

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
Факультет електроніки та інформаційних технологій

Кафедра комп'ютерних наук

Кваліфікаційна робота бакалавра
ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ ДІЯЛЬНОСТІ ФІРМИ
З ПРОКАТУ АВТОМОБІЛІВ

Здобувач освіти гр. ІН – 81

Едуард БРИЖКО

Науковий керівник

Дмитро ПРИЛЕПА

Завідувач кафедри
доктор техн.наук, професор

Анатолій ДОВБИШ

Суми 2022

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних наук

Затверджую _____

Зав. кафедри Довбиш А.С.

“ _____ ” _____ 2022 р.

ЗАВДАННЯ

до кваліфікаційної роботи

Здобувача вищої освіти четвертого курсу, групи ІН – 81 спеціальності
“Комп'ютерні науки” денної форми навчання.

Тема: «Інформаційна система для автоматизації діяльності фірми з прокату автомобілів»

Затверджена наказом по СумДУ

№ _____ від _____ 2022 р.

Зміст пояснювальної записки: 1) Огляд технологій, що застосовуються в розробці систем керування ресурсами; 2) Постановка завдання й формування завдань дослідження; 3) Огляд можливостей Dynamics 365 Business Central; 4) Моделювання інформаційної системи; 5) Розробка веб ресурсу; 6) Аналіз результатів.

Дата видачі завдання “ _____ ” _____ 2022 р.

Керівник кваліфікаційної роботи _____ Прилепа Д.В.

Завдання прийняв до виконання _____ Брижко Е.О.

РЕФЕРАТ

Записка: 85 стор., 37 рис., 1 таблиця, 1 додаток, 19 літературних джерел.

Об'єкт дослідження — процеси інформаційного аналізу і синтезу системи для автоматизації діяльності фірми з прокату автомобілів

Мета роботи — розробка SaaS рішення на базі Microsoft Dynamics 365 для фірми з прокату автомобілів

Результати — проведений аналіз інформаційних джерел, методів та інструментів, які дозволяють розробляти системи керування ресурсами котрі підтримують різні платформи. Після ознайомлення з існуючими рішеннями була обрана система керування ресурсами котру було модифіковано під потреби користувача, а саме достврено необхідні об'єкти системи та звіти. Дана розробка дозволяє проводити операції вставки, оновлення, видалення з усіма сутностями системи, аналізувати дані, що містяться в системі, друкувати або переглядати на основі цих даних звіти в форматі Word, PDF. Додаток був реалізований за допомогою мов програмування AL та RDL.

ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ АВТОМАТИЗАЦІЇ ДІЯЛЬНОСТІ
ФІРМИ З ПРОКАТУ АВТОМОБІЛІВ, INFORMATION SYSTEM FOR
AUTOMATION OF CAR RENTAL FIRM ACTIVITIES, AL, BUSINESS
CENTRAL, SSR

ЗМІСТ

ВСТУП.....	5
1 ІНФОРМАЦІЙНИЙ ОГЛЯД.....	7
1.1 Огляд існуючих технологій.....	7
1.2 Огляд аналогів.....	8
1.3 Постановка задачі.....	9
2 ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ.....	11
2.1 Аналіз предметної області.....	11
2.2 Структура бази даних.....	12
2.3 Бізнес логіка.....	19
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ.....	21
3.1 Налаштування середовища розробки.....	21
3.2 Короткий опис програмної реалізації.....	26
3.3 Тестування системи.....	31
ВИСНОВКИ.....	42
СПИСОК ЛІТЕРАТУРИ.....	43
ДОДАТОК.....	45

ВСТУП

Одним з основних чинників розвитку технологій являється спрощення та поліпшення повсякденного життя. Автоматизація процесу не тільки економить час на виконання тієї чи іншої роботи, а й зменшує ймовірні похибки при виконанні процесу. Варто розуміти, що чим більше підприємство тим більше грошей буде заощаджено від автоматизації того чи іншого процесу, адже вартість оплати людської праці для опрацювання великої кількості інформації буде значно більшою ніж купівля ліцензії будь якої з ERP систем. Сучасні ERP системи також мають основні аспекти захисту інформації, а саме: конфіденційності, цілісності, доступності [1].

Сфера оренди автомобілів являється досить популярною як в світі так і в Україні, адже автомобіль являється одним з найдоступніших та найпопулярніших засобів для пересування. Враховуючи це, компанія «Douchaty» шукає зручну систему для ведення звітності, адже розуміє що оперування інформацією в табличному редакторі MS Excel значно ускладнить та удорожчить цей процес.

Для полегшення процесу оренди автомобілей необхідно створити інформаційну систему котра пришвидшить самі процеси та допоможе уникнути можливих помилок.

Для досягнення мети необхідно:

- Проаналізувати предметну область, ознайомитись з усіма можливими процесами.
- Провести порівняння програмних продуктів які виконують схожі завдання.
- Сформулювати мету та методи її реалізації
- Змодельовати інформаційну систему та спроектувати базу даних
- Виконати програмну реалізацію поставленої задачі.

Це головні етапи котрим варто приділити особливу увагу. Аналіз предметної області дасть розуміння які варіанти використання системи можуть

буду взагалі, це допоможе при проектуванні бази даних. Проектування бази даних один з найважливіших етапів розробки. Від побудови бази даних залежить її гнучкість та можливість масштабування [2].

Також важливим пунктом є не тільки зручне введення інформації, а і її виведення, саме тому додатковим пунктом буде створення звіту в форматах PDF та Word котрий буде містити дані про конкретне замовлення.

1 ІНФОРМАЦІЙНИЙ ОГЛЯД

1.1 Огляд існуючих технологій

ERP – планування ресурсів підприємства. Система призначена для автоматизації обліку керування. Використовується як малими так і великими бізнесами на багатьох рівнях керування задачами [3].

ERP забезпечують керування наступними задачами:

- Керування виробництвом
- Керування розподілом запасів
- Керування фінансами
- Керування постачання
- Керування звітністю
- Керування взаємодією з клієнтами

З основних переваг:

- Використання одної інтегрованої програми замість декількох розрізнених.
- Забезпечення безпеки, внутрішньої та зовнішньої.

Недоліки:

- Проблема міграції даних.
- Необхідність достатньої кваліфікації співробітників
- Помилки в розробці програми

SaaS (Software as a service) – модель поширення продукту споживачам, при якій постачальник розробляє веб-додаток. Споживач платить не за володіння програмою, а за її використання [3].

Основні переваги:

- швидкість розгортання та впровадження
- багатоплатформність
- відсутність необхідності встановлення програмного забезпечення
- збереження даних в хмарі, а не на носіях

- всі оновлення відбуваються на стороні розробника та не потребують участі користувача

З основних недоліків:

- необхідність постійного підключення до мережі інтернет
- безпека захисту інформації

Програмне забезпечення – програма або сукупність програм системи для оброблення інформації та перелік документів необхідних для експлуатації цих програм [5].

Вирізняють три типи програмного забезпечення:

- системне
- інструментальне
- прикладне

ERP системи відносяться до прикладного програмного забезпечення адже вони не обслуговують потреби комп'ютера та не автоматизують процес розробки нових програм, а розв'язують задачі користувача.

З основних недоліків програмного забезпечення порівняно з SaaS моделлю, це необхідність додаткових витрат на обслуговування обладнання на якому буде запущене програмне забезпечення, підтримання необхідного рівню захисту інформації, необхідність в додатковому персоналі котрий буде слідкувати за оновленнями системи та забезпечення доставки цих оновлень на обладнання та слідкувати за сумісністю всього програмного забезпечення що використовується компанією.

1.2 Огляд аналогів

Універсальним інструментом ведення обліку є MS Excel. Має безліч формул для розрахунку будь яких показників, від звичайної суми, до розрахунку внутрішньої норми дохідності котра використовується в бухгалтерському обліку фондів приватних інвестицій. Використання макросів дозволяє завантажувати інформації з інших MS Excel документів. З недоліків виділяють відсутність бази даних, ддя поширення одного документу між працівниками та підтримання

актуальності інформації потрібно використовувати сторонню систему, наприклад Outlook [6].

Однією з найрозповсюджених ERP систем на території СНД є 1С: ERP. З основних причин такої розповсюдженості є невелика ціна на продукт та досить велика кількість розробників для 1С. Головним недоліком вважається низький рівень безпеки та захисту конфіденційної інформації [7].

Однією з найрозповсюджених ERP систем в світі є MS Dynamics ERP в склад якої входить декілька продуктів котрі в сукупності забезпечують єдину систему для автоматизації та аналізу бізнес процесів [8].

З основних переваг виділяють:

- постійні оновлення системи, а саме два рази на рік
- високий рівень безпеки та захисту конфіденційної інформації
- можливість розмежувати інформацію на рівні приватності
- мультивалютність та консолідація
- багатомовний інтерфейс
- гнучкість системи
- підтримка API

Одним з головних недоліків вважається велика ціна продукту порівняно з 1С та вартість додаткової розробки під індивідуальні потреби бізнеса також значно більша [9].

MS Dynamics ERP це SAAS система в той час, як для старих версії 1С: ERP необхідно встановити програмне забезпечення.

1.3 Постановка задачі

За результатами проведеного аналітичного огляду можна зробити висновок, що задача з розробки та провадження інформаційних систем в сфері обслуговування є актуальною. Тому метою даної роботи є розробка та програмна реалізація ресурсу для фірми з прокату автомобілів. Для досягнення поставленої мети необхідно виконати такі завдання:

1. Створення інформаційної моделі веб ресурсу компанії;

2. Виконати проектування та нормалізацію структури бази даних веб ресурсу фірми;
3. Визначити основні правила бізнес логіки системи та засоби її програмної реалізації;
4. Виконати проектування та реалізацію, що включає прототипування та конструювання графічного інтерфейсу користувача;
5. Визначити засоби для програмної реалізації веб ресурсу;
6. Програмна реалізація та тестування працездатності інформаційної системи.

2 ВИБІР МЕТОДУ РОЗВ'ЯЗАННЯ ЗАДАЧІ

2.1 Аналіз предметної області

«Douihaty» - молода компанія, котра надає послуги в сфері оренди автомобілей в таких містах України як: Київ, Суми, Севастополь, Львів, Черкаси та Рівне. «Douihaty» пропонує найкращі послуги оренди транспортних засобів по всій Україні. Клієнти можуть орендувати автомобілі, мікроавтобуси та вантажівки. Паркінги є в кожному місці. У деяких містах є декілька центрів для парковки. Парковки в Києві, Сумах, Черкасах мають місця для вантажівок, тоді як у Севастополі, Львові та Рівному - немає.

Компанії потрібна система планування ресурсів підприємства для забезпечення відповідної якості обслуговування клієнтів, комфортної роботи для своїх працівників і при цьому максимальному зменшенню витрат на ці потреби.

Часті зміни вартості оренди у зв'язку зі змінами вартості бензину, проблеми зі зручністю оформлення заявки на оренду, проблеми з цілісністю зберігання інформації, все це може призвести до великих проблем з обслуговуванням, що в свою чергу призведе до значних збитків та погіршення іміджу компанії.

За результатами обговорення проекту з представниками компанії з прокату автомобілів були визначені основні вимоги до веб ресурсу.

При цьому програмне забезпечення повинно задовольняти наступним вимогам:

- Створювати та змінювати транспортні засоби, місця та паркувальні центри та мати можливість переглянути загальний список створених сутностей.
- Створювати та закривати замовлення на оренду.
- Переглядати всі замовлення для конкретного клієнта.
- Переглядати всі транспортні засоби в певному місці;
- Переглядати всі транспортні засоби, які використовуються;

- Переглядати кількість вільних транспортних засобів на всіх паркувальних центрах;
- Кожен центр паркування повинен мати додатковий ідентифікатор на основі його назви в такому форматі:
«Центр паркування А» -> «ЦЕНТР-ПАРКУВАННЯ-А»
- В останній день кожного місяця підприємство проводить технічне обслуговування, в цей день неможливо орендувати або повернути транспортний засіб.
- Можливість створення звіту в форматі PDF та Word котрий буде містити в собі дані про конкретне замовлення або їх перелік.

2.2 Структура бази даних

Схема бази даних описує зміст та структуру даних для створення бази даних. Для реляційних баз даних схема визначає таблиці [10].

На схемі зазвичай визначають такі об'єкти:

- таблиці
- ключі
- індекси

База даних – сукупність даних збережених в відповідності до схеми даних [11].

Класифікація баз даних по моделі даних:

- реляційна
- ієрархічна
- об'єктна
- об'єктно-реляційна
- мережева

Класифікація за розміщенням даних:

- локальна
- розподілена

Структурована БД використовує структурований опис типу фактів за допомогою схеми даних.

Неструктурована БД це база даних яка містить тексти статей чи книг з функцією швидкого пошуку.

Система управління базами даних – сукупність програм призначених для створення та управління базами даних. Система забезпечує безпеку зберігання даних [12].

Основні функції СУБД:

- керування даними
- підтримка мов маніпуляції даними

СУБД складаються з:

- ядра
- процесору мови БД
- сервісних програм

Згідно з методом надання доступу до БД СУБД поділяються на:

- вбудовані;
- «Клієнт-сервер»
- «Файл-сервер»

За рівнем розподіленості СУБД бувають:

- розподіленими
- локальними

DFD – діаграма потоків даних [13].

Типи графічних елементів представлених на діаграмі:

- Процеси
- Сховища даних
- Зовнішні сутності
- Потоки даних між елементами

DFD діаграма має декілька рівнів деталізації.

DFD 0-го рівня також називається контекстною схемою. Це найпростіший спосіб зображення модельованих систем і процесів.

DFD 1-го рівня дає більш детальне уявлення про елементи контекстної схеми та виділяє основні функції системи [13].

ERD – діаграма відношення сутностей. Дозволяє понуритись в логіку бази даних затративши на це мінімальний час та сили [14].

Типи графічних елементів представлених на діаграмі:

- сутність
- атрибути
- зв'язок

Сутність – це будь який об'єкт в базі даних.

Атрибут – характеристика за допомогою якої можна описати певну сутність.

На сутності можуть бути представлені як ключові атрибути так і увесь список атрибутів. До ключових, відносять атрибути котрі унікально ідентифікують сутність.

Існує декілька типів зв'язків:

- один до одного
- один до багатьох
- багато до багатьох

Для побудови DFD та ERD діаграм я буду використовувати CaseStudioPortable. Інтерфейс програми досить зрозумілий та зручний (див. Рис 2.1).

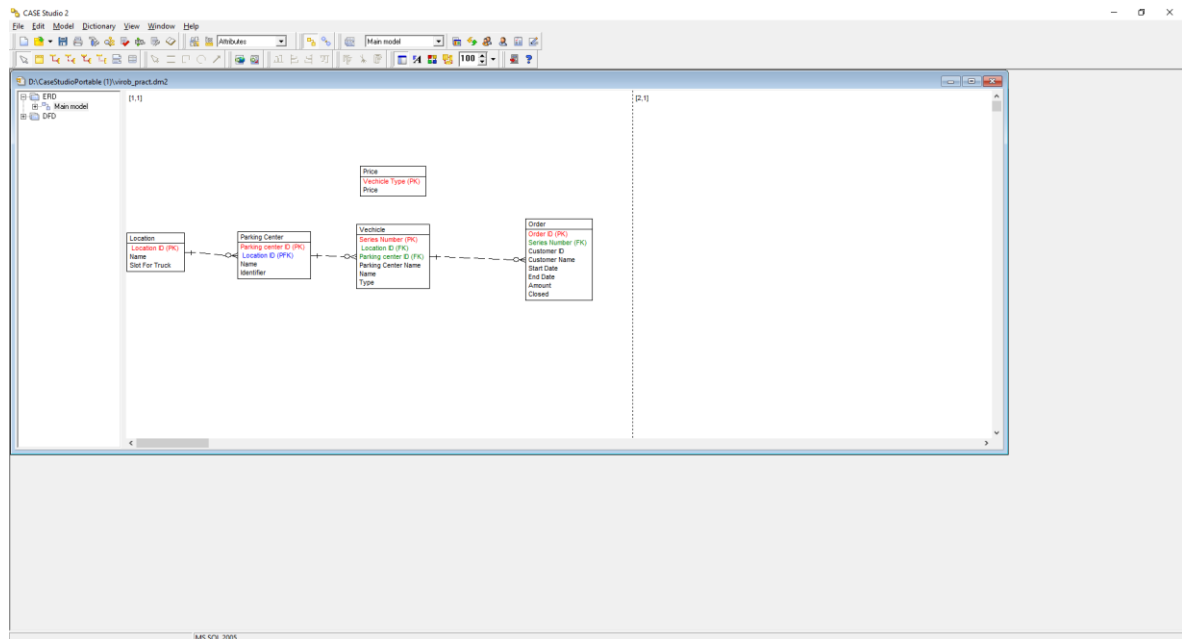


Рисунок 2.1 – Інтерфейс CaseStudio

В цій програмі є можливість будувати обидва типи діаграм та генерувати скрипт бази даних.

Для побудови DFD діаграми є всі необхідні елементи, а саме сховище даних, зовнішні сутності, процеси та потоки даних (див. Рис 2.2).

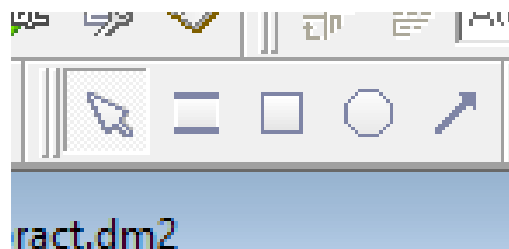


Рисунок 2.2 – Елементи для побудови DFD діаграми

Основними користувачами системи будуть оператори, бухгалтери та керуючий склад компанії. Оператори та бухгалтери будуть наповнювати систему основною інформацією, створювати відповідні документи, звіти тощо. Керуючий склад компанії буде аналізувати бізнес процеси та ґрунтуючись на своїх спостереженнях покращувати деякі з них.

Діаграма 0-го рівня описує загальні процеси обміном інформацією між користувачами і системою (див. Рис 2.3).

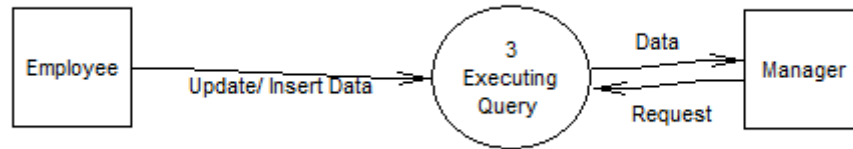


Рисунок 2.3 – DFD 0-го рівня

Діаграма 1-го рівня більш детально описує потік даних (див. Рис 2.4).

