

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ
КАФЕДРА ЕЛЕКТРОНІКИ, ІНФОРМАЦІЙНИХ СИСТЕМ ТА
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Кваліфікаційна робота

перший (бакалаврський)

(рівень вищої освіти)

на тему **Розробка інформаційної системи транспортної компанії
Ірландії**

Виконав: студентка 5 курсу, групи ПЗ-19-1бз

спеціальності 121 Інженерія програмного

забезпечення

(код і назва спеціальності)

освітньої програми Інженерія програмного

забезпечення

(код і назва освітньої програми)

Є. М. Довганік

(ініціали та прізвище)

Керівник доцент, к.т.н. О.М. Міхайлуца

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Рецензент директор ТОВ «Дискус»

Р.О. Лютий

(посада, вчене звання, науковий ступінь, підпис, ініціали та прізвище)

Запоріжжя

2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ІНЖЕНЕРНИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ім. Ю.М. Потебні
ЗАПОРІЗЬКОГО НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ

Кафедра електроніки, інформаційних систем та програмного забезпечення
Рівень вищої освіти _____ перший (бакалавський) _____
Спеціальність _____ 121 Інженерія програмного забезпечення _____
(код та назва)
Освітня програма _____ Інженерія програмного забезпечення _____
(код та назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри _____ Т. В. Критська
“ 01 ” _____ березня _____ 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ

Довганік Євгенії Максимівні
(прізвище, ім'я, по батькові)

1. Тема роботи Розробка інформаційної системи транспортної компанії Ірландії
керівник роботи _____ Міхайлуца Олена Миколаївна, к.т.н., доцент
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом ЗНУ від _____ від 26.12.2023 № 2212-с
2. Строк подання студентом кваліфікаційної роботи _____ 21.05.2024
3. Вихідні дані дипломної роботи
 - комплект нормативних документів ;
 - технічне завдання до роботи.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
 - огляд та збір літератури стосовно теми дипломної роботи;
 - огляд та аналіз існуючих рішень та аналогів;
 - дослідження проблеми створення інформаційних систем;
 - створення програмного продукту та його опис;
 - перелік вимог для роботи програми;
 - тестування програмної системи та розробка висновків та пропозицій.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
_____ слайдів презентації

6. Консультанти розділів бакалаврської роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата
		Завдання прийняв

7. Дата видачі завдання 01.03.2024**КАЛЕНДАРНИЙ ПЛАН**

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів кваліфікаційної роботи	Примітка
1	Аналіз предметної області	01.03-10.03.24	виконано
2	Формулювання основної задачі дипломної роботи та узгодження її з науковим керівником	11.03-12.03.24	виконано
3	Аналіз існуючих методів рішення	13.03-14.03.24	виконано
4	Аналіз методів розробки інформаційних систем	15.03-20.03.24	виконано
5	Аналіз сучасних систем управління базами даних	21.03-26.03.24	виконано
6	Узгодження подальших дій з науковим керівником	27.03-28.03.24	виконано
7	Проектування інформаційної системи	29.03-05.04.24	виконано
8	Програмна реалізація застосунку	06.04-16.04.24	виконано
9	Представлення отриманих результатів науковому керівнику та узгодження плану подальшого дослідження	17.04-19.04.24	виконано
10	Реалізація користувацького інтерфейсу для комп'ютерної системи	20.04-01.05.24	виконано
11	Тестування застосунку	02.05-7.05.24	виконано
12	Оформлення звіту	8.05-15.05.24	виконано
13	Оформлення презентації. Отримання рецензій від опонентів.	16.05-20.05.24	виконано

Студент _____ Є.М. Довганік
(підпис) (прізвище та ініціали)Керівник роботи _____ О.М. Міхайлуца
(підпис) (прізвище та ініціали)**Нормоконтроль пройдено**Нормоконтролер _____ І.А. Скрипник
(підпис) (прізвище та ініціали)

АНОТАЦІЯ

Сторінок – 68

Рисунків – 28

Джерел – 15

Довганік Є. М. Розробка інформаційної системи транспортної компанії Ірландії : кваліфікаційна робота бакалавра спеціальності 121 «Інженерія програмного забезпечення» / наук. керівник О. М. Михайлуца. Запоріжжя : ЗНУ, 2024. 68 с.

В сучасних умовах динамічного розвитку транспортної галузі, чітка і злагоджена взаємодія між підрозділами компанії є ключовим фактором успіху. Підвищення ефективності цієї взаємодії стає одним із пріоритетних напрямків удосконалення діяльності будь-якої транспортної компанії. Автоматизація інформаційних потоків та процесів може суттєво оптимізувати роботу та підвищити загальну продуктивність.

Мета і завдання роботи полягають у проведенні дослідження особливостей функціонування транспортних компаній, визначенні практичних рекомендації щодо впровадження сучасних інформаційних технологій в управління транспортними підприємствами, аналізі існуючих програмних рішень для автоматизації інформаційних потоків та процесів в транспортній галузі.

Проведено порівняльний аналіз існуючих підходів та технологій до розробки складних інформаційних систем. На основі проведеного аналізу розроблено інтелектуальну інформаційну систему управління транспортною компанією з використанням сучасних методологій. Створений програмний комплекс може бути адаптований до потреб будь-якої транспортної компанії з метою оптимізації її роботи.

Ключові слова: *інформаційна система, транспортна компанія, MS SQL Server.*

ABSTRACT

Pages – 68

Drawings – 28

Sources – 15

Dovhanik Y. M. Development of the information system of the transport company of Ireland: qualifying thesis of the bachelor of specialty 121 "Software engineering" / Science. manager O. M. Mikhailutsa. Zaporizhzhia: ZNU, 2024. 68 p.

In today's conditions of dynamic development of the transport industry, clear and coordinated interaction between the company's divisions is a key success factor. Increasing the efficiency of this interaction becomes one of the priority directions for improving the activities of any transport company. Automation of information flows and processes can significantly optimize work and increase overall productivity.

The purpose and tasks of the work consist in conducting a study of the peculiarities of the functioning of transport companies, determining practical recommendations for the introduction of modern information technologies in the management of transport enterprises, analyzing existing software solutions for automating information flows and processes in the transport industry.

A comparative analysis of existing approaches and technologies to the development of complex information systems was conducted. On the basis of the conducted analysis, an intelligent information system of transport company management was developed using modern methodologies. The created software complex can be adapted to the needs of any transport company in order to optimize its work.

Keywords: *information system, transport company, MS SQL Server.*

ЗМІСТ

ВСТУП	7
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	14
1.1 Огляд ключових тенденцій розвитку транспортної логістики	14
1.2 Аналіз програмних продуктів-аналогів	15
1.3 Постановка завдання	29
2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ	31
2.1 Аналіз сучасних систем управління базами даних.....	31
2.2 Обґрунтування вибору методології та технології проектування.....	32
2.3 Мова програмування C#.....	36
2.4 Технологія для доступу до баз даних Entity Framework Core	38
3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ТРАНСПОРТНОЇ КОМПАНІЇ	40
3.1 Опис предметної області	40
3.2 Архітектура системи.....	41
3.3 Функціональні вимоги системи	42
3.4 Проектування інформаційної системи.....	44
3.5 Програмна реалізація застосунку	54
3.6 Розробка інтерфейсу інформаційної системи	60
ВИСНОВКИ.....	66
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	67

ВСТУП

Актуальність теми

Сьогодні більшість сучасних підприємств та організацій використовують різні інформаційні системи для покращення своєї діяльності. Інформаційні системи впроваджуються в усі сфери діяльності підприємств, включаючи бухгалтерський облік, управління персоналом та надання різноманітних послуг клієнтам.

Інформаційні технології допомагають людям виконувати роботу в різних сферах діяльності, адже сучасні інформаційні технології дозволяють автоматизувати більшість завдань, що призводить до покращення економічних показників підприємства. За допомогою інформаційних технологій можна виконувати складні для людини обчислення за лічені секунди, при цьому уникаючи помилок.

Крім того, інформаційні системи обробляють величезні масиви взаємопов'язаної інформації, які потребують зберігання, оновлення, коригування, а також дозволяють виконувати різноманітні запити до баз даних. Залежно від специфіки предметної області, інформаційні системи можуть дуже відрізнятися одна від одної за своїми функціями, архітектурою, реалізацією, але все-таки можна виділити деякі загальні властивості, що характерні для інформаційних систем:

- інформаційні системи призначені для збирання, зберігання та обробки інформації, тому в основі будь-якої з них лежить середовище зберігання та доступу до даних;
- інформаційні системи орієнтуються на кінцевого користувача, що не має високої кваліфікації в галузі застосування обчислювальної техніки, тому клієнтські додатки повинні мати простий, зручний інтерфейс, який надає можливість кінцевому користувачеві виконувати всі необхідні для роботи функції, але в той же час не дає йому виконувати зайві дії.

Незважаючи на те, що інформаційні системи можуть відрізнятися за функціоналом, архітектурою та реалізацією залежно від специфіки предметної області, існують деякі спільні властивості, характерні для всіх ІС:

1. Збирання, зберігання та обробка інформації:

- В основі будь-якої ІС лежить середовище зберігання та доступу до даних.
- ІС призначені для збирання, зберігання та обробки інформації, яка використовується для прийняття рішень, управління процесами та надання послуг.

2. Орієнтація на кінцевого користувача:

- ІС розробляються для користувачів, які не мають глибоких знань в ІТ-сфері.
- Тому клієнтські додатки повинні мати простий та зручний інтерфейс, який дозволяє користувачеві виконувати всі необхідні функції без зайвих дій.

3. Задачі, які вирішуються при розробці ІС:

- Розробка бази даних, яка призначена для зберігання інформації, що використовується ІС. При розробці бази даних необхідно визначити структуру даних, способи доступу до них та методи забезпечення їх цілісності та безпеки.
- Розробка графічного інтерфейсу користувача, який забезпечує взаємодію користувача з ІС. При його розробці необхідно враховувати ергономіку, простоту використання та естетичність інтерфейсу.

Крім цих основних завдань, при розробці ІС також вирішуються й інші:

- Розробка програмного забезпечення, яке реалізує логіку роботи ІС. Воно повинне бути надійним, ефективним та відповідати вимогам користувачів.
- Тестування та впровадження ІС. Перед впровадженням ІС необхідно провести ретельне тестування, щоб переконатися в її працездатності та

відповідності вимогам. Впровадження ІС потребує навчання користувачів та внесення змін до бізнес-процесів підприємства.

- Супровід та підтримка ІС. Після впровадження ІС необхідно забезпечити її супровід та підтримку. Це включає в себе виправлення помилок, оновлення програмного забезпечення та навчання нових користувачів.

Отже, розробка інформаційної системи — це складний процес, який потребує ретельного планування, кваліфікованих фахівців та значних ресурсів. Однак, успішно розроблена та впроваджена ІС може значно покращити ефективність роботи підприємства, підвищити його конкурентоспроможність та надати нові можливості для розвитку.

Транспортна сфера Ірландії динамічно розвивається останніми роками, що призводить до жорсткої конкуренції між компаніями. Ефективна інформаційна система може допомогти транспортним компаніям отримати конкурентну перевагу, покращивши їхні операції, обслуговування клієнтів та прийняття рішень.

Існує декілька ключових факторів, які роблять розробку ІС для транспортної компанії Ірландії актуальною:

1. Зростаюча конкуренція:

- Кількість транспортних компаній в Ірландії зростає, що призводить до жорсткої конкуренції за клієнтів.
- ІС може допомогти транспортним компаніям диференціюватися від своїх конкурентів та пропонувати кращі послуги та ціни.

2. Зростаючі потреби клієнтів:

- Клієнти очікують зручного та ефективного обслуговування від транспортних компаній.
- ІС може допомогти транспортним компаніям краще задовольнити ці потреби, надаючи клієнтам доступ до інформації про рейси, бронювання, відстеження вантажів та ін.

3. Покращення операційної ефективності:

- ІС може допомогти транспортним компаніям автоматизувати багато завдань, що може призвести до значної економії часу та коштів.
- Це може покращити загальну ефективність операцій компанії та її прибутковість.

4. Підвищення безпеки:

- ІС може допомогти транспортним компаніям покращити безпеку своїх операцій, відстежуючи вантажі, транспортні засоби та водіїв.
- Це може допомогти запобігти аваріям та іншим інцидентам.

5. Використання нових технологій:

- Транспортний сектор швидко впроваджує нові технології, такі як штучний інтелект, Інтернет речей та блокчейн.
- ІС може допомогти транспортним компаніям використовувати ці технології для покращення своїх операцій та обслуговування клієнтів.

6. Вплив COVID-19:

- Пандемія COVID-19 призвела до значних змін у транспортному секторі.
- ІС може допомогти транспортним компаніям адаптуватися до цих змін та краще обслуговувати своїх клієнтів у нових умовах.

7. Вимоги законодавства:

- Існують різні законодавчі акти, які регулюють транспортну сферу в Ірландії.
- ІС може допомогти транспортним компаніям дотримуватися цих вимог.

8. Підтримка стійкості:

- Транспортний сектор відіграє важливу роль у зміні клімату.
- ІС може допомогти транспортним компаніям стати більш стійкими, оптимізуючи свої маршрути, скорочуючи викиди парникових газів та використовуючи більш екологічні транспортні засоби.

Розробка інформаційної системи для транспортної компанії Ірландії є актуальною темою з багатьох причин. Така система може допомогти компаніям отримати конкурентну перевагу, покращити обслуговування клієнтів, підвищити операційну ефективність, посилити безпеку,

використовувати нові технології, адаптуватися до змін, дотримуватися законодавства та підтримувати стійкість. Інвестиція в розробку ІС може допомогти транспортним компаніям Ірландії залишатися конкурентоспроможними та успішними в динамічному та мінливому середовищі.

Мета дослідження

Розробка інформаційної системи транспортної компанії Ірландії з використанням сучасних методологій.

Завдання дослідження

Розглянути особливості та існуючі рішення (аналоги) для ведення діяльності транспортних компаній. Проаналізувати існуючі підходи та технології до розробки складних інформаційних систем та вибрати відповідну для реалізації інформаційної системи транспортної компанії.

Об'єкт дослідження

Об'єктом дослідження є процес розробки інформаційної системи транспортної компанії.

Предмет дослідження

Предметом дослідження є процес автоматизації бізнес-процесів у компанії з вантажоперевезень та управління заявками на надання послуг з вантажних перевезень, а також виведення необхідної звітної документації.

Практичне значення одержаних результатів

Практична значимість роботи представляється у розробці інформаційної системи для компанії з вантажоперевезень, яка дозволить заощадити матеріальні та тимчасові ресурси та відповідно підвищити прибуток.

Глосарій

Інформаційна система — Комп'ютеризована система, яка збирає, зберігає, обробляє та надає інформацію, необхідну для прийняття рішень, управління процесами та надання послуг.

Транспортна компанія — Підприємство, яке надає послуги з перевезення пасажирів, вантажів або обох.

Графічний інтерфейс користувача — Програмний компонент, який дозволяє користувачеві взаємодіяти з ІС за допомогою візуальних елементів, таких як іконки, меню та кнопки.

Логістика — Процес планування, організації та управління переміщенням товарів та інформації від точки виробництва до точки споживання.

Управління ланцюгом постачання — Мережа організацій, що співпрацюють для доставляння товарів та послуг від виробника до кінцевого споживача.

Система відстеження рейсів — Програмне забезпечення, яке використовується для відстеження розташування та стану транспортних засобів, вантажів та пасажирів.

Система бронювання — Програмне забезпечення, яке використовується для бронювання місць у транспортних засобах, готелях та інших закладах.

Система диспетчеризації — Програмне забезпечення, яке використовується для призначення транспортних засобів, водіїв та інших ресурсів для виконання завдань.

Система управління персоналом — Програмне забезпечення, яке використовується для управління такими аспектами роботи з персоналом, як набір кадрів, навчання, оплата праці та оцінка результатів роботи.

