

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)

Кафедра _____ Штучного інтелекту _____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

рівень вищої освіти _____ другий (магістерський) _____

_____ Система оцінки ефективності управління заявками для _____
логістичного підприємства _____
(тема)

Виконала:
студентка 2 курсу, групи _____ СШМ-20-1 _____
Левицька М. В. _____
(прізвище, ініціали)

Спеціальність 122 Комп'ютерні науки _____
(код і повна назва спеціальності)

Тип програми _____ освітньо-професійна _____
(освітньо-наукова або освітньо-професійна)

Освітня програма _____ Комп'ютерні науки _____
(повна назва освітньої програми)

Керівник _____ доц. Шевченко О. Ю. _____
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____
(підпис)

_____ В. О. Філатов _____
(прізвище, ініціали)

2021 р.

Харківський національний університет радіоелектроніки

Факультет _____ Комп'ютерних наук _____
(повна назва)
Кафедра _____ Штучного інтелекту _____
(повна назва)
Рівень вищої освіти _____ другий (магістерський) _____
Спеціальність _____ 122 Комп'ютерні науки _____
(повна назва)
Тип програми _____ освітньо-професійна _____
(освітньо-професійна або освітньо-наукова)
Освітня програма _____ Системи штучного інтелекту (СШІ) _____
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)

« _____ » _____ 20 ____ р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентці _____ Левицькій Марині Василівні _____
(прізвище, ім'я, по батькові)

1. Тема роботи Система оцінки ефективності управління заявками для логістичного підприємства

затверджена наказом по університету від 8 листопада 2021 р. № 1695 Ст.

2. Термін подання студентом роботи до екзаменаційної комісії 9 грудня 2021 р.

3. Вихідні дані до роботи процеси електронного обліку вантажоперевезень, оформлення і виконання замовлень користувача, оцінка ефективності управління заявками, технічна документація з розробки веб-додатків, технічна документація з розробки додатків з використанням Spring та Hibernate

4. Перелік питань, що потрібно опрацювати в роботі _____

1) Аналіз предметної області та постановка задачі;

2) Розробка вимог до інформаційної системи;

3) Опис прийнятих проектних рішень та програмної реалізації;

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри)_____

6. Консультанти розділів роботи (п.6 включається до завдання за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Аналіз предметної галузі і постановка завдання	01.09.2021 – 27.09.2020	виконано
2	Дослідження галузі	28.09.2021 – 10.10.2021	виконано
3	Дослідження технологій та методів реалізації	29.10.2021 – 01.11.2021	виконано
4	Розробка вимог до інформаційної системи	02.11.2021 – 03.11.2021	виконано
5	Програмна реалізація	04.11.2021 – 10.11.2021	виконано
6	Тестування моделі та аналіз результатів	11.11.2021 – 12.11.2021	виконано
7	Оформлення пояснювальної записки	13.11.2021 – 03.12.2021	виконано
8	Попередній захист	07.12.2021	виконано
9	Захист перед ЕК	09.12.2021	

Дата видачі завдання 01 вересня 2021 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Шевченко О. Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка: 95 с., 35 рис., 2 дод., 20 джерел.

АНАЛІТИЧНА СИСТЕМА, БАЗА ДАНИХ,
ВАНТАЖОПЕРЕВЕЗЕННЯ, ВЕБ-СЕРВІС, ЛОГІСТИКА, СИСТЕМА
ОБЛІКУ, HIBERNATE, JPA, MYSQL, SPRING.

Об'єкт дослідження – процеси електронного обліку вантажоперевезень, оформлення і виконання замовлень користувача у рамках логістичного підприємства, оцінка ефективності характеристик системи.

Предмет дослідження – аналітична система оцінки ефективності управління заявками для логістичного підприємства у вигляді надбудови до веб-сервісу для надання послуг.

Метою роботи є розробка веб-сервісу для вантажоперевезень та системи оцінки ефективності керування заявками для логістичного підприємства.

Методи дослідження – методи структурного аналізу і моделювання реляційних баз даних, методи реляційної алгебри і реляційного числення, методи проектування клієнт-серверних додатків для Інтернет мережі, методи об'єктно-орієнтованого програмування.

В результаті проведених досліджень вирішено задачу проектування і реалізації веб-сервісу для логістичного підприємства, а також реалізації аналітичної системи оцінки ефективності керування заявками у системі вантажоперевезень.

Отримані результати використовуються у галузі підтримки електронного обліку заявок для логістичного підприємства.

В якості фреймворку для розробки веб-додатку були використані Spring та Hibernate, які використовують мову програмування Java як базу.

РЕФЕРАТ

Пояснительная записка: 95 с., 35 рис., 2 прил., 20 источников.

АНАЛИТИЧЕСКАЯ СИСТЕМА, БАЗА ДАННЫХ, ВЕБ-СЕРВИС, ГРУЗОПЕРЕВОЗКИ, ЛОГИСТИКА, СИСТЕМА УЧЕТА, HIBERNATE, JPA, MYSQL, SPRING.

Объект исследования – процессы электронного учета грузоперевозок, оформление и исполнение заказов пользователя в рамках перевозок, оценка эффективности характеристик системы.

Предмет исследования – аналитическая система оценки эффективности управления заявками для логистического предприятия в виде надстройки к веб-сервису для предоставления услуг.

Цель работы – разработка веб-сервиса и системы оценки эффективности управления заявками для логистического предприятия.

Методы исследования – методы структурированного анализа и моделирования реляционных баз данных, методы проектирования клиент-серверных приложений для сети Интернет, методы объектно-ориентированного программирования.

В результате проведенных исследований была решена задача проектирования и реализации веб-сервиса для логистического предприятия, а также реализации аналитической системы оценки эффективности управления заявками.

Полученные результаты используются в области поддержки электронного учета заявок для логистического предприятия.

В качестве фреймворка для разработки веб-приложения были использованы Spring и Hibernate, которые используют язык программирования Java как базу.

ABSTRACT

Explanatory note: 95 p., 35 fig., 2 ann., 20 sources.

ANALYTICAL SYSTEM, ACCOUNT SYSTEM, CARGO TRANSPORTATION, DATABASE, HIBERNATE, JPA, LOGISTICS, MYSQL, SPRING, WEB-SERVICE.

Object of research is the processes of electronic accounting of freight, registration and execution of user orders in the framework of transportation, evaluation of the effectiveness of the characteristics of the system.

Subject of research is an analytical system to assess the effectiveness of applications management for the logistics company in the form of an add-on to a web service for the provision of services.

The purpose of the graduation project is to develop a web service for freight and a system to assess the effectiveness of application management for the logistics enterprise.

Research methods are the structuring analysis and design of relational databases, design methods of client-server applications for the Internet, object-oriented programming methods.

As a result of research, the problem of designing and implementing a web service for the logistics enterprise, as well as the implementation of the analytical system of evaluation of the effectiveness of application management in the freight transport system was solved.

The end work's results can be useful for the support of electronic accounting requests for the logistics enterprise.

As a framework, the Spring and Hibernate were used, that use the Java programming language as a base.

ЗМІСТ

Перелік умовних позначень, символів, одиниць, скорочень і термінів	8
Вступ	9
1 Аналіз предметної області та постановка задачі	10
1.1 Аналіз предметного середовища	10
1.2 Огляд наявних аналогів	13
1.3 Постановка задачі	14
2 Розробка вимог до інформаційної системи	15
2.1 Опис функціональної моделі	15
2.2 Вхідні дані	21
2.3 Вихідні дані	22
2.4 Проектування бази даних	23
2.5 Побудова об'єктно-орієнтованої моделі	26
2.6 Математичне та алгоритмічне забезпечення	32
3 Опис прийнятих проектних рішень та програмної реалізації	37
3.1 Засоби розробки	37
3.2 Вимоги до технічного та програмного забезпечення	38
3.3 Опис програмної реалізації	38
3.4 Розробка інтерфейсу додатку	53
Висновки	65
Перелік джерел посилання	67
Додаток А Текст програми	69
Додаток Б Відомість кваліфікаційної роботи магістра	95

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

БД – База даних;

СУБД – Система управління базами даних;

DAO – Data Access Object – об'єкт доступу до даних;

HQL – Hibernate Query Language – мова запитів Hibernate;

HTML – HyperText Markup Language – мова гіпертекстової розмітки;

IoC – Inversion of Control – інверсія контролю;

JPA – Java Persistence API – програмний інтерфейс Java Persistence;

ORM – Object-Relational Mapping – об'єктно-реляційне відображення;

SQL – Structed Query Language – структурована мова запитів;

UML – Unified Modeling Language – уніфікована мова моделювання.