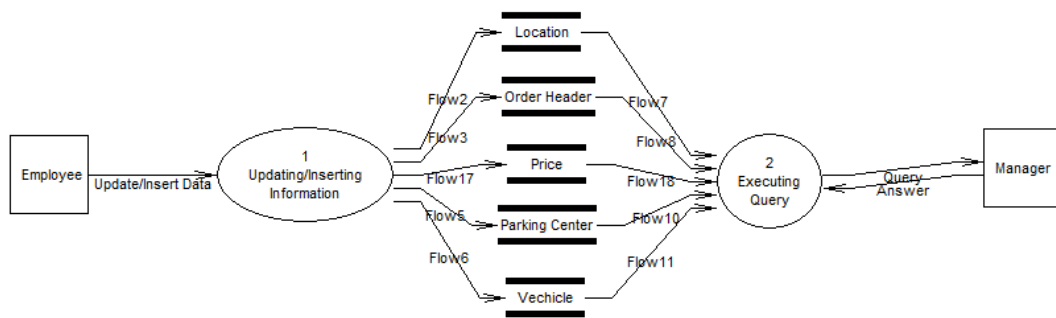


Рисунок 2.4 – DFD 1-го рівня

Для обробки процесів модель на рис. 2.4 має такі сховища даних як: «Location», «Order Header», «Order Line», «Parking Center», «Price», «Customer» і «Vechicle». Всі запити від співробітників діляться на процес «Updating/Inserting Information», який змінює сховище даних і процес «Executing Query», який обробляє запити від керуючого складу компанії, знаходить необхідну інформацію в сховищах даних і формує відповідь на запит.

Для побудови ERD діаграми в CaseStudio є наступні елементи: сутність та різні типи зв'язків (див. Рис 2.5). Розміщувати елементи користувач може в будь якій кількості та сам обирає кінцеве розташування кожної сутності, проте варто

пам'ятати що від цього залежить як швидко розробник може зрозуміти загальну структуру та логіку процесу.



Рисунок 2.5 – Елементи для побудови ERD діаграми

Згідно до вимог розробника в деяких містах України паркінги мають місця для вантажівок, а деякі ні, тому в таблицю «Location» було додане поле «Slot For Truck» типу Boolean.

Також зазначено, що в деяких містах може бути декілька паркінг центрів, тому зв'язок від сутності «Location» до «Parking Center» один до багатьох. Для сутності «Parking Center» також необхідно додати додатковий атрибут «Identifier» про який теж згадувалось в вимогах. В одному паркінг центрі може стояти декілька автомобілів тому і зв'язок між сутністю «Parking Center» та «Vechicle» буде один до багатьох.

Для зручнішого користування інформаційною системою денормалізуємо сутність «Order», а саме додамо окрім атрибуту «Vechicle Series Number», що є первинним ключом таблиці «Vechicle», додаткові атрибути «Location ID» та «Parking Center ID», котрі являються первинними ключами таблиць «Location» та «Parking Center» відповідно. Зроблено це для того, щоб оператор при заповненні цих полів звужував список транспортних засобів котрі будуть фільтруватися за вказаними параметрами міста та парковочного центру, тобто цим порушується третя нормальна форма, а саме залежність всіх атрибутів від первинного ключа. Таким чином інформаційна система приведена до другої нормальної форми, при цьому важливо встановити додаткові перевірки для усунення потенційної суперечливості і надмірності даних. Також важливо додати перевірки на видалення батьківських сутностей для підтримання цілісності даних.

На ERD діаграмі відображені вищезазначені сутності системи, їх атрибути та зв'язок між сутностями (див. Рис 2.6).

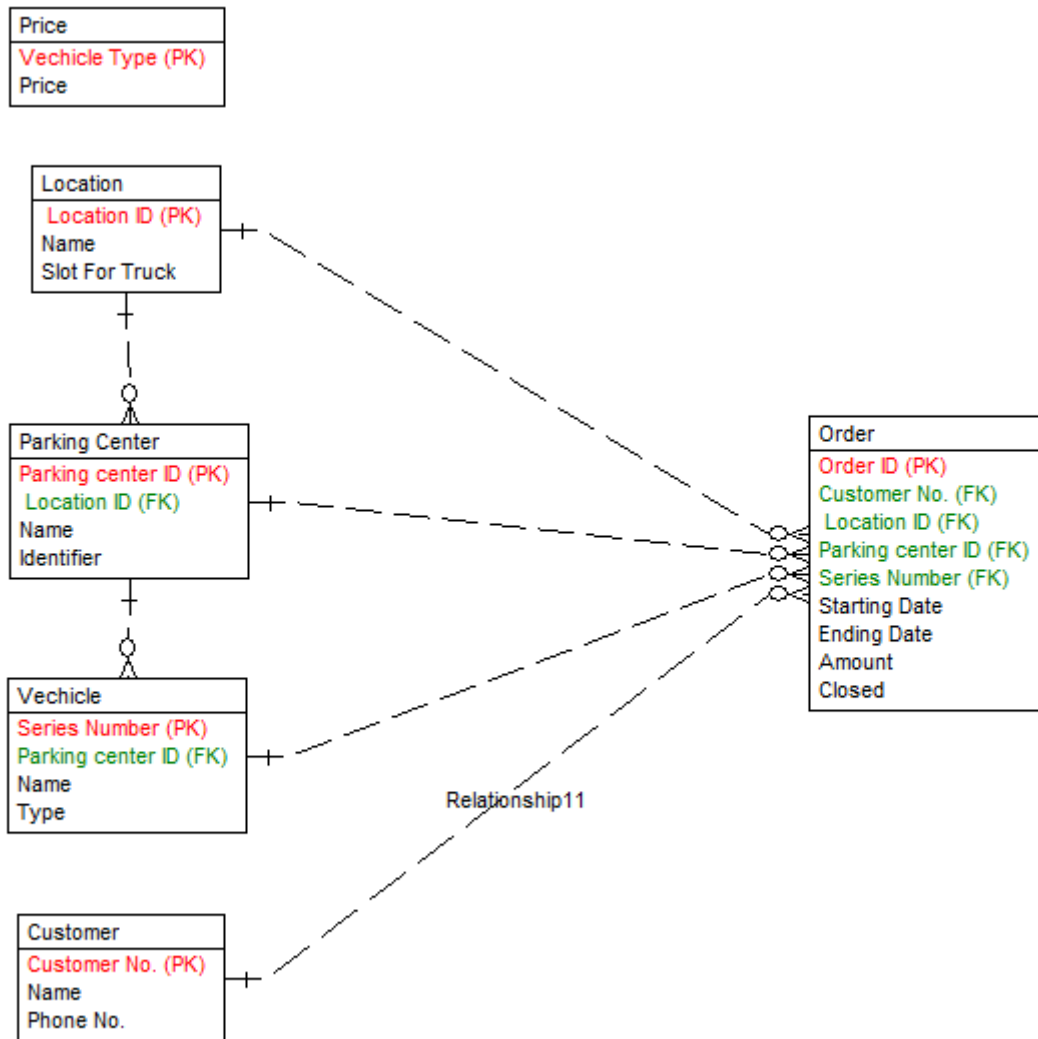


Рисунок 2.6 – ERD діаграма

Цих діаграм достатньо для аналізу загальної логіки процесів та побудови бази даних. Після побудови користувач має можливість згенерувати скрипт обравши потрібну йому базу даних (див. Рис 2.7), проте в нашому випадку це не потрібно.

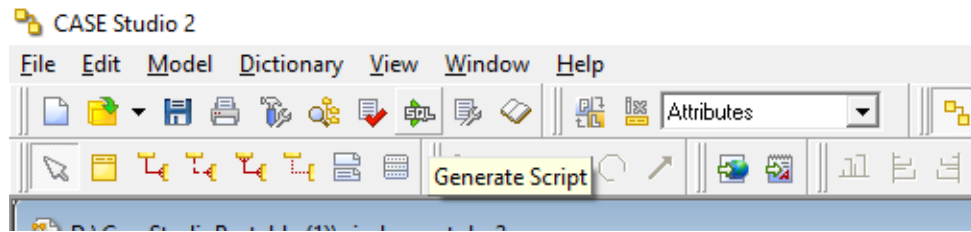


Рисунок 2.7 – Генерація скрипта

Побудувавши всі необхідні діаграми можна приступити до визначень загальної бізнес логіки.

2.3 Бізнес логіка

В розробленій базі даних встановлені наступні обмеження та вимоги:

- Повертати орендований автомобіль потрібно в той самий паркінг центр з якого він був орендований
- В останній день кожного місяця підприємство проводить технічне обслуговування, в цей день неможливо орендувати або повернути транспортний засіб

Також в таблиці представлені імена необхідних таблиць, атрибутів, типи та обмеження за довжиною (див. табл. 2.1).

Таблиця 2.1 – Структура бази даних

Назва таблиці	Назва поля	Тип	Ключ
Location	ID	Integer	PK
Location	Name	Text[100]	
Location	Slot For Truck	Boolean	
Parking Center	ID	Integer	PK
Parking Center	Location ID	Integer	FK
Parking Center	Name	Text[100]	
Parking Center	Identifier	Code[100]	

Продовження табл. 2.1

Vehicle	Series Number	Code[20]	PK
Vehicle	Parking Center ID	Integer	FK
Vehicle	Name	Text[100]	
Vehicle	Type	Text[50]	
Customer	No.	Code[20]	PK
Customer	Name	Text[100]	
Customer	Phone No.	Code[20]	
Order	ID	Integer	PK
Order	Customer No.	Code[20]	FK
Order	Location ID	Integer	FK
Order	Parking Center ID	Integer	FK
Order	Vehicle Series Number	Code[20]	FK
Order	Date of rent	Date	
Order	Date of return	Date	
Order	Amount	Decimal	
Order	Closed	Boolean	

Після визначення обмежень бази даних можна перейти до практичної реалізації, а саме розробки інформаційної системи.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ

3.1 Налаштування середовища розробки

Для розробника в першу чергу важливий рівень безпеки даних та можливе масштабування бізнесу в інші країни, тому доцільним буде обрати саме MS Dynamics ERP за їх переваги. Інтерфейс Business Central представлений на рисунку 3.1.

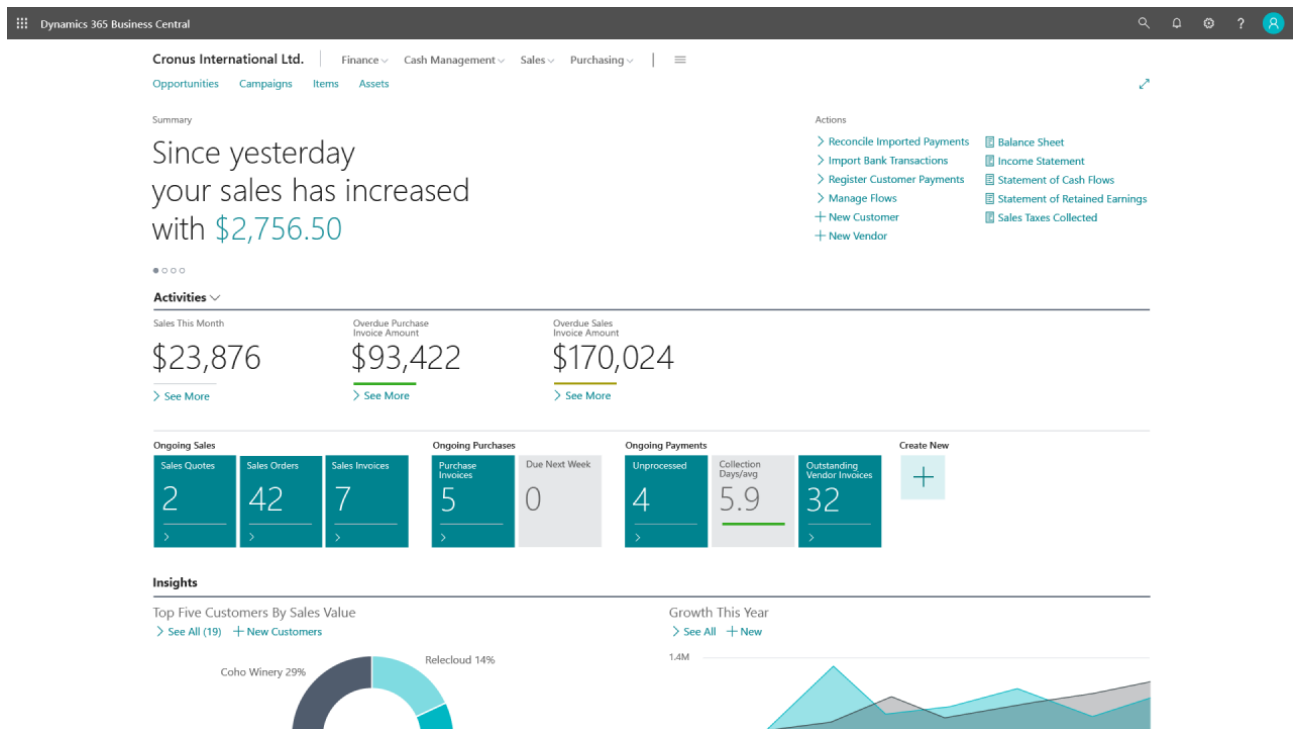


Рисунок 3.1 – Інтерфейс Dynamics 365 Business Central

Переглянувши основні можливості MS Dynamics ERP я прийшов висновку, що потрібно модифікувати систему згідно до вимог розробника оскільки базові можливості їх не покривають. Для цього в Dynamics 365, що є частиною MS Dynamics ERP передбачені розширення системи.

AL – це об'єктно базована мова програмування яка використовується для маніпулювання даними (наприклад, отримання, вставлення та зміна записів) у базі даних Dynamics 365 Business Central [16].

Розробник може створити будь-який з наступних об'єктів з власним наповненням:

- таблиця
- сторінка картка
- сторінка список
- перелічення
- звіт
- інтерфейс

Кожен об'єкт має свій тип, унікальний номер та унікальне ім'я. Стандартні об'єкти використовують діапазон унікальних номерів від 0 до 49999, для кастомізованих об'єктів виділений діапазон з 50000 до 99999, а отже всі об'єкти котрі будуть створені повинні мати унікальний номер в межах від 50000 до 99999 [16].

Для самої ж розробки потрібно створити безпечне середовище. В безпечному середовищі можна тестувати написаний код, або усувати неполадки, не порушуючи робочих процесів компанії або бізнес-даних. Таке середовище називається пісочницею. Пісочниця — це місце, де можна безпечно досліджувати, вивчати, демонструвати, розробляти й тестувати службу без ризику вплинути на дані та налаштування вашого робочого середовища [17].

Створювати пісочницю буду за допомогою Docker. Docker є провідним міжплатформним контейнерним середовищем програмного забезпечення. Це дозволяє користувачам пакувати програми з усіма їх залежностями та розповсюджувати цей пакет на інші машини. Ці машини називаються хостами, і вони можуть запускати пакет як повністю ізольоване середовище, яке є віртуалізованим. Пакет називається зображенням [17].

Образ Docker — це шаблон з мінімальною кількістю файлів операційної системи та файлів програми, які йому необхідно запустити. Все, що вам потрібно для запуску програми, є на зображенні. Образи Docker зберігаються в реєстрі Docker, куди користувачі можуть завантажувати та завантажувати зображення. Для Business Central Microsoft розміщує власний реєстр [17].

Для розробки та тестування можна запускати контейнери в ізоляції процесів у Windows 10 Pro та Enterprise, якщо ви хочете запустити Docker у виробництві, вам завжди знадобиться Windows Server як операційна система хоста але Microsoft не підтримує контейнери Docker для Business Central у виробничому середовищі.

Розміщувати контейнер буду з використанням Hyper-V ізоляції. Цей режим забезпечує підвищену безпеку та кращу сумісність між операційною системою хоста та операційною системою базового образу контейнера. У цьому режимі буде запущена високо оптимізована віртуальна машина, яка матиме власне ядро. Контейнер буде працювати всередині даної віртуальної машини.

Образ Docker можна плутати з віртуальною машиною, але вони досить різні. Для запуску образу не потрібна повна операційна система, оскільки вона використовує ядро операційної системи хоста. Також у зображенні немає графічного інтерфейсу який є в віртуальній машині.

При запуску зображення, створюється його екземпляр. В результаті всі створені зміни зберігаються поверх зображення, а тому саме зображення не буде змінено. Примірник зображення називається контейнером. Ви можете створити екземпляр кількох контейнерів з одного образу, і всі вони зберігатимуть свої зміни в різних файлах і не будуть заважати один одному.

Основними компонентами Docker є двигун Docker і клієнт Docker. Двигун — це процес або додаток, що реалізує фактичну роботу, тоді як клієнт використовується для зв'язку з двигуном [17].

Всі контейнери Business Central створюються з базою даних Cronus котра являється тестовою базою даних і містить в собі мінімальний набір даних для тестування основних компонентів системи. Ви можете запустити контейнер, розробити та протестувати свою функцію в базі даних Cronus, або ж створити власну базу даних, де при створенні можна обрати один з декількох режимів її наповнення або взагалі без тестових даних.

Для розробки друкованого звіту буду використовувати SQL Server Reporting Services (SSRS). Служби Reporting Services пов'язані зі

«традиційними» звітами з розбивкою на сторінки, що ідеально підходять для документів із фіксованим макетом, оптимізованих для друку, таких як файли PDF та Word [18].

Даний серіс можна встановити в якості додатку для Visual Studio 2019 та Visual Studio 2022.

Під час написання коду варто дотримуватись певних стандартів адже гарно написаний код набагато легше підтримувати та розширювати - такий код називається чистим [19].

Чистий код має такі особливості:

- проходить всі тести
- не містить дублювання
- містить мінімальну кількість сутностей

Гарний код має читатись як книга, адже з читабельністю приходить розширюваність та ремонтпридатність [19]. Для цього потрібно використовувати деякі правила неймінгу, а саме:

- використовуйте імена що розкривають наміри
- використовуйте імена які можна вимовити
- виберіть одне слово для кожного поняття

Та уникайте цього при виборі назв:

- уникайте дезінформації
- остерігайтесь використання імен які незначно відрізняються

При написанні функцій варто дотримуватись таких принципів:

- не повторюйте себе (уникайте дублювання)
- уникайте побічних ефектів
- використовуйте описові назви
- розташовуйте функцій в порядку їх виклику

Поганий код це одна з найголовніших мотивацій залишити коментар. Тож замість написання коментаря витратьте час на покращення логіки коду [19].

Команда розробників повинна домовитись про єдиний стиль форматування, а потім кожен член цієї команди повинен використовувати цей стиль.