Стійкість — Здатність системи витримувати або відновлюватися після збоїв, потрясінь або інших змін.

Законодавство — Система законів, правил та норм, які регулюють транспортну сферу.

Конкурентна перевага — Фактор, який дає одній компанії перевагу над іншими на ринку.

Обслуговування клієнтів — Процес надання допомоги та підтримки клієнтам.

Операційна ефективність — Здатність компанії виробляти товари та послуги з мінімальними витратами та ресурсами.

Інтерфейс — це сукупність засобів, методів та правил, призначених для взаємодії елементів системи (або цілих систем) між собою.

Архітектура — це базова організація системи, втілена в її компонентах, їхніх відносинах між собою і з оточенням, а також принципи, що визначають проектування та розвиток системи програми, видимі ззовні властивості цих елементів і зв'язки між ними.

Прив'язка даних — процес приєднання даних в одному об'єкті до іншого об'єкту. Він забезпечує зручний шлях передачі даних між різними рівнями застосунку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Огляд ключових тенденцій розвитку транспортної логістики

Транспортна логістика — це динамічна галузь, яка постійно розвивається, адаптуючись до мінливих потреб світової економіки. За останні роки спостерігається стрімкий розвиток цієї сфери, обумовлений низкою ключових факторів:

1. Зростання електронної комерції: бурхливий розвиток онлайн-торгівлі призвів до збільшення обсягу товарів, які потребують доставки «з пункту А до пункту Б». Це, в свою чергу, стимулює попит на логістичні послуги, пов'язані з складуванням, упаковкою, доставкою та управлінням замовленнями.

2. Глобалізація: світ стає все більш взаємопов'язаним, що веде до зростання міжнародної торгівлі та ланцюгів постачання. Це робить логістику необхідною ланкою для будь-якого бізнесу, який прагне вийти на міжнародний ринок.

3. Інновації: логістична галузь активно впроваджує новітні технології, такі як штучний інтелект, машинне навчання, Інтернет речей (IoT) та роботизація. Ці технології дозволяють оптимізувати маршрути, автоматизувати рутинні завдання, покращити видимість ланцюгів постачання та підвищити загальну ефективність.

4. Зростання екологічної свідомості: споживачі та компанії все більше зацікавлені в екологічно відповідальній логістиці. Це стимулює розвиток «зелених» логістичних рішень, таких як використання електричного транспорту, оптимізація маршрутів для зменшення викидів CO₂ та використання екологічно чистих упаковок.

5. Зміна споживчих очікувань: сучасні споживачі очікують швидкої, надійної та зручної доставки. Це змушує логістичні компанії постійно вдосконалювати свої послуги та пропонувати нові опції, такі як доставка того ж дня, гнучкі варіанти доставки та прозоре відстеження замовлень.

Ці ключові тенденції обумовлюють ряд важливих змін в логістичній галузі:

- Збільшення використання мультимодальних перевезень: перевезення вантажів декількома видами транспорту (наприклад, авто плюс залізниця) стає все більш популярним завдяки своїй економічності та екологічності.
- Зростання популярності складів «міського типу»: ці склади розташовуються в межах міст, що дозволяє скоротити час доставки товарів кінцевим споживачам.
- Розвиток омніканальної логістики: логістичні компанії змушені адаптуватися до того, що споживачі використовують декілька каналів (онлайн, офлайн) для здійснення покупок. Це потребує інтеграції логістичних процесів в єдину систему.
- Персоналізація доставки: логістичні компанії все частіше пропонують персоналізовані послуги доставки, враховуючи індивідуальні потреби та побажання клієнтів.

1.2 Аналіз програмних продуктів-аналогів

SAP Supply Chain Management (SAP SCM)

SAP Supply Chain Management — це комплексна програмна платформа, що допомагає компаніям управляти своїми ланцюгами постачання. Вона пропонує широкий спектр модулів, які охоплюють всі аспекти ланцюжка постачання, від планування та закупівлі до виробництва, дистрибуції та обслуговування клієнтів (див. Рис. 1).

Основні можливості SAP SCM включають:

- Планування ланцюжка постачання: дозволяє компаніям прогнозувати попит, оптимізувати запаси та планувати виробництво.
- Виробництво: дозволяє компаніям планувати та оптимізувати виробничі процеси, відстежувати виробництво та контролювати якість.

- **Управління закупівлями:** допомагає компаніям знаходити постачальників, вести переговори про ціни та оптимізувати процес закупівлі.
- **Дистрибуція:** допомагає компаніям управляти складами, планувати доставку та відстежувати замовлення.
- **Обслуговування клієнтів:** дозволяє компаніям керувати замовленнями клієнтів, надавати підтримку клієнтам та аналізувати дані про клієнтів.



Рисунок 1 — SAP Supply Chain Management

Однак SAP SCM також має деякі недоліки:

- **Висока вартість:** впровадження та обслуговування SAP SCM може бути дорогим, особливо для малих та середніх підприємств.
- **Складність:** SAP SCM є складною платформою, яка потребує значних знань та досвіду для її використання.
- **Жорстка структура:** SAP SCM може бути складною для налаштування та адаптації до конкретних потреб бізнесу.
- **Необхідність інтеграції:** SAP SCM може потребувати інтеграції з іншими системами ERP та CRM, що може бути дорогим та складним.

Загалом, SAP SCM є потужним інструментом, який може допомогти компаніям покращити ефективність ланцюжка постачання, знизити витрати та підвищити рентабельність інвестицій.

Oracle Supply Chain Management Cloud (Oracle SCM Cloud)

Oracle SCM Cloud — це хмарна служба управління ланцюгом постачання (УЛП), розроблена компанією Oracle. Вона допомагає підприємствам оптимізувати всі етапи ланцюжка постачання, від планування попиту та закупівель до виробництва, дистрибуції та обслуговування клієнтів (див. Рис. 2).

Основні можливості Oracle SCM Cloud:

- Планування попиту та постачання: дозволяє прогнозувати попит, оптимізувати запаси та планувати виробництво на основі аналізу даних та машинного навчання.
- Логістика та дистрибуція: дозволяє планувати маршрути доставки, відстежувати транспортні засоби та вантажі в режимі реального часу, оптимізувати витрати на транспортування.
- Обслуговування клієнтів: сприяє ефективному управлінню замовленнями клієнтів, надає підтримку клієнтам та аналізує дані про клієнтів для покращення взаємовідносин.

Переваги використання Oracle SCM Cloud:

- Збільшення ефективності ланцюга постачання: Oracle SCM Cloud допомагає оптимізувати всі етапи ланцюжка постачання, що може призвести до зниження витрат, підвищення точності прогнозів попиту та скорочення часу виконання замовлень.
- Підвищення гнучкості та масштабованості: хмарна природа Oracle SCM Cloud дозволяє компаніям легко масштабувати систему відповідно до потреб бізнесу.
- Зниження витрат на IT-інфраструктуру: оскільки Oracle SCM Cloud є хмарною службою, компаніям не потрібно інвестувати в дороге обладнання та програмне забезпечення.
- Покращена видимість ланцюга постачання: Oracle SCM Cloud забезпечує повну видимість всіх етапів ланцюжка постачання, що дозволяє компаніям приймати більш обґрунтовані рішення.

- Підтримка інновацій: Oracle SCM Cloud постійно оновлюється новими функціями та можливостями, що дозволяє компаніям залишатися на чолі інновацій у сфері управління ланцюгом постачання.



Рисунок 2 — Oracle Supply Chain Management Cloud

При виборі рішення для управління ланцюгом постачання важливо порівняти Oracle SCM Cloud з іншими варіантами, такими як SAP Supply Chain Management (SAP SCM). Обидві платформи є потужними рішеннями, але вони мають свої переваги та недоліки.

Oracle SCM Cloud, незважаючи на свої переваги, має також певні недоліки, які варто врахувати перед його впровадженням. Деякі з них:

1. Залежність від хмарних сервісів: Oracle SCM Cloud є хмарною платформою, що означає залежність від стабільного інтернет-з'єднання. Перебої з інтернетом можуть вплинути на роботу системи та призвести до затримки або помилок у роботі.

2. Можливі обмеження функціональності: хоча Oracle SCM Cloud пропонує широкий спектр функціональних можливостей, вони можуть не повністю відповідати специфічним потребам усіх бізнесів. Можлива необхідність додаткової настройки або інтеграції з іншими системами для задоволення унікальних потреб вашої компанії.

3. Міграція даних та інтеграція: міграція даних з існуючих систем до Oracle SCM Cloud може бути складним та тривалим процесом. Також може знадобитися інтеграція з іншими системами вашої компанії, що потребує додаткових витрат часу та ресурсів.

4. Безпека даних: хоча Oracle забезпечує безпеку хмарних сервісів, деякі компанії можуть бути стурбовані розміщенням критичних даних ланцюга постачання за межами їхнього локального контролю.

5. Залежність від постачальника: як і з будь-яким хмарним рішенням, користувач залежить від Oracle щодо доступності, продуктивності та безпеки платформи.

6. Вартість володіння: хоча Oracle SCM Cloud пропонує модель підписки з оплатою за користування, загальна вартість володіння може бути значною, враховуючи витрати на ліцензії, налаштування, інтеграцію та навчання користувачів.

7. Необхідність змін у бізнес-процесах: впровадження Oracle SCM Cloud може потребувати змін у ваших існуючих бізнес-процесах, що може викликати опір у співробітників. Важливо забезпечити ефективне навчання та підтримку для успішного переходу.

Загалом, Oracle SCM Cloud є потужним інструментом для управління ланцюгом постачання, але важливо усвідомлювати його потенційні недоліки перед прийняттям рішення про впровадження. Обов'язково ретельно оцініть свої потреби та ресурси, щоб визначити, чи підходить ця система для вашої компанії.

JDA Transportation Management (JDA TMS)

JDA Transportation Management — це система управління транспортом (TMS), яка допомагає компаніям планувати, оптимізувати та виконувати свої транспортні операції. Її можливості включають планування маршрутів, диспетчеризацію, відстеження вантажів, управління тарифами та аудит (див. Рис. 3).

До основних можливостей JDA TMS відноситься наступне:

- Планування маршрутів: дозволяє оптимізувати маршрути доставки, враховуючи такі фактори, як місцезнаходження складів, вантажів, водіїв та транспортних засобів, а також обмеження часу та витрат.
- Диспетчеризація: дозволяє відстежувати транспортні засоби та вантажі в режимі реального часу, координувати дії водіїв та оптимізувати маршрути доставки.
- Відстеження вантажів: дозволяє відстежувати статус вантажів протягом усього ланцюжка постачання, від моменту завантаження до моменту доставки.
- Управління тарифами: дозволяє автоматизувати розрахунок та застосування фрахтових ставок, а також управляти контрактами з перевізниками.
- Аудит: дозволяє відстежувати та аналізувати транспортні дані для виявлення неефективності та можливостей для покращення.

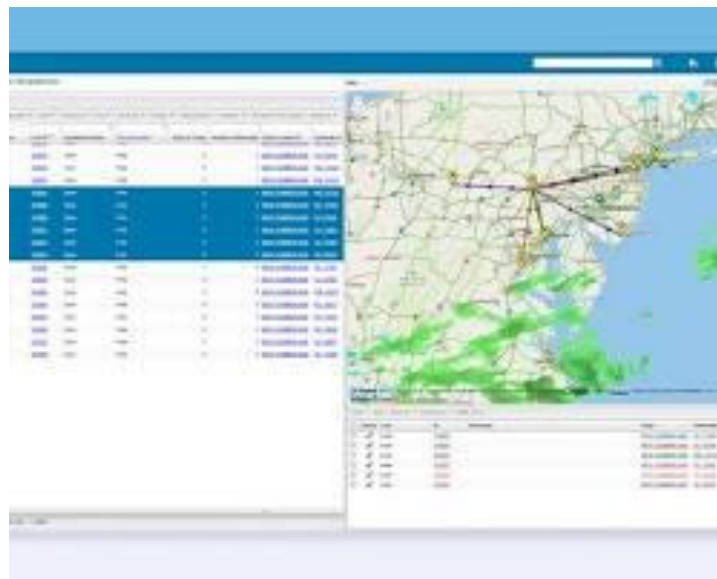


Рисунок 3 — JDA Transportation Management

Перевагами використання JDA TMS є:

- Зниження витрат на транспорт: JDA TMS може допомогти компаніям економити на транспортних витратах за рахунок оптимізації маршрутів, зниження простою та покращення використання транспортних засобів.

- Покращення обслуговування клієнтів: JDA TMS може допомогти компаніям покращити обслуговування клієнтів за рахунок більш точного прогнозування часу доставки та підвищення видимості вантажів.
- Покращена ефективність ланцюжка постачання: JDA TMS може допомогти компаніям покращити ефективність ланцюжка постачання за рахунок оптимізації транспортних операцій та зменшення затримок.
- Підвищення видимості та контролю: JDA TMS забезпечує повну видимість транспортних операцій, що дозволяє компаніям приймати більш обґрунтовані рішення.
- Зниження ризиків: JDA TMS може допомогти компаніям знизити ризики, пов'язані з транспортними операціями, за рахунок кращого планування та моніторингу.

Однак JDA TMS також має деякі недоліки:

- Висока вартість: JDA TMS є дорогою системою, що може зробити її недоступною для малих та середніх підприємств.
- Складність: JDA TMS є складною системою, яка потребує значних знань та досвіду для її використання.
- Жорстка структура: JDA TMS може бути складною для налаштування та адаптації до конкретних потреб бізнесу.
- Необхідність інтеграції: JDA TMS може потребувати інтеграції з іншими системами ERP та CRM, що може бути дорогим та складним.

Microlise Fleet Management

Microlise Fleet Management — це комплексна система управління автопарком, яка допомагає компаніям оптимізувати свої транспортні операції (див. Рис. 4).

До основних можливостей Microlise Fleet Management можна віднести наступне:

- Відстеження та моніторинг транспортних засобів: система дозволяє відстежувати місцезнаходження, швидкість та інші параметри транспортних засобів в режимі реального часу.
- Управління поведінкою водія: система допомагає водіям їздити більш економно та безпечно, надаючи їм рекомендації та попередження.
- Планування маршрутів та оптимізація: система дозволяє планувати оптимальні маршрути для транспортних засобів, враховуючи такі фактори, як місцезнаходження пунктів призначення, час доби та дорожні умови.
- Управління паливом: система допомагає контролювати витрати палива та виявляти випадки нецільового використання.
- Технічне обслуговування та ремонт: система допомагає планувати та організувати технічне обслуговування та ремонт транспортних засобів.
- Звіти та аналітика: система надає детальні звіти про роботу транспортних засобів та водіїв, що дозволяє приймати обґрунтовані рішення щодо покращення ефективності.

The screenshot shows the Microlise website's main page. At the top, there is a navigation menu with the Microlise logo on the left and links for 'Products', 'Industries', 'IIoT & OEM', 'Resources', 'Company', and a blue 'Request a Demo' button on the right. Below the navigation bar, the page is organized into three main content blocks:

- Transport Management System (TMS):** The first block features the title 'Transport Management System (TMS)', a sub-headline 'Transform the way you manage, optimise, and track your transportation processes, and streamline your supply chain like never before', and a blue 'LEARN MORE' button. To the right of this text is a photograph of a truck driving on a road through a wooded area.
- Fleet Telematics:** The second block has the title 'Fleet Telematics', a sub-headline 'Maximise driver efficiency and ensure your vehicles are being operated at an optimum level with Microlise fleet telematics', and a blue 'LEARN MORE' button.
- Safety, Health & Compliance:** The third block includes the title 'Safety, Health & Compliance', a sub-headline 'Is your fleet meeting and exceeding legal obligations? Do you have the systems in place to ensure the safety of your team operating out on the road and in the field? Use our safety, health and compliance products to ensure the maximum up-time of your fleet and safeguard your staff.', and a blue 'LEARN MORE' button. To the right of this text is a photograph of a multi-lane highway with a truck, overlaid with a 'SAFETY CHECKS' box listing 'Driver', 'Vehicle', and 'Weather'.