ВСТУП

В останні часи спостерігається різке збільшення обсягів інформації в суспільстві взагалі та інформації, використовуваної на підприємствах і в організаціях, зокрема. Це пов'язано з появою нових технологій, швидкою їх змінюваністю, активним накопиченням теоретичних знань і практичного досвіду. Інтернет сьогодні став найбільш ефективним засобом реклами і просування інформації. Інтернет може задовільнити усі потреби сучасного користувача, це – покупки, укладання ділових відносин, пошук клієнтів тощо. Саме тому щоб підвищити ефективність роботи будь-якої компанії, успішно конкурувати в сьогоднішніх умовах, варто розглянути створення веб та/або мобільного додатка, адже це допоможе не тільки організувати контакт між усіма учасниками бізнесу, а й надати нові можливості для розширення діяльності і зниження видатків.

Актуальність кваліфікаційної роботи визначається тим, що для будь-якої компанії, зокрема підприємства з вантажоперевезень необхідні засоби швидкого доступу до інформації, а також аналіз ефективності роботи підприємства, що і забезпечує веб-сервіс. Більшість організацій та підприємств дбають про розробку власних ресурсів, адже вони займають передову позицію у стратегії розвитку будь-якої компанії. Тому важливо, щоб сервіс відповідав вимогам відвідувачів. Користувачеві приємно відвідувати ті веб-сторінки, які мають стильний інтерфейс, не перенасичені графікою і анімацією, які швидко завантажуються і правильно відображені у вікні браузера, які містять цікавий і корисний контент.

Метою роботи є реалізація веб-сервісу для логістичного підприємства, а саме, для вантажоперевезень та створення аналітичної системи оцінки ефективності керування заявками для підприємства у вигляді надбудови для веб-сервісу.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ

1.1 Аналіз предметного середовища

Використання інформаційних технологій призвело до кардинальних змін традиційних способів ведення бізнесу та виникнення нового виду економічної діяльності – електронного бізнесу.

Транспортування – це ключова комплексна активність, пов'язана з переміщенням матеріальних ресурсів, незавершеного виробництва або готової продукції певним транспортним засобом у логістичному ланцюзі, яка складається, у свою чергу, з комплексних та елементарних активностей, що охоплюють експедирування, вантажопереробку, упакування, передачу прав власності на вантаж, страхування і т. ін.

Транспортні послуги можна умовно розділити на два сегменти – логістика і пасажирські перевезення. Обидві категорії вимагають налагодженої взаємодії між трьома сторонами:

- компаніями або приватними особами, які надають послуги;
- перевізниками – виконавцями;
- клієнтами – користувачами сервісу.

Для бізнесу розробка додатка для транспорту вигідна тим, що:

- автоматизуються бізнес-процеси;
- знімається навантаження з адміністраторів, знижується гостра потреба в них;
- завдання чіткі і деталізовані, так як більшість полів замовлення (за бажанням всі) – обов'язкові для заповнення;
- з'являється додатковий канал постійної реклами;
- відображаються всі пересування, зупинки, доручення виконавців;

– можна систематизувати доходи, отримувати звіти і виводити статистику, збирати відгуки і оцінки клієнтів щодо якості роботи виконавців;

– продукт забезпечує додатковий прибуток за рахунок монетизації.

Транспортно-логістичні підприємства з вантажних перевезень є яскравим прикладом систем масового обслуговування і є дуже актуальними на сьогоднішній день, особливо у великих містах.

У даній роботі підприємство з вантажних перевезень моделюється як багатоканальна система масового обслуговування з чергою, в якій відмова відбувається через деякий час очікування. Каналами обслуговування слугуватимуть зареєстровані в системі водії з призначеним авто. Відмова замовлень здійснюватиметься диспетчером, якщо всі канали зайняті і заявка вже знаходиться в черзі певний час.

Системи масового обслуговування (СМО) – це такі системи, в які у випадкові моменти часу надходять заявки на обслуговування[2]. При цьому заявки виконуються за допомогою наявних у розпорядженні системи каналів обслуговування. З позиції моделювання процесу масового обслуговування ситуації, коли утворюються черги заявок (вимог) на обслуговування, виникають таким чином. Вимога, надходячи в обслуговуючу систему, приєднується до черги з тих, що надійшли раніше. Канал обслуговування вибирає вимогу з наявних у черзі і розпочинає її обслуговування. Після завершення обслуговування чергової вимоги канал обслуговування приступає до обслуговування наступної вимоги, якщо вона є у блоці очікування. Даний цикл функціонування системи масового обслуговування повторюється багаторазово протягом всього періоду роботи обслуговуючої системи. При цьому передбачається, що перехід системи на обслуговування чергової вимоги після завершення обслуговування попередньої вимоги відбувається миттєво, у випадковий момент часу.

Системи масового обслуговування класифікують за різними ознаками.

Перша класифікація по наявності черг:

- системи з відмовами (без черги) – заявка, яка надійшла в момент, коли всі канали зайняті, отримує відмову, покидає СМО і надалі не обслуговується;

- системи з очікуванням (з чергою) – заявка, що прийшла в момент, коли всі канали зайняті, не відкидається, а стає в чергу і чекає можливості бути обслугованою. В свою чергу класифікують:

- по довжині черги – з обмеженою довжиною черги, які допускають чергу, але з обмеженим числом місць в ній;

- за часом очікування – з обмеженим часом очікування;

- по дисципліні обслуговування – з обслуговуванням по пріоритету, що допускають чергу, але деякі заявки обслуговуються поза чергою (тобто по пріоритету).

Друга класифікація за числом каналів обслуговування:

- одноканальні;

- багатоканальні.

Третя класифікація за місцем знаходження джерела вимог:

- розімкнені, коли джерело знаходиться поза системою;

- замкнуті, коли джерело знаходиться в самій системі.

Для оцінки ефективності керування потоками в системі масового обслуговування можна використати показники її функціонування:

- інтенсивність навантаження;

- середній час обслуговування заявок;

- ймовірність, що канали обслуговування вільні;

- ймовірність відхилення заявки;

- середній час очікування заявки в черзі;

- кількість зайнятих каналів обслуговування;

- кількість каналів в режимі простою;

- пропускна здатність СМО (кількість заявок, що обслуговується за одиницю часу);

– відносна величина витрат.

1.2 Огляд наявних аналогів

Існує декілька додатків, які надають можливості керування потоками на основі розрахунку характеристик системи масового обслуговування.

LimeTech – програмне і технічне забезпечення для керування чергами. Дає змогу регулювати потоки відвідувачів, аналізувати статистику сервісу і отримувати відгуки відвідувачів про рівень обслуговування, але встановлюється лише на термінали, що забезпечуються компанією LimeTech.

NeuroniQ (Система Управління Чергою) – це сукупність обладнання і програмного забезпечення, що дозволяє впорядкувати прийом відвідувачів, позбутися «живих» черг, а також вирішити проблему обліку та підвищення продуктивності праці персоналу. Також надає детальну статистичну інформацію щодо функціонування системи. Знову ж надається лише у комплекті із технічним забезпеченням у вигляді терміналів.