Для того щоб допомогти розробнику дотримуватись деяких з правил написання чистого коду існує безліч розширень для Visual Studio Code, я буду використовувати набір розширень для AL, можна завантажити одразу пак в якому зібрані одні з найкращих екстеншенів для AL та повимикати непотрібні вам, прикладом такого паку можу слугувати Al Extension Pack котрий включає в себе наступні розширення (див рисунок 3.2).

AL Extension Pack

This pack includes extensions to improve your D365BC AL development.

Extensions

The AL Extension Pack includes:

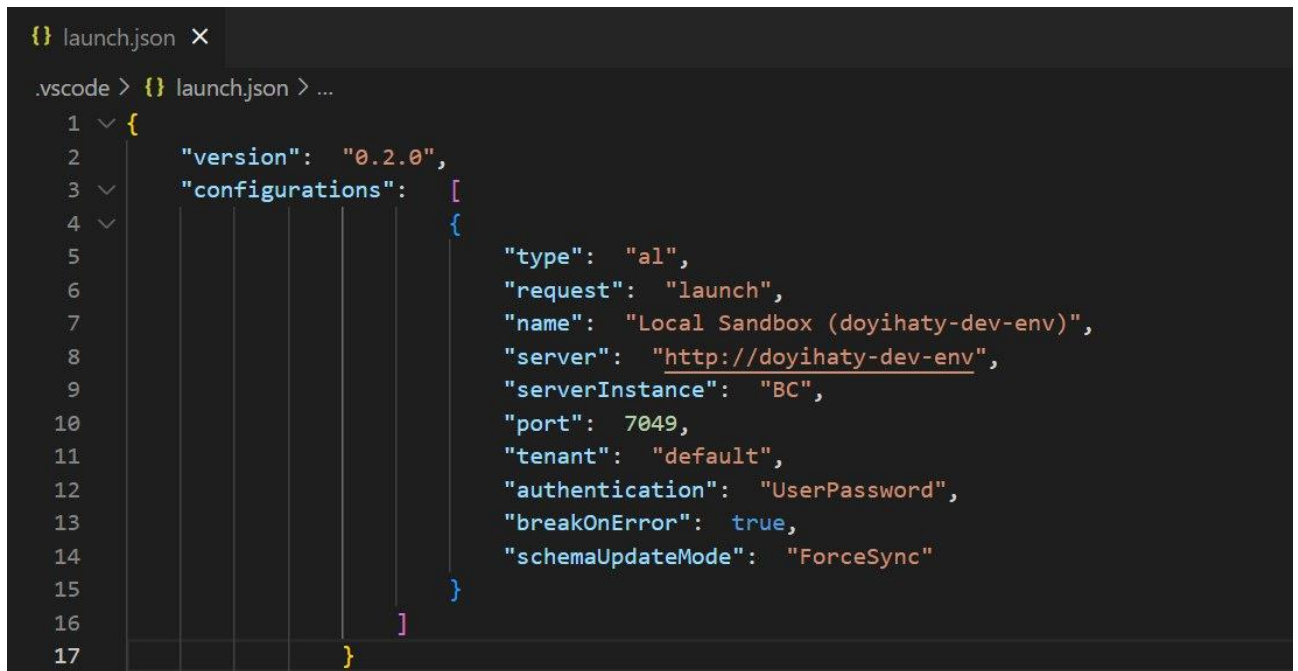
- [AL CodeActions](#)
 - [AL Formatter](#)
 - [AL Language](#)
 - [AL Navigator](#)
 - [AL Object Designer](#)
 - [AL Object ID Ninja](#)
 - [AL Structure Creator](#)
 - [AL Test Runner](#)
 - [AL Toolbox](#)
 - [AL Variable Helper](#)
 - [AL XML Documentation](#)
 - [AZ AL Dev Tools/AL Code Outline](#)
 - [Azure Repos](#)
 - [Better Comments](#)
 - [Create GUID](#)
 - [Debugger for Chrome](#)
 - [Docker](#)
 - [Docker Explorer](#)
 - [Dynamics NAV C/AL](#)
 - [Git History](#)
 - [GitLens — Git supercharged](#)
 - [Live Share](#)
 - [PowerShell](#)
 - [Remote - Containers](#)
 - [REST Client](#)
 - [TSLint](#)
 - [VSCode-Icons](#)
 - [Waldo's CRS AL Language Extension](#)
 - [XLIFF Sync](#)
-

Рисунок 3.2 – Список розширень з Al Extension Pack

В переліку даних розширень знаходяться розширення для більш комфортної роботи як з кодом AL так і з Docker, Git, PowerShell.

3.2 Короткий опис програмної реалізації

Для підключення до локального середовища розробки необхідного в файлі `launch.json` котрий створюється при створенні проекту прописати наступні конфігурації (див. рисунок 3.3)



```
{
  "version": "0.2.0",
  "configurations": [
    {
      "type": "al",
      "request": "launch",
      "name": "Local Sandbox (doyihaty-dev-env)",
      "server": "http://doyihaty-dev-env",
      "serverInstance": "BC",
      "port": 7049,
      "tenant": "default",
      "authentication": "UserPassword",
      "breakOnError": true,
      "schemaUpdateMode": "ForceSync"
    }
  ]
}
```

Рисунок 3.3 – Вміст конфігураційного файлу

Проаналізувавши наявні об'єкти бази даних Dynamics 365 Business Central було прийнято рішення використовувати стандартні об'єкти котрі пов'язані з сутністю Customer, а саме: таблиця, сторінка список та сторінка картка. Для всіх інших сутностей таблиці та сторінки були створені власноруч. Проаналізувавши ERD діаграму було прийнято рішення створити об'єкт типу перелічення для типу техніки, котре буде мати такі поля як автомобіль, мікроавтобус та вантажівка (див. рисунок 3.4)

```

1  enum 59800 "DHT Vechicle Type"
2  {
3      Caption = 'Vechicle Type';
4
5      0 references
6      value(10; Car)
7      {
8          Caption = 'Car';
9      }
10     0 references
11     value(20; Microbus)
12     {
13         Caption = 'Microbus';
14     }
15     0 references
16     value(30; Truck)
17     {
18         Caption = 'Truck';
19     }
20 }

```

Рисунок 3.4 – Перелічення типів техніки

Для вирішення проблеми додаткові перевірки для усунення потенційної суперечливості і надмірності даних необхідно не забути вказати властивість `TableRelation` яка дозволяє встановлювати пошук в інших таблицях та не дозволяє ввести значення в поле котрого немає в вказаній в цій властивості таблиці. Як приклад (див. рисунок 3.5) поле «Vechicle Series Number» значення якого залежить від значення поля «Parking Center ID» в цій же таблиці, тобто фактично ми вказуємо батьківську таблицю, поле батьківської таблиці в яке потрібно дивитись та за потреби накладаємо фільтр на батьківську таблицю.

```

19 references
field(40; "Vechicle Series Number"; Code[20])
{
    Caption = 'Vechicle Series Number';
    TableRelation = "DHT Vechicle"."Series Number" where("Parking Center ID" = field("Parking Center ID"));

    trigger OnValidate()
    begin
        CalcFields("Vechicle Name", "Vechicle Type");
    end;
}

```

Рисунок 3.5 – Використання властивості `TableRelation`

Для вирішення проблеми цілісності даних було використано додаткові функції котрі були розміщені в OnDelete тригері.

Тригер OnDelete запускається якщо користувач хоче видалити якийсь запис через інтерфейс користувача або при виклику функції Delete(true), де параметр функції відповідає за виклик тригеру, значення за замовченням false. Тобто під час виклику тригеру OnDelete іде перевірка на існування дочірніх записів і якщо такі існують вони будуть видалені разом з батьківським записом. Для прикладу приведу функцію з OnDelete тригера таблиці Location (див. рисунок 3.6)

```

trigger OnDelete()
begin
    DeleteRelatedParkingCentersWithConfirmation();
end;

1 reference
local procedure DeleteRelatedParkingCentersWithConfirmation()
var
    DHTLocationMgt: Codeunit "DHT Location Mgt";
    RelatedRecordExistMsg: Label 'All related records will be deleted. Do you want to continue?';
begin
    if DHTLocationMgt.RelatedParkingCentersExist(ID) then
        if GuiAllowed() then
            if Confirm(RelatedRecordExistMsg, false) then
                DHTLocationMgt.DeleteRelatedParkingCenters(ID)
            else
                Error('');
end;

```

Рисунок 3.6 – Функція для видалення дочірніх сутностей

Також в вимогах було зазначено, що в деяких містах немає паркувальних місць для вантажівок, для реалізації цієї вимоги потрібно в OnValidate тригер полів «Type» та «Parking Center ID» таблиці «Vehicle» поставити відповідну перевірку (див. рисунок 3.7). Тригер OnValidate запускається коли дані вводяться в поле для якого він вказаний.

```

field(30; Type; Enum "DHT Vechicle Type")
{
    Caption = 'Vechicle Type';

    trigger OnValidate()
    begin
        CheckVechicleType();
    end;
}

```

Рисунок 3.7 – Перевірка типу машини

Для створення звіту потрібно створити об'єкт з типом звіт та обрати один з двох варіантів його макету, а саме RDLC або Word. Я обрав RDLC адже він має більше можливостей для налаштування під потреби користувач. Загальний вигляд звіту показаний на рисунку 3.8.

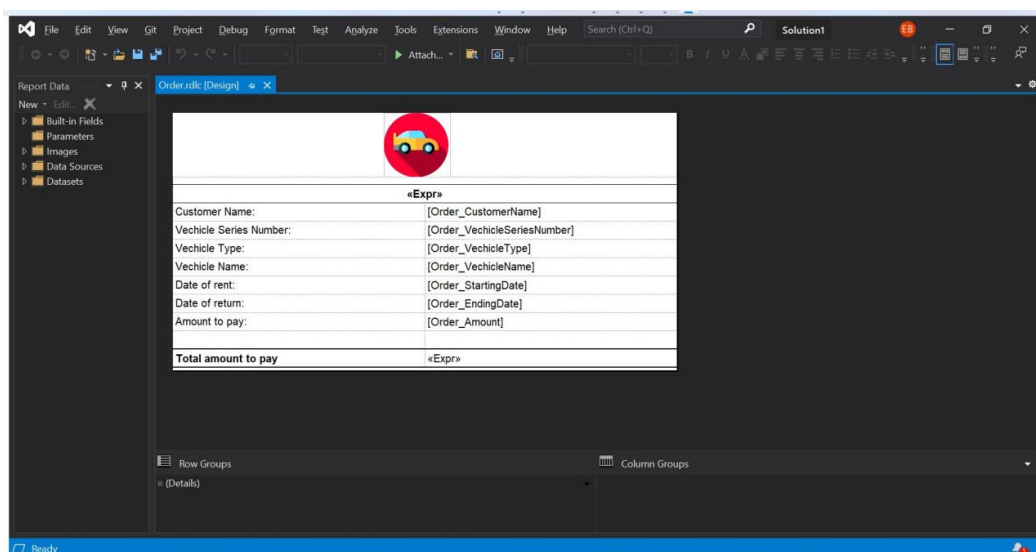


Рисунок 3.8 – Загальний вигляд звіту

Також одним з плюсів SSRS звітів є можливість використання вбудованих функцій. В моєму випадку я використовую функцію Sum (див. рисунок 3.9) для вирахування загальної суми до сплати котра відображена в останньому рядку звіту.

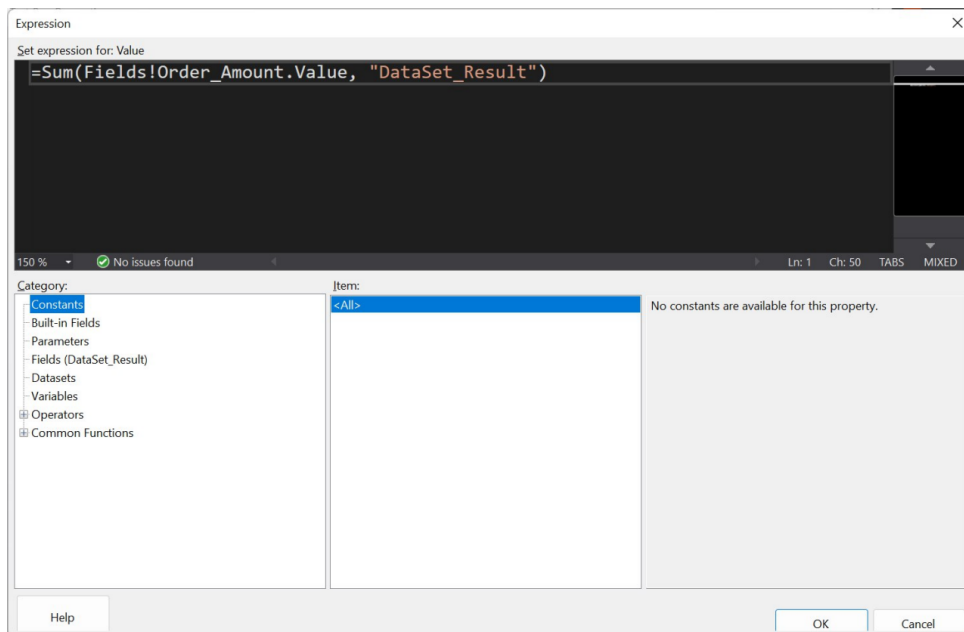


Рисунок 3.9 – Використання вбудованої функції

Першою сторінкою котру буде бачити користувач при авторизації буде Role Center. Role Center - точка входу користувача та домашня сторінка для Dynamics 365. В Dynamics 365 є можливість налаштувати декілька Role Center сторінок. Якщо система використовується декількома типами користувачів, можна створити окремий Role Center під кожен тип користувача. Це зробить домашню сторінку більш адаптивною під потреби користувача. Для поточного замовника це було менш пріоритетно, тому з домашньої сторінки було прибрано всі зайві компоненти та додані вкладки для швидкого переходу на потрібні сторінки та звіти (див. рисунок 3.10)

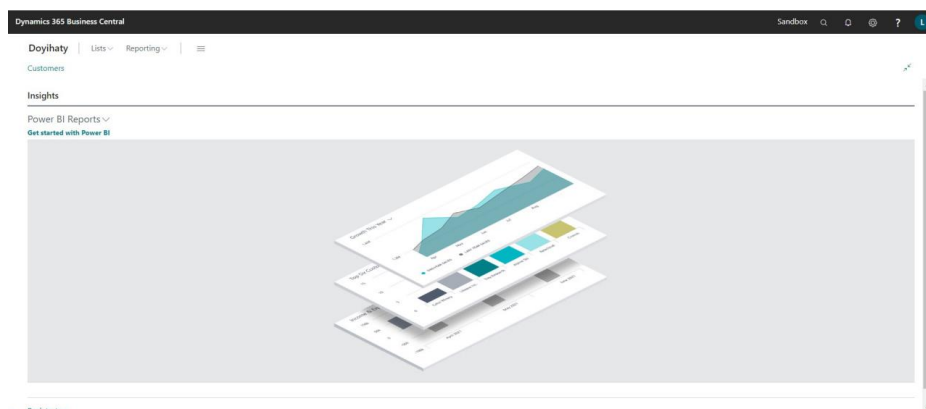


Рисунок 3.10 – Role Center

3.3 Тестування системи

Одним з основних вимог клієнта була можливість зручно створювати та переглядати сутності. Для цього на домашній сторінці була введена вкладка з усіма сутностями (див. рисунок 3.11)

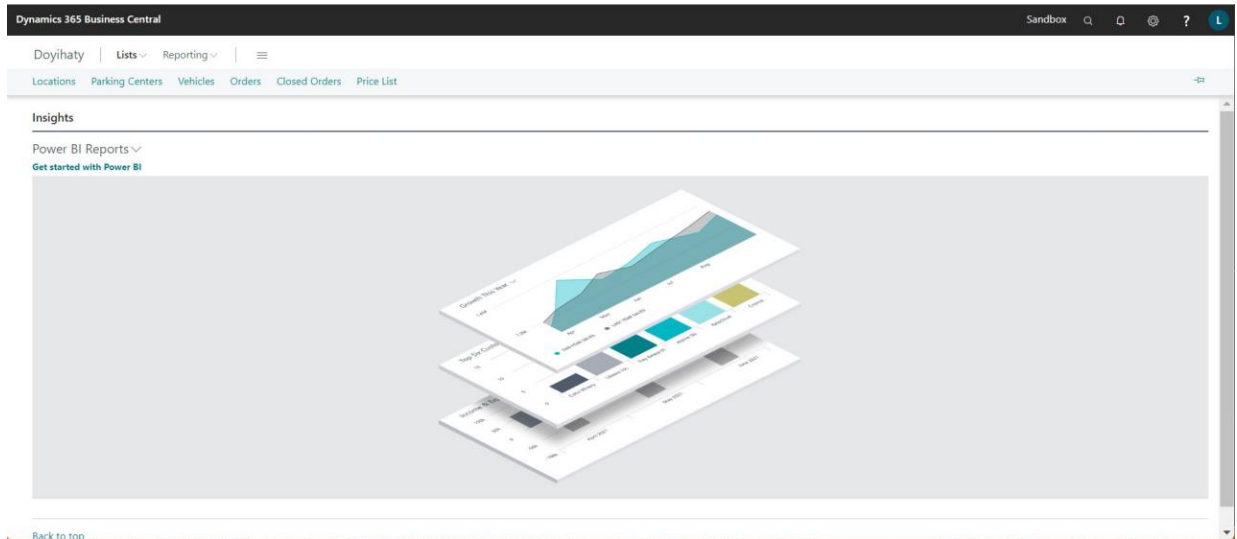


Рисунок 3.11 – Домашня сторінка з вкладками

Є можливість проглянути всі локації рисунок 3.12

Name	Slot For Truck
Sumy	<input checked="" type="checkbox"/>
Cherkasy	<input checked="" type="checkbox"/>
Kyiv	<input checked="" type="checkbox"/>
Lviv	<input type="checkbox"/>
Rivne	<input type="checkbox"/>
Sevastopol	<input type="checkbox"/>

Рисунок 3.12 – Список локацій

Створити нову, відрегувати існуючу або видалити локацію рисунок 3.13

The screenshot shows a mobile application interface for editing a location. At the top, there is a navigation bar with a back arrow, the title 'DHT Location', and several action icons (edit, share, add, delete). Below the navigation bar, the number '5' is displayed. The main form area is titled 'GroupName' and contains two fields: 'Name' with the value 'Sumy' and 'Slot For Truck' which is a toggle switch currently turned on. The interface is clean and modern with a light gray background.

Рисунок 3.13 – Картка локації

Аналогічні сторінки створені для паркінг центрів, транспорту, замовлень, переліку цін та користувачів.