Рисунок 4 — Головна сторінка Microlise Fleet Management

До переваг використання Microlise Fleet Management можна віднести:

- Зниження витрат: система може допомогти компаніям значно знизити витрати на транспортні операції за рахунок економії палива, зменшення простою, продовження терміну служби транспортних засобів та зниження ризику аварій.
- Покращення безпеки: система допомагає покращити безпеку водіння та знизити ризик аварій.
- Підвищення ефективності: система дозволяє оптимізувати транспортні операції, що призводить до підвищення ефективності та продуктивності.
- Покращення обслуговування клієнтів: система допомагає покращити обслуговування клієнтів за рахунок більш точного прогнозування часу доставки та підвищення видимості вантажів.
- Підвищення видимості та контролю: система забезпечує повну видимість транспортних операцій, що дозволяє компаніям приймати більш обґрунтовані рішення.

Microlise Fleet Management підходить для різних галузей, включаючи транспортні та логістичні компанії, компанії з доставки. Однак Microlise Fleet Management також має деякі недоліки, серед яких варто відмітити:

- Висока вартість: система може бути дорогим рішенням для малих та середніх підприємств.
- Складність: система може бути складною для впровадження та використання.
- Необхідність інтеграції: система може потребувати інтеграції з іншими системами ERP та CRM, що може бути дорогим та складним.
- Залежність від постачальника: компанії залежать від Microlise щодо підтримки та оновлень системи.

Verizon Connect

Verizon Connect — це комплексна платформа управління автопарком, яка допомагає компаніям відстежувати, оптимізувати та керувати своїми транспортними операціями. Вона пропонує широкий спектр функціональних

можливостей для підвищення ефективності, безпеки та продуктивності автопарку (див. Рис. 5).

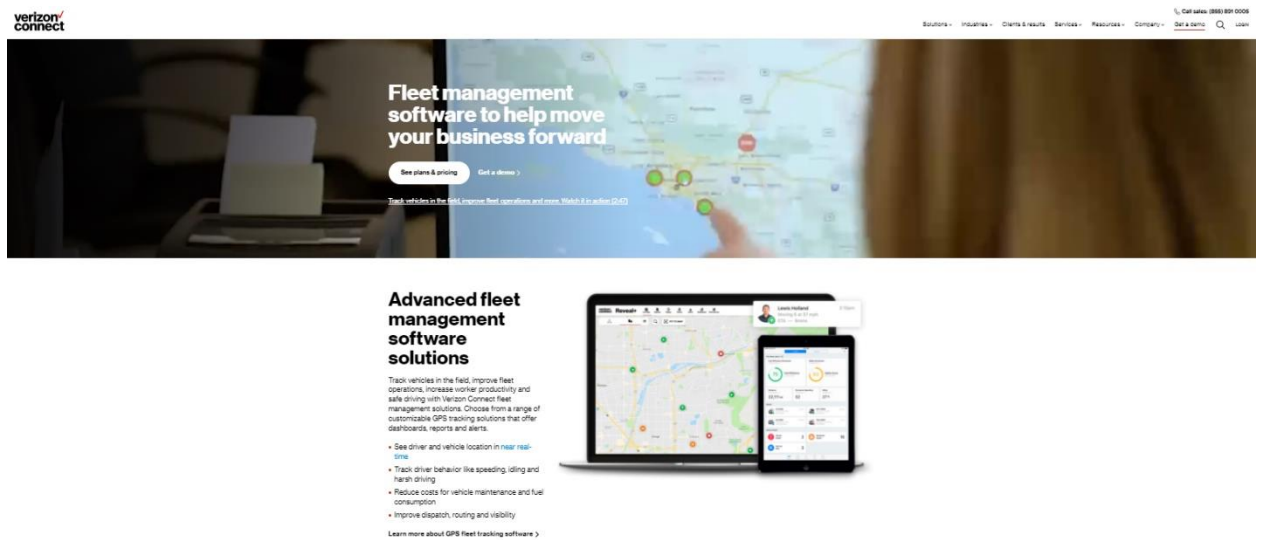


Рисунок 5 — Головна сторінка Verizon Connect

Основні можливості Verizon Connect:

- Відстеження GPS в реальному часі: відстеження місцезнаходження, швидкості, напрямку та простою транспортних засобів 24/7.
- Моніторинг поведінки водія: отримання даних про різкі гальмування, прискорення, повороти та холостий хід для покращення стилю водіння.
- Оптимізація маршрутів: планування ефективних маршрутів з урахуванням дорожнього трафіку, зупинок та обмежень.
- Управління паливом: моніторинг витрат палива, виявлення неефективного використання та оптимізація заправки.
- Обслуговування транспортних засобів: нагадування про технічне обслуговування на основі пробігу або часу, управління ремонтами.
- Безпека водіїв: оповіщення про порушення правил дорожнього руху, такі як відволічене керування та перевищення швидкості.
- Звіти та аналітика: створення детальних звітів про ефективність, безпеку та продуктивність автопарку.

До переваг використання Verizon Connect можна віднести:

- Зниження витрат: скорочення витрат на паливо, покращення технічного обслуговування, зниження простоїв та ризику аварій.
- Підвищення безпеки: моніторинг поведінки водія для запобігання аваріям та створення безпечнішого робочого середовища.
- Покращення ефективності: оптимізація маршрутів, скорочення часу доставки та підвищення продуктивності.
- Підвищення продуктивності водіїв: надання диспетчерам даних для кращого управління роботою водіїв.
- Покращення обслуговування клієнтів: точне прогнозування часу доставки та підвищення інформованості клієнтів.

Verizon Connect підходить для різних галузей, включаючи транспортні та логістичні компанії, компанії з доставки, комунальні служби, а також таксі та служби прокату автомобілів.

Однак, Verizon Connect також має деякі недоліки:

- Вартість: абонентська плата може бути високою для малих підприємств.
- Складність налаштування: система налаштовується під конкретні потреби, що може потребувати часу та ресурсів.
- Залежність від стійкого сигналу: деякі функції, наприклад, відстеження в реальному часі, залежать від надійного стільникового зв'язку.
- Інтеграція з сторонніми системами: інтеграція з програмами бухгалтерського обліку або планування маршрутів може потребувати додаткових витрат.

Загалом, Verizon Connect є потужним інструментом для управління автопарком, який може допомогти компаніям оптимізувати свої операції, знизити витрати та підвищити безпеку.

GPS Insight

GPS Insight — це хмарна платформа управління автопарком, яка допомагає компаніям відстежувати свої транспортні засоби, оптимізувати маршрути, підвищувати безпеку водіїв та знижувати витрати. Вона пропонує

широкий спектр функцій для ефективного управління автопарком будь-якого розміру (див. Рис. 6).

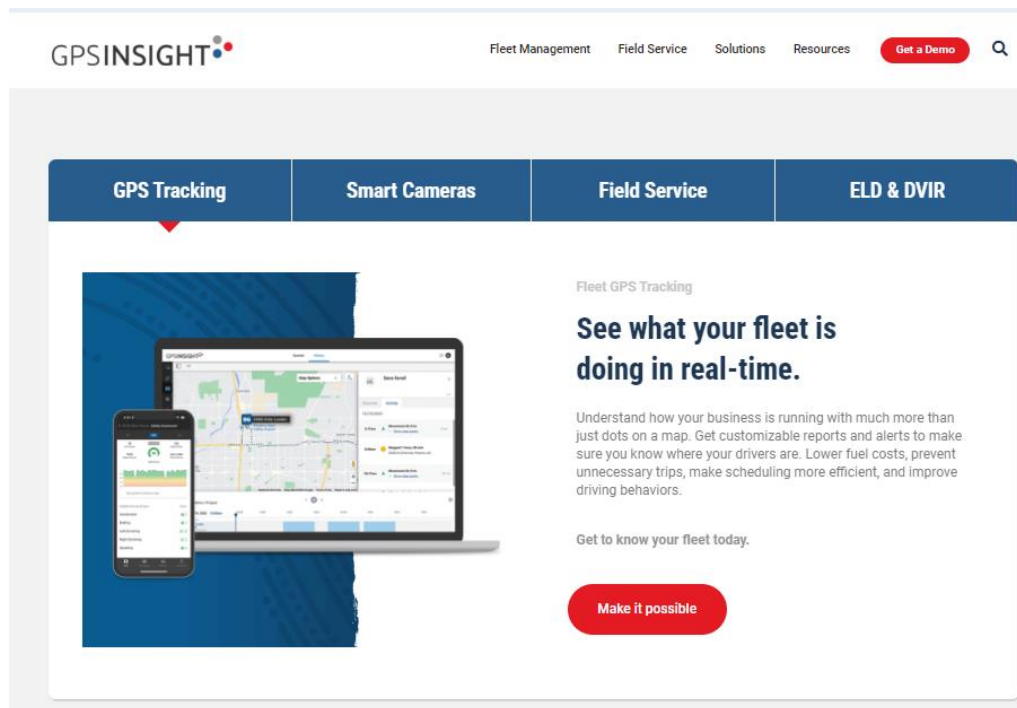


Рисунок 6 — Сторінка GPS Insight

Основні можливості GPS Insight:

- Відстеження GPS в реальному часі: цілодобовий моніторинг місцезнаходження, швидкості, напрямку та простою транспортних засобів.
- Оптимізація маршрутів: планування найефективніших маршрутів з урахуванням дорожнього трафіку, зупинок, обмежень та габаритно-вагових характеристик транспортних засобів.
- Моніторинг поведінки водія: отримання даних про різкі гальмування, прискорення, повороти та холостий хід для поліпшення стилю водіння та зниження витрат на паливо.
- Управління паливом: спостереження за витратами палива, виявлення неефективного використання пального та оптимізація заправки.

- Обслуговування транспортних засобів: нагадування про технічне обслуговування на основі пробігу або часу, управління ремонтами та історією обслуговування.
- Безпека водіїв: оповіщення про порушення правил дорожнього руху, такі як відволічене керування та перевищення швидкості, для запобігання аваріям.
- Звіти та аналітика: створення детальних звітів про ефективність, безпеку, продуктивність та витрати автопарку.
- Мобільний додаток: доступ до даних про автопарк та управління завданнями безпосередньо з мобільного пристрою.

До переваги використання GPS Insight можна віднести:

- Зниження витрат: скорочення витрат на паливо, покращення технічного обслуговування, оптимізація маршрутів та зниження простоїв.
- Підвищення безпеки: спостереження за поведінкою водія для запобігання аваріям, поліпшення стилю водіння та створення безпечнішого робочого середовища.
- Покращення ефективності: оптимізація маршрутів, скорочення часу доставки та підвищення продуктивності співробітників.
- Краще обслуговування клієнтів: точне прогнозування часу доставки та підвищення інформованості клієнтів про місцезнаходження вантажу.
- Підвищення продуктивності водіїв: надання диспетчерам даних для кращого управління роботою водіїв.

Серед недоліків GPS Insight можна вказати наступне:

- Вартість: абонентська плата може бути високою для малих підприємств, особливо за розширені функції.
- Складність налаштування: система налаштовується під конкретні потреби бізнесу, що може потребувати часу та ресурсів від ІТ-відділу.
- Залежність від підключення до інтернету: деякі функції, наприклад, відстеження в реальному часі та мобільний додаток, залежать від надійного підключення до інтернету.

- Інтеграція з сторонніми системами: інтеграція з програмами бухгалтерського обліку або планування маршрутів може потребувати додаткових витрат.

Загалом, GPS Insight є потужним хмарним рішенням для управління автопарком, яке може допомогти компаніям оптимізувати свої операції, знизити витрати та підвищити безпеку.

Системи відстеження рейсів

Системи відстеження рейсів — це програмні комплекси, які використовуються для моніторингу та управління пересуванням транспортних засобів (див. Рис. 7). Їх застосовують у різних сферах, таких як:

- Логістика та вантажні перевезення: для відстеження вантажів, оптимізації маршрутів та контролю часу доставки.
- Таксі та служби прокату: для відстеження водіїв та забезпечення безпеки пасажирів.
- Служби доставки: для моніторингу кур'єрів та контролю часу доставки замовлень.



Рисунок 7 — Сторінка GPS Insight

До основних функцій систем відстеження рейсів відноситься:

- Відстеження GPS: визначення місцезнаходження транспортного засобу в режимі реального часу.
- Відображення маршрутів: відображення пройденого та запланованого маршруту на карті.

- Історія поїздок: зберігання інформації про попередні поїздки, включаючи час, маршрут та інші дані.
- Недоліки систем відстеження рейсів:
- Вартість: впровадження та обслуговування системи може бути дорогим, особливо для малих підприємств.
 - Складність: деякі системи можуть бути складними для налаштування та використання.
 - Залежність від технологій: система може бути недоступною при відсутності мобільного зв'язку або Інтернету.

Системи відстеження рейсів — це ефективний інструмент для підвищення ефективності роботи транспортних компаній.

1.3 Постановка завдання

Метою дипломної роботи є створення інформаційної системи з вантажоперевезень, яка дозволить зменшити тимчасові витрати на отримання та опрацювання інформації, що дасть можливість клієнтам отримувати всю інформацію, що їх цікавить, і новини про компанію у зручний час за допомогою Internet. Крім того дана інформаційна система дозволить зменшити витрати компанії як матеріальні, так і тимчасові ресурси, на формування звітної документації.

Для досягнення поставленої в роботі мети потрібно вирішити ряд задач, а саме:

1. Створення концептуальної моделі (ER-моделі), для чого:
 - проаналізувати предметну область;
 - ретельно вивчити бізнес-процеси, потреби та вимоги транспортної компанії до інформаційної системи;
 - визначити ключові сутності, що описують об'єкти та дані, з якими буде працювати система;

- для кожної сутності визначити всі необхідні атрибути, включаючи ідентифікуючі (первинний ключ), які однозначно описують кожен екземпляр сутності;
- визначити зв'язки між сутностями, що описують їх взаємозв'язок та взаємозалежність;
- описати зв'язки, включаючи кардинальність (кількість екземплярів однієї сутності, які можуть бути пов'язані з одним екземпляром іншої сутності);
- описати ER-модель, створити ER-діаграму, яка візуально представляє сутності, атрибути та зв'язки між ними. ER-діаграма має бути чіткою, лаконічною та зрозумілою.

2. Створення бази даних на MS SQL Server:

- використовуючи ER-модель як основу, створити структуру бази даних в MS SQL Server;
- створити таблиці, визначити поля, типи даних, обмеження та ключі;
- заповнити базу даних тестовими даними для перевірки та налагодження роботи.

3. Створення інтерфейсу користувача:

- розробити зручний та інтуїтивно зрозумілий інтерфейс користувача, який дозволить користувачам транспортної компанії ефективно взаємодіяти з інформаційною системою;
- інтерфейс користувача повинен відповідати вимогам користувачів та забезпечувати легкий доступ до необхідних функцій та даних.

2 ДОСЛІДЖЕННЯ ПРОГРАМНИХ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Аналіз сучасних систем управління базами даних

Для розробки інформаційної системи вантажоперевезень, яка відповідає поставленим цілям, важливо правильно вибрати систему управління базами даних (СУБД). СУБД буде слугувати основою для зберігання та обробки даних про вантажоперевезення, клієнтів, маршрути та інші аспекти роботи компанії.

