіз», Національний транспортний університет. Київ, 2016.

39. Городецький М.Я. Аналітичне оцінювання управління конкурентоспроможністю автосервісних підприємств України. / М.Я. Городецький Микола. Аналітичне оцінювання управління конкурентоспроможністю автосервісних підприємств України. Інститут бухгалтерського обліку, контроль та аналіз в умовах глобалізації. 2019. Випуск 3-4. С. 54-67. Інститут бухгалтерського обліку, контроль та аналіз в умовах глобалізації. 2019. Випуск 3-4. С. 54-67

40. Городецький М.Я. Детермінанти впливу на конкурентоспроможність автосервісів. LXXII Наукова конференція професорсько-викладацького складу,

аспірантів, студентів та співробітників відокремлених структурних підрозділів у
Національному транспортному університеті. Київ, 2016.

41. Марков О.Д. Станции технического обслуживания автомобилей. К.:
Кондор, 2008. 536 с.

42. Андрусенко С.І. Організація фірмового обслуговування: навч. посіб. Київ,
1996. 216 с.

43. Бідняк М.Н., Городецький М.Я. Методичні засади підвищення
конкурентоспроможності підприємств автосервісу. Вісник НТУ. 2016. Вип. 2. 4.

Комп'ютерно-інформаційні технології автосервісу. URL: posibnyky.vntu.edu.ua
С.

44. Офіційний сайт «Укравто». [Електронний ресурс] // – Режим доступу:
<http://www.ukravto.ua/>.

45. Офіційний сайт «Богдан». [Електронний ресурс] // – Режим доступу:
<http://bogdan.ua/>.

46. Системи автосервісу. [Електронний ресурс] // – Режим доступу:
<http://nara.org.ua/index.php/2-uncategorised/39-sistemy-avtoservisa.html>.

47. Компанець К.А. Організаційно-економічне обґрунтування принципів та
методів управління підприємством (на прикладі туристичних організацій): дис.

канд. екон. наук. спец. 08.00.04. К., 2015. 235 с.

48. Петроченко С.М., Яковенко О.В. Організація комплексного автосервісу.
М.: Транспорт, 2007. 240 с.

49. Фастовцев Г.Ф. Автотехобслуговування. М.: Машинобудування, 2008.
256с.

50. Управління персоналом організації - Підручник для студентів вузів / За
ред. А.Я. Кифанова. М.: ИНФРА, 2006. 512 с.

51. Д. Дубровський «Відкриваємо автосервіс» 256 стор Пітер 2009

52. Бармашове Лариса Вікторівна «Фірмовий автосервіс» освітній сайт
МГТУ.

53. Шестопадов С.К., Шестопадов К.С. Легкові автомобілі. - М.: Транспорт,
2008. - 240 с.