При створенні транспортного засобу якщо вказати тип автомобілю вантажівка та обрати паркінг центр з міста де немає місць для вантажівки користувач отримає помилку (див. рисунок 3.14)

The screenshot shows a mobile application interface for editing a vehicle. At the top, there is a navigation bar with a back arrow, the title 'Vechicle', and several action icons (edit, share, add, delete). Below the navigation bar, the text 'Not saved' is displayed. The main form area is titled 'GroupName' and contains several fields: 'Series Number' (AA0001AA), 'Name' (BMW 3i), 'Vehicle Type' (Truck), 'Parking Center ID' (7), 'Parking Center', and 'Location Name'. A yellow error banner at the top of the form area reads: 'The page has an error. Refresh (F5) to undo the change, or correct the error.' A red 'X' icon is visible next to the 'Parking Center ID' field, and a 'Validation Results' dialog box is open, displaying the message: 'The defined location doesn't contains slots for truck'.

Рисунок 3.14 – Виведення помилки про неправильний тип

При створенні замовлення на оренду при заповненні поля «Location ID» поле «Parking Center ID» відфільтровується та показуються лише паркінг центри для відповідної локації (див. рисунок 3.15)

The screenshot shows a web form titled 'Order' with a '0' indicator. The form is divided into several sections:

- Customer:** Customer No. (1), Customer Name (Eduard Bryzhko).
- Location:** Location ID (5), Location Name (Sumy).
- Parking Center:** Parking Center ID (0), Parking Center Name (—).
- Vehicle:** A dropdown menu is open, showing a list of parking centers:

Name	Location	Identifier
Parking Center A	Sumy	PARKING-C...
Parking Center B	Sumy	PARKING-C...
- Dates:** Starting Date and Ending Date fields with asterisks indicating required fields.
- Amount To Pay:** Amount (0.00).

Рисунок 3.15 – Фільтрація паркінг центрів за локацією

Аналогічно при заповненні поля «Parking Center ID» поле «Vehicle Series Number» відфільтровується лише для даного паркінг центру.

При заповненні даних в полях «Renting Date» та «Returning Date» перевіряється коректність введених даних (див. рисунок 3.16)

The screenshot shows the 'Dates' section of the form. The 'Renting Date' field contains '2/1/2022' and the 'Returning Date' field contains '1/2/2022'. A red error message is displayed:

Validation Results
 Renting Date is bigger than Returning Date. Please, define date in correct format to proceed.

Below the dates, the 'Amount To Pay' section shows an amount of '-9,300.00'.

Рисунок 3.16 – Перевірка правильності введення дати

Також однієї з вимог було обмежити можливість повернути або взяти в оренду автомобіль в останній день місяця (див рисунок 3.17)

The screenshot shows a form with two main sections: 'Dates' and 'Amount To Pay'. In the 'Dates' section, the 'Renting Date' is set to 1/1/2022 and the 'Returning Date' is set to 1/31/2022. A red 'x' icon is next to the return date field, indicating an error. A 'Validation Results' pop-up box is displayed over the return date field, containing the text: 'Sorry it is impossible to rent or return a car on the last day of month due to internal policy. Please, choose another day.' In the 'Amount To Pay' section, the 'Amount' field shows 0.00.

Рисунок 3.17 – Перевірка на останній день місяця

Після введення даних в коректному форматі буде автоматично розрахована сума до сплати (див. рисунок 3.18)

The screenshot shows a completed rental order form. At the top, there is a navigation bar with a back arrow, the word 'Order', and several icons (edit, share, add, delete). Below this is a 'Close Order' button. The form is divided into several sections: 'Customer' with fields for 'Customer No.' (1) and 'Customer Name' (Eduard Bryzhko); 'Location' with fields for 'Location ID' (5) and 'Location Name' (Sumy); 'Parking Center' with fields for 'Parking Center ID' (5) and 'Parking Center Name' (Parking Center A); 'Vehicle' with fields for 'Vehicle Series Number' (AA0001AA), 'Vehicle Type' (Truck), and 'Vehicle Name' (BMW 3i); 'Dates' with fields for 'Renting Date' (2/1/2022) and 'Returning Date' (2/5/2022); and 'Amount To Pay' with a field for 'Amount' (1,200.00).

Рисунок 3.18 – Сформоване замовлення

Також під час вибору машини та внесені дат перевіряється її доступність, тобто чи не використовується вона в визначеному проміжку часу (див. рисунок 3.19)

Order

0

The page has an error. [Refresh \(F5\)](#) to undo the change, or correct the error.

Customer

Customer No. Customer Name

Location

Location ID Location Name

Parking Center

Parking Center ID Parking Center Name

Vehicle

Vehicle Series Number Vehicle Type

Vehicle Name

Dates

Renting Date Returning Date

Amount To Pay

Amount

Validation Results
Current Vehicle is in use on specified date, please choose other vehicle or dates

Рисунок 3.19 – Перевірка машини на доступність

Після того як клієнт поверне машину та оплатить замовлення, замовлення можна буде закрити відповідною кнопкою «Close Order» (див. рисунок 3.17). Після натискання кнопки замовлення буде перенесене до завершених, а сторінка

закрита. Це замовлення тепер можна буде знайти на сторінці «Closed Orders» (див. рисунок 3.20)

ID ↑	Customer Name	Location Name	Parking Center Name	Renting Date	Returning Date	Amount
8	Eduard Bryzhko	Sumy	Parking Center A	2/1/2022	2/5/2022	1,200.00

Рисунок 3.20 – Завершені замовлення

Ціна замовлення розраховується відповідно до вказаних даних (див. рисунок 3.21)

Type ↑	Price
Car	100.00
Microbus	200.00
Truck	300.00

Рисунок 3.21 – Список цін

Для перегляду всіх замовлень для конкретного клієнта потрібно з домашньої сторінки зайти на вкладку «Reporting» де обрати необхідну сторінку «Customer Orders». Де після заповнення відповідних полів з'явиться кнопка показати результат котра виведе перелік всіх замовлень для обраного клієнта (див. рисунок 3.22)

Customer Orders ✓ Saved

Filters

Customer No. Closed

Show Result

Рисунок 3.22 – Фільтрація замовлень по клієнтам

Після натиснення на кнопку було отримано наступний результат (див. рисунок 3.23)

Customer Orders ✓ Saved

Filters
Customer No. ... Closed

	ID ↑	Location Name	Parking Center Name	Vehicle Series Number	Vehicle Name	Vehicle Type
→	8	Sumy	Parking Center A	AA0001AA	BMW 3i	Truck

Рисунок 3.23 – Результат фільтрації по клієнту

Для перегляду всіх транспортних засобів по локації була реалізована сторінка з подібною логікою для закріплення у замовника певного користувацького досвіду (див. рисунок 3.24)

Vehicle On Location ✓ Saved

Filters
Location ... Vehicle Type ...
Parking Center ...

	Parking Center Name	Vehicle Series Number	Vehicle Name	Vehicle Series Number
→	Parking_Center_A	AA0002AA	Mercedes 999	Car

Рисунок 3.24 – Результат фільтрації по локації

Для перегляду всіх транспортних засобів які використовуються потрібно вказати дати оренди та повернення для більш зручного пошуку (див. рисунок 3.25)

The screenshot shows the 'DHT Vehicles In Use' interface. At the top, there are navigation icons (back, edit, share, add, delete) and a title 'DHT Vehicles In Use'. Below the title is the vehicle ID 'AA0004'. A 'Filters' section contains two date pickers: 'Renting Date' set to '1/1/2022' and 'Returning Date' set to '2/1/2022'. Below the filters is a table of vehicles:

Series Number ↑	Name	Vehicle Type	Parking Center	Location Name
AA0001AA	BMW 3i	Truck	Parking Center A	Sumy
AA0002AA	Mercedes 999	Car	Parking Center A	Sumy
→ AA0004	Volvo 888	Microbus	Parking Center A	Sumy

Рисунок 3.25 – Результат фільтрації транспортних засобів що використовуються

Для того щоб отримати кількість доступних автомобілів на даний момент потрібно натиснути на кнопку «Show Number of Available Vehicles» котра знаходиться на сторінці з переліком всіх автомобілів. В результаті на екран буде виведено повідомлення (див. рисунок 3.26)

The screenshot shows the 'Doyihaty' interface. At the top, there are navigation icons (back, search, new, delete, edit list, show number of available vehicles) and a title 'Doyihaty'. Below the title is a 'Vehicles' section with a dropdown menu set to 'All'. Below the 'Vehicles' section is a table of vehicles:

Series Number ↑	Name	Vehicle Type	Parking Center ID	Parking Center	Location Name
AA0001AA	BMW 3i	Truck	5	Parking Center A	Sumy
AA0002AA	Mercedes 999	Car	5	Parking Center A	Sumy
AA0004	Volvo 888	Microbus	5	Parking Center A	Sumy

A notification dialog box is displayed in the foreground with the text: "Number of Available Vehicles on Current Date: 3". The dialog box has an information icon and an 'OK' button.

Рисунок 3.26 – Повідомлення з кількістю доступних автомобілів

Для того щоб отримати документ в форматі Word, PDF або XML потрібно зайти на сторінку «Print Order» на вкладці «Reporting». На сторінці вказати необхідні параметри для фільтрації, при чому якщо ви не бажаєте фільтрувати по якомусь з параметрів просто залиште його порожнім (див. рисунок 3.27) та натисніть «Send to...» де оберіть необхідний вам параметр (див. рисунок 3.28)

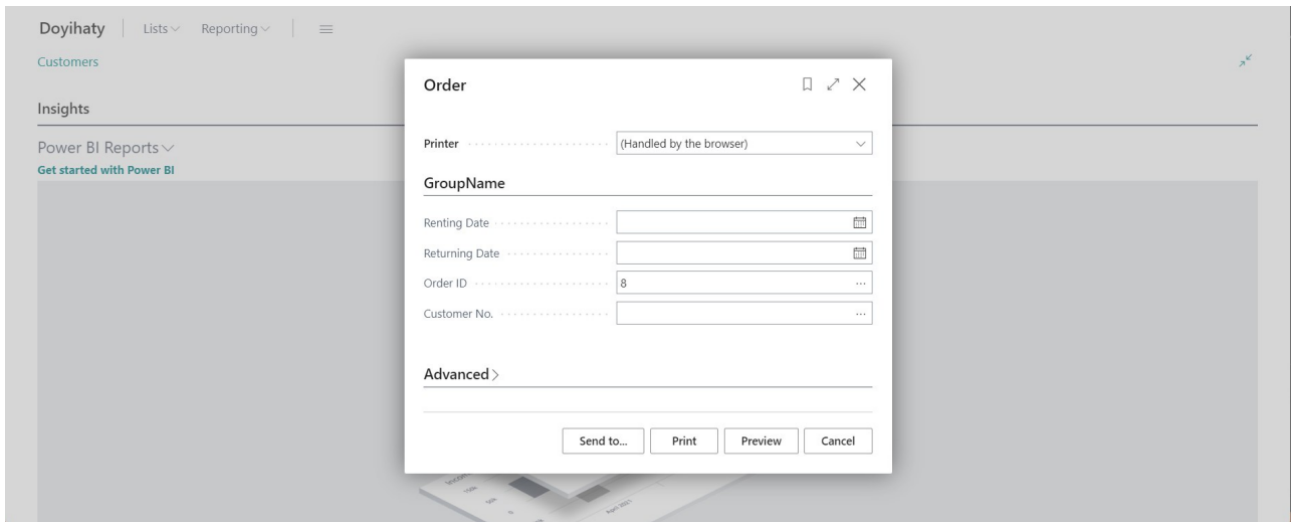


Рисунок 3.27 – Сторінка для фільтрації

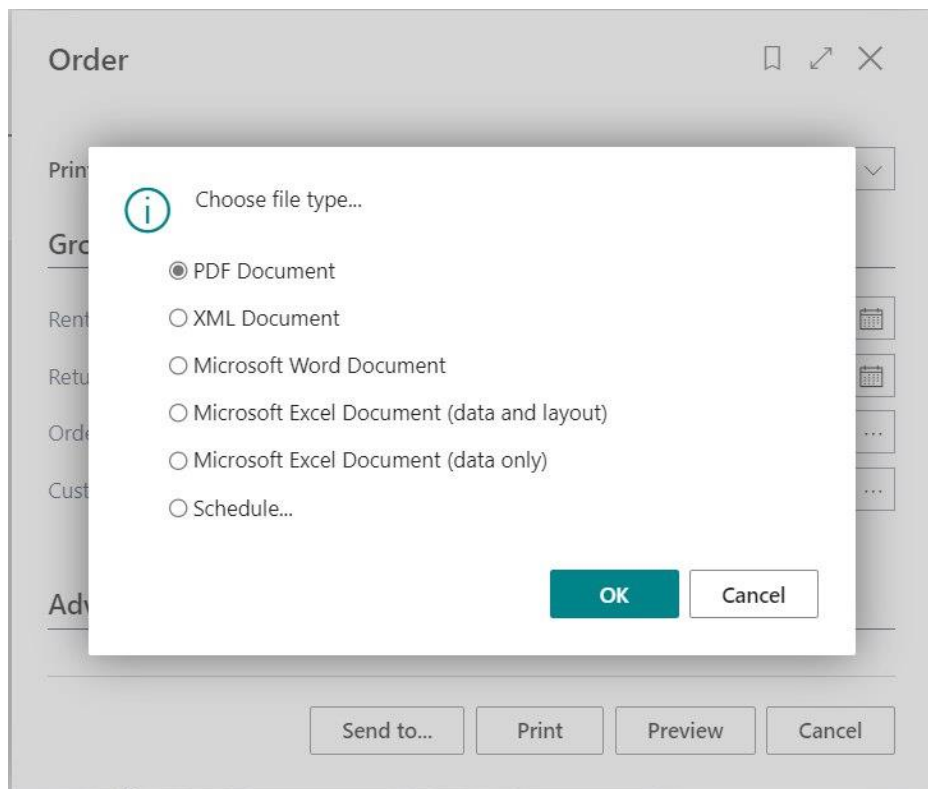


Рисунок 3.28 – Вибір вихідного файлу

В результаті буде згенеровано файл з вказаним форматом (див. рисунок 3.29)

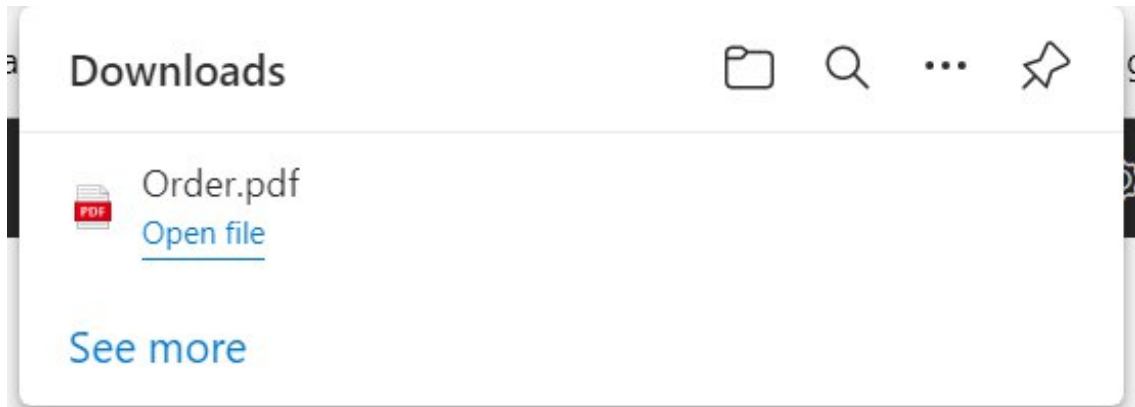


Рисунок 3.29 – Вихідний файл

Та відповідною інформацією (див. рисунок 3.30)

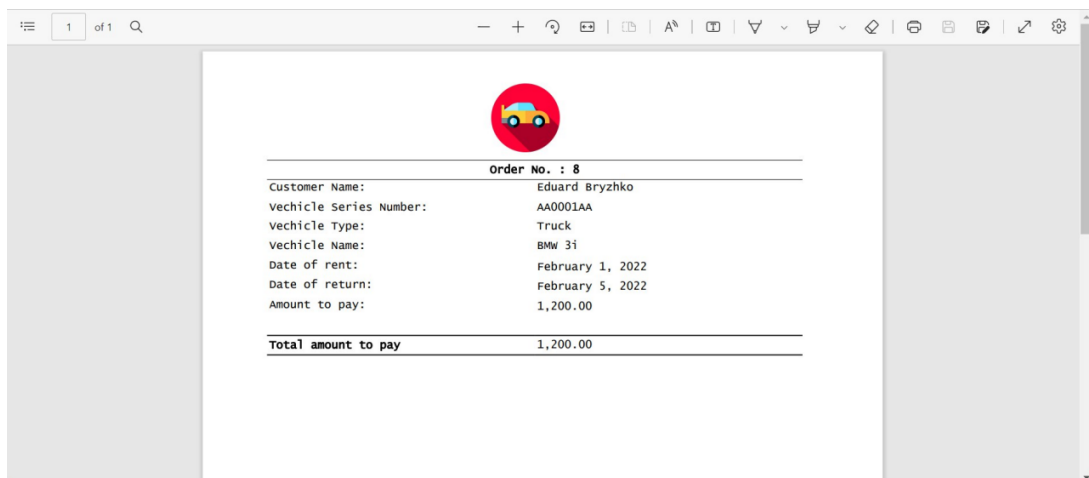


Рисунок 3.30 – Контент сгенерованого файлу

Також варто зазначити що на всіх сторінках користувач може натиснувши на три крапки в кінці поля може побачити список всіх існуючих записів. Наприклад, якщо на сторінці «Vehicle On Location» в полі «Location» натиснути на три крапки з'явиться сторінка з усіма записами з таблиці «Location» (див. рисунок 3.31).