При виборі СУБД для інформаційної системи вантажоперевезень слід враховувати такі фактори:

- Масштабованість: Чи зможе СУБД обробляти зростаючий обсяг даних та навантаження?
- Продуктивність: Яка швидкість доступу до даних та виконання запитів?
- Функціональність: Які функції пропонує СУБД (підтримка складних типів даних, геопросторові функції, аналітика даних тощо)?
- Надійність: Чи гарантує СУБД цілісність та доступність даних?
- Безпека: Які механізми захисту даних пропонує СУБД?
- Вартість: Чи є СУБД безкоштовною або комерційною, скільки коштує ліцензування та обслуговування?
- Підтримка спільноти: Наскільки велика та активна спільнота користувачів та розробників СУБД?

Рекомендовані СУБД для інформаційної системи вантажоперевезень:

- MySQL: відкрита РСУБД з високою продуктивністю та масштабованістю, використовується в багатьох веб-проектах.
- PostgreSQL: відкрита РСУБД з акцентом на надійність та функціональність, підтримує складні типи даних та геопросторові функції.
- Microsoft SQL Server: комерційна РСУБД з широкими можливостями, використовується в корпоративних системах.

Вибір СУБД залежить від конкретних потреб компанії та її бюджету.

- MySQL може бути хорошим вибором, якщо потрібна безкоштовна та масштабована СУБД з базовою функціональністю.
- PostgreSQL може бути кращим вибором, якщо вам потрібна більш надійна та функціональна СУБД, яка підтримує складні типи даних та геопросторові функції.
- Microsoft SQL Server може бути кращим вибором, якщо вам потрібна комерційна СУБД з широкими можливостями та підтримкою від Microsoft.

Отже, правильний вибір СУБД має важливе значення для успішної розробки та експлуатації інформаційної системи вантажоперевезень.

2.2 Обґрунтування вибору методології та технології проектування

На сьогоднішній день широко використовуються технології проектування інформаційної системи компанії та її компонентів на основі:

- Процесного підходу: розбивка роботи на етапи з чітко визначеними завданнями та результатами.
- Спіральної моделі життєвого циклу інформаційних систем: ітеративний підхід, який передбачає повторне проходження етапів проектування з поступовим уточненням вимог та покращенням системи.

Ця модель життєвого циклу інформаційних систем складається з наступних етапів:

1. Аналіз проблеми: чітке визначення проблеми, яку має вирішити інформаційна система.
2. Проектування інформаційної системи: розробка детального плану та специфікацій для інформаційної системи.
3. Реалізація інформаційної системи: створення програмного забезпечення та інших компонентів інформаційної системи.
4. Впровадження ІС: розгортання ІС та навчання користувачів.

5. Супровід інформаційної системи: підтримка та вдосконалення інформаційної системи протягом її життєвого циклу.

Аналіз компанії

В основі аналізу компанії лежить поняття бізнес-моделі ІС, яка базується на наступних рівнях опису системи:

1. Концептуальний рівень: змістовне опису ІС на основі структурного підходу.
2. Логічний рівень: формалізоване опису ІС на основі об'єктно-орієнтованого підходу.
3. Фізичний рівень: програмно-апаратна реалізація ІС.

Декомпозиція системи

Різні принципи декомпозиції є основою для концептуального моделювання. Розглянемо основні підходи:

1. Структурний підхід: декомпозиція системи на функції, які необхідно автоматизувати, і розбивка їх на конкретні завдання та процедури.
2. Об'єктно-орієнтований підхід: представляє систему у вигляді взаємодіючих об'єктів.

Методології проектування

При структурному аналізі використовують наступні методології:

- SADT-методологія: основа методології IDEF0, призначена для побудови функціональної моделі предметної області, що відображає функціональну структуру об'єкта моделювання, операції, що виконуються системою, та зв'язки між ними.
- DFD-методологія: визначає модель системи як ієрархію діаграм потоків даних, які описують процеси перетворення вхідної інформації у вихідну.
- IDEF3-методологія: механізм документування та збору інформації про процеси в системі.

Для аналізу інформаційної моделі компанії з вантажоперевезень будемо використовувати IDEF0-модель, яка дозволяє проводити об'єктивний аналіз бізнес-процесів, що здійснюються в компанії з вантажоперевезень.

Для розглянутої в дипломній роботі предметної області, а саме інформаційної системи автоматизації обліку діяльності транспортного підприємства з перевезення вантажів, розроблено ієрархію діаграм потоків даних використовуючи нотацію Гейна Сарсона (див. Рис. 8).



Рисунок 8 — DFD для інформаційної системи автоматизації обліку діяльності транспортного підприємства з перевезення вантажів

Слід зазначити, що DFD-методологія в основному використовується для опису руху документів у компанії з вантажоперевезень. Для прикладу наведено DFD системи обліку складу (див. Рис. 9). Для створення діаграми потоків даних виділено основних акторів, основні процеси і основні сховища даних. Головними акторами буде сам Клієнт, Особа, що доставляє замовлення, і також працівник складу. Основним сховищем даних буде База Даних зі всіма замовленнями. Основний процесом буде система обліку складу. Користувач зможе зробити заказ, та отримувати оновлення про цей заказ. Система обліку повинна надавати користувачу оновлення заказу, оновлювати його у базі

даних. Працівник доставки повинен отримати заказ на доставку від системи обліку, і надавати апдейти щодо доставки. А робітник складу повинен підтвердити прибуття товару.

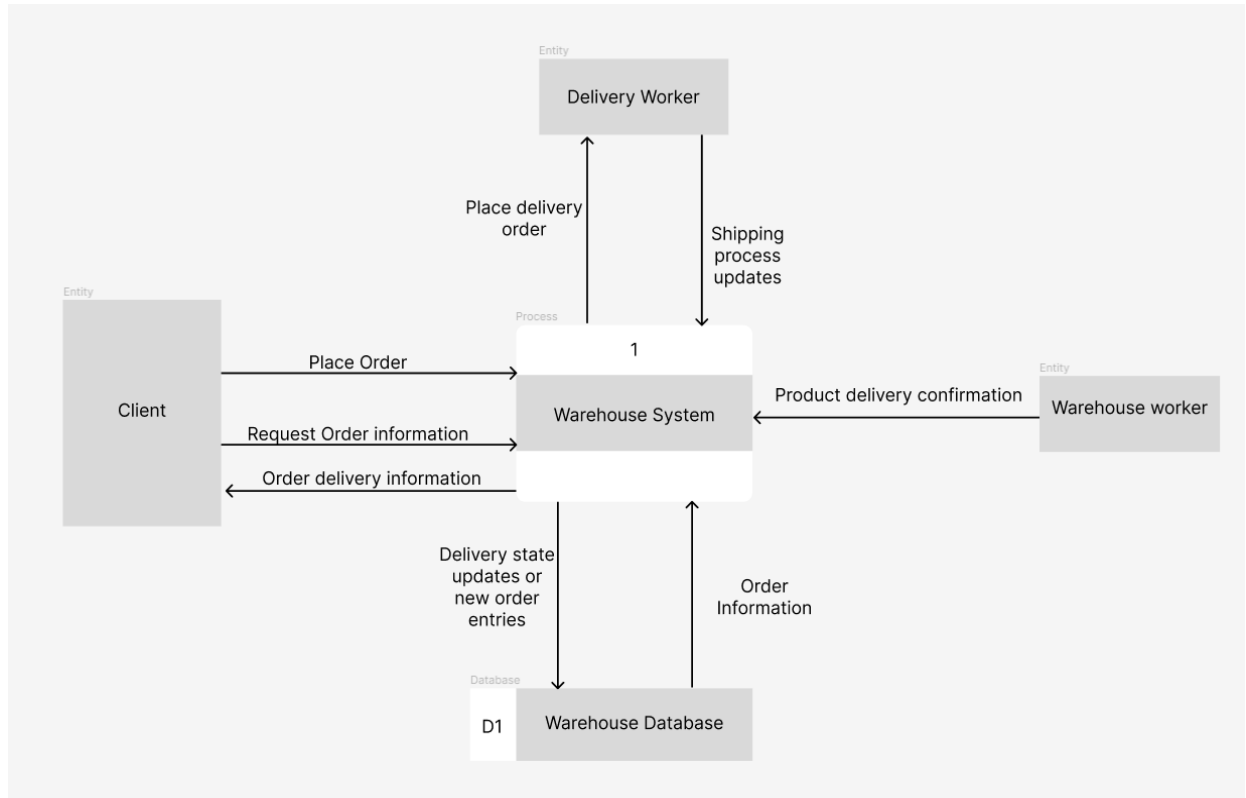


Рисунок 9 — DFD системи обліку складу

Це перший рівень діаграми потоків даних. Розширимо її далі. Розіб'ємо процес обліку на: процес формування заказу, систему управління БД, систему управління процесом доставки, і також на веб-інтерфейс, в якому користувач може отримати оновлення щодо стану заказу. Тепер користувач створює заказ через процес створення заказу, що надсилає заказ системі управління БД. Ця система вносить заказ до бази даних, і створює заказ на доставку, який вона передає процесу управління процесом доставки. Цей процес отримує список вільних працівників з БД працівників, обирає певного працівника, модифікує його в базі даних (він вже не вільний), і передає йому інформацію про доставку. Він передає назад інформацію про стан доставки. Система управління процесом доставки передає ці дані в систему управління

центральною базою даних, що вносить зміни до бази даних складу, і надсилає інформацію на веб-сервіс, який вже передає інформацію замовнику. Працівник складу передає інформацію про отримання замовлення до системи управління центральною базою даних, що вносить зміни до БД складу, і, знову, надсилає нову інформацію до веб-застосунку (див. Рис. 10).

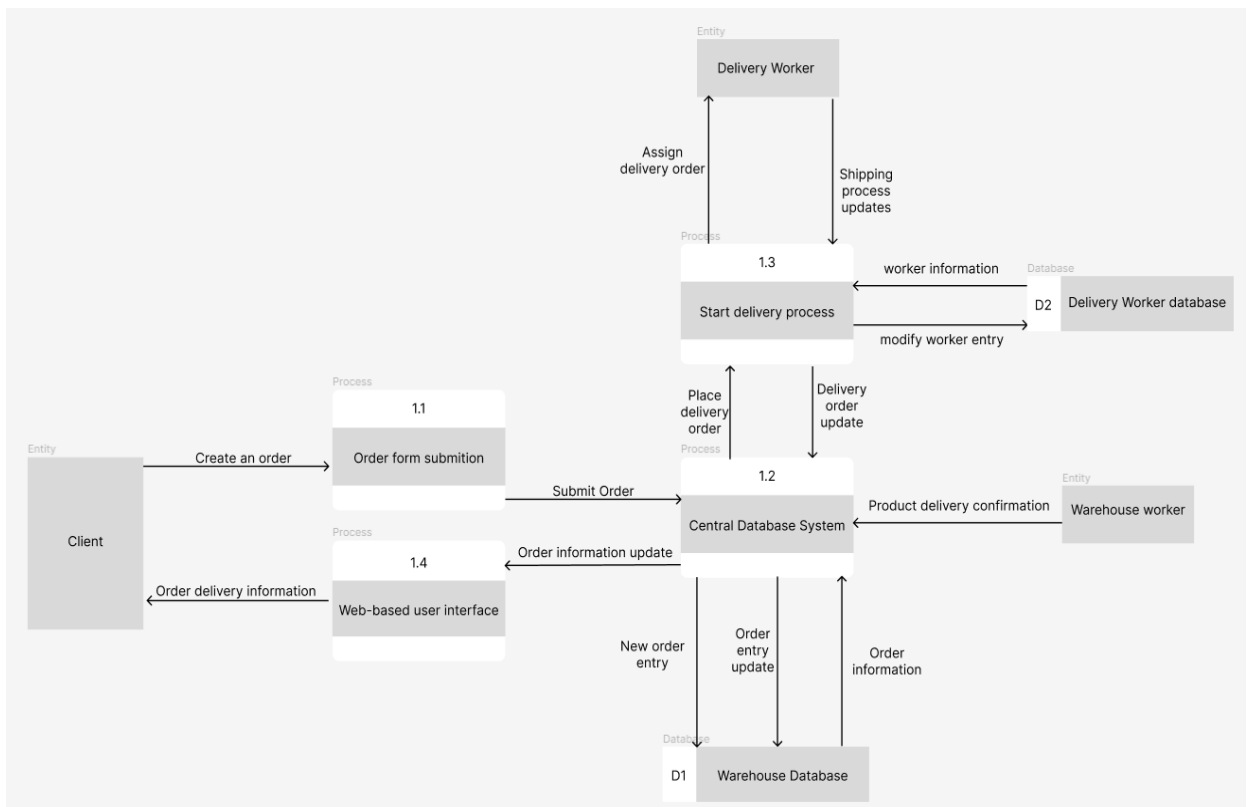


Рисунок 10 — DFD другого рівня

2.3 Мова програмування C#

C# — це гнучка та потужна мова програмування, розроблена компанією Microsoft для платформи .NET. Її простота у вивченні поєднується з широким спектром можливостей, що робить її популярним вибором для створення різноманітних програмних рішень. До ключових характеристик C# можна віднести:

- C# полегшує моделювання складних систем та організацію коду, ґрунтуючись на принципах ООП;

- завдяки жорсткій системі типізації C# допомагає уникнути помилок під час розробки та робить код більш надійним;
- C# має чіткий та зрозумілий синтаксис, схожий на C++ і Java, що робить її доступною для початківців;
- C# підтримує широкий спектр функцій, включаючи поліморфізм, перевантаження операторів, делегати, події та атрибути;
- C# тісно інтегрована з платформою .NET, що надає доступ до великої бібліотеки класів та потужних інструментів розробки.

Серед переваг використання C# виділяють ключові моменти, а саме:

- C# використовується для створення веб-сайтів, десктопних програм, мобільних додатків, ігор та багато іншого.
- C# має велику та активну спільноту розробників, що робить доступним широкий спектр ресурсів та підтримки;
- C# офіційно підтримується компанією Microsoft, що гарантує її довгостроковий розвиток та сумісність з новими технологіями.

Система Common Language Runtime (CLR)

CLR (Common Language Runtime), або «загальне середовище виконання мов», є ключовим компонентом платформи .NET Framework. Це віртуальна машина, на якій виконуються всі мови, що підтримуються .NET, включаючи C#.

C# тісно інтегрована з CLR, і багато її особливостей залежать від можливостей CLR, а саме:

- CLR визначає правила типізації для .NET, і C# використовує ці правила для забезпечення безпеки та надійності коду;
- CLR надає C# доступ до таких можливостей, як збірка сміття, винятки та безпека.

CLR є фундаментальною частиною платформи .NET Framework, і вона відіграє важливу роль у роботі C#. Завдяки CLR, C# пропонує простоту розробки, безпеку, надійність, продуктивність та широкий спектр

можливостей, що робить її популярним вибором для розробників програмного забезпечення.

2.4 Технологія для доступу до баз даних Entity Framework Core

Entity Framework Core (EF Core) — це об'єктно-орієнтована технологія від Microsoft для доступу до даних, що вирізняється легкістю, відкритістю та можливістю розширення [5]. EF Core є ORM-інструментом (Object-Relational Mapping — відображення об'єктів на реляційні дані). Це означає, що EF Core дозволяє працювати з базами даних, але надає більш високий рівень абстракції:

- Можна абстрагуватися від самої бази даних та її таблиць.
- Працювати з даними незалежно від типу сховища.

На відміну від фізичного рівня, де оперують таблицями, індексами, первинними та зовнішніми ключами, на концептуальному рівні, який пропонує EF Core, працюють з об'єктами (див. Рис. 11).