Qmatic – програмне забезпечення для керування віртуальними чергами. Надає можливості віртуальної реєстрації, обміну повідомленнями, дистанційного керування чергами через мобільний додаток. Дає змогу аналізувати статистику і генерувати звіти з кількісними показниками роботи системи, але непридатний для логістичних підприємств.

Wavetec – програмне і технічне забезпечення для керування чергами. Дає змогу гнучко налаштовувати параметри системи, інтегруватися з іншими платформами і складати детальні звіти з результатами роботи системи. Встановлюється на спеціальні термінали від компанії Wavetec.

Враховуючи недоліки перерахованих аналогів стає доцільною розробка аналітичної системи оцінки ефективності керування потоками в системах масового обслуговування у вигляді зручного налаштування для веб-додатку логістичного підприємства, яка надавала би змогу генерувати

детальний аналіз ефективності керування потоками в системі у вигляді показових графів, таблиць і кількісних показників.

1.3 Постановка задачі

Метою роботи є створення веб-сервісу для підприємства з вантажоперевезень та аналітичної системи оцінки ефективності управління заявками у вигляді веб-додатку, яка надавала би змогу генерувати детальний аналіз ефективності керування потоками в системі у вигляді показових графів, таблиць і кількісних показників.

Для досягнення мети проекту необхідно виконати:

- аналіз предметної області;
- виділити основні вхідні та вихідні об'єкти;
- розробку методу розрахунку характеристик системи масового обслуговування;
- розробку методу аналізу ефективності керування потоками заявок в системі;
- проектування БД в якій містяться записи про користувачів і замовлення;
- реалізацію веб-сервіса, а також аналітичну систему оцінки ефективності за допомогою вибраних програмних компонентів.

Отже, в даному розділі був проведений аналіз предметного середовища, виконаний огляд наявних аналогів, визначена і описана мета роботи, а також постановка задачі.

2 РОЗРОБКА ВИМОГ ДО ІНФОРМАЦІЙНОЇ СИСТЕМИ

2.1 Опис функціональної моделі

Для проектування схем використання спочатку необхідно визначити дійових осіб, а потім дії, які у системі може виконувати кожен з них.

Нижче наведений список акторів системи і їх можливості.

Користувач має можливість:

- здійснювати замовлення на вантажоперевезення.

Диспетчер має можливість:

- авторизуватися;
- назначати водіїв на замовлення;
- відхиляти заявки;
- переглянути список поточних замовлень;
- переглянути список водіїв з призначеним авто.

Водій має можливість:

- авторизуватися;
- переглядати список своїх замовлень;
- підтверджувати виконання замовлення (завершувати замовлення).

Адміністратор має можливість:

- авторизуватися;
- переглядати список всіх замовлень;
- переглядати та змінювати список існуючих ресурсів (авто, водії);
- призначати водіям авто;
- переглядати список замовлень з їх поточним статусом;
- переглядати статистичні і аналітичні дані щодо ефективності

роботи підприємства.

Діаграма варіантів використання (Use Case Diagram) призначена для визначення функціональних вимог до системи, описує взаємовідносини та залежності між варіантами використання та дійовими особами, що беруть

участь у процесі, демонструє те, що система повинна виконувати, не вказуючи застосовані методи [10].

Була складена діаграма варіантів використання для акторів системи (рисунок 2.1):

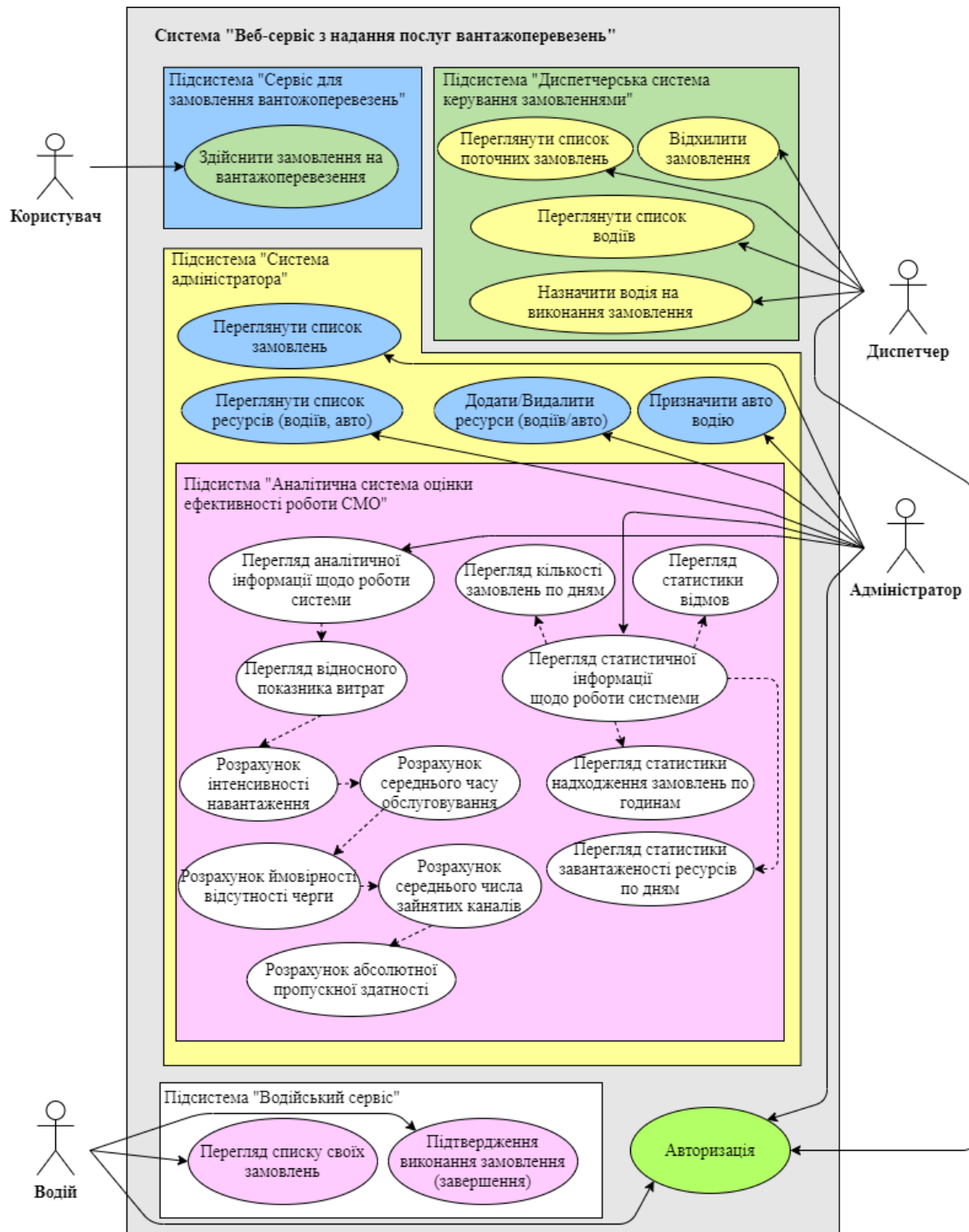


Рисунок 2.1 – Схема структурна варіантів використання

Розглянемо дії з варіантів використання:

- підтвердження виконання замовлення – можливість водія змінити статус замовлення на виконаний, тобто завершити замовлення;
- переглянути список ресурсів – можливість адміністратора переглядати доступні ресурси системи, а саме список водіїв і авто;
- додавання/Видалення ресурсів в системі – можливість адміністратора додавати і видаляти ресурси, а саме водіїв і авто;
- призначення авто водію – можливість адміністратора призначати авто з наявного автопарку водіям, зареєстрованим в системі;
- перегляд аналітичної інформації щодо ефективності роботи підприємства як системи – можливість адміністратора переглядати характеристики ефективності роботи СМО (керування потоками заявок в системі), а саме переглядати відносний показник витрат, інтенсивності навантаження, середнього часу обслуговування, абсолютної пропускної здатності і т.д.;
- перегляд статистичної інформації щодо роботи СМО – можливість адміністратора переглядати кількісну статистику роботи системи. Цей варіант включає в себе перегляд статистики кількості замовлень по дням, перегляд статистики відмов, перегляд статистики надходження замовлень по годинам, а також перегляд статистики завантаженості ресурсів по дням.