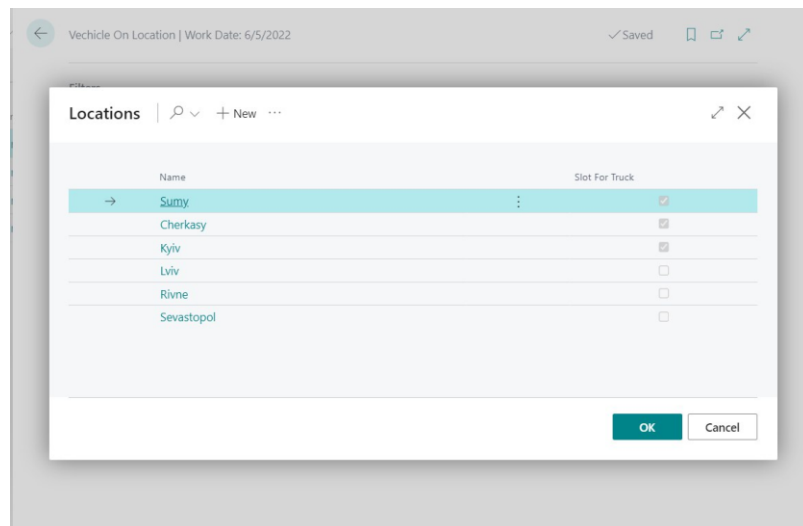


Рисунок 3.31 – Допоміжна сторінка

Це значно полегшує та пришвидшує процес створення будь якої з сутностей, зменшує можливість появи помилок про некоректність введеної інформації та також дає можливість до створювати деякі сутності не закриваючи поточної сторінки.

ВИСНОВКИ

В ході аналізу декількох ERP систем та врахування всіх їх недоліків та переваг була обрана та, котра найбільш підходить до вимог замовника та були виконані наступні завдання:

- Створено інформаційної моделі веб ресурсу компанії;
- Було виконано проектування та нормалізацію структури бази даних веб ресурсу фірми, а саме побудовані DFD діаграми нульового та першого рівнів та ERD діаграма;
- Визначено основні правила бізнес логіки системи та засоби її програмної реалізації;
- Виконано проектування та реалізацію, що включає прототипування та конструювання графічного інтерфейсу користувача;
- Визначено засоби для програмної реалізації веб ресурсу;
- Було створене безпечне середовище розробки та виконана програмна реалізація та тестування працездатності інформаційної системи.

Розроблений веб ресурс задовольняє всім вимогам, поставленим на етапі постановки задачі. Інформаційна система вийшла гнучкою та розширювальною.

СПИСОК ЛІТЕРАТУРИ

1. Tarik K. ERP Implementation: Requirements Engineering for ERP Product Customization. 2020
2. Modeling Styles in Business Process Modeling / J. Pinggera et al. Enterprise, Business-Process and Information Systems Modeling. Berlin, Heidelberg, 2012. P. 151–166. URL: https://doi.org/10.1007/978-3-642-31072-0_11
3. Gautam A. Enterprise Resource Planning. Digitising Enterprise. 2021. P. 161–172. URL: <https://doi.org/10.1201/9781003203131-14>
4. Matt C. Software as a service. Controlling & Management. 2009. Vol. 53, no. 3. P. 151–152. URL: <https://doi.org/10.1007/s12176-009-0048-5>
5. Software Development Methodologies, Educational Software Development Processes. Encyclopedia of Education and Information Technologies. Cham, 2020. P. 1564. URL: https://doi.org/10.1007/978-3-030-10576-1_300612
6. Katz A. Introduction to Excel. Beginning Microsoft Excel 2010. Berkeley, CA, 2010. P. 1–15. URL: https://doi.org/10.1007/978-1-4302-2956-8_1
7. Babu K. V. S. N. J., Bezawada M. Enterprise Resource Planning. SSRN Electronic Journal. 2012. URL: <https://doi.org/10.2139/ssrn.2142346>
8. Critchley S. Dynamics 365 Essentials. Berkeley, CA : Apress, 2020. URL: <https://doi.org/10.1007/978-1-4842-5911-5>
9. Yapa S. Out-of-the-Box Customizations. Customizing Dynamics 365. Berkeley, CA, 2019. P. 69–107. URL: https://doi.org/10.1007/978-1-4842-4379-4_3
10. Özsu M. T. Database. Encyclopedia of Database Systems. New York, NY, 2018. P. 931–932. URL: https://doi.org/10.1007/978-1-4614-8265-9_80734
11. Biskup J., Brüggemann H. H. Towards Designing Acyclic Database Schemes. Advances in Data Base Theory. Boston, MA, 1984. P. 3–25. URL: https://doi.org/10.1007/978-1-4615-9385-0_1
12. Coulombe H. N. Data base development: Overview. Washington, DC : U.S. Environmental Protection Agency, Office of Research and Development

13. Li Q., Chen Y.-L. Data Flow Diagram. Modeling and Analysis of Enterprise and Information Systems. Berlin, Heidelberg, 2009. P. 85–97. URL: https://doi.org/10.1007/978-3-540-89556-5_4
14. Kamps T. Diagram Design / Thomas Kamps., 1999. – (Springer-Verlag Berlin Heidelberg).
15. Programming Microsoft Dynamics 365 Business Central – Birmingham: Packt Publishing Ltd, 2019. – 511 c. – (1300419), ISBN 978-1-78913-779-8.
16. Customizing Dynamics 365, 2019. – (Apress). – (978-1-4842-4378-7).
17. Vohra D. Hello Docker. Pro Docker. Berkeley, CA, 2016. P. 1–18. URL: https://doi.org/10.1007/978-1-4842-1830-3_1
18. Aitchison A. Reporting Services. Pro Spatial with SQL Server 2012. Berkeley, CA, 2012. P. 445–470. URL: https://doi.org/10.1007/978-1-4302-3492-0_17
19. Clean Code: A Handbook of Agile Software Craftsmanship. Pearson Education, Limited, 2009.

ДОДАТОК

```

table 59800 "DHT Location"
{
    Caption = 'Location';
    DrillDownPageId = "DHT Location";

    fields
    {
        field(10; ID; Integer)
        {
            AutoIncrement = true;
            Caption = 'ID';
            Editable = false;
            MinValue = 0;
        }
        field(20; Name; Text[10])
        {
            Caption = 'Name';
        }
        field(30; SlotForTruck; Boolean)
        {
            Caption = 'Slot For Truck';
        }
    }

    keys
    {
        key(PK; ID)
        {
            Clustered = true;
        }
    }

    fieldgroups
    {
        fieldgroup(DropDown; Name, SlotForTruck)
        {
        }
    }

    trigger OnDelete()
    begin
        DeleteRelatedParkingCentersWithConfirmation();
    end;

    local procedure DeleteRelatedParkingCentersWithConfirmation()
    var
        DHTLocationMgt: Codeunit "DHT Location Mgt";
        RelatedRecordExistMsg: Label 'All related records will be delated. Do you want to
continue?';
    begin
        if DHTLocationMgt.RelatedParkingCentersExist(ID) then
            if GuiAllowed() then
                if Confirm(RelatedRecordExistMsg, false) then
                    DHTLocationMgt.DeleteRelatedParkingCenters(ID)
                else
                    Error('');
            end;
    end;
}
page 59800 "DHT Location"
{
    ApplicationArea = All;
    PageType = Card;
    SourceTable = "DHT Location";
    UsageCategory = Administration;
}

```

```

layout
{
    area(Content)
    {
        group(GroupName)
        {
            field(Name; Rec.Name)
            {
                ApplicationArea = All;
            }
            field(SFT; Rec.SlotForTruck)
            {
                ApplicationArea = All;
            }
        }
    }
}

page 59801 "DHT Locations"
{
    ApplicationArea = All;
    Caption = 'Locations';
    CardPageId = "DHT Location";
    Editable = false;
    PageType = List;
    SourceTable = "DHT Location";
    UsageCategory = Lists;

    layout
    {
        area(Content)
        {
            repeater(control)
            {
                field(Name; Rec.Name)
                {
                    ApplicationArea = All;
                }
                field(SlotForTruck; Rec.SlotForTruck)
                {
                    ApplicationArea = All;
                }
            }
        }
    }

    procedure GetSelectionFilter() Result: Text
    var
        DHTLocation: Record "DHT Location";
        SelectionFilterManagement: Codeunit SelectionFilterManagement;
        RecordRef: RecordRef;
    begin
        CurrPage.SetSelectionFilter(DHTLocation);
        RecordRef.GetTable(DHTLocation);
        Result := SelectionFilterManagement.GetSelectionFilter(RecordRef,
DHTLocation.FieldNo("ID"));
    end;
}

codeunit 59801 "DHT Location Mgt"
{
    procedure RelatedParkingCentersExist(LocationID: Integer) Result: Boolean

```

```

var
  DHTParkingCenter: Record "DHT Parking Center";
begin
  DHTParkingCenter.SetRange("Location ID", LocationID);
  Result := not DHTParkingCenter.IsEmpty();
end;

procedure DeleteRelatedParkingCenters(LocationID: Integer)
var
  DHTParkingCenter: Record "DHT Parking Center";
  DHTVehicle: Record "DHT Vehicle";
begin
  DHTParkingCenter.SetRange("Location ID", LocationID);
  if DHTParkingCenter.FindSet() then
    repeat
      DHTVehicle.SetRange("Parking Center ID", DHTParkingCenter.ID);
      if DHTVehicle.FindSet() then
        repeat
          DHTVehicle.DeleteRelatedOpenOrders();
          DHTVehicle.Delete();
          until DHTVehicle.Next() = 0;
        DHTParkingCenter.Delete();
      until DHTParkingCenter.Next() = 0;
    end;
  }
table 59802 "DHT Parking Center"
{
  Caption = 'Parking Center';
  DrillDownPageId = "DHT Parking Center";

  fields
  {
    field(10; ID; Integer)
    {
      AutoIncrement = true;
      Caption = 'ID';
      Editable = false;
      MinValue = 0;
    }
    field(20; Name; Text[100])
    {
      Caption = 'Name';

      trigger OnValidate()
      var
        begin
          DefineIdentifier;
        end;
    }
    field(30; "Location ID"; Integer)
    {
      Caption = 'Location ID';
      TableRelation = "DHT Location".ID;

      trigger OnValidate()
      begin
        CalcFields("Location Name");
      end;
    }
    field(40; "Location Name"; Text[100])
    {
      CalcFormula = lookup("DHT Location".Name where(ID = field("Location ID")));
      Caption = 'Location';
      Editable = false;
      FieldClass = FlowField;
    }
    field(60; Identifier; Code[100])
    {
      Editable = false;

```

```

    }
}

keys
{
    key(PK; ID)
    {
        Clustered = true;
    }
}

fieldgroups
{
    fieldgroup(DropDown; Name, "Location Name", Identifier)
    {
    }
}

local procedure DefineIdentifier()
begin
    Identifier := UpperCase(Name);
    while StrPos(Identifier, ' ') > 0 do
        Identifier := (DelStr(Identifier, StrPos(Identifier, ' ')) + '-' +
CopyStr(Identifier, StrPos(Identifier, ' ') + StrLen(' ')));
end;

trigger OnDelete()
begin
    DeleteRelatedVehiclesWithCheck();
end;

local procedure DeleteRelatedVehiclesWithCheck()
var
    DHTParkingCenterMgt: Codeunit "DHT Parking Center Mgt";
    RelatedRecordExistMsg: Label 'All related records will be deleted. Do you want to
continue?';
begin
    if DHTParkingCenterMgt.RelatedVehicleExists(ID) then
        if GuiAllowed() then
            if Confirm(RelatedRecordExistMsg, false) then
                DHTParkingCenterMgt.DeleteRelatedVehicle(ID)
            else
                Error('');
end;

procedure DeleteRelatedVehicles()
var
    DHTParkingCenterMgt: Codeunit "DHT Parking Center Mgt";
begin
    DHTParkingCenterMgt.DeleteRelatedVehicle(ID)
end;
}
page 59803 "DHT Parking Center"
{
    ApplicationArea = All;
    Caption = 'Parking Center';
    PageType = Card;
    SourceTable = "DHT Parking Center";
    UsageCategory = Administration;

    layout
    {
        area(Content)
        {
            group(GroupName)
            {
                field(Name; Rec.Name)
                {
                    ApplicationArea = All;

```



```

end;

procedure DeleteRelatedVechicle(ParkingCenterID: Integer)
var
    DHTVechicle: Record "DHT Vechicle";
begin
    DHTVechicle.SetRange("Parking Center ID", ParkingCenterID);
    if DHTVechicle.FindSet() then
        repeat
            DHTVechicle.DeleteRelatedOpenOrders();
            DHTVechicle.Delete();
        until DHTVechicle.Next() = 0;
    end;
}

table 59803 "DHT Vechicle"
{
    Caption = 'Vechicle';
    DrillDownPageId = "DHT Vechicle";

    fields
    {
        field(10; "Series Number"; Code[20])
        {
            Caption = 'Series Number';
        }
        field(20; Name; Text[100])
        {
            Caption = 'Name';
        }
        field(30; Type; Enum "DHT Vechicle Type")
        {
            Caption = 'Vechicle Type';

            trigger OnValidate()
            begin
                CheckVechicleType();
            end;
        }
        field(40; "Parking Center ID"; Integer)
        {
            Caption = 'Parking Center ID';
            TableRelation = "DHT Parking Center".ID;

            trigger OnValidate()
            var
                ParCenter: Record "DHT Parking Center";
            begin
                CheckVechicleType();
                CalcFields("Parking Center Name", "Location ID");
                DefineLocationName();
            end;
        }
        field(50; "Parking Center Name"; Text[100])
        {
            CalcFormula = lookup("DHT Parking Center".Name where(ID = field("Parking Center
ID"))));
            Caption = 'Parking Center';
            Editable = false;
            FieldClass = FlowField;
        }
        field(60; "Location ID"; Integer)
        {
            CalcFormula = lookup("DHT Parking Center"."Location ID" where(ID =
field("Parking Center ID"))));
            Caption = 'Location';
            Editable = false;
            FieldClass = FlowField;
        }
    }
}

```

```

        field(70; "Location Name"; Text[100])
        {
            Caption = 'Location Name';
            Editable = false;
            TableRelation = "DHT Parking Center"."Location Name" where(ID = field("Parking
Center ID"));
        }
    }

keys
{
    key(PK; "Series Number")
    {
        Clustered = true;
    }
}

fieldgroups
{
    fieldgroup(DropDown; Name, Type, "Parking Center ID")
    {
    }
}

var
    ClosedOrdersExistErr: Label 'Related closed orders exists, current record can't be
deleted';

trigger OnDelete()
begin
    DeleteRelatedOpenOrdersWithConfirmation();
end;

procedure DeleteRelatedOpenOrdersWithConfirmation()
var
    DHTVehicleMgt: Codeunit "DHT Vehicle Mgt";
    RelatedRecordExistMsg: Label 'All related Orders will be deleted. Do you want to
continue?';
begin
    if DHTVehicleMgt.RelatedClosedOrdersExists("Series Number") then
        Error(ClosedOrdersExistErr);

    if DHTVehicleMgt.RelatedOpenOrdersExists("Series Number") then
        if GuiAllowed() then
            if Confirm(RelatedRecordExistMsg, false) then
                DHTVehicleMgt.DeleteRelatedOrders("Series Number")
            else
                Error('');
end;

procedure DeleteRelatedOpenOrders()
var
    DHTVehicleMgt: Codeunit "DHT Vehicle Mgt";
begin
    if DHTVehicleMgt.RelatedClosedOrdersExists("Series Number") then
        Error(ClosedOrdersExistErr);

    DHTVehicleMgt.DeleteRelatedOrders("Series Number");
end;

local procedure DefineLocationName()
var
    DHTLocation: Record "DHT Location";
begin
    Rec.CalcFields("Location ID");
    if DHTLocation.Get("Location ID") then
        Validate("Location Name", DHTLocation.Name);
end;

```

```

local procedure CheckVechicleType()
var
    DHTVechicleMgt: Codeunit "DHT Vechicle Mgt";
begin
    DHTVechicleMgt.CheckVechicleType(Rec);
end;
}

page 59805 "DHT Vechicle"
{
    ApplicationArea = All;
    Caption = 'Vechicle';
    PageType = Card;
    SourceTable = "DHT Vechicle";
    UsageCategory = Administration;

    layout
    {
        area(Content)
        {
            group(GroupName)
            {
                field("Series Number"; Rec."Series Number")
                {
                    ApplicationArea = All;
                }
                field(Name; Rec.Name)
                {
                    ApplicationArea = All;
                }
                field(Type; Rec.Type)
                {
                    ApplicationArea = All;
                }
                field(ParkingCenter; Rec."Parking Center ID")
                {
                    ApplicationArea = All;
                }
                field(ParkingCenterName; Rec."Parking Center Name")
                {
                    ApplicationArea = All;
                }
                field(LocationName; Rec."Location Name")
                {
                    ApplicationArea = All;
                }
            }
        }
    }
}

page 59806 "DHT Vechicles"
{
    AccessByPermission = page "DHT Vechicles" = X;
    AdditionalSearchTerms = 'Vechicles,cars,trucks,microbuses';
    ApplicationArea = All;
    CardPageId = "DHT Vechicle";
    PageType = List;
    SourceTable = "DHT Vechicle";
    UsageCategory = Administration;

    layout
    {
        area(Content)
        {
            repeater(control)
            {
                field("Series Number"; Rec."Series Number")
            }
        }
    }
}

```

```

    {
        ApplicationArea = All;
    }
    field(Name; Rec.Name)
    {
        ApplicationArea = All;
    }
    field(Type; Rec.Type)
    {
        ApplicationArea = All;
    }
    field(ParkingCenter; Rec."Parking Center ID")
    {
        ApplicationArea = All;
    }
    field(ParkingCenterName; Rec."Parking Center Name")
    {
        ApplicationArea = All;
    }
    field(LocationName; Rec."Location Name")
    {
        ApplicationArea = All;
    }
    }
}
}
actions
{
    area(Processing)
    {
        action(ShowResult)
        {
            ApplicationArea = All;
            Caption = 'Show Number of Available Vechicles';
            Promoted = true;
            PromotedCategory = Process;
            PromotedIsBig = true;
            PromotedOnly = true;

            trigger OnAction()
            var
                DHTVechicle: Record "DHT Vechicle";
                DHTOrder: Record "DHT Order";
                Counter: Integer;
            begin
                If DHTVechicle.FindSet() then
                    repeat
                        DHTOrder.SetRange("Vechicle", DHTVechicle."Series Number",
DHTVechicle."Series Number");
                        DHTOrder.SetFilter("Renting Date", '<=%1', WorkDate());
                        DHTOrder.SetFilter("Returning Date", '>=%1', WorkDate());
                        DHTOrder.SetRange(Closed, false);
                        if DHTOrder.IsEmpty() then
                            Counter += 1;
                        until DHTVechicle.Next() = 0;

                        Message('Number of Available Vechicles on Current Date: %1', Counter);
                    end;
                }
            }
        }
    }
}

procedure GetSelectionFilter() Result: Text //TODO: Remove if not needed
var
    DHTVechicle: Record "DHT Vechicle";
    SelectionFilterManagement: Codeunit SelectionFilterManagement;
    RecordRef: RecordRef;
begin
    CurrPage.SetSelectionFilter(DHTVechicle);