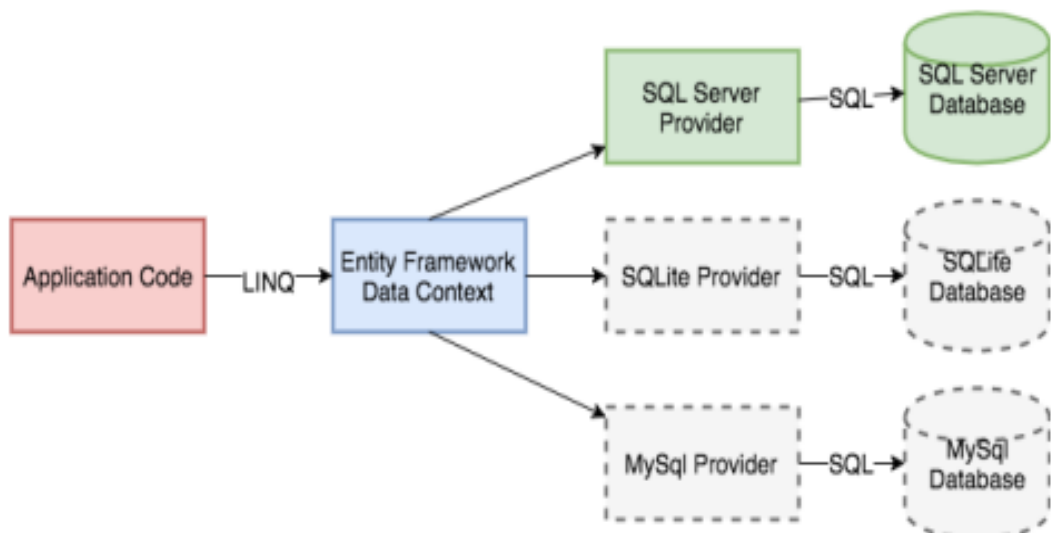


Рисунок 11 — Структура роботи технології EF Core

До переваг EF Core можна віднести наступне:

- Підтримка множини СУБД: EF Core дає можливість працювати з будь-якою СУБД, якщо для неї існує відповідний провайдер [6]. Microsoft надає ряд вбудованих провайдерів: для MS SQL Server, SQLite, PostgreSQL. Крім того, існують провайдери від сторонніх постачальників, наприклад, для MySQL.
- Легкість використання: EF Core має простий та зрозумілий API, що робить його зручним для розробників.
- Відкритість: EF Core є відкритим проектом, що гарантує його прозорість та можливість доопрацювання.
- Можливість розширення: EF Core можна розширювати за допомогою додаткових пакетів та інструментів.

Вибір EF Core для розробки програмного продукту:

Враховуючи поставлене завдання та методи його вирішення, було вирішено розробити програмний продукт у вигляді десктопного додатку з використанням технології Entity Framework Core для роботи з ним.

Обґрунтування:

- EF Core відповідає вимогам ORM-інструменту, а саме: легкість використання, відкритість, можливість розширення.
- EF Core підтримує СУБД Microsoft SQL Server 2018: ця СУБД обрана для створення інформаційної системи керування туристичним комплексом.
- EF Core дозволяє абстрагуватися від фізичного рівня бази даних, що спрощує розробку та робить код більш стійким до змін.

Отже, вибір інструментів та технологій є ключовим фактором при розробці програмного забезпечення. Entity Framework Core, як потужний та зручний ORM-інструмент, чудово підходить для створення програмного продукту, що відповідає поставленим вимогам. У цьому випадку, для розробки інформаційної системи транспортної компанії було обрано СУБД Microsoft SQL Server 2018.

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ТРАНСПОРТНОЇ КОМПАНІЇ

3.1 Опис предметної області

В умовах жорсткої конкуренції на ринку транспортно-логістичних послуг, компанії змушені постійно вдосконалювати свою роботу. Це особливо актуально з огляду на розгалужену транспортну систему країни, яка робить планування маршрутів перевезень складним завданням.

Для забезпечення високого рівня сервісу, оперативності та якості доставки логістичній компанії необхідно автоматизувати всі процеси. Саме тому виникає потреба у спеціалізованій інформаційній системі (ІС), яка допоможе оптимізувати всі ланки логістичного ланцюжка. Наразі існує безліч програм для транспортних компаній, але не всі з них здатні забезпечити необхідну функціональність.

Логістична компанія отримує велику кількість заявок від постійних клієнтів та разових замовників. З постійними клієнтами також ведуться довгострокові проекти. Для ефективного управління всіма цими процесами необхідна інформаційна система, яка:

- Дозволить менеджеру зручно працювати з великою кількістю інформації.

- Забезпечить ефективне управління процесами вантажоперевезень.

Процес управління вантажоперевезеннями включає:

- Обробку заявок клієнтів.
- Планування маршрутів.
- Планування роботи водіїв та автотранспорту.
- Планування консолідації вантажів (об'єднання вантажів).
- Складання звітів.

Всі ці процеси повинні бути автоматизовані за допомогою ІС, яка може бути розбита на модулі:

- Управління вантажоперевезеннями: планування рейсів, планування роботи водіїв та автотранспорту, контроль доставки.
- Звітність: аналіз ефективності роботи компанії з усіх боків, складання звітів різних видів.

Впровадження спеціалізованої ІС дозволить транспортній компанії: підвищити ефективність роботи, знизити витрати, покращити якість послуг, збільшити конкурентоспроможність на ринку.

3.2 Архітектура системи

Інформаційна система ведення діяльності транспортної компанії представлена в даній роботі, підрозділяється на дві частини:

- серверна база даних;
- клієнт – підсистема, яка безпосередньо відповідає за ведення діяльності транспортної компанії, що дозволяє документувати дані про співробітників та їх посади, парк автомобілів та їх маршрути, контрагентів, договорів, точок маршруту.

Діаграма розгортання (див. Рис. 12) наочно демонструє фізичне розташування компонентів інформаційної системи транспортної компанії:

- Сервер: розташований на окремому комп'ютері, який слугує центральним сховищем даних для всієї системи.
- Клієнтські машини: встановлені на робочих станціях користувачів, що дозволяють їм отримувати доступ до даних та керувати функціоналом системи.
- Локальна обчислювальна мережа (ЛОМ): з'єднує сервер та клієнтські машини, забезпечуючи їм можливість обмінюватися даними.

Завдяки ЛОМ:

- Користувачі можуть отримувати доступ до актуальної інформації з сервера.

- Зміни, внесені користувачами на своїх клієнтських машинах, синхронізуються з сервером.
- Система працює злагоджено та ефективно.

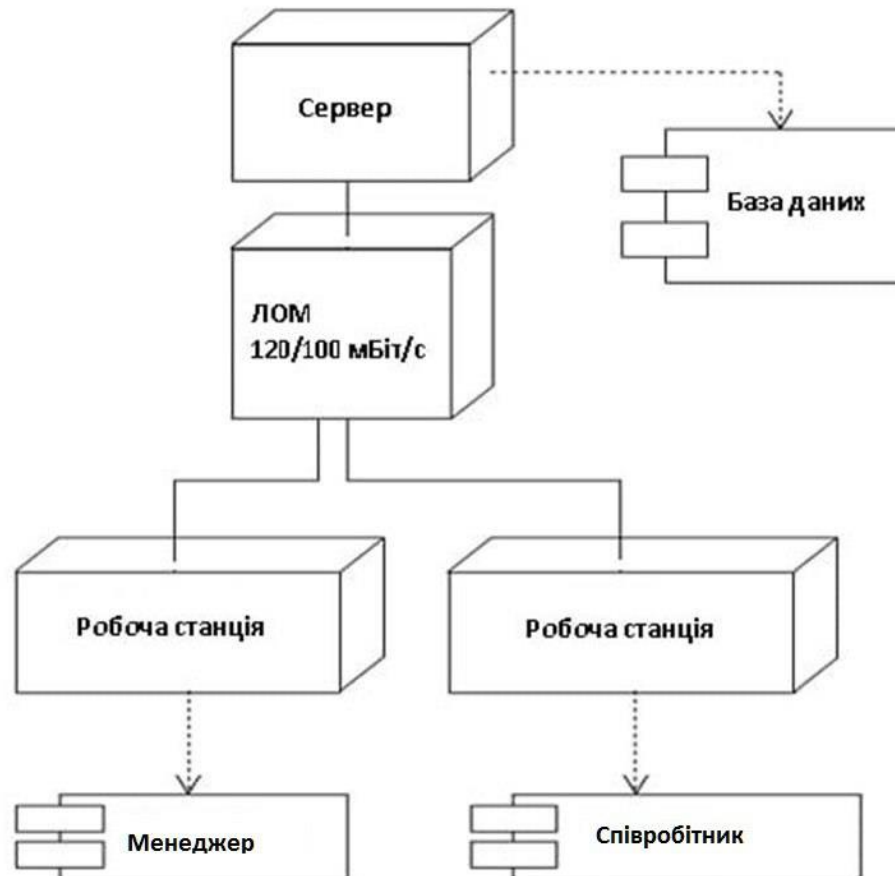


Рисунок 12 — Діаграма розгортання системи

3.3 Функціональні вимоги системи

Згідно з архітектурою, система поділяється на дві частини:

- Серверна частина, яка зберігає всі дані та забезпечує доступ до них.
- Клієнтська частина, що надає користувацькому інтерфейс для роботи з системою.

Можна виділити наступні функціональні вимоги до бази даних транспортної компанії:

1. Авторизація та доступ до системи:

- Користувачі (співробітники) повинні мати можливість створювати облікові записи з унікальними логінами та паролями.

- Система повинна мати різні рівні доступу для різних груп користувачів (наприклад, адміністратори, диспетчери, водії).

- Користувачі повинні мати можливість змінювати свої паролі та відновлювати їх у разі втрати.

2. Управління співробітниками:

- Система повинна дозволяти додавати, редагувати та видаляти інформацію про співробітників (ПІБ, ПІН, дата народження, стать).

- Система повинна зберігати історію посад, які займав кожен співробітник (дата призначення, дата звільнення, назва посади).

3. Управління автопарком:

- Система повинна дозволяти додавати, редагувати та видаляти інформацію про транспортні засоби (марка, модель, рік випуску, тип, номерний знак).

- Система повинна зберігати інформацію про поточний стан транспортних засобів (на ходу, в ремонті, на ТО).

4. Планування маршрутів:

- Система повинна дозволяти диспетчерам створювати та редагувати маршрути перевезень, включаючи: ID автотранспортного засобу; ID водія; дату та час початку та закінчення рейсу; список адрес (пунктів навантаження/розвантаження) з порядковим номером.

5. Управління контрагентами:

- Система повинна дозволяти додавати, редагувати та видаляти інформацію про контрагентів (назва, ПІН, додатковий статус).

6. Управління адресами:

- Система повинна дозволяти додавати, редагувати та видаляти інформацію про адреси (графство, місто, вулиця, номер будинку, номер квартири).

7. Система повинна дозволяти створювати та редагувати договори перевезень, включаючи: ID контрагента-відправника; ID контрагента-отримувача; ID адреси отримання; ID адреси доставки; суму договору; вагу, висоту, ширину та глибину вантажу.

8. Система повинна дозволяти диспетчерам відстежувати статус рейсів у режимі реального часу (на ходу, очікування завантаження/розвантаження, закінчено).

- Система повинна генерувати звіти про виконані рейси, включаючи: дату та час рейсу; маршрут; вантаж; водія; витрачений час.

9. Система повинна генерувати різні звіти за запитом, наприклад: звіти про виконані рейси; звіти про завантаженість автопарку; звіти про роботу водіїв.

3.4 Проектування інформаційної системи

На етапі проектування системи було проведено структурно-функціональне моделювання. Це дозволило детально проаналізувати всі бізнес-процеси, які буде підтримувати система, та визначити можливості для покращення роботи [10].

Головною метою проектування є створення ефективної інформаційної системи, яка буде відповідати всім поставленим вимогам, враховуючи наявні обмеження та технологічні можливості [12].

Для визначення основних функціональних можливостей системи були створені діаграми варіантів використання для різних типів користувачів (акторів). Ці діаграми наочно демонструють дії, які може виконувати кожен користувач в системі [14]. Наведемо приклади Use Case для деяких функціональних вимог.

Функціонал доступу до даних: Система має дозволяти авторизованим як співробітники компанії користувачам отримувати потрібні дані про

замовлення, вантажі, клієнтів та маршрути за запитом за умови їх наявності у базі даних (див. Рис. 13).

Елемент	Деталі
ID	0001
Назва Use Case	Доступ до даних
Ініціатор	Співробітник компанії
Учасники	Співробітник компанії, система
Передумови	Співробітник компанії авторизований у системі.
Основний потік подій	<ol style="list-style-type: none"> 1. Співробітник компанії отримує за запитом всі дані про замовлення 2. Співробітник компанії отримує за запитом всі дані про вантажі 3. Співробітник компанії отримує за запитом всі дані про клієнтів 4. Співробітник компанії отримує за запитом всі дані про маршрути
Альтернативні потоки (опціонально)	<ol style="list-style-type: none"> 1а. Співробітник компанії не отримує за запитом всі дані про замовлення, якщо такого замовлення не існує 2а. Співробітник компанії не отримує за запитом всі дані про вантажі, якщо таких вантажів не існує 3а. Співробітник компанії не отримує за запитом всі дані про клієнтів, якщо таких клієнтів не існує 4а. Співробітник компанії не отримує за запитом всі дані про маршрути, якщо таких маршрутів не існує
Постумови	Співробітник отримав потрібну інформацію
Винятки (опціонально)	<ol style="list-style-type: none"> 1. Помилки з'єднання з сервером під час виконання запиту 2. Намагання отримати інформацію про неіснуючі замовлення, вантажі, клієнтів та маршрути

Приклад коду для побудови Use Case «Функціонал доступу до даних» для Plant UML наведено в лістингу 1.

Лістинг 1 Код для побудови Use Case «Функціонал доступу до даних» для Plant UML

```
@startuml
left to right direction
```

```

actor "Співробітник компанії" as user
rectangle "Система" {
  usecase "Логін в систему" as UC1
  usecase "Доступ до бази даних" as UC2
  usecase "Отримає дані про вантаж" as UC3
  usecase "Отримає дані про клієнта" as UC4
  usecase "Отримає дані про маршрут" as UC5
  usecase "Отримає дані про замовлення" as UC6
  usecase "Перегляд повідомлення про\nвідсутність таких
даных" as UC7
}
user --> UC1 : Відкриває додаток\nі логіниться
user --> UC2 : Переходить до 'Дані'
user --> UC3 : Запитує дані про вантаж
user --> UC4 : Запитує дані про клієнта
user --> UC5 : Запитує дані про маршрут
user --> UC6 : Запитує дані про замовлення
user -[#red]> UC7 : Намагається отримати\nнеіснуючі дані
UC3 .[#green]> UC7 : Виняток
UC4 .[#green]> UC7 : Виняток
@enduml

```

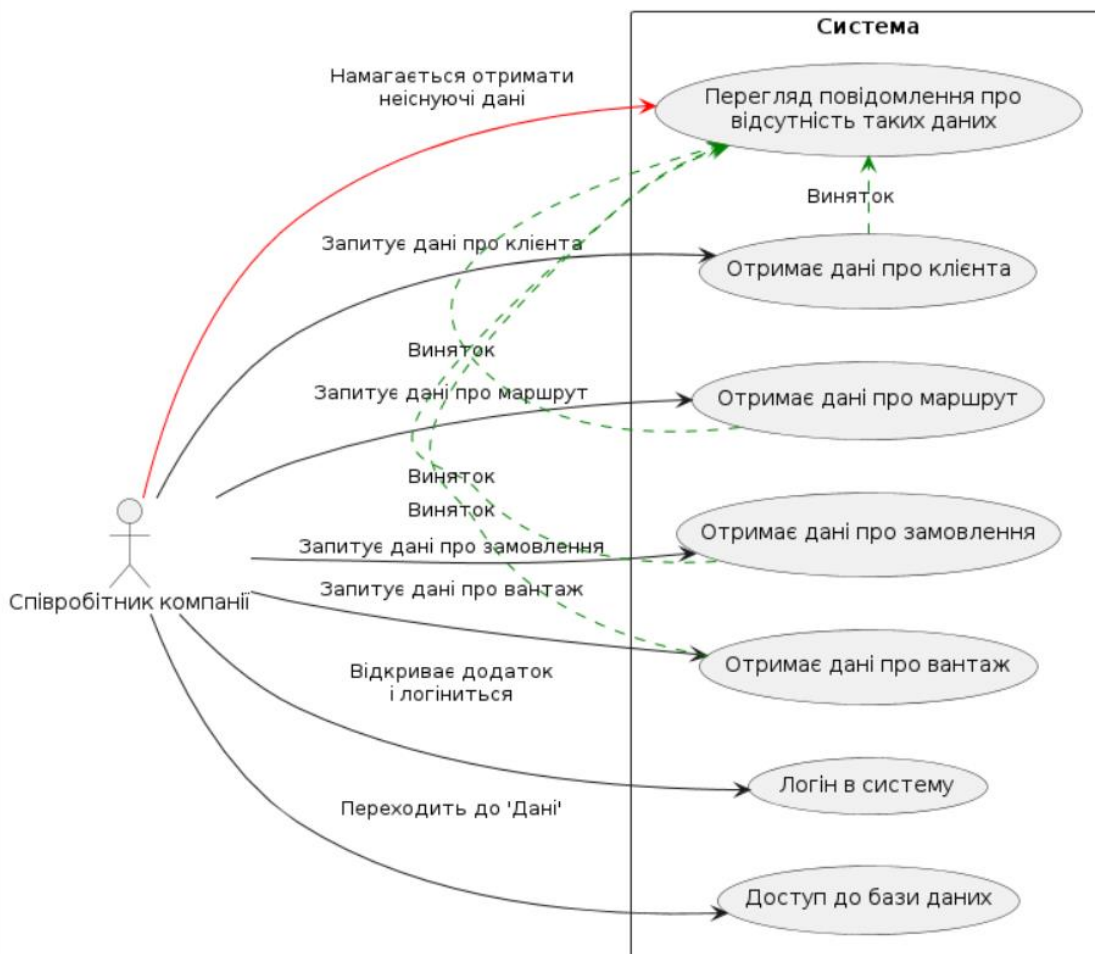


Рисунок 13 — Use Case Функціонал доступу до даних

Функціонал автоматизації процесів: Система має дозволяти авторизованим як співробітники компанії користувачам введення даних, генерування звітів та планування маршрутів, відповідно до їх прав за посадою (див. Рис. 14).