Перед тим як користувач побачить вміст сайту у себе на екрані, браузер робить запит на сервер щоб отримати вміст.

Клієнтська частина веб-додатку – це графічний інтерфейс. Тобто те, що користувач побачить на сторінці. Графічний інтерфейс відображається в браузері. Користувач взаємодіє з веб-додатком саме через браузер.

Серверна частина веб-додатку – це програма на сервері, що обробляє запити користувача (точніше, запити браузера). При кожному переході користувача по посиланню, браузер відправляє запит до сервера. Сервер обробляє цей запит, і формує веб-сторінку, описану через HTML, і відсилає

клієнтові по мережі. Браузер відразу відображає отриманий результат у вигляді чергової веб-сторінки.

База даних (БД, або система управління базами даних, СУБД) – програмне забезпечення на сервері, що займається зберіганням даних та їх наданням в потрібний момент. База даних розташовується на сервері.

Серверна частина веб-додатку звертається до бази даних, витягуючи дані, які необхідні для формування сторінки, що запитує користувач.

Найкращим методом для відображення процесу діяльності системи є побудова діаграми діяльності UML (Unified Modeling Language – уніфікована мова моделювання). UML був створений для визначення, візуалізації, проектування й документування в основному програмних систем. UML не є мовою програмування, але за допомогою UML моделей можна виконати кодогенерацію. Мова моделювання підходить для різноманітного класу програмних систем, різних розмірів проектів та областей додатків.

Діаграма діяльності (або діаграма активності) в UML (Activity Diagram) – візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів, вершинами якого є певні дії, а переходи відбуваються по завершенню дій. Діаграма діяльності (Activity Diagram) демонструє послідовність бізнес-процесів або дій, яку виконує система для досягнення мети, при чому послідовність робіт може залежати від прийняття певних рішень залежно від умов, що склалися. Кожний стан на діаграмі діяльності відповідає виконанню деякої операції, а перехід виконується тільки після її завершення [10].

Розглянемо схему, у якій показано, які дії має виконати користувач для створення замовлення в системі (рисунк 2.2):

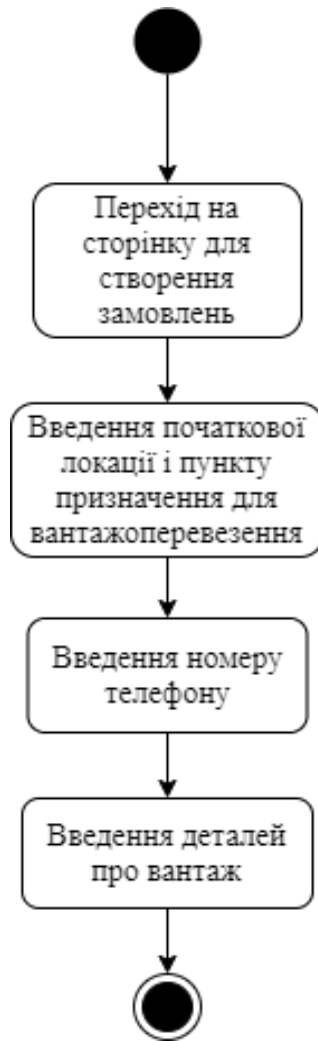


Рисунок 2.2 – Діаграма діяльності користувача

Для коректної роботи сервісу користувач заповнює певну інформацію про своє замовлення (номер телефону, початковий пункт і пункт призначення, деталі про вантаж). При створенні замовлення, в базу даних заноситиметься час створення запису для визначення кількості часу витраченого на виконання замовлення.

Розглянемо хід дій, який здійснює диспетчер в диспетчерській системі керування замовленнями (рисунок 2.3):

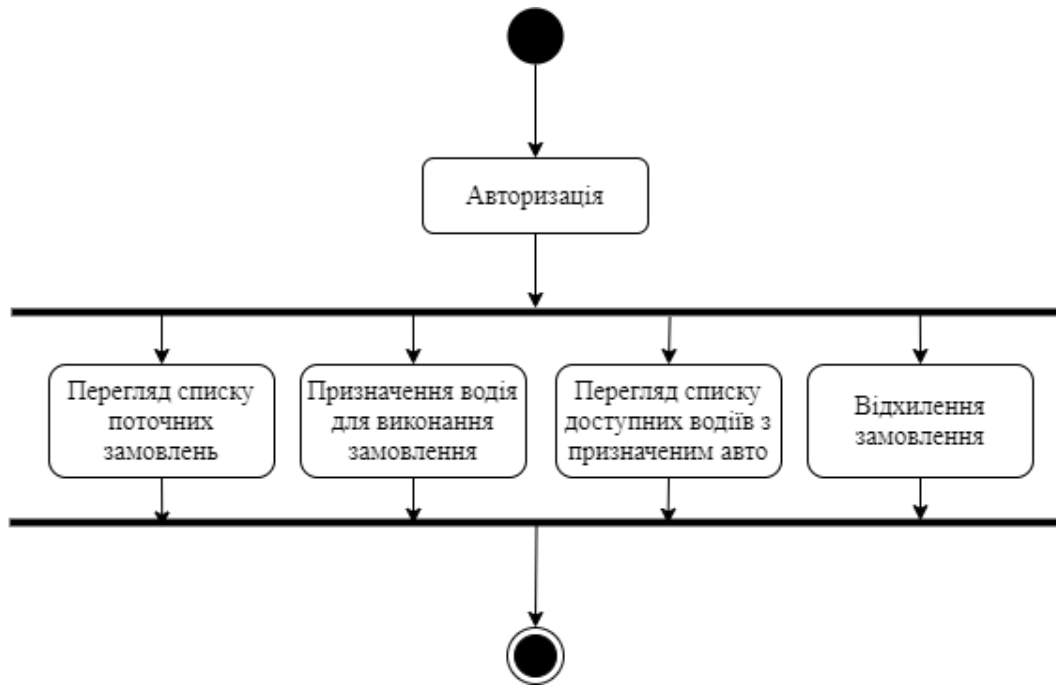


Рисунок 2.3 – Діаграма діяльності диспетчера

Свої стандартні функції, а саме перегляд списку поточних замовлень, відхилення замовлень, призначення водіїв і перегляд їх списку, диспетчер має можливість виконувати лише після авторизації в системі.

Розглянемо хід дій водія в водійському сервісі (рисунок 2.4).

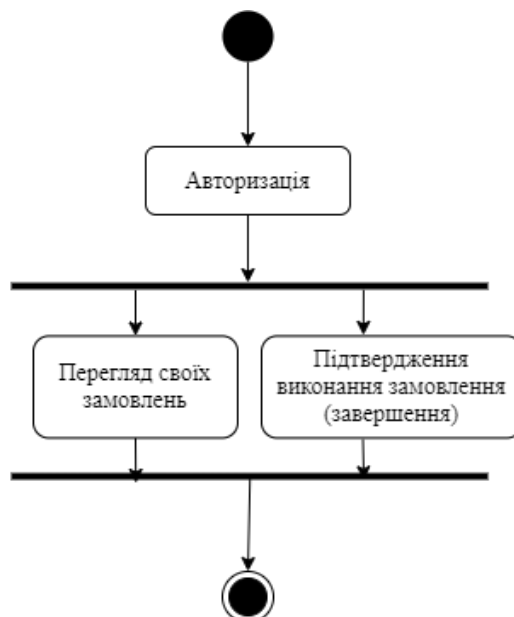


Рисунок 2.4 – Діаграма діяльності водія

Переглядати список своїх замовлень і змінювати статус замовлень водій може тільки після авторизації. Облікові записи водіїв створюються адміністраторами системи.