```

```

        RecordRef.GetTable(DHTVehicle);
        Result := SelectionFilterManagement.GetSelectionFilter(RecordRef,
DHTVehicle.FieldNo("Series Number"));
    end;
}

enum 59800 "DHT Vehicle Type"
{
    Caption = 'Vehicle Type';

    value(10; Car)
    {
        Caption = 'Car';
    }
    value(20; Microbus)
    {
        Caption = 'Microbus';
    }
    value(30; Truck)
    {
        Caption = 'Truck';
    }
}

pageextension 59800 RoleCenterPageExt extends "Business Manager Role Center"
{
    actions
    {
        addlast(sections)
        {
            group("Lists")
            {
                action("Locations")
                {
                    ApplicationArea = All;
                    Caption = 'Locations';
                    RunObject = page "DHT Locations";
                }
                action("Parking Centers")
                {
                    ApplicationArea = All;
                    Caption = 'Parking Centers';
                    RunObject = page "DHT Parking Centers";
                }
                action("Vehicles")
                {
                    ApplicationArea = All;
                    Caption = 'Vehicles';
                    RunObject = page "DHT Vehicles";
                }
                action("Orders")
                {
                    ApplicationArea = All;
                    Caption = 'Orders';
                    RunObject = page "DHT Orders";
                }
                action(ClosedOrders)
                {
                    ApplicationArea = All;
                    Caption = 'Closed Orders';
                    RunObject = page "DHT Closed Orders";
                }
                action(PriceList)
                {
                    ApplicationArea = All;
                    Caption = 'Price List';
                    RunObject = page "DHT Price List";
                }
            }
        }
        group(Reporting)
    }
}

```

```

    {
        action(CustomerOrders)
        {
            ApplicationArea = All;
            Caption = 'Customer Orders';
            RunObject = page "DHT Customer Orders";
        }
        action(VehicleOnLocation)
        {
            ApplicationArea = All;
            Caption = 'Vehicle On Location';
            RunObject = page "DHT Vehicle On Location";
        }
        action(VehiclesInUse)
        {
            ApplicationArea = All;
            Caption = 'Vehicles In Use';
            RunObject = page "DHT Vehicles In Use";
        }
        action(PrintOrder)
        {
            ApplicationArea = All;
            Caption = 'Print Order';
            RunObject = report "DHT Order";
        }
    }
}
table 59801 "DHT Order"
{
    Caption = 'Order';
    DrillDownPageId = "DHT Order";

    fields
    {
        field(10; ID; Integer)
        {
            AutoIncrement = true;
            Caption = 'ID';
            Editable = false;
            MinValue = 0;
        }
        field(20; "Customer No."; Code[20])
        {
            Caption = 'Customer No.';
            TableRelation = Customer."No.";

            trigger OnValidate()
            begin
                CalcFields("Customer Name");
            end;
        }
        field(30; "Customer Name"; Text[100])
        {
            CalcFormula = lookup(Customer.Name where("No." = field("Customer No.")));
            Caption = 'Customer Name';
            Editable = false;
            FieldClass = FlowField;
        }
        field(40; "Vehicle Series Number"; Code[20])
        {
            Caption = 'Vehicle Series Number';
            TableRelation = "DHT Vehicle"."Series Number" where("Parking Center ID" =
field("Parking Center ID"));

            trigger OnValidate()
            begin

```

```

        CalcFields("Vechicle Name", "Vechicle Type");
    end;
}
field(50; "Vechicle Name"; Text[100])
{
    CalcFormula = lookup("DHT Vechicle".Name where("Series Number" =
field("Vechicle Series Number")));
    Caption = 'Vechicle Name';
    Editable = false;
    FieldClass = FlowField;
}
field(60; "Vechicle Type"; Enum "DHT Vechicle Type")
{
    CalcFormula = lookup("DHT Vechicle".Type where("Series Number" =
field("Vechicle Series Number")));
    Caption = 'Vechicle Type';
    Editable = false;
    FieldClass = FlowField;
}
field(70; "Parking Center ID"; Integer)
{
    Caption = 'Parking Center ID';
    TableRelation = "DHT Parking Center".ID where("Location ID" = field("Location
ID"));

    trigger OnValidate()
    begin
        CalcFields("Parking Center Name");
    end;
}
field(80; "Parking Center Name"; Text[100])
{
    CalcFormula = lookup("DHT Parking Center".Name where(ID = field("Parking Center
ID")));
    Caption = 'Parking Center Name';
    Editable = false;
    FieldClass = FlowField;
}
field(90; "Location ID"; Integer)
{
    Caption = 'Location ID';
    TableRelation = "DHT Location".ID;

    trigger OnValidate()
    begin
        CalcFields("Location Name");
    end;
}
field(100; "Location Name"; Text[100])
{
    CalcFormula = lookup("DHT Location".Name where(ID = field("Location ID")));
    Caption = 'Location Name';
    Editable = false;
    FieldClass = FlowField;
}
field(110; "Renting Date"; Date)
{
    Caption = 'Renting Date';
}
field(120; "Returning Date"; Date)
{
    Caption = 'Returning Date';
}
field(130; Amount; Decimal)
{
    Caption = 'Amount';
    Editable = false;
}
field(140; Closed; Boolean)

```



```

        {
            Caption = 'Closed';
        }
    }

    keys
    {
        key(PK; ID)
        {
            Clustered = true;
        }
    }
}

page 59809 "DHT Order"
{
    Caption = 'Order';
    PageType = Card;
    SourceTable = "DHT Order";

    layout
    {
        area(Content)
        {
            group(Customer)
            {
                Caption = 'Customer';

                field("Customer No."; Rec."Customer No.")
                {
                    ApplicationArea = All;
                    ShowMandatory = true;

                    trigger OnValidate()
                    begin
                        if Rec."Customer No." <> xRec."Customer No." then
                            UpdateVisibility();
                    end;
                }
                field("Customer Name"; Rec."Customer Name")
                {
                    ApplicationArea = All;
                }
            }
            group(Location)
            {
                Caption = 'Location';

                field("Location ID"; Rec."Location ID")
                {
                    ApplicationArea = All;
                    ShowMandatory = true;

                    trigger OnValidate()
                    begin
                        if Rec."Location ID" <> xRec."Location ID" then begin
                            Rec.Validate("Parking Center ID", 0);
                            Rec.Validate("Vechicle Series Number", '');
                            UpdateVisibility();
                        end;
                    end;
                }
                field("Location Name"; Rec."Location Name")
                {
                    ApplicationArea = All;
                }
            }
            group(ParkingCenter)
            {

```

```

Caption = 'Parking Center';
field("Parking Center ID"; Rec."Parking Center ID")
{
  ApplicationArea = All;
  ShowMandatory = true;

  trigger OnValidate()
  begin
    if Rec."Parking Center ID" <> xRec."Parking Center ID" then begin
      Rec.Validate("Vehicle Series Number", '');
      UpdateVisibility();
    end;
  end;
}
field("Parking Center Name"; Rec."Parking Center Name")
{
  ApplicationArea = All;
}
}
group(Vehicle)
{
  Caption = 'Vehicle';

  field("Vehicle Series Number"; Rec."Vehicle Series Number")
  {
    ApplicationArea = All;

    trigger OnValidate()
    begin
      if Rec."Vehicle Series Number" <> xRec."Vehicle Series Number"
then
          CheckVehicleInUse();
          Rec.Validate(Amount, GetAmount());
        end;
    }
  field("Vehicle Name"; Rec."Vehicle Name")
  {
    ApplicationArea = All;
  }
  field("Vehicle Type"; Rec."Vehicle Type")
  {
    ApplicationArea = All;
  }
}
group(Dates)
{
  Caption = 'Dates';

  field("Renting Date"; Rec."Renting Date")
  {
    ApplicationArea = All;
    ShowMandatory = true;

    trigger OnValidate()
    begin
      if Rec."Renting Date" <> xRec."Renting Date" then begin
        ValidateDate();
        CheckVehicleInUse();
        Rec.Validate(Amount, GetAmount());
        UpdateVisibility();
      end;
    end;
  }
  field("Returning Date"; Rec."Returning Date")
  {
    ApplicationArea = All;
    ShowMandatory = true;

    trigger OnValidate()

```

```

begin
    if Rec."Returning Date" <> xRec."Returning Date" then begin
        ValidateDate();
        CheckVechicleInUse();
        Rec.Validate(Amount, GetAmount());
        UpdateVisibility();
    end;
end;
}
}
group(AmountToPay)
{
    Caption = 'Amount To Pay';

    field(Amount; Rec.Amount)
    {
        ApplicationArea = All;
    }
}
}
}

actions
{
    area(Processing)
    {
        action(Close)
        {
            ApplicationArea = All;
            Caption = 'Close Order';
            Promoted = true;
            PromotedCategory = Process;
            PromotedIsBig = true;
            PromotedOnly = true;
            Visible = MandatoryFieldsPopulated;

            trigger OnAction()
            begin
                CloseOrder();
            end;
        }
    }
}

var
    MandatoryFieldsPopulated: Boolean;

trigger OnOpenPage()
begin
    UpdateVisibility();
    CurrPage.Editable(not Rec.Closed);
end;

trigger OnAfterGetRecord()
begin
    UpdateVisibility();
end;

local procedure UpdateVisibility()
begin
    MandatoryFieldsPopulated := (Rec."Customer No." <> '') and (Rec."Location ID" <> 0)
and (Rec."Vechicle Series Number" <> '') and (Rec."Parking Center ID" <> 0) and (Rec."Renting Date"
<> 0D) and (Rec."Returning Date" <> 0D) and (not Rec.Closed);
end;

local procedure ValidateDate()
var
    DHTOrderMgt: Codeunit "DHT Order Mgt";

```

```

begin
    DHTOrderMgt.ValidateDate(Rec);
end;

local procedure GetAmount() Result: Decimal
var
    DHTOrderMgt: Codeunit "DHT Order Mgt";
begin
    Result := DHTOrderMgt.GetAmount(Rec);
end;

local procedure CloseOrder()
var
    DHTOrderMgt: Codeunit "DHT Order Mgt";
begin
    DHTOrderMgt.CloseOrder(Rec);
    CurrPage.Close();
end;

local procedure CheckVechicleInUse()
var
    DHTOrderMgt: Codeunit "DHT Order Mgt";
begin
    DHTOrderMgt.CheckVechicleInUse(Rec);
end;
}

page 59811 "DHT Closed Orders"
{
    ApplicationArea = All;
    CardPageId = "DHT Order";
    DeleteAllowed = false;
    Editable = false;
    InsertAllowed = false;
    PageType = List;
    SourceTable = "DHT Order";
    SourceTableView = where(Closed = const(true));
    UsageCategory = Lists;

    layout
    {
        area(Content)
        {
            repeater(Group)
            {
                field(ID; Rec.ID)
                {
                    ApplicationArea = All;
                }
                field("Customer Name"; Rec."Customer Name")
                {
                    ApplicationArea = All;
                }
                field("Location Name"; Rec."Location Name")
                {
                    ApplicationArea = All;
                }
                field("Parking Center Name"; Rec."Parking Center Name")
                {
                    ApplicationArea = All;
                }
                field("Renting Date"; Rec."Renting Date")
                {
                    ApplicationArea = All;
                }
                field("Returning Date"; Rec."Returning Date")
                {
                    ApplicationArea = All;
                }
            }
        }
    }
}

```

```

        field(Amount; Rec.Amount)
        {
            ApplicationArea = All;
        }
    }
}

procedure GetSelectionFilter() Result: Text
var
    DHTOrder: Record "DHT Order";
    SelectionFilterManagement: Codeunit SelectionFilterManagement;
    RecordRef: RecordRef;
begin
    CurrPage.SetSelectionFilter(DHTOrder);
    RecordRef.GetTable(DHTOrder);
    Result := SelectionFilterManagement.GetSelectionFilter(RecordRef,
DHTOrder.FieldNo("ID"));
end;
}
codeunit 59800 "DHT Order Mgt"
{
    procedure ValidateDate(DHTOrder: Record "DHT Order")
    var
        IncorrectDateErr: Label 'Renting Date is bigger then Returning Date. Please, define
date in correct format to proceed.';
        WrongDateErr: Label 'Sorry it is impossible to rent or return a car on the last day
of month due to internal policy. Please, choose another day.';
    begin
        if (DHTOrder."Renting Date" > DHTOrder."Returning Date") and (DHTOrder."Returning
Date" <> 0D) then
            Error(IncorrectDateErr);

        if DHTOrder."Renting Date" <> 0D then
            if CalcDate('<CM>', DHTOrder."Renting Date") = DHTOrder."Renting Date" then
                Error(WrongDateErr);

        if DHTOrder."Returning Date" <> 0D then
            if CalcDate('<CM>', DHTOrder."Returning Date") = DHTOrder."Returning Date" then
                Error(WrongDateErr);
    end;

    procedure GetAmount(DHTOrder: Record "DHT Order") Result: Decimal
    var
        DHTPrice: Record "DHT Price";
        UndefinedTypePriceErr: Label 'Please, define renting price for selected type on
PriceList page.';
    begin
        if (DHTOrder."Renting Date" = 0D) or (DHTOrder."Returning Date" = 0D) or
(DHTOrder."Vechicle Series Number" = '') then
            exit;

        DHTOrder.CalcFields("Vechicle Type");
        if not DHTPrice.Get(DHTOrder."Vechicle Type") then
            Error(UndefinedTypePriceErr);

        if DHTOrder."Renting Date" = DHTOrder."Returning Date" then
            Result := DHTPrice.Price
        else
            Result := (DHTOrder."Returning Date" - DHTOrder."Renting Date") *
DHTPrice.Price;
    end;

    procedure CloseOrder(DHTOrder: Record "DHT Order")
    begin
        DHTOrder.Validate(Closed, true);
        DHTOrder.Modify(true);
    end;
}

```

```

procedure CheckVechicleInUse(DHTOrder: Record "DHT Order")
var
  WorkingDHTOrder: Record "DHT Order";
  DHTVechicleMgt: Codeunit "DHT Vechicle Mgt";
  VechicleInUseErr: Label 'Current Vechicle is in use on specified date, please choose
other vechicle or dates';
begin
  if (DHTOrder."Renting Date" = 0D) or (DHTOrder."Returning Date" = 0D) or
(DHTOrder."Vechicle Series Number" = '') then
    exit;

    DHTVechicleMgt.SetInUseFiltersOnOrder(WorkingDHTOrder, DHTOrder."Vechicle Series
Number", DHTOrder."Renting Date", DHTOrder."Returning Date");
  if not WorkingDHTOrder.IsEmpty() then
    Error(VechicleInUseErr);
end;
}
table 59806 "DHT Customer Order Temp"
{
  Caption = 'Order';
  TableType = Temporary;

  fields
  {
    field(10; ID; Integer)
    {
      AutoIncrement = true;
      Caption = 'ID';
      Editable = false;
      MinValue = 0;
    }
    field(20; "Vechicle Series Number"; Code[20])
    {
      Caption = 'Vechicle Series Number';
      Editable = false;
      TableRelation = "DHT Vechicle"."Series Number" where("Parking Center ID" =
field("Parking Center ID"));

      trigger OnValidate()
      begin
        CalcFields("Vechicle Name", "Vechicle Type");
      end;
    }
    field(30; "Vechicle Name"; Text[100])
    {
      Caption = 'Vechicle Name';
      Editable = false;
    }
    field(40; "Vechicle Type"; Enum "DHT Vechicle Type")
    {
      Caption = 'Vechicle Type';
      Editable = false;
    }
    field(50; "Parking Center ID"; Integer)
    {
      Caption = 'Parking Center ID';
      Editable = false;
      TableRelation = "DHT Parking Center".ID where("Location ID" = field("Location
ID"));
    }
    field(60; "Parking Center Name"; Text[100])
    {
      Caption = 'Parking Center Name';
      Editable = false;
    }
    field(70; "Location ID"; Integer)
    {
      Caption = 'Location ID';

```

```

        Editable = false;
        TableRelation = "DHT Location".ID;
    }
    field(80; "Location Name"; Text[100])
    {
        Caption = 'Location Name';
        Editable = false;
    }
    field(90; "Renting Date"; Date)
    {
        Caption = 'Renting Date';
        Editable = false;
    }
    field(100; "Returning Date"; Date)
    {
        Caption = 'Returning Date';
        Editable = false;
    }
    field(110; Amount; Decimal)
    {
        Caption = 'Amount';
        Editable = false;
    }
}

keys
{
    key(PK; ID)
    {
        Clustered = true;
    }
}
}

page 59812 "DHT Customer Orders"
{
    DeleteAllowed = false;
    InsertAllowed = false;
    PageType = Worksheet;
    SourceTable = "DHT Customer Order Temp";
    SourceTableTemporary = true;

    layout
    {
        area(Content)
        {
            group(Filters)
            {
                Caption = 'Filters';

                field("Customer No."; CustomerNoFilter)
                {
                    ApplicationArea = All;
                    Caption = 'Customer No.';

                    trigger OnValidate()
                    begin
                        ResetView();
                        UpdateActionVisible();
                    end;

                    trigger OnLookup(var Text: Text): Boolean
                    var
                        CustomerList: Page "Customer List";
                    begin
                        CustomerList.LookupMode(true);
                        if CustomerList.RunModal() = Action::LookupOK then begin
                            Text := CustomerList.GetSelectionFilter();
                        end;
                    end;
                }
            }
        }
    }
}

```

```

        exit(true);
    end;
end;
}
field(Closed; ClosedFilter)
{
    ApplicationArea = All;
    Caption = 'Closed';

    trigger OnValidate()
    begin
        ResetView();
        UpdateActionVisible();
    end;
}
}
repeater(GroupName)
{
    Visible = MatrixVisible;

    field(ID; Rec.ID)
    {
        ApplicationArea = All;
    }
    field("Location Name"; Rec."Location Name")
    {
        ApplicationArea = All;
    }
    field("Parking Center Name"; Rec."Parking Center Name")
    {
        ApplicationArea = All;
    }
    field("Vehicle Series Number"; Rec."Vehicle Series Number")
    {
        ApplicationArea = All;
    }
    field("Vehicle Name"; Rec."Vehicle Name")
    {
        ApplicationArea = All;
    }
    field("Vehicle Type"; Rec."Vehicle Type")
    {
        ApplicationArea = All;
    }
    field("Renting Date"; Rec."Renting Date")
    {
        ApplicationArea = All;
    }
    field("Returning Date"; Rec."Returning Date")
    {
        ApplicationArea = All;
    }
    field(Amount; Rec.Amount)
    {
        ApplicationArea = All;
    }
}
}
}
actions
{
    area(Processing)
    {
        action>ShowResult)
        {
            ApplicationArea = All;
            Caption = 'Show Result';
            Promoted = true;
        }
    }
}