Елемент	Деталі
ID	0002
Назва Use Case	Автоматизація процесів
Ініціатор	Співробітник компанії
Учасники	Співробітник компанії, система
Передумови	Співробітник компанії авторизований у системі.
Основний потік подій	<ol style="list-style-type: none"> 1. Співробітник компанії може вводити дані 2. Співробітник компанії може генерувати звіти 3. Співробітник компанії може планувати маршрути
Альтернативні потоки (опціонально)	1,2,3а. Співробітник компанії не може виконати дію якщо його посада не передбачає таких прав
Постумови	Співробітник виконав дію, чим зекономив свій час
Винятки (опціонально)	<ol style="list-style-type: none"> 1. Помилки з'єднання з сервером під час виконання запиту 2. Намагання виконати дію що не передбачена посадою

Приклад коду для побудови Use Case «Функціонал автоматизації процесів» для Plant UML наведено в лістингу 2.

Лістинг 2 Код для побудови Use Case «Функціонал автоматизації процесів» для Plant UML

```
@startuml
left to right direction
actor "Співробітник компанії" as user
rectangle "Система" {
    usecase "Логін в систему" as UC1
    usecase "Вводить дані" as UC2
}
```

```

usecase "Генерує звіт" as UC3
usecase "Планує маршрут" as UC4
usecase "Перегляд повідомлення про\nвідсутність
прав на такі дії" as UC5
}
user --> UC1 : Відкриває додаток\nі логіниться як
співробітник
user --> UC2 : Переходить до 'ввести дані'
user --> UC3 : Переходить до 'генерувати звіт'
user --> UC4 : Переходить до 'запланувати маршрут'
user -[#red]> UC5 : Намагається виконати\nнедозволену
дію
UC2 .[#green]> UC5 : Виняток
UC3 .[#green]> UC5 : Виняток
UC4 .[#green]> UC5 : Виняток
@enduml

```

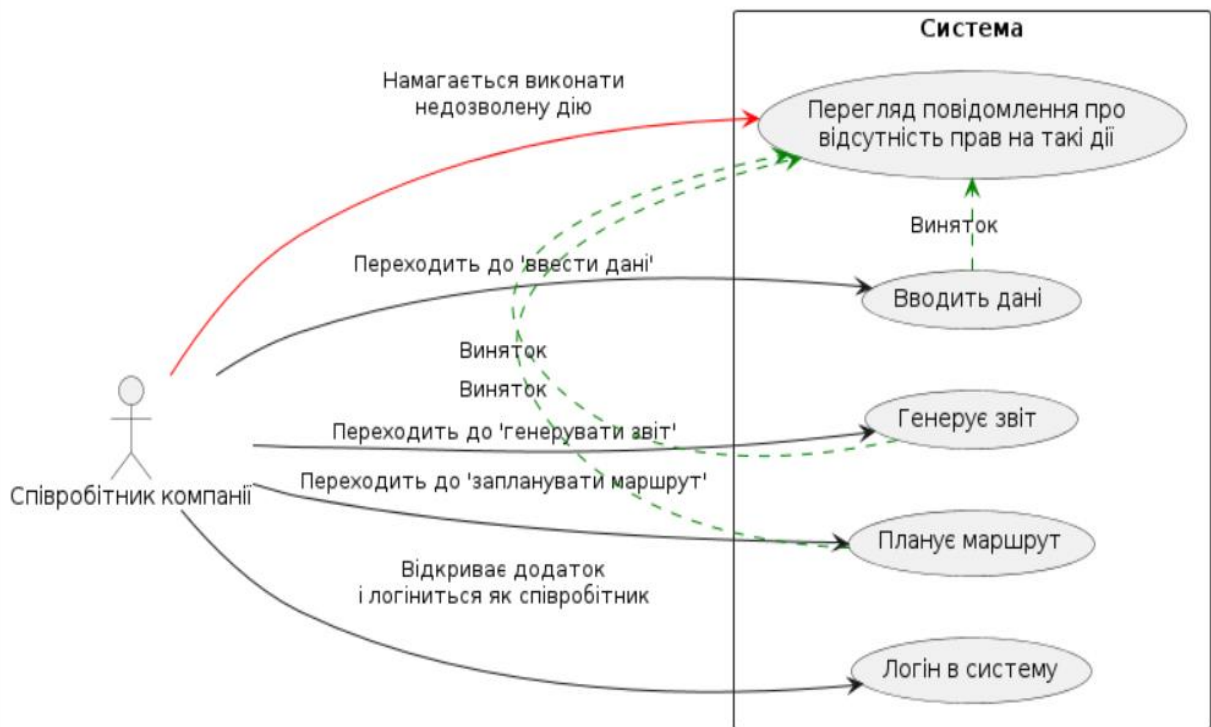


Рисунок 14 — Use Case Функціонал автоматизації процесів

Під час проектування бази даних використовувалася модель «сутність-зв'язок» (Entity-Relationship Diagram, ERD). Ця модель дозволяє візуально зобразити зв'язки між різними об'єктами (сутностями), які зберігаються в базі даних.

Виокремлення сутностей, їх характеристик (атрибутів) та опис зв'язків між ними дає чітке уявлення про логічну структуру бази даних [11].

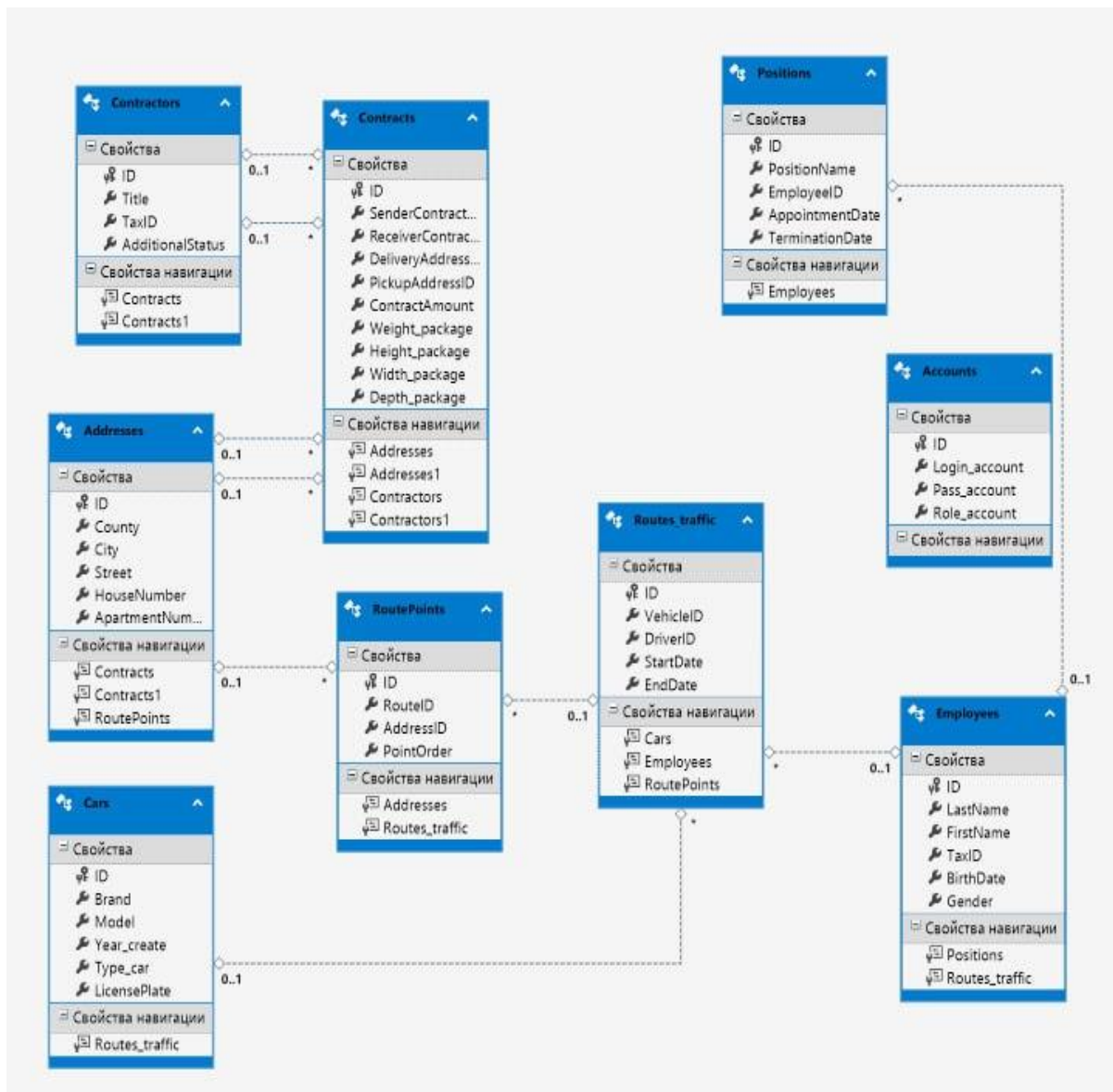


Рисунок 15 — ER-діаграма бази даних

Для опису моделі предметної області дипломної роботи наведено ER-діаграму бази даних транспортної компанії (див. Рис.15), яка візуально представляє зв'язки між різними об'єктами (сутностями) в базі даних, а також структуру таблиць реляційної бази даних, яка містять детальну інформацію про кожну сутність, включаючи її атрибути та зв'язки з іншими сутностями.

На рисунку 15 зображені зв'язки між виділеними сутностями. Ця діаграма допомагає чітко зрозуміти, як різні об'єкти в базі даних пов'язані між собою.

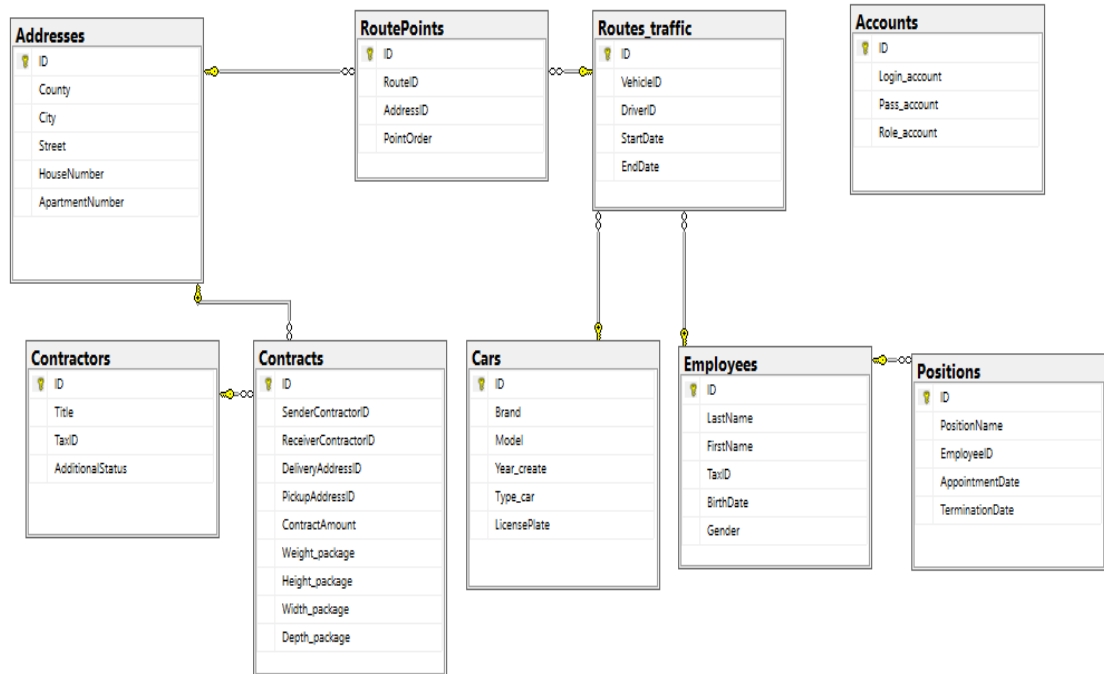


Рисунок 16 — Схема бази даних

База даних транспортної компанії складається з 9 таблиць (див. Рис. 16). Кожна таблиця відповідає певній сутності. Для кожної сутності в базі даних є первинний ключ. Це поле, яке однозначно ідентифікує кожен запис у таблиці. Використання первинних ключів забезпечує унікальність записів у базі даних та полегшує зв'язування таблиць між собою.

Сутність «Співробітники» містить інформацію про співробітників, що працюють в компанії, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 1.

Сутність «Посади» містить інформацію про посади співробітників, які працюють в компанії, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 2.

Таблиця 1 — Атрибути сутності «Employees»

Назва	Тип	Зв'язок	Опис
Id	int	-	Первинний ключ
LastName	nvarchar(50)	-	Прізвище
FirstName	nvarchar(50)	-	Ім'я
TaxID	nvarchar(20)	-	Ідентифікатор податку
BirthDate	date	-	Дата народження
Gender	nvarchar(10)	-	Стать

Таблиця 2 — Атрибути сутності «Positions»

Назва	Тип	Зв'язок	Опис
Id	int	-	Первинний ключ
PositionName	nvarchar (100)	-	Назва посади
EmployeeID	int	Employees	ID_співробітника
AppointmentDate	date		Дані призначення
TerminationDate	date		Дані звільнення

Сутність «Автомобілі» призначена для зберігання інформації про парк автомобілів, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 3.

Таблиця 3 — Атрибути сутності «Cars»

Назва	Тип	Зв'язок	Опис
Id	int	-	Первинний ключ
Brand	nvarchar (50)	-	Марка
Model	nvarchar (50)	-	Модель
Year_create	int	-	Рік випуску
Type_car	nvarchar (50)	-	Тип
LicensePlate	nvarchar (20)	-	Державний номер

Сутність «Маршрути» містить інформацію про рейси водіїв, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 4.