Розглянемо хід дій адміністратора в системі (рисунок 2.5).

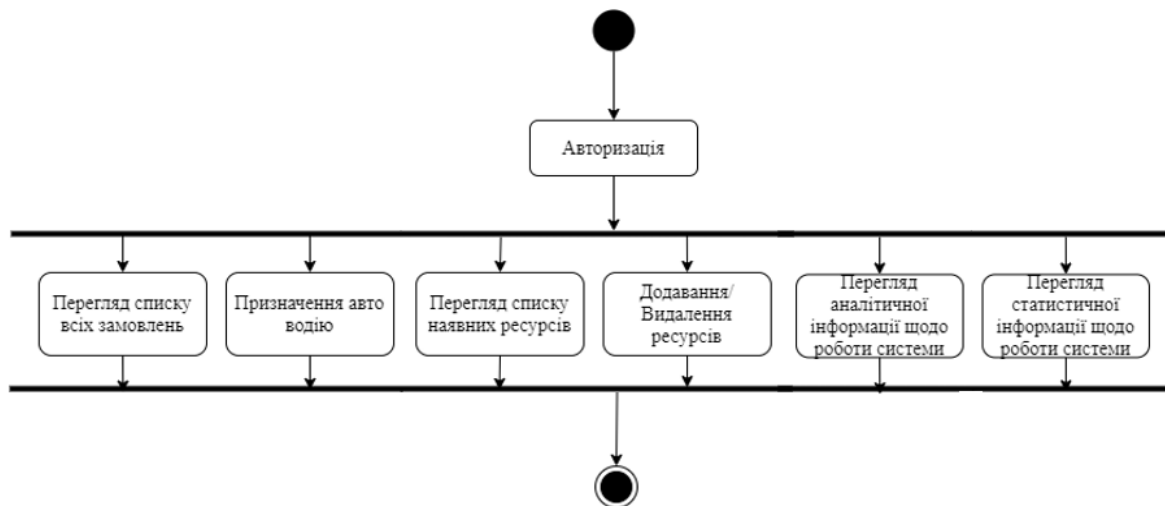


Рисунок 2.5 – Діаграма діяльності адміністратора в системі

2.2 Вхідні дані

Вхідні дані в систему можна поділити на наступні групи:

а) дані форм реєстрації водіїв в веб-сервісі:

- 1) ім'я (строковий тип з обмеженням в 20 символів);
- 2) прізвище (строковий тип обмеженням в 20 символів);
- 3) email (строковий тип обмеженням в 255 символів);
- 4) телефон (строковий тип обмеженням в 20 символів);
- 5) пароль (строковий тип обмеженням в 255 символів);

б) дані форм реєстрації авто в веб-сервісі:

- 1) модель (строковий тип з обмеженням в 20 символів);
- 2) довжина грузового відсіку (дійсне додаткове число [0; 1000]);
- 3) висота грузового відсіку (дійсне додаткове число [0; 1000]);

- 4) об'єм грузового відсіку (дійсне додаткове число [0; 1000]);
 - 5) максимальна вага (дійсне додаткове число [0; 1000]);
 - 6) номер авто (строковий тип з обмеженням в 20 символів);
- в) дані форми створення замовлення в веб-сервісі:
- 1) номер телефону (строковий тип з обмеженням в 20 символів);
 - 2) початкова локація (строковий тип обмеженням в 255 символів);
 - 3) пункт призначення (строковий тип обмеженням в 255 символів);
 - 4) додаткова інформація (строковий тип обмеженням в 255 символів);
 - 5) дата створення замовлення (тип поля «Дата й час»).

2.3 Вихідні дані

Вихідними даними є інформація про розраховані характеристики моделі масового обслуговування, а саме:

- інтенсивність навантаження (Дійсне додаткове число [0; 100000]);
- середній час обслуговування (Дійсне додаткове число [0; 100000]);
- ймовірність, що канал вільний (Дійсне додаткове число [0; 100]);
- ймовірність обслуговування заявок (ймовірність того, що клієнт буде обслугований) [0; 100]);
- середнє число незайнятих каналів (Дійсне додаткове число [0; 100000]);
- середнє число каналів, зайнятих обслуговуванням [0; 100000]);
- середнє число каналів, що перебувають в режимі простою (Дійсне додаткове число [0; 100000]);
- абсолютна пропускна здатність (Дійсне додаткове число [0; 100000]);

- відносна величина витрат (Дійсне додаткове число [0; 100000]).

Вихідні дані відображаються у форматі списку з показниками. До кожного показника надана інформація, що описує його для повного аналізу характеристик моделі.

2.4 Проектування бази даних

Якість проектування бази даних може впливати на роботу з нею. З добре спроектованою базою даних легше працювати, легше писати до неї запити. Для якісного проектування бази даних існують різні методики, різні послідовності кроків або етапів, які багато в чому схожі. І в цілому ми можемо виділити наступні етапи: виділення сутностей і їх атрибутів, які зберігатимуться у базі даних, і формування по них таблиць, атомізація складних атрибутів на простіші, визначення унікальних ідентифікаторів (первинних ключів) об'єктів, які зберігаються в рядках таблиці, визначення стосунків між таблицями за допомогою зовнішніх ключів та нормалізація бази даних.

База даних складається з 5 таблиць (сутностей):

- таблиця `users`, що містить записи з зареєстрованими в системі користувачами;
- таблиця `trucks`, що містить записи з занесеними в систему авто;
- таблиця `truck_orders`, що містить записи за замовленнями користувачів;
- таблиця `roles`, що містить записи з ролями;
- таблиця `user_role`, що містить записи з посиланнями на існуючих користувачів і їх список ролей.

Схема бази даних наведена на рисунку 2.6:

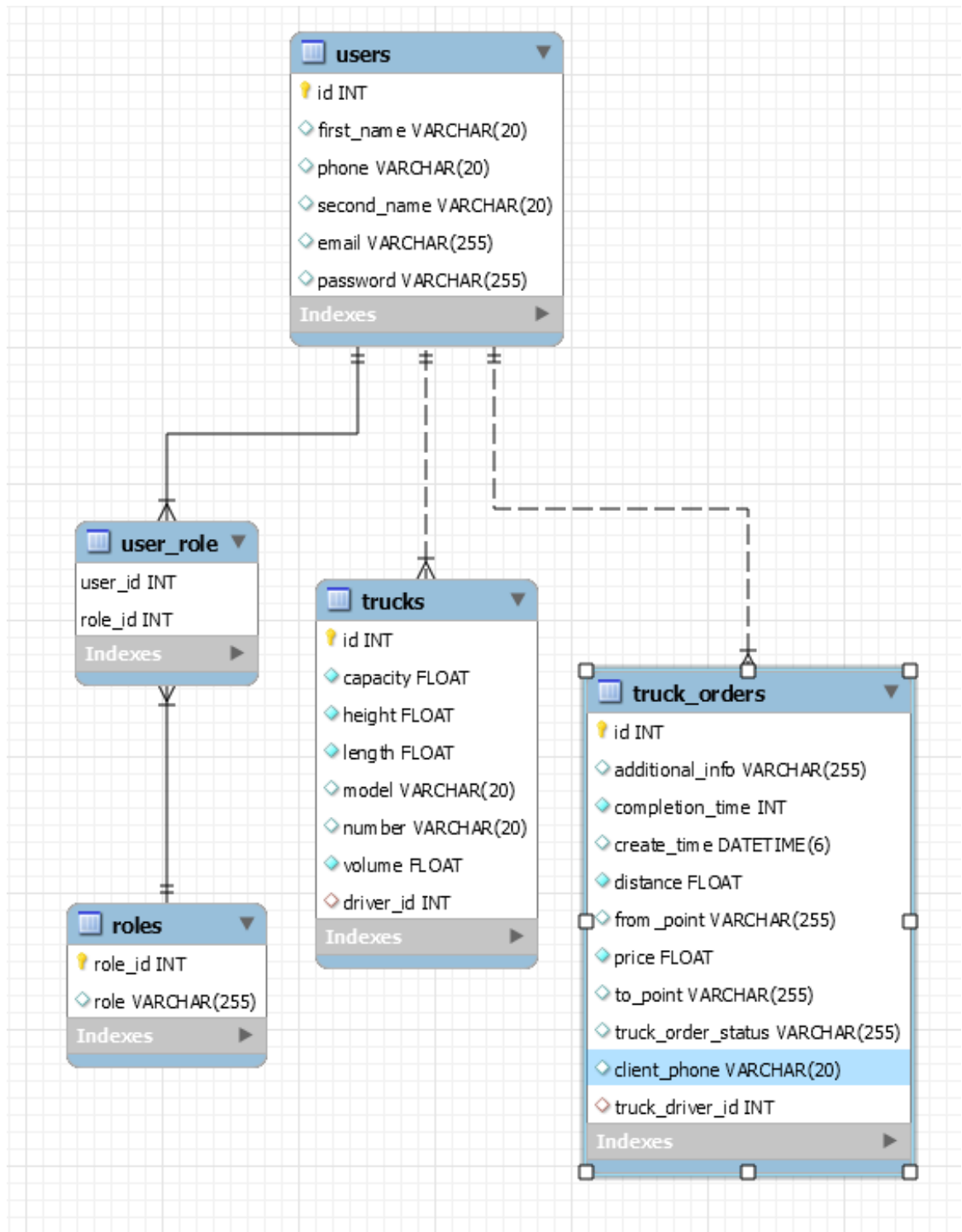


Рисунок 2.6 – Схема бази даних

Таблиці **users** і **roles** пов'язані з іншими за допомогою полей ідентифікаторів. Ці поля є унікальними ідентифікаторами для кожного запису.

Таблиця users містить в собі наступну інформацію:

- ідентифікатор запису (id) – генерується системою для унікальності запису (тип даних INT);
- ім'я (first_name) (тип даних VARCHAR);
- прізвище (second_name) (тип даних VARCHAR);
- телефон (phone) (тип даних VARCHAR);
- електронна пошта (email) (тип даних VARCHAR);
- пароль (password) (тип даних VARCHAR).

Таблиця trucks містить в собі наступну інформацію:

- ідентифікатор запису (id) – генерується системою для унікальності запису (тип даних INT);
- максимальна вага (capacity) (тип даних FLOAT);
- об'єм грузового відсіку (volume) (тип даних FLOAT);
- висота грузового відсіку (height) (тип даних FLOAT);
- довжина грузового відсіку (length) (тип даних FLOAT);
- модель авто (model) (тип даних VARCHAR);
- номер (number) (тип даних VARCHAR);
- водій (driver_id) – зв'язок з таблицею users. Є посиланням на зареєстрованого в системі водія (тип даних INT).

Таблиця truck_orders містить в собі наступну інформацію:

- ідентифікатор запису (id) – генерується системою для унікальності запису (тип даних INT);
- номер телефону (phone) (тип даних VARCHAR);
- пункт відправлення (from_point) – Адреса пункту відправлення (тип даних VARCHAR);
- пункт призначення (to_point) – Адреса пункту призначення (тип даних VARCHAR);
- дата і час створення замовлення (create_time) – (тип даних DATETIME);

- ціна (price) (тип даних FLOAT);
- довжина шляху в кілометрах (distance) (тип даних FLOAT);
- час виконання замовлення в хвилинах (completion_time) (тип даних FLOAT);
- статус замовлення (truck_order_status) (тип даних VARCHAR);
- призначений на виконання водій (truck_driver_id) – зв’язок з таблицею users. Є посиланням на зареєстрованого в системі водія (тип даних INT).

Таблиця roles містить в собі наступну інформацію:

- ідентифікатор запису (role_id) – генерується системою для унікальності запису (тип даних INT);
- назва ролі (role) – назва ролі, яка визначає доступні можливості користувача (тип даних VARCHAR).

Таблиця user_role містить в собі наступну інформацію:

- користувач системи (user_id) – посилання на існуючого в системі користувача (тип даних INT);
- роль (role_id) – посилання на існуючу в системі роль (тип даних INT).

2.5 Побудова об’єктно-орієнтованої моделі

Для написання додатку було вирішено застосовувати архітектурний шаблон проектування MVC. Це фундаментальний паттерн, який знайшов застосування у багатьох технологіях, дав розвиток новим технологіям та полегшує розробку кожного дня. Шаблон проектування MVC передбачає поділ даних програми, призначеного для користувача інтерфейсу і керуючої логіки на три окремих компоненти: Модель, Представлення і Контролер – таким чином, що модифікація кожного компонента може здійснюватися незалежно.

Модель – це бізнес логіка застосунку. Вона є центральним компонентом шаблону MVC і відображає поведінку застосунку, незалежну від інтерфейсу користувача. Модель повинна бути незалежною від інших частин продукту.

Представлення відповідає за відображення даних, отриманих з моделі. Приклад представлення – HTML-сторінка, WPF форма, Windows Form. Одночасно можуть співіснувати кілька виглядів (представлень) однієї і тієї ж інформації, наприклад гістограма для керівництва компанії й таблиці для бухгалтерії.

Контролер одержує вхідні дані й перетворює їх на команди для моделі чи представлення.

Схема шаблону MVC має наступний вигляд (рисунок 2.7):

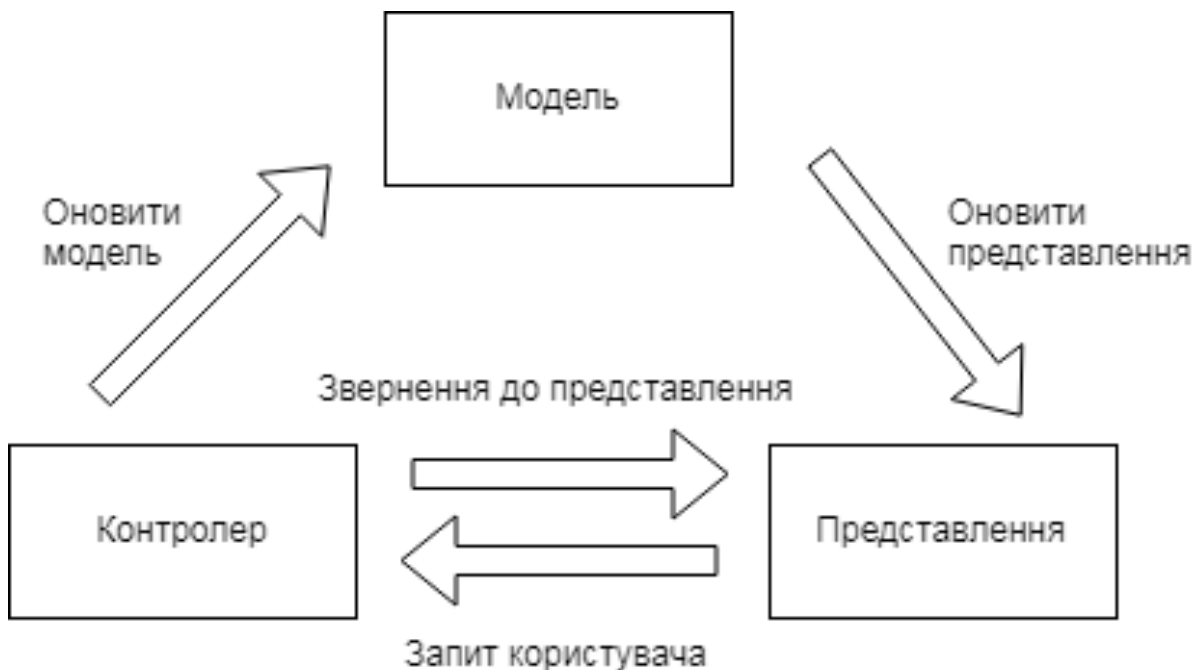


Рисунок 2.7 – Шаблон проектування модель–представлення–контролер

Діаграма класів системи має наступний вигляд (рисунок 2.8):