```



```

        PromotedCategory = Process;
        PromotedIsBig = true;
        PromotedOnly = true;
        Visible = ShowResultVisible;

        trigger OnAction()
        begin
            InitView();
        end;
    }
}

var
    CustomerNoFilter: Text;
    ClosedFilter: Boolean;
    ShowResultVisible, MatrixVisible : Boolean;

trigger OnOpenPage()
begin
    UpdateActionVisible();
end;

trigger OnAfterGetRecord()
begin
    UpdateActionVisible();
end;

local procedure InitView()
begin
    ClearMatrix();
    BuildMartix();
    MatrixVisible := true;
end;

local procedure ResetView()
begin
    ClearMatrix();
    MatrixVisible := false;
end;

local procedure ClearMatrix()
begin
    Rec.Reset();
    Rec.DeleteAll(true);
end;

local procedure BuildMartix()
var
    DHTOrder: Record "DHT Order";
begin
    DHTOrder.SetFilter("Customer No.", CustomerNoFilter);
    DHTOrder.SetRange(Closed, ClosedFilter);
    DHTOrder.SetAutoCalcFields("Vechicle Name", "Vechicle Type", "Parking Center Name",
"Location Name");
    if DHTOrder.FindSet() then
        repeat
            Rec.Init();
            Rec.ID := DHTOrder.ID;
            Rec."Vechicle Series Number" := DHTOrder."Vechicle Series Number";
            Rec."Vechicle Name" := DHTOrder."Vechicle Name";
            Rec."Vechicle Type" := DHTOrder."Vechicle Type";
            Rec."Parking Center ID" := DHTOrder."Parking Center ID";
            Rec."Parking Center Name" := DHTOrder."Parking Center Name";
            Rec."Location ID" := DHTOrder."Location ID";
            Rec."Location Name" := DHTOrder."Location Name";
            Rec."Renting Date" := DHTOrder."Renting Date";
            Rec."Returning Date" := DHTOrder."Returning Date";
            Rec.Amount := DHTOrder.Amount;
        until DHTOrder.NoMoreRecords;
    end;
end;

```

```

        Rec.Insert();
        until DHTOrder.Next() = 0;
    end;

    local procedure UpdateActionVisible()
    begin
        ShowResultVisible := (CustomerNoFilter <> '') and (Rec.IsEmpty());
    end;
}

page 59813 "DHT Vechicle On Location"
{
    ApplicationArea = All;
    DeleteAllowed = false;
    InsertAllowed = false;
    PageType = Worksheet;
    SourceTable = "DHT Vechicle On Location Temp";
    SourceTableTemporary = true;
    UsageCategory = Administration;

    layout
    {
        area(Content)
        {
            group(Filter)
            {
                Caption = 'Filters';

                field(LocationIDFilter; LocationIDFilter)
                {
                    ApplicationArea = All;
                    Caption = 'Location';
                    ShowMandatory = true;

                    trigger OnValidate()
                    begin
                        ResetView();
                        UpdateActionVisible();
                        Clear(ParkingCenterFilter);
                    end;

                    trigger OnLookup(var Text: Text): Boolean
                    var
                        DHTLocations: Page "DHT Locations";
                    begin
                        DHTLocations.LookupMode(true);
                        if DHTLocations.RunModal() = Action::LookupOK then begin
                            Text := DHTLocations.GetSelectionFilter();
                            exit(true);
                        end;
                    end;
                }

                field(ParkingCenterIDFilter; ParkingCenterFilter)
                {
                    ApplicationArea = All;
                    Caption = 'Parking Center';

                    trigger OnValidate()
                    begin
                        ResetView();
                        UpdateActionVisible();
                    end;

                    trigger OnLookup(var Text: Text): Boolean
                    var
                        DHTParkingCenter: Record "DHT Parking Center";
                        DHTParkingCenters: Page "DHT Parking Centers";

```

```

begin
    DHTParkingCenter.SetFilter("Location ID", LocationIDFilter);
    DHTParkingCenters.SetTableView(DHTParkingCenter);
    DHTParkingCenters.SetRecord(DHTParkingCenter);
    DHTParkingCenters.LookupMode(true);
    if DHTParkingCenters.RunModal() = Action::LookupOK then begin
        Text := DHTParkingCenters.GetSelectionFilter();
        exit(true);
    end;
end;
}

field(VechicleTypFilter; VechicleType)
{
    ApplicationArea = All;
    Caption = 'Vechicle Type';

    trigger OnValidate()
    begin
        ResetView();
        UpdateActionVisible();
    end;
}
}
repeater(GroupName)
{
    Visible = MatrixVisible;

    field("Parking Center Name"; Rec."Parking Center Name")
    {
        ApplicationArea = All;
    }
    field("Vechicle Series Number"; Rec."Vechicle Series Number")
    {
        ApplicationArea = All;
    }
    field("Vechicle Name"; Rec."Vechicle Name")
    {
        ApplicationArea = All;
    }
    field("Vechicle Type"; Rec."Vechicle Type")
    {
        ApplicationArea = All;
    }
}
}
}

actions
{
    area(Processing)
    {
        action>ShowResult)
        {
            ApplicationArea = All;
            Caption = 'Show Result';
            Promoted = true;
            PromotedCategory = Process;
            PromotedIsBig = true;
            PromotedOnly = true;
            Visible = ShowResultVisible;

            trigger OnAction()
            begin
                InitView();
            end;
        }
    }
}
}

```

```

var
    LocationIDFilter, ParkingCenterFilter : Text;
    VechicleType: Enum "DHT Vechicle Type";
    ShowResultVisible, MatrixVisible : Boolean;

trigger OnOpenPage()
begin
    UpdateActionVisible();
end;

trigger OnAfterGetRecord()
begin
    UpdateActionVisible();
end;

local procedure InitView()
begin
    ClearMatrix();
    BuildMartix();
    MatrixVisible := true;
end;

local procedure ResetView()
begin
    ClearMatrix();
    MatrixVisible := false;
end;

local procedure ClearMatrix()
begin
    Rec.Reset();
    Rec.DeleteAll(true);
end;

local procedure BuildMartix()
var
    DHTVechicle: Record "DHT Vechicle";
begin
    DHTVechicle.SetFilter("Location ID", LocationIDFilter);
    DHTVechicle.SetFilter("Parking Center ID", ParkingCenterFilter);
    DHTVechicle.SetRange(Type, VechicleType);
    DHTVechicle.SetAutoCalcFields("Parking Center Name");
    if DHTVechicle.FindSet() then
        repeat
            Rec.Init();
            Rec.ID += 1;
            Rec."Parking Center ID" := DHTVechicle."Parking Center ID";
            Rec."Parking Center Name" := DHTVechicle."Parking Center Name";
            Rec."Vechicle Series Number" := DHTVechicle."Series Number";
            Rec."Vechicle Name" := DHTVechicle.Name;
            Rec."Vechicle Type" := DHTVechicle.Type;
            Rec.Insert(true);
        until DHTVechicle.Next() = 0;
    end;

    local procedure UpdateActionVisible()
    begin
        ShowResultVisible := (LocationIDFilter <> '') and (Rec.IsEmpty());
    end;
}
table 59807 "DHT Vechicle On Location Temp"
{
    Caption = 'Vechicle On Location';
    TableType = Temporary;

    fields
    {
        field(10; ID; Integer)
    }
}

```

```

    {
        AutoIncrement = true;
        Caption = 'ID';
        Editable = false;
    }
    field(20; "Parking Center ID"; Integer)
    {
        Caption = 'Parking Center ID';
        Editable = false;
        TableRelation = "DHT Parking Center".ID;
    }
    field(30; "Parking Center Name"; Text[100])
    {
        Caption = 'Parking Center Name';
        Editable = false;
    }
    field(40; "Vechicle Series Number"; Code[20])
    {
        Caption = 'Vechicle Series Number';
        Editable = false;
    }
    field(50; "Vechicle Name"; Text[100])
    {
        Caption = 'Vechicle Name';
        Editable = false;
    }
    field(60; "Vechicle Type"; Enum "DHT Vechicle Type")
    {
        Caption = 'Vechicle Series Number';
        Editable = false;
    }
}

keys
{
    key(PK; ID)
    {
        Clustered = true;
    }
}
}
page 59814 "DHT Vechicles In Use"
{
    ApplicationArea = All;
    PageType = Card;
    SourceTable = "DHT Vechicle";
    SourceTableTemporary = true;
    UsageCategory = Administration;

    layout
    {
        area(Content)
        {
            group(Filter)
            {
                Caption = 'Filters';

                field(RentingDateFilter; RentingDate)
                {
                    ApplicationArea = All;
                    Caption = 'Renting Date';
                    ShowMandatory = true;

                    trigger OnValidate()
                    begin
                        ResetView();
                        UpdateActionVisible();
                    end;
                }
            }
        }
    }
}

```

```

    field(ReturningDateFilter; ReturningDate)
    {
        ApplicationArea = All;
        Caption = 'Returning Date';
        ShowMandatory = true;

        trigger OnValidate()
        begin
            ResetView();
            UpdateActionVisible();
        end;
    }
}
repeater(GroupName)
{
    Visible = MatrixVisible;

    field("Series Number"; Rec."Series Number")
    {
        ApplicationArea = All;
    }
    field(Name; Rec.Name)
    {
        ApplicationArea = All;
    }
    field("Type"; Rec."Type")
    {
        ApplicationArea = All;
    }
    field("Parking Center Name"; Rec."Parking Center Name")
    {
        ApplicationArea = All;
    }
    field("Location Name"; Rec."Location Name")
    {
        ApplicationArea = All;
    }
}
}
}

actions
{
    area(Processing)
    {
        action>ShowResult)
        {
            ApplicationArea = All;
            Caption = 'Show Result';
            Promoted = true;
            PromotedCategory = Process;
            PromotedIsBig = true;
            PromotedOnly = true;
            Visible = ShowResultVisible;

            trigger OnAction()
            begin
                InitView();
            end;
        }
    }
}

var
    RentingDate, ReturningDate : Date;
    ShowResultVisible, MatrixVisible : Boolean;

trigger OnOpenPage()
begin

```

```

        UpdateActionVisible();
    end;

    trigger OnAfterGetRecord()
    begin
        UpdateActionVisible();
    end;

    local procedure InitView()
    begin
        ClearMatrix();
        BuildMartix();
        MatrixVisible := true;
    end;

    local procedure ResetView()
    begin
        ClearMatrix();
        MatrixVisible := false;
    end;

    local procedure ClearMatrix()
    begin
        Rec.Reset();
        Rec.DeleteAll(true);
    end;

    local procedure BuildMartix()
    var
        DHTVehicleMgt: Codeunit "DHT Vechicle Mgt";
        DHTVehicle: Record "DHT Vechicle";
        DHTOrder: Record "DHT Order";
    begin
        DHTVehicle.SetAutoCalcFields("Parking Center Name", "Location ID");
        if DHTVehicle.FindSet() then
            repeat
                DHTVehicleMgt.SetInUseFiltersOnOrder(DHTOrder,          DHTVehicle."Series
Number", RentingDate, ReturningDate);
                if not DHTOrder.IsEmpty() then begin
                    Rec := DHTVehicle;
                    Rec.Insert();
                end;
            until DHTVehicle.Next() = 0;
        end;

        local procedure UpdateActionVisible()
        begin
            ShowResultVisible := (RentingDate <> 0D) and (ReturningDate <> 0D) and
(Rec.IsEmpty());
        end;
    }
    <?xml version="1.0" encoding="utf-8"?>
    <Report xmlns="http://schemas.microsoft.com/sqlserver/reporting/2016/01/reportdefinition"
xmlns:rd="http://schemas.microsoft.com/SQLServer/reporting/reportdesigner">
    <AutoRefresh>0</AutoRefresh>
    <DataSources>
    <DataSource Name="DataSource">
    <ConnectionProperties>
    <DataProvider>SQL</DataProvider>
    <ConnectionString />
    </ConnectionProperties>
    <rd:DataSourceID>4019a1ab-4786-4468-b625-fc28c5971fb5</rd:DataSourceID>
    </DataSource>
    </DataSources>
    <ReportSections>
    <ReportSection>
    <Body>
    <ReportItems>
    <Tablix Name="Tablix1">

```

```

<TableBody>
  <TableColumns>
    <TableColumn>
      <Width>8.255cm</Width>
    </TableColumn>
    <TableColumn>
      <Width>8.255cm</Width>
    </TableColumn>
  </TableColumns>
  <TableRows>
    <TableRow>
      <Height>0.24722cm</Height>
      <TableCells>
        <TableCell>
          <CellContents>
            <Textbox Name="Textbox27">
              <CanGrow>true</CanGrow>
              <KeepTogether>true</KeepTogether>
              <Paragraphs>
                <Paragraph>
                  <TextRuns>
                    <TextRun>
                      <Value />
                      <Style>
                        <FontWeight>Bold</FontWeight>
                      </Style>
                    </TextRun>
                  </TextRuns>
                  <Style>
                    <TextAlign>Center</TextAlign>
                  </Style>
                </Paragraph>
              </Paragraphs>
              <rd:DefaultName>Textbox27</rd:DefaultName>
              <Style>
                <Border>
                  <Style>None</Style>
                </Border>
                <BottomBorder>
                  <Style>Solid</Style>
                  <Width>0.5pt</Width>
                </BottomBorder>
                <VerticalAlign>Middle</VerticalAlign>
                <PaddingLeft>2pt</PaddingLeft>
                <PaddingRight>2pt</PaddingRight>
                <PaddingTop>2pt</PaddingTop>
                <PaddingBottom>2pt</PaddingBottom>
              </Style>
            </Textbox>
            <ColSpan>2</ColSpan>
          </CellContents>
        </TableCell>
        <TableCell />
      </TableCells>
    </TableRow>
    <TableRow>
      <Height>0.6cm</Height>
      <TableCells>
        <TableCell>
          <CellContents>
            <Textbox Name="Textbox5">
              <CanGrow>true</CanGrow>
              <KeepTogether>true</KeepTogether>
              <Paragraphs>
                <Paragraph>
                  <TextRuns>
                    <TextRun>
                      <Value>"Order No. : " + Fields!Order_ID.Value</Value>
                      <Style>

```



```

        <FontWeight>Bold</FontWeight>
    </Style>
</TextRun>
</TextRuns>
</TextRuns>
<Style>
    <TextAlign>Center</TextAlign>
</Style>
</Paragraph>
</Paragraphs>
<rd:DefaultName>Textbox5</rd:DefaultName>
<ZIndex>4</ZIndex>
<Style>
    <Border>
        <Style>None</Style>
    </Border>
    <BottomBorder>
        <Style>Solid</Style>
        <Width>0.5pt</Width>
    </BottomBorder>
    <VerticalAlign>Middle</VerticalAlign>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
</Style>
</Textbox>
<ColSpan>2</ColSpan>
</CellContents>
</TablixCell>
<TablixCell />
</TablixCells>
</TablixRow>
<TablixRow>
    <Height>0.6cm</Height>
    <TablixCells>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox30">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value>Customer Name:</Value>
                                    <Style />
                                </TextRun>
                            </TextRuns>
                        </Paragraph>
                    </Paragraphs>
                    <rd:DefaultName>Textbox30</rd:DefaultName>
                    <Style>
                        <Border>
                            <Style>None</Style>
                        </Border>
                        <PaddingLeft>2pt</PaddingLeft>
                        <PaddingRight>2pt</PaddingRight>
                        <PaddingTop>2pt</PaddingTop>
                        <PaddingBottom>2pt</PaddingBottom>
                    </Style>
                </Textbox>
            </CellContents>
        </TablixCell>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox1">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>

```

```

<Paragraphs>
  <Paragraph>
    <TextRuns>
      <TextRun>
        <Value>=Fields!Order_CustomerName.Value</Value>
        <Style />
      </TextRun>
    </TextRuns>
  </Paragraph>
</Paragraphs>
<rd:DefaultName>Textbox1</rd:DefaultName>
<Style>
  <Border>
    <Style>None</Style>
  </Border>
  <PaddingLeft>2pt</PaddingLeft>
  <PaddingRight>2pt</PaddingRight>
  <PaddingTop>2pt</PaddingTop>
  <PaddingBottom>2pt</PaddingBottom>
</Style>
</Textbox>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
  <Height>0.6cm</Height>
  <TablixCells>
    <TablixCell>
      <CellContents>
        <Textbox Name="Textbox18">
          <CanGrow>true</CanGrow>
          <KeepTogether>true</KeepTogether>
          <Paragraphs>
            <Paragraph>
              <TextRuns>
                <TextRun>
                  <Value      EvaluationMode="Constant">Vehicle      Series
Number:</Value>
                  <Style />
                </TextRun>
              </TextRuns>
            </Paragraph>
          </Paragraphs>
          <rd:DefaultName>Textbox18</rd:DefaultName>
          <Style>
            <Border>
              <Style>None</Style>
            </Border>
            <PaddingLeft>2pt</PaddingLeft>
            <PaddingRight>2pt</PaddingRight>
            <PaddingTop>2pt</PaddingTop>
            <PaddingBottom>2pt</PaddingBottom>
          </Style>
        </Textbox>
      </CellContents>
    </TablixCell>
  </TablixCells>
  <CellContents>
    <Textbox Name="Textbox19">
      <CanGrow>true</CanGrow>
      <KeepTogether>true</KeepTogether>
      <Paragraphs>
        <Paragraph>
          <TextRuns>
            <TextRun>
              <Value>=Fields!Order_VehicleSeriesNumber.Value</Value>

```

```

        <Style />
    </TextRun>
</TextRuns>
</TextRuns>
    <Style />
</Paragraph>
</Paragraphs>
</Paragraphs>
<rd:DefaultName>Textbox19</rd:DefaultName>
<Style>
    <Border>
        <Style>None</Style>
    </Border>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
</Style>
</Textbox>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
    <Height>0.6cm</Height>
    <TablixCells>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox24">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value>Vechicle Type:</Value>
                                <Style />
                            </TextRun>
                        </TextRuns>
                    </Paragraph>
                </Paragraphs>
                <rd:DefaultName>Textbox24</rd:DefaultName>
                <Style>
                    <Border>
                        <Style>None</Style>
                    </Border>
                    <PaddingLeft>2pt</PaddingLeft>
                    <PaddingRight>2pt</PaddingRight>
                    <PaddingTop>2pt</PaddingTop>
                    <PaddingBottom>2pt</PaddingBottom>
                </Style>
            </Textbox>
        </CellContents>
    </TablixCell>
    <TablixCell>
        <CellContents>
            <Textbox Name="Textbox25">
                <CanGrow>true</CanGrow>
                <KeepTogether>true</KeepTogether>
                <Paragraphs>
                    <Paragraph>
                        <TextRuns>
                            <TextRun>
                                <Value>=Fields!Order_VechicleType.Value</Value>
                            <Style />
                        </TextRun>
                    </TextRuns>
                </Paragraph>
            </Paragraphs>
        </CellContents>
    </TablixCell>