Таблиця 4 — Атрибути сутності «Routes_traffic»

Назва	Тип	Зв'язок	Опис
Id	int	-	Первинний ключ
VehicleID	int	Cars	ID_авто
DriverID	int	Employees	ID_водія
StartDate	datetime	-	Дата початку рейсу
EndDate	datetime	-	Дата закінчення рейсу

Сутність «Підрядники» містить інформацію про підрядників компанії, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 5.

Таблиця 5 — Атрибути сутності «Contractors»

Назва	Тип	Зв'язок	Опис
Id	int	-	Первинний ключ
Title	nvarchar (100)	-	Назва
TaxID	nvarchar (20)	-	Державний ідентифікатор
AdditionalStatus	nvarchar (100)	-	Додатковий статус

Сутність «Адреси» містить інформацію про адреси маршрутів, відображена в базі даних відповідною таблицею, містить атрибути, зазначені в таблиці 6.

Сутність «Договори» містить інформацію про договори між підрядниками замовниками та підрядниками отримувачами в якому вказуються адреси, сума договору та габарити вантажу (див. Табл.7).

Таблиця 6 — Атрибути сутності «Addresses»

Назва	Тип	Зв'язок	Опис
Id	int	-	Первинний ключ
County	nvarchar (100)	-	Графство
City	nvarchar (100)	-	Місто
Street	nvarchar (100)	-	Вулиця
HouseNumber	nvarchar (20)	-	Будинок
ApartmentNumber	nvarchar (20)	-	Квартира

Таблиця 7 — Атрибути сутності «Contracts»

Назва	Тип	Зв'язок	Опис
Id	int	-	Первинний ключ
SenderContractorID	nvarchar(255)	Contractors	ID_контрагента-відправника
ReceiverContractorID	int	Contractors	ID_контрагента-отримувача
DeliveryAddressID	int	Addresses	ID_адреси_отримання
PickupAddressID	int	Addresses	ID_адреси_доставки
ContractAmount	decimal(18, 2)	-	Сума договору
Weight_package	decimal(18, 2)	-	Вага
Height_package	decimal(18, 2)	-	Висота
Width_package	decimal(18, 2)	-	Ширина
Depth_package	decimal(18, 2)	-	Глибина

Таблиця 8 — Атрибути сутності «RoutePoints»

Назва	Тип	Зв'язок	Опис
Id	int	-	Первинний ключ
RouteID	int	Routes_traffic	ID_маршруту
AddressID	int	Addresses	ID_адреси
PointOrder	int	-	Порядковий номер точки

3.5 Програмна реалізація застосунку

Кожній таблиці в базі даних присвоюється свій власний номер за допомогою enum `TablesTypeEnum`. Потім, за допомогою кортежу `Tuple` їй присвоюється і назва всередині програми. Таким чином, можна через номер таблиці отримати її назву, приклад для виведення на заголовок вікна наведено у лістингу 3.

Лістинг 3 Виведення на заголовок вікна назви таблиці через номер

```
public enum TablesTypeEnum
{
    Addresses= 1,
    Cars = 2,
    Contractors = 3,
    Contracts = 4,
    Employees = 5,
    Positions = 6,
    RoutePoints = 7,
    Routes_traffic = 8,
    Accounts = 9
}
public class TablesTypes
{
    public static Tuple<int, string> Addresses { get; } =
new Tuple<int, string>(1, "Addresses");
    public static Tuple<int, string> Cars { get; } = new
Tuple<int, string>(2, "Cars");
    public static Tuple<int, string> Contractors { get; }
= new Tuple<int, string>(3, "Contractors");
    public static Tuple<int, string> Contracts { get; } =
new Tuple<int, string>(4, "Contracts");
    public static Tuple<int, string> Employees { get; } =
new Tuple<int, string>(5, "Employees");
    public static Tuple<int, string> Positions { get; } =
new Tuple<int, string>(6, "Positions");
    public static Tuple<int, string> RoutePoints { get; }
= new Tuple<int, string>(7, "RoutePoints");
    public static Tuple<int, string> Routes_traffic { get;
} = new Tuple<int, string>(8, "Routes_traffic");
    public static Tuple<int, string> Accounts { get; } =
new Tuple<int, string>(9, "Accounts");
```

```

public static string GetTableName(int number)
{
    switch ((TablesTypeEnum) number)
    {
        case TablesTypeEnum.Addresses:
            return TablesTypes.Addresses.Item2;
        case TablesTypeEnum.Cars:
            return TablesTypes.Cars.Item2;
        case TablesTypeEnum.Contractors:
            return TablesTypes.Contractors.Item2;
        case TablesTypeEnum.Contracts:
            return TablesTypes.Contracts.Item2;
        case TablesTypeEnum.Employees:
            return TablesTypes.Employees.Item2;
        case TablesTypeEnum.Positions:
            return TablesTypes.Positions.Item2;
        case TablesTypeEnum.RoutePoints:
            return TablesTypes.RoutePoints.Item2;
        case TablesTypeEnum.Routes_traffic:
            return TablesTypes.Routes_traffic.Item2;
        case TablesTypeEnum.Accounts:
            return TablesTypes.Accounts.Item2;
        default:
            return TablesTypes.Employees.Item2;
    }
}

```

Коли ми натискаємо кнопку видалення, нас перекидає у цей метод. Він аналізує, який тут номер таблиці, знаходить відповідний запис і видалляє його. Реалізація наведена у лістингу 4.

Лістинг 4 Видалення відповідного запису

```

public bool DeleteRows(int tableNumber, List<int>
recordsIds)
{
    try
    {
        switch ((TablesTypeEnum) tableNumber)
        {
            case TablesTypeEnum.Addresses:
                foreach (var record in recordsIds)
                {
                    entities.Addresses.Remove(entities.Addresses.Find(record));
                }
            }
        }
    }
}

```

```

        }
        entities.SaveChanges();
        return true;
    case TablesTypeEnum.Cars:
        foreach (var record in recordsIds)
        {
            entities.Cars.Remove(entities.Cars.Find(record));
        }
        entities.SaveChanges();
        return true;

```

Користувач може отримати всіх робітників із бази даних, конкретного робітника за id, додавати робітників, оновлювати робітників. Реалізація наведена у лістингу 5.

Лістинг 5 Отримати даних про робітників із бази даних

```

public BindingSource GetEmployes()
{
    BindingSource bs = new BindingSource();
    var query = from obj in entities.Employees
                select new
                {
                    Id = obj.ID,
                    LastName = obj.LastName,
                    FirstName = obj.FirstName,
                    TaxID = obj.TaxID,
                    BirthDate = obj.BirthDate,
                    Gender = obj.Gender
                };
    bs.DataSource = query.ToList();
    return bs;
}

public Employees GetEmployees(int id)
{
    return entities.Employees.Find(id);
}

public bool AddEmployees(Employees AW)
{
    try
    {
        entities.Employees.Add(AW);
    }
}

```



```

        entities.SaveChanges();
        return true;
    }
    catch (Exception e)
    {
        MessageBox.Show("Помилка зв'язаності таблиць.",
e.Message); return false;
    }
}

public bool UpdateEmployees(Employees AW)
{
    try
    {
        var record =
entities.Employees.SingleOrDefault(rec => rec.ID == AW.ID);

        if (record != null)
        {
            record.LastName = AW.LastName;
            record.FirstName = AW.FirstName;
            record.TaxID = AW.TaxID;
            record.BirthDate = AW.BirthDate;
            record.Gender = AW.Gender;
            entities.SaveChanges();
            return true;
        }
        else
            return false;
    }
    catch (Exception e)
    {
        MessageBox.Show("Помилка зв'язаності таблиць.",
e.Message); return false;
    }
}
}

```

Копіювання властивостей із бази даних для того, щоб вивести їх на форму для оновлення (операція Update). Реалізація наведена у лістингу 6.

Лістинг 6 Копіювання властивостей із бази даних

```

private void CopyEntity(Employees newValue)
{
    employees.ID = newValue.ID;
    employees.LastName = newValue.LastName;
    employees.FirstName = newValue.FirstName;
    employees.TaxID = newValue.TaxID;
}

```

```

        employees.BirthDate = newValue.BirthDate;
        employees.Gender = newValue.Gender;
    }

```

Здійснюється перевірка перед додаванням нового робітника у базу даних. Якщо є порожні поля, то додавання не відбувається. Реалізація наведена у лістингу 7.

Лістинг 7 Перевірка перед додаванням нового робітника

```

try
{
    if (textBoxSurname.Text.Length == 0 ||
        textBoxName.Text.Length == 0 ||
            textBoxTax.Text.Length == 0 ||
                textBoxSex.Text.Length == 0)
        {
            MessageBox.Show("Поля не можуть бути
порожніми!");
            return;
        }
    else
        {
            employees.LastName =
textBoxSurname.Text.ToString();
            employees.FirstName =
textBoxName.Text.ToString();
            employees.TaxID = textBoxTax.Text.ToString();
            employees.BirthDate =
dateTimePicker_birthDate.Value;
            employees.Gender = textBoxSex.Text.ToString();
            DialogResult = DialogResult.OK;
        }
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Змінення",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
}

```

При подвійному натисканні на рядку таблиці відкривається вікно зміни даних. Якщо внесені зміни задовольняють перевірки, то інформація вноситься в базу даних. Реалізація наведена у лістингу 8.

Лістинг 8 Внесення інформації в базу даних через рядки таблиці

```
private void dataGridView1_CellDoubleClick(object sender,
DataGridViewCellEventArgs e)
{
    if (dataGridView1.SelectedRows.Count > 0)
    {
        switch ((TablesTypeEnum)CargoDB.currentTable)
        {
            case TablesTypeEnum.Addresses:
                Addresses addresses =
CargoDB.GetAddresses(Convert.ToInt32(dataGridView1.Selected
Rows[0].Cells[0].Value));
                if (addresses != null)
                {
                    using (FormAddresses form = new
FormAddresses(addresses))
                    {
                        DialogResult dialogResult =
form.ShowDialog();
                        if (dialogResult ==
DialogResult.OK)
                        {
                            if
(!CargoDB.UpdateAddresses(form.addresses))
                                MessageBox.Show("Змінення
обраних даних неможливе", "Змінення", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                            else
                                LoadTables(CargoDB.currentTable);
                        }
                    }
                }
                break;
            }
        }
    }
}
```

При натисканні на кнопку додавання відкривається відповідна форма додавання для обраної таблиці. Після перевірки, у разі задоволення вимогам, відбувається додавання у базу даних. Реалізація наведена у лістингу 9.

Лістинг 9 Додавання у базу даних через форму

```
private void buttonAdd_Click(object sender, EventArgs e)
{
    switch ((TablesTypeEnum)CargoDB.currentTable)
    {
        case TablesTypeEnum.Addresses:
            using (FormAddresses form = new FormAddresses())
            {
                DialogResult dialogResult =
form.ShowDialog();
                if (dialogResult == DialogResult.OK)
                {
                    if
(!CargoDB.AddAddresses(form.addresses))
                        MessageBox.Show("Додавання введених
даних неможливе", "Додавання", MessageBoxButtons.OK,
MessageBoxIcon.Error);
                    else
                        LoadTables(CargoDB.currentTable);
                }
            }
            break;
    }
}
```

3.6 Розробка інтерфейсу інформаційної системи

Клієнтська програма розроблена з урахуванням усіх функціональних вимог, що були визначені на попередньому етапі.

Для доступу до ресурсів системи користувачі повинні пройти аутентифікацію та авторизацію (див. Рис. 17).



Рисунок 17 — Вікно аутентифікації

Після верифікації особистості користувачеві стає доступним вікно інформаційної системи транспортної компанії. (див. Рис. 18).

Id	LastName	FirstName	TaxID	BirthDate	Gender
1	Petersen	Mette	456123789	10/03/1988	Female
2	Andersen	Jens	789456123	25/11/1982	Male
3	Nielsen	Lise	321654987	18/07/1995	Female
4	Pedersen	Morten	654987321	02/09/1987	Male
5	Jensen	Anna	987321654	05/12/1993	Female
6	Madsen	Anders	159753468	30/06/1984	Male
7	Christensen	Sofie	753159468	12/04/1991	Female
8	Larsen	Henrik	369852147	22/10/1989	Male
9	Rasmussen	Louise	852369741	14/02/1996	Female
10	Møller	Thomas	147852369	08/08/1983	Male
11	Knudsen	Maria	654321987	25/01/1994	Female
12	Holm	Nikolaj	258741369	12/05/1986	Male
13	Thomsen	Camilla	987123654	20/09/1992	Female
14	Lassen	Peter	741852963	15/11/1981	Male
15	Andreasen	Sara	852963741	28/03/1997	Female

Рисунок 18 — Вікно інформаційної системи транспортної компанії

Користувач може самостійно обирати, які таблиці будуть доступні для перегляду в поточному вікні, залежно від його рівня доступу. Щоб додати новий запис до обраної таблиці, слід скористатися кнопкою «Додати запис до таблиці» (див. Рис. 19).

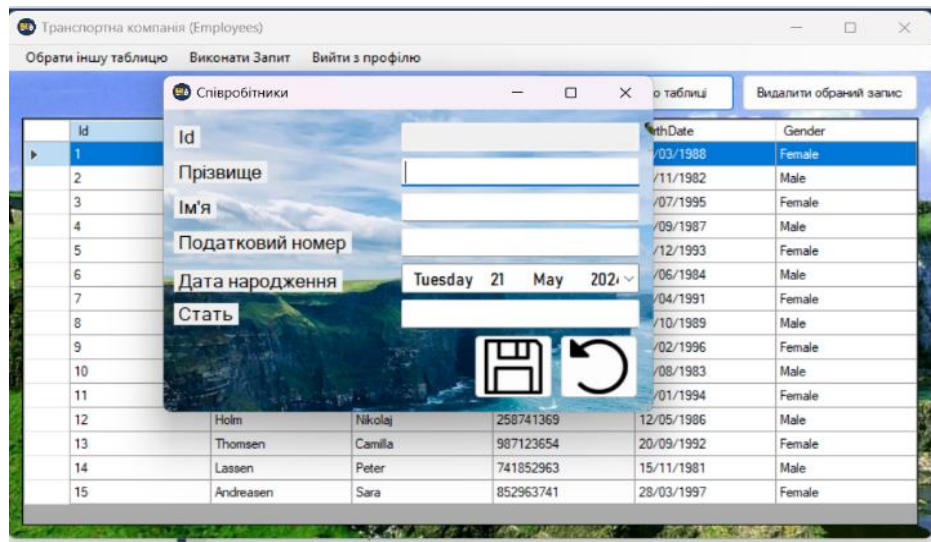


Рисунок 19 — Додавання запису

Календарний інтерфейс робить вибір дат зручним та інтуїтивно зрозумілим (див. Рис. 20).

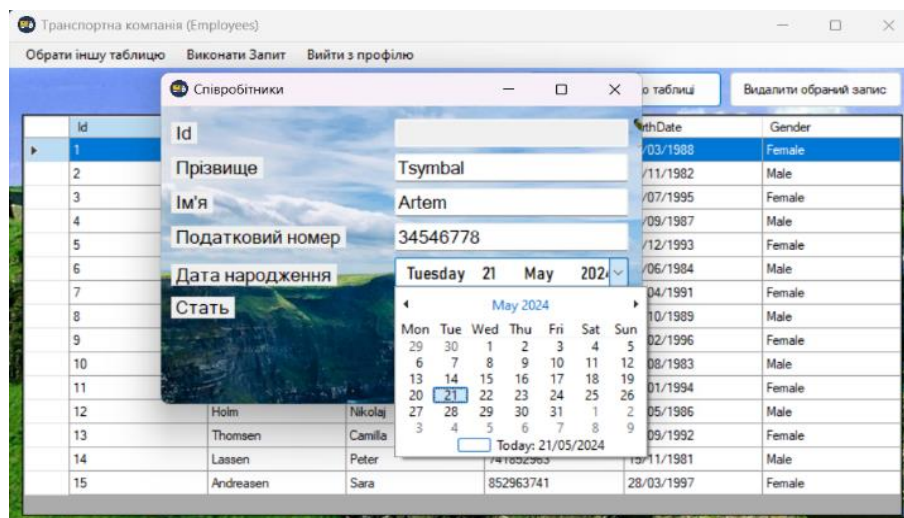


Рисунок 20 — Інтерфейс календаря

Додавання запису неможливе без заповнення всіх обов'язкових полів. У такому випадку з'явиться нагадування про необхідність заповнити всі поля (див. Рис.21).