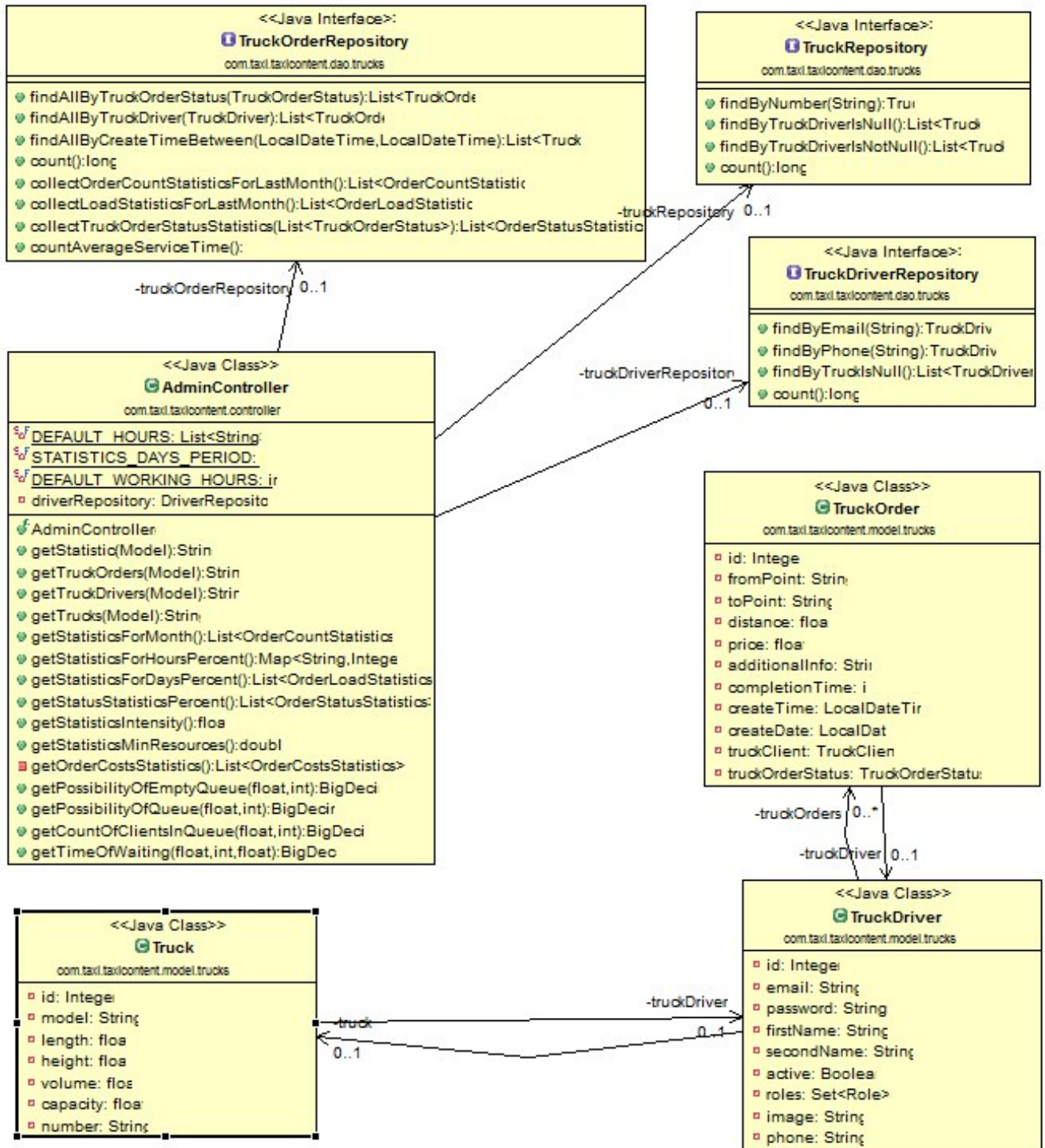


Рисунок 2.8 – Діаграма класів аналітичної системи оцінки ефективності керування потоками замовлень

Клас AdminController відповідає за надання інформації щодо системи і її функціонування. Має наступний набір методів:

- getStatistic – метод для надання даних про поточку кількість ресурсів (авто, водії), кількість замовлень а також відносну величину витрат;
- getTruckOrders – метод для надання списку замовлень;
- getTruckDrivers – метод для надання списку зареєстрованих в системі водіїв;
- getTrucks – метод для надання списку зареєстрованих в системі авто;
- getStatisticsForMonth – надає статистику кількості замовлень на день за останній місяць;
- getStatisticsForHoursPercent – надає статистику надходжень замовлень по годинам у відсотковому співвідношенні;
- getStatisticsForDaysPercent – надає статистику завантаженості ресурсів за день в відсотковому співвідношенні;
- getStatusStatisticsPercent – надає відсоткове співвідношення виконаних і відхилених замовлень;
- getStatisticsIntensity – розраховує інтенсивність навантаження;
- getStatisticsMinResources – розраховує мінімальну необхідну кількість ресурсів (каналів) для коректного функціонування системи;
- getPossibilityOfEmptyQueue – розраховує ймовірність, що канал вільний;
- getPossibilityOfQueue – розраховує ймовірність, що канал зайнятий;
- getCountOfClientsInQueue – розраховує середню кількість клієнтів в черзі;
- getTimeOfWaiting – розраховує середій час очікування клієнтів.

Клас `TruckOrderRepository` відповідає за взаємодіє з базою даних, а саме створює, видаляє і шукає записи з замовленнями. Окрім стандартних методів (`Create`, `Read`, `Update`, `Delete`) Має наступний набір доступних методів:

- `findAllByTruckOrderStatus` – надає список замовлень за статусом;
- `findAllByTruckDriver` – надає список замовлень, призначених конкретному водію;
- `findAllByCreateTimeBetween` – надає список замовлень між заданими датами;
- `count` – надає кількість замовлень, що поступили в систему;
- `collectOrderCountStatisticsForLastMonth` – надає статистичні дані про кількість замовлень по дням за останній місяць;
- `collectLoadStatisticsForLastMonth` – надає статистичні дані щодо завантаження ресурсів по дням за останній місяць;
- `collectTruckOrderStatusStatistics` – надає кількість замовлень для кожного статусу;
- `countAverageServiceTime` – надає інформацію щодо середнього часу виконання замовлення.

Клас `TruckRepository` відповідає за взаємодіє з базою даних, а саме створює, видаляє і шукає записи з авто. Окрім стандартних методів (`Create`, `Read`, `Update`, `Delete`) має наступний набір доступних методів:

- `findByNumber` – шукає авто за його номером;
- `findByTruckDriverIsNull` – шукає авто без призначеного водія;
- `findByTruckDriverIsNotNull` – шукає авто з призначеним водієм;
- `count` – надає кількість зареєстрованих в системі авто.

Клас `TruckDriverRepository` відповідає за взаємодіє з базою даних, а саме створює, видаляє і шукає записи з водіями. Окрім стандартних методів (`Create`, `Read`, `Update`, `Delete`) має наступний набір доступних методів:

- `findByEmail` – шукає водія за його поштою;

- findByPhone – шукає водія за його телефоном;
- findByTruckIsNull – шукає водія без призначеного транспорту;
- count – надає кількість зареєстрованих в системі водіїв.