```

```

        <rd:DefaultName>Textbox25</rd:DefaultName>
        <Style>
            <Border>
                <Style>None</Style>
            </Border>
            <PaddingLeft>2pt</PaddingLeft>
            <PaddingRight>2pt</PaddingRight>
            <PaddingTop>2pt</PaddingTop>
            <PaddingBottom>2pt</PaddingBottom>
        </Style>
    </Textbox>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
    <Height>0.6cm</Height>
    <TablixCells>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox21">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value>Vechicle Name:</Value>
                                    <Style />
                                </TextRun>
                            </TextRuns>
                        </Paragraph>
                    </Paragraphs>
                    <rd:DefaultName>Textbox21</rd:DefaultName>
                    <Style>
                        <Border>
                            <Style>None</Style>
                        </Border>
                        <PaddingLeft>2pt</PaddingLeft>
                        <PaddingRight>2pt</PaddingRight>
                        <PaddingTop>2pt</PaddingTop>
                        <PaddingBottom>2pt</PaddingBottom>
                    </Style>
                </Textbox>
            </CellContents>
        </TablixCell>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox22">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value>=Fields!Order_VechicleName.Value</Value>
                                    <Style />
                                </TextRun>
                            </TextRuns>
                        </Paragraph>
                    </Paragraphs>
                    <rd:DefaultName>Textbox22</rd:DefaultName>
                    <Style>
                        <Border>
                            <Style>None</Style>
                        </Border>
                        <PaddingLeft>2pt</PaddingLeft>

```

```

        <PaddingRight>2pt</PaddingRight>
        <PaddingTop>2pt</PaddingTop>
        <PaddingBottom>2pt</PaddingBottom>
    </Style>
</Textbox>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
<Height>0.6cm</Height>
<TablixCells>
<TablixCell>
<CellContents>
<Textbox Name="Textbox2">
<CanGrow>true</CanGrow>
<KeepTogether>true</KeepTogether>
<Paragraphs>
<Paragraph>
<TextRuns>
<TextRun>
<Value>Date of rent: </Value>
<Style />
</TextRun>
</TextRuns>
<Style />
</Paragraph>
</Paragraphs>
<rd:DefaultName>Textbox2</rd:DefaultName>
<ZIndex>1</ZIndex>
<Style>
<Border>
<Style>None</Style>
</Border>
<PaddingLeft>2pt</PaddingLeft>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
<PaddingBottom>2pt</PaddingBottom>
</Style>
</Textbox>
</CellContents>
</TablixCell>
<TablixCell>
<CellContents>
<Textbox Name="Textbox12">
<CanGrow>true</CanGrow>
<KeepTogether>true</KeepTogether>
<Paragraphs>
<Paragraph>
<TextRuns>
<TextRun>
<Value>=Fields!Order_RentingDate.Value</Value>
<Style />
</TextRun>
</TextRuns>
<Style>
<TextAlign>Left</TextAlign>
</Style>
</Paragraph>
</Paragraphs>
<rd:DefaultName>Textbox12</rd:DefaultName>
<Style>
<Border>
<Style>None</Style>
</Border>
<VerticalAlign>Middle</VerticalAlign>
<PaddingLeft>2pt</PaddingLeft>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>

```

```

        <PaddingBottom>2pt</PaddingBottom>
    </Style>
</Textbox>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
<Height>0.6cm</Height>
<TablixCells>
<TablixCell>
<CellContents>
<Textbox Name="Textbox3">
<CanGrow>true</CanGrow>
<KeepTogether>true</KeepTogether>
<Paragraphs>
<Paragraph>
<TextRuns>
<TextRun>
<Value EvaluationMode="Constant">Date of return:</Value>
<Style />
</TextRun>
</TextRuns>
<Style />
</Paragraph>
</Paragraphs>
<rd:DefaultName>Textbox3</rd:DefaultName>
<ZIndex>2</ZIndex>
<Style>
<Border>
<Style>None</Style>
</Border>
<PaddingLeft>2pt</PaddingLeft>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
<PaddingBottom>2pt</PaddingBottom>
</Style>
</Textbox>
</CellContents>
</TablixCell>
<TablixCell>
<CellContents>
<Textbox Name="Textbox14">
<CanGrow>true</CanGrow>
<KeepTogether>true</KeepTogether>
<Paragraphs>
<Paragraph>
<TextRuns>
<TextRun>
<Value>=Fields!Order_ReturningDate.Value</Value>
<Style />
</TextRun>
</TextRuns>
<Style>
<TextAlign>Left</TextAlign>
</Style>
</Paragraph>
</Paragraphs>
<rd:DefaultName>Textbox14</rd:DefaultName>
<Style>
<Border>
<Style>None</Style>
</Border>
<VerticalAlign>Middle</VerticalAlign>
<PaddingLeft>2pt</PaddingLeft>
<PaddingRight>2pt</PaddingRight>
<PaddingTop>2pt</PaddingTop>
<PaddingBottom>2pt</PaddingBottom>
</Style>

```

```

        </Textbox>
    </CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
    <Height>0.6cm</Height>
    <TablixCells>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox4">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value EvaluationMode="Constant">Amount to pay:</Value>
                                    <Style />
                                </TextRun>
                            </TextRuns>
                        </Paragraph>
                    </Paragraphs>
                    <rd:DefaultName>Textbox4</rd:DefaultName>
                    <ZIndex>3</ZIndex>
                    <Style>
                        <Border>
                            <Style>None</Style>
                        </Border>
                        <PaddingLeft>2pt</PaddingLeft>
                        <PaddingRight>2pt</PaddingRight>
                        <PaddingTop>2pt</PaddingTop>
                        <PaddingBottom>2pt</PaddingBottom>
                    </Style>
                </Textbox>
            </CellContents>
        </TablixCell>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox16">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value>=Fields!Order_Amount.Value</Value>
                                    <Style>
                                        <Format>#,0.00;(#,0.00)</Format>
                                    </Style>
                                </TextRun>
                            </TextRuns>
                        </Paragraph>
                    </Paragraphs>
                    <rd:DefaultName>Textbox16</rd:DefaultName>
                    <Style>
                        <Border>
                            <Style>None</Style>
                        </Border>
                        <VerticalAlign>Middle</VerticalAlign>
                        <PaddingLeft>2pt</PaddingLeft>
                        <PaddingRight>2pt</PaddingRight>
                        <PaddingTop>2pt</PaddingTop>
                        <PaddingBottom>2pt</PaddingBottom>
                    </Style>
                </Textbox>
            </CellContents>
        </TablixCell>
    </TablixCells>
</TablixRow>
</Table>

```

```

        </Textbox>
    </CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
    <Height>0.6cm</Height>
    <TablixCells>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox9">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value />
                                    <Style />
                                </TextRun>
                            </TextRuns>
                        </Paragraph>
                    </Paragraphs>
                    <rd:DefaultName>Textbox9</rd:DefaultName>
                    <Style>
                        <Border>
                            <Style>None</Style>
                        </Border>
                        <PaddingLeft>2pt</PaddingLeft>
                        <PaddingRight>2pt</PaddingRight>
                        <PaddingTop>2pt</PaddingTop>
                        <PaddingBottom>2pt</PaddingBottom>
                    </Style>
                </Textbox>
            </CellContents>
        </TablixCell>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox10">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value />
                                    <Style />
                                </TextRun>
                            </TextRuns>
                        </Paragraph>
                    </Paragraphs>
                    <rd:DefaultName>Textbox10</rd:DefaultName>
                    <Style>
                        <Border>
                            <Style>None</Style>
                        </Border>
                        <PaddingLeft>2pt</PaddingLeft>
                        <PaddingRight>2pt</PaddingRight>
                        <PaddingTop>2pt</PaddingTop>
                        <PaddingBottom>2pt</PaddingBottom>
                    </Style>
                </Textbox>
            </CellContents>
        </TablixCell>
    </TablixCells>
</TablixRow>
<TablixRow>

```



```

<Height>0.6cm</Height>
<TablixCells>
  <TablixCell>
    <CellContents>
      <Textbox Name="Textbox6">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Total amount to pay</Value>
                <Style>
                  <FontWeight>Bold</FontWeight>
                </Style>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
        <rd:DefaultName>Textbox6</rd:DefaultName>
        <Style>
          <Border>
            <Style>None</Style>
          </Border>
          <TopBorder>
            <Style>Solid</Style>
          </TopBorder>
          <BottomBorder>
            <Style>Solid</Style>
          </BottomBorder>
          <VerticalAlign>Middle</VerticalAlign>
          <PaddingLeft>2pt</PaddingLeft>
          <PaddingRight>2pt</PaddingRight>
          <PaddingTop>2pt</PaddingTop>
          <PaddingBottom>2pt</PaddingBottom>
        </Style>
      </Textbox>
    </CellContents>
  </TablixCell>
  <TablixCell>
    <CellContents>
      <Textbox Name="Textbox7">
        <CanGrow>true</CanGrow>
        <KeepTogether>true</KeepTogether>
        <Paragraphs>
          <Paragraph>
            <TextRuns>
              <TextRun>
                <Value>Sum(Fields!Order_Amount.Value,
"DataSet_Result")</Value>
                <Style>
                  <Format>#,0.00;(#,0.00)</Format>
                </Style>
              </TextRun>
            </TextRuns>
          </Paragraph>
        </Paragraphs>
        <rd:DefaultName>Textbox7</rd:DefaultName>
        <Style>
          <Border>
            <Style>None</Style>
          </Border>
          <TopBorder>

```

```

        <Style>Solid</Style>
    </TopBorder>
    <BottomBorder>
        <Style>Solid</Style>
    </BottomBorder>
    <VerticalAlign>Middle</VerticalAlign>
    <PaddingLeft>2pt</PaddingLeft>
    <PaddingRight>2pt</PaddingRight>
    <PaddingTop>2pt</PaddingTop>
    <PaddingBottom>2pt</PaddingBottom>
</Style>
</Textbox>
</CellContents>
</TablixCell>
</TablixCells>
</TablixRow>
<TablixRow>
    <Height>0.10611cm</Height>
    <TablixCells>
        <TablixCell>
            <CellContents>
                <Textbox Name="Textbox8">
                    <CanGrow>true</CanGrow>
                    <KeepTogether>true</KeepTogether>
                    <Paragraphs>
                        <Paragraph>
                            <TextRuns>
                                <TextRun>
                                    <Value />
                                    <Style>
                                        <FontWeight>Bold</FontWeight>
                                    </Style>
                                </TextRun>
                            </TextRuns>
                        <Style>
                            <TextAlign>Left</TextAlign>
                        </Style>
                    </Paragraph>
                </Paragraphs>
                <rd:DefaultName>Textbox8</rd:DefaultName>
                <Style>
                    <Border>
                        <Style>None</Style>
                    </Border>
                    <PaddingLeft>2pt</PaddingLeft>
                    <PaddingRight>2pt</PaddingRight>
                    <PaddingTop>2pt</PaddingTop>
                    <PaddingBottom>2pt</PaddingBottom>
                </Style>
            </Textbox>
            <ColSpan>2</ColSpan>
        </CellContents>
    </TablixCell>
    <TablixCell />
</TablixCells>
</TablixRow>
</TablixRows>
</TablixBody>
<TablixColumnHierarchy>
    <TablixMembers>
        <TablixMember />
        <TablixMember />
    </TablixMembers>
</TablixColumnHierarchy>
<TablixRowHierarchy>
    <TablixMembers>
        <TablixMember>
            <KeepWithGroup>After</KeepWithGroup>
        </TablixMember>
    </TablixMembers>
</TablixRowHierarchy>

```

```

    <TablixMember>
      <Group Name="Details" />
      <TablixMembers>
        <TablixMember />
        <TablixMember />
        <TablixMember />
        <TablixMember />
        <TablixMember />
        <TablixMember />
        <TablixMember />
        <TablixMember />
        <TablixMember />
      </TablixMembers>
    </TablixMember>
    <TablixMember>
      <KeepWithGroup>Before</KeepWithGroup>
    </TablixMember>
    <TablixMember>
      <KeepWithGroup>Before</KeepWithGroup>
    </TablixMember>
  </TablixMembers>
</TablixRowHierarchy>
<KeepTogether>true</KeepTogether>
<DataSetName>DataSet_Result</DataSetName>
<Top>0cm</Top>
<Left>0cm</Left>
<Height>6.35333cm</Height>
<Width>16.51cm</Width>
<Style>
  <Border>
    <Style>None</Style>
  </Border>
</Style>
</Tablix>
</ReportItems>
<Height>6.35333cm</Height>
<Style />
</Body>
<Width>16.51cm</Width>
<Page>
  <PageHeader>
    <Height>2.09903cm</Height>
    <PrintOnFirstPage>true</PrintOnFirstPage>
    <PrintOnLastPage>true</PrintOnLastPage>
    <ReportItems>
      <Image Name="Image1">
        <Source>Embedded</Source>
        <Value>photo5235727887770958803</Value>
        <Sizing>FitProportional</Sizing>
        <Top>0cm</Top>
        <Left>6.92009cm</Left>
        <Height>2.09903cm</Height>
        <Width>2.15264cm</Width>
        <Style>
          <Border>
            <Style>None</Style>
          </Border>
        </Style>
      </Image>
    </ReportItems>
  <Style>
    <Border>
      <Style>None</Style>
    </Border>
  </Style>
</PageHeader>
<PageHeight>29.7cm</PageHeight>
<PageWidth>21cm</PageWidth>
<LeftMargin>2cm</LeftMargin>

```

```

        <RightMargin>2cm</RightMargin>
        <TopMargin>1cm</TopMargin>
        <BottomMargin>1cm</BottomMargin>
        <ColumnSpacing>1.27cm</ColumnSpacing>
        <Style />
    </Page>
</ReportSection>
</ReportSections>
<Code>Public Function BlankZero(ByVal Value As Decimal)
    if Value = 0 then
        Return ""
    end if
    Return Value
End Function

Public Function BlankPos(ByVal Value As Decimal)
    if Value > 0 then
        Return ""
    end if
    Return Value
End Function

Public Function BlankZeroAndPos(ByVal Value As Decimal)
    if Value >= 0 then
        Return ""
    end if
    Return Value
End Function

Public Function BlankNeg(ByVal Value As Decimal)
    if Value < 0 then
        Return ""
    end if
    Return Value
End Function

Public Function BlankNegAndZero(ByVal Value As Decimal)
    if Value <= 0 then
        Return ""
    end if
    Return Value
End Function

</Code>
    <EmbeddedImages>
        <EmbeddedImage Name="autocarlogotemplatevectoricon">
            <MimeType>image/png</MimeType>
            <ImageData>
                MOYZNek6SrojoPemtnadp5rLWaKplacJMjy6FwDrKug403t2HGv830nNUXfIjmg+kd9jrPLR/cdvMvOmNz27iyXkjk
                q29hw604y4ptxJSpJsQearNdgURjfJC7uEchpPSX/t24rVcm1yTvshVkJ+jaHRQ0chzJVPktyYjy2XmzdC0GxGmp7rkeQERK+
                Ay7qEpA5CvPA60GxmZDSHmXEuNrF0rQQpKhtBHDqdMbnt3FkvJHJVt7Dh1pxlxTbiSlSTYg8Na0NoUtaglKRcqJsAMV3PjEfS
                YpYDrmovHoJ80deJMmRMew/IdU44s3UpRuefoGba7lpy8CWQ0tDTC+U0rypxQN16iz9FqqtKgvfpi62j+YxEmRJ7KX4kht5pw
                pbagtJ9I4Vtpjc9u4s15I5KtvYc0tOMuKbcSUqSbEHgPPMx2y484htA1qWQkD0nFUz7TImkiEkynNvrBHpOvFVzBVkyo/Cnz
                d7hpHJQPR9jgVSo0t3foMt605tbWU38tteKXuv5lhaKJjbE1A61J3tfrRin7stAfsJsOVGVtFnUesWOIm6Bk6aBvdYYT203a/
                rAwzWqPJALFRiOA6tB5CvwoEutr6K0nyG+F0to6S0jynFZfobzRU7UYjLqRxKW8hPoNziRmegRSQ5UWDb9Wd9/oviVuhUhm4j
                svPq8gQn1nE3dAq79xGbaJp29NfrPFiXpmz16cqQ46rq01E28nN//9k=</ImageData>
            </EmbeddedImage>
        </EmbeddedImages>
    <Language>=User!Language</Language>
    <ConsumeContainerWhitespace>true</ConsumeContainerWhitespace>
    <rd:ReportUnitType>Cm</rd:ReportUnitType>
    <rd:ReportID>0eeb6585-38ae-40f1-885b-8d50088d51b4</rd:ReportID>
    <DataSets>
        <DataSet Name="DataSet_Result">
            <Fields>
                <Field Name="Order_ID">
                    <DataField>Order_ID</DataField>
                </Field>
                <Field Name="Order_CustomerNo">
                    <DataField>Order_CustomerNo</DataField>
            </Fields>
        </DataSet>
    </DataSets>

```

```
</Field>
<Field Name="Order_CustomerName">
  <DataField>Order_CustomerName</DataField>
</Field>
<Field Name="Order_LocationName">
  <DataField>Order_LocationName</DataField>
</Field>
<Field Name="Order_ParkingCenterName">
  <DataField>Order_ParkingCenterName</DataField>
</Field>
<Field Name="Order_RentingDate">
  <DataField>Order_RentingDate</DataField>
</Field>
<Field Name="Order_ReturningDate">
  <DataField>Order_ReturningDate</DataField>
</Field>
<Field Name="Order_VechicleSeriesNumber">
  <DataField>Order_VechicleSeriesNumber</DataField>
</Field>
<Field Name="Order_VechicleType">
  <DataField>Order_VechicleType</DataField>
</Field>
<Field Name="Order_VechicleName">
  <DataField>Order_VechicleName</DataField>
</Field>
<Field Name="Order_AmountTxt">
  <DataField>Order_AmountTxt</DataField>
</Field>
<Field Name="Order_Amount">
  <DataField>Order_Amount</DataField>
</Field>
<Field Name="Order_AmountFormat">
  <DataField>Order_AmountFormat</DataField>
</Field>
</Fields>
<Query>
  <DataSourceName>DataSource</DataSourceName>
  <CommandText />
</Query>
</DataSet>
</DataSets>
</Report>
```