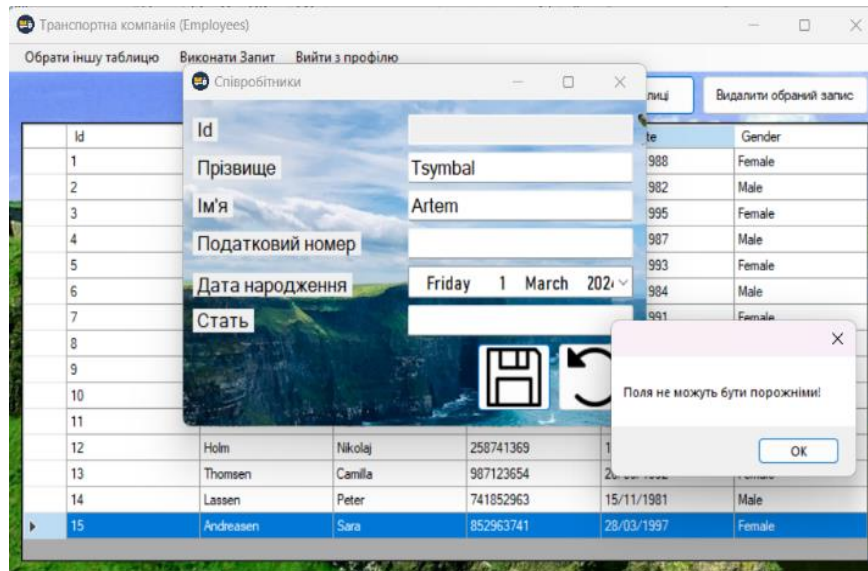


Рисунок 21 — Повідомлення про помилку

Після успішного заповнення всіх необхідних полів інформація про нового співробітника буде додана до відповідної таблиці (див. Рис.22).

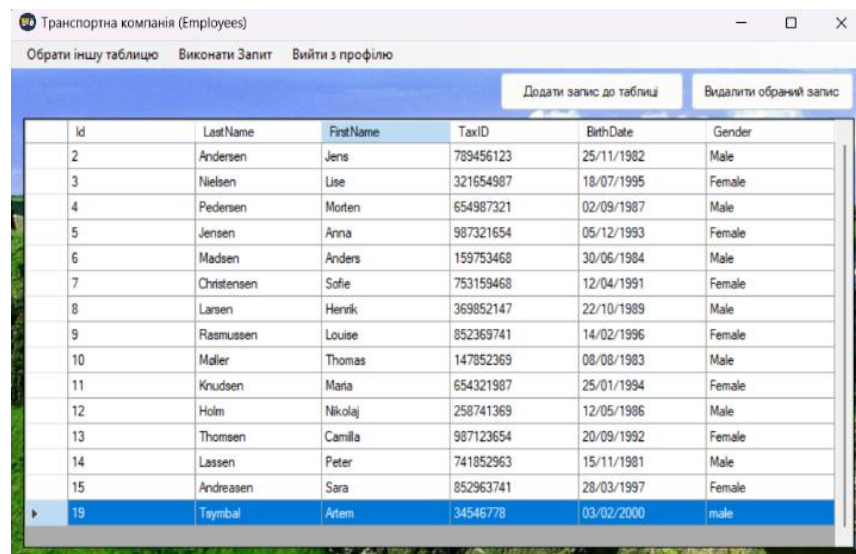


Рисунок 22 — Додавання нового співробітника до відповідної таблиці

Система надає можливість видалення співробітників. Перед остаточним видаленням програма запропонує вам підтвердити свою дію (див. Рис. 23).

Користувацький інтерфейс дозволяє обирати інші таблиці для роботи з даними (див. Рис. 24).

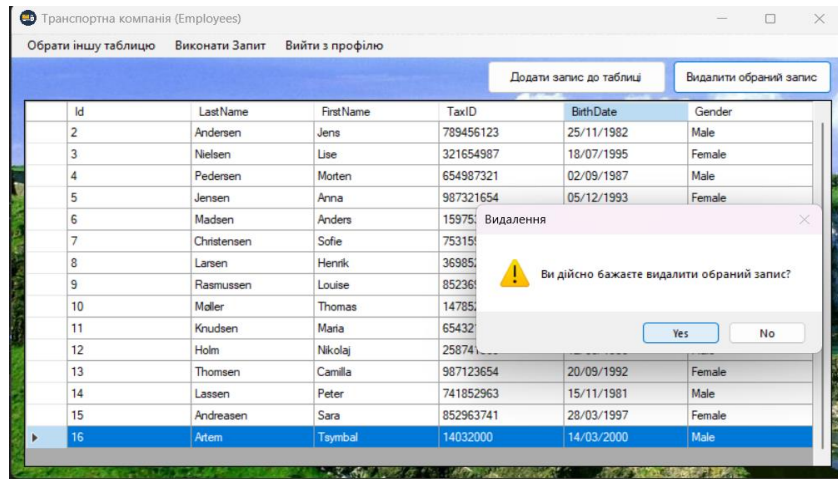


Рисунок 23 — Підтвердження дій про видалення запису

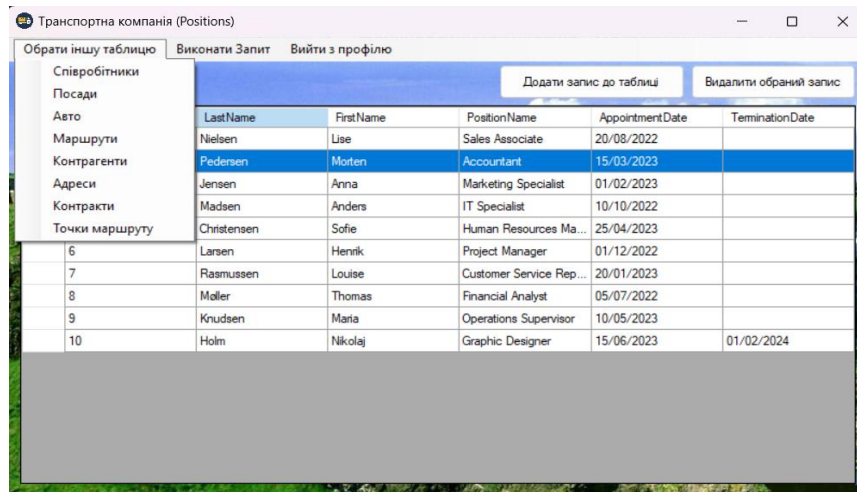


Рисунок 24 — Вікно вибору таблиць

Система надає можливість здійснювати пошук інформації в таблицях (див. Рис. 25-27).

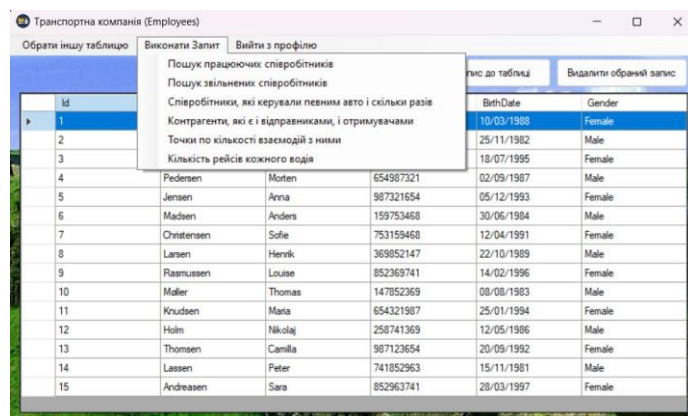


Рисунок 25 — Пошук інформації в таблицях

ID	LastName	FirstName	TaxID	BirthDate	Gender
3	Nielsen	Lise	321654987	18/07/1995	Female
4	Pedersen	Morten	654987321	02/09/1987	Male
5	Jensen	Anna	987321654	05/12/1993	Female
6	Madsen	Anders	159753468	30/06/1984	Male
7	Christensen	Sofie	753159468	12/04/1991	Female
8	Larsen	Henrik	369852147	22/10/1989	Male
9	Rasmussen	Louise	852369741	14/02/1996	Female
10	Moller	Thomas	147852369	08/08/1983	Male
11	Knudsen	Maria	654321987	25/01/1994	Female

Рисунок 26 — Результат запиту «Пошук працюючих співробітників»

ID	County	City	Street	HouseNumber	ApartmentNumber	TotalInteractions
1	Trondheim	Trondheim	Kongens gate	5	10	6
2	Stavanger	Stavanger	Kongsgata	8	22	6
3	Tromsø	Tromsø	Strandgata	15	7	5
4	Drammen	Drammen	Gronland	25	3A	5
5	Kristiansand	Kristiansand	Skippergata	20	3A	5
6	Sandnes	Sandnes	Langgata	14	5	4
7	Alesund	Alesund	Nedre Strandgate	7	3A	3
8	Skien	Skien	Sannergata	30	3A	3
9	Bode	Bode	Storgata	18	3A	3

Рисунок 27 — Результат запиту «Точки по кількості взаємодії з ними»

Вхід до системи за логіном та паролем менеджера надає йому обмежений доступ до функціональності (див. Рис. 28).

Last Name	First Name	TaxID	BirthDate	Gender
Fetersen	Mette	456123789	10/03/1988	Female
Andersen	Jens	789456123	25/11/1982	Male
Nielsen	Lise	321654987	18/07/1995	Female
Pedersen	Morten	654987321	02/09/1987	Male
Jensen	Anna	987321654	05/12/1993	Female
Madsen	Anders	159753468	30/06/1984	Male
Christensen	Sofie	753159468	12/04/1991	Female
Larsen	Henrik	369852147	22/10/1989	Male
Rasmussen	Louise	852369741	14/02/1996	Female
Moller	Thomas	147852369	08/08/1983	Male
Knudsen	Maria	654321987	25/01/1994	Female
Holm	Nikolaj	258741369	12/05/1986	Male
Thomsen	Camilla	987123654	20/09/1992	Female
Lassen	Peter	741852963	15/11/1981	Male
Andreasen	Sara	852963741	28/03/1997	Female
Aasen	Tyrrbal	14032000	14/03/2000	Male

Рисунок 28 — Доступ до функціональності для менеджера

ВИСНОВКИ

1. У ході реалізації кваліфікаційної роботи було проведено огляд наукових робіт та існуючих продуктів, що стосуються інтелектуальних інформаційних систем для транспортних компаній, проаналізовано функціональні можливості та особливості аналогів, виявлено актуальні потреби та проблеми в цій сфері.
2. На основі аналізу визначено мету розробки власного програмного забезпечення, яке буде відповідати специфічним вимогам транспортної компанії. Очікується, що розробка нового продукту допоможе підвищити ефективність її роботи.
3. Розроблено технічне завдання, яке чітко описує функціональні можливості, вимоги до системи та план робіт. Спроектовано архітектуру системи, яка враховує всі необхідні компоненти та їх взаємодію.
4. Розроблено базу даних, яка буде зберігати всю необхідну інформацію про роботу компанії.
5. Створено десктопний застосунок, який дозволяє користувачам взаємодіяти з системою та виконувати всі необхідні дії.
6. Існує можливість розширення функціону системи в майбутньому. Створення веб-сайту з унікальною системою планування маршрутів та корисною інформацією для водіїв та експедиторів є одним із перспективних напрямків розвитку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Головіна О. Сучасні технології в управлінні транспортною логістикою. *International Science Journal of Management, Economics & Finance*. Vol. 2, No. 3, 2023, pp. 35-42.
2. Особливості автомобільних вантажних перевезень у Європі : веб-сайт. URL: <https://www.cargosupport.com.ua/ua/osoblivosti-avtomobilnih-vantazhnih-perevezen-u-evropi/> (дата звернення: 10.03.2024).
3. Демченко Є. Б., Дорош А. С., Сковрон І. Я. Сучасні інформаційні системи на ринку вантажних перевезень України. *Транспортні системи та технології перевезень*. Дніпро, 2022. Вип. 23. С. 79–88.
4. Розробка інформаційної системи для транспортної компанії : електронний ресурс. URL: <https://trans-atlas.com.ua/ua/article/259> (дата звернення: 07.03.2024).
5. Jon P. Smith. *Entity Framework Core in Action* : Packt, 2018. 520 с.
6. Entity Framework Core documentation : веб-сайт. URL: <https://learn.microsoft.com/en-us/ef/> (дата звернення: 07.03.2024).
7. Відомості про проекти і рішення на мові C# : електронний ресурс. URL: <https://docs.microsoft.com/ru-ru/visualstudio/get-started/tutorial-projectssolutions?view=vs-2019> (дата звернення: 10.03.2024).
8. Paul DuBois. *MySQL Cookbook*. 3rd edition. O'Reilly, 2014. 837 р.
9. MySQL Workbench : веб-сайт. URL: <https://www.mysql.com/products/workbench> (дата звернення: 10.03.2023).
10. Метод функціонального моделювання : веб-сайт. URL: http://ni.biz.ua/3/3_20/3_20501_metod-funktsionalnogo-modelirovaniya-SADT-IDEF.html (дата звернення: 10.03.2023).
11. What is an Entity Diagram(ERD)? веб-сайт. URL: <https://medium.com/@soni.dumitru/what-is-an-entity-relationship-diagram-erd-13daee5b2a>.

12. Мулеса О. Ю. Інформаційні системи та реляційні бази даних : Ужгород, 2018. 118 с.

13. Top 10 Programming Languages for Desktop Apps In 2021. Decipherzone : веб-сайт. URL: <https://www.decipherzone.com/blog-detail/top-programming-languages-for-desktop-apps-in-2021> (дата звернення: 07.04.2024).

14. Карпенко М. Ю., Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем : навч. посіб. Харків : ХНУМГ ім. О. М. Бекетова, 2017. 93 с.

15. Use Case Diagrams | Unified Modeling Language (UML) : веб-сайт. URL: <https://www.geeksforgeeks.org/use-case-diagram/> (дата звернення: 07.04.2024).

Декларація
академічної доброчесності
здобувача ступеня вищої освіти ЗНУ

Я, Довганік Євгенія Максимівна, студентка 5 курсу, форми навчання заочної, Інженерного навчально-наукового інституту, спеціальність 121 Інженерія програмного забезпечення, адреса електронної пошти vanjjdm@gmail.com, — підтверджую, що написана мною кваліфікаційна робота на тему «**Розробка інформаційної системи транспортної компанії Ірландії**» відповідає вимогам академічної доброчесності та не містить порушень, що визначені у ст.42 Закону України «Про освіту», зі змістом яких ознайомлений.

- заявляю, що надана мною для перевірки електронна версія роботи є ідентичною її друкованій версії;

згодна на перевірку моєї роботи на відповідність критеріям академічної доброчесності у будь-який спосіб, у тому числі за допомогою інтернет-системи, а також на архівування моєї роботи в базі даних цієї системи.

Дата 25.05.2024 Підпис _____ Довганік Євгенія Максимівна
(студент)

Дата 25.05.2024 Підпис _____ Міхайлуца Олена Миколаївна
(науковий керівник)