Клас Truck є моделлю транспорту в системі і має наступні поля:

- id – унікальний ідентифікатор;
- model – назва моделі транспорту;
- length – довжина багажного відсіку авто;
- height – висота багажного відсіку авто;
- volume – об'єм багажного відсіку авто;
- capacity – максимальна вага для транспортування;
- number – номер авто;
- truckDriver – призначений водій.

Клас TruckDriver є моделлю водія в системі і має наступні поля:

- id – унікальний ідентифікатор;
- email – пошта водія;
- password – пароль для особового запису;
- firstName – ім'я;
- secondName – прізвище;
- active – флаг, що визначає чи є запис діючим в системі;
- image – фото водія;
- phone – номер телефону;
- truck – призначене авто;
- truckOrders – призначені замовлення.

Клас TruckOrder є моделлю замовлень в системі і має наступні поля:

- id – унікальний ідентифікатор;
- fromPoint – пункт відправлення;
- toPoint – пункт призначення;
- price – розрахована ціна;
- additionalInfo – додаткова інформація;
- completionTime – час виконання замовлення;

- createDate – дата і час створення замовлення;
- truckDriver – водій;
- truckClient – клієнт;
- truckOrderStatus – поточний статус замовлення.

2.6 Математичне та алгоритмічне забезпечення

Математичне забезпечення – сукупність математичних методів, моделей і алгоритмів обробки інформації, використовуваних при вирішенні функціональних завдань і в процесі автоматизації проектувальних робіт.

Кожна СМО включає в свою структуру деяке число обслуговуючих пристроїв (одиниць, приладів, ліній), які називають каналами обслуговування. Роль каналів можуть грати особи, які виконують ті чи інші операції (касири, оператори, продавці, лінії зв'язку, авто) [2].

Кожна СМО призначена для обслуговування (виконання) деякого потоку заявок (або вимог), що надходять на вхід системи здебільшого не регулярно, а в випадкові моменти часу. Обслуговування заявок, в загальному випадку, також триває не постійно, а в випадкові проміжки часу. Після обслуговування заявки канал звільнюється і стає готовим до прийому наступної заявки. Випадковий характер потоку і часу їх обслуговування призводить до нерівномірної завантаженості СМО: в деякі проміжки часу на вході СМО можуть накопичуватися необслуговувані заявки (вони або стають в чергу, або залишають СМО необслугованими), в інші ж періоди при вільних каналах на вході СМО заявок не буде, що призводить до недовантаження СМО, тобто до простою каналів [3].

Схема СМО зображена на рисунку 2.9.

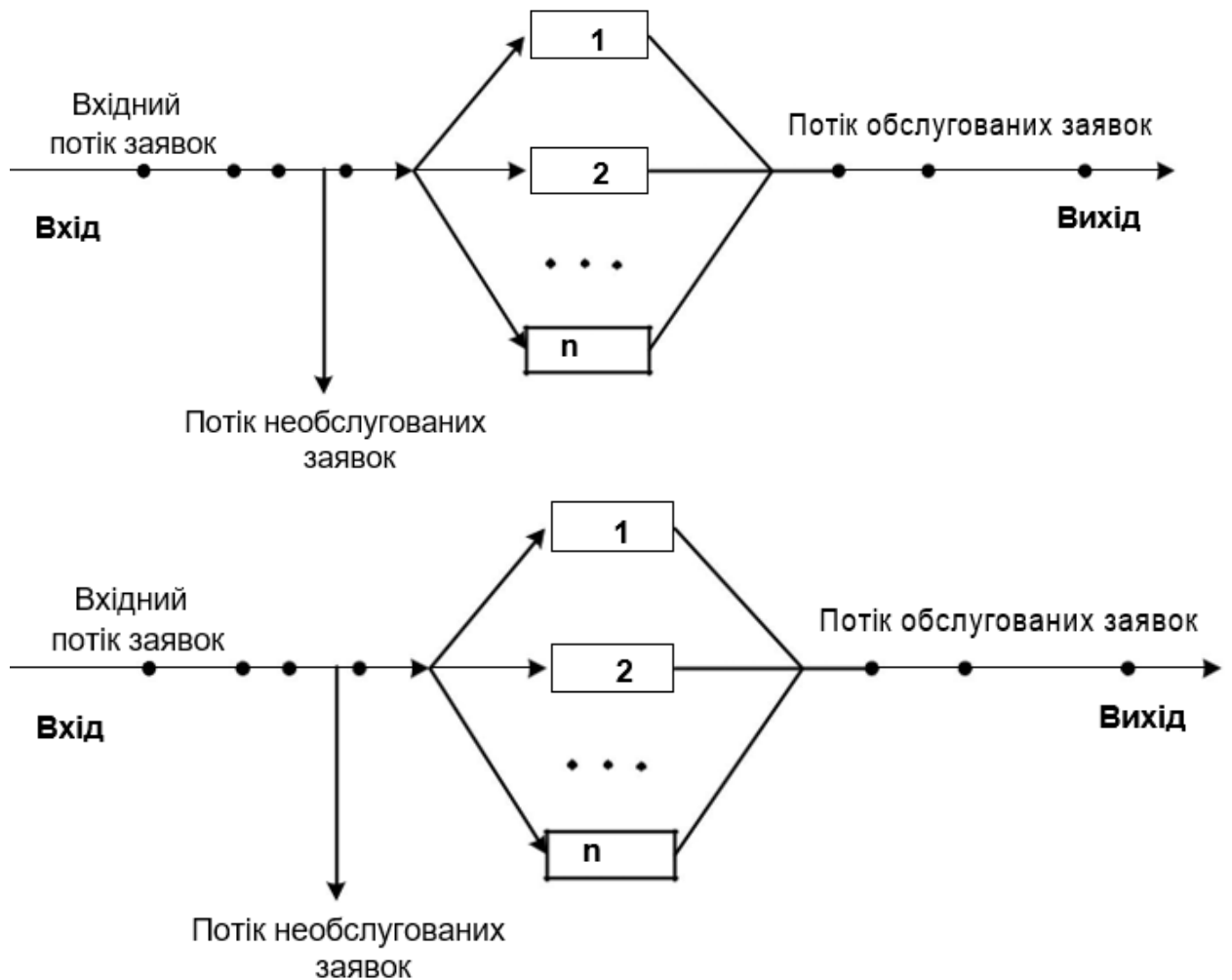


Рисунок 2.9 – Схема роботи СМО

Таким чином, у будь-якій СМО можна виділити наступні основні елементи:

- вхідний потік заявок;
- чергу;
- канали обслуговування;
- вихідний потік обслугованих заявок.

Кожна СМО в залежності від своїх параметрів: характеру потоку заявок, числа каналів обслуговування і їх продуктивності, а також від правил організації роботи, має певну ефективність функціонування

(пропускну спроможність), що дозволяє їй успішно справлятися з потоком заявок [4].

Для рішення задачі розрахунку показників ефективності керування потоками заявок в моделі системи масового обслуговування для дипломної роботи використовуються наступні формули:

Інтенсивність навантаження:

$$\rho = \frac{\lambda}{\mu} \quad (2.1)$$

де μ – інтенсивність потоку обслуговування;

λ – інтенсивність потоку запитів.

Інтенсивність навантаження показує ступінь узгодженості вхідного і вихідного потоків заявок каналу обслуговування і визначає стійкість системи масового обслуговування [5].

Час обслуговування:

$$t_{\text{обсл}} = \frac{1}{\mu} \quad (2.2)$$

де μ – інтенсивність потоку обслуговування;

ймовірність, що канал вільний.

$$p_0 = \frac{1}{\sum_{k=0}^n \frac{\rho^k}{k!}} \quad (2.3)$$

де n – кількість каналів обслуговування;

ρ – інтенсивність навантаження.

Ймовірність відмови (ймовірність того, що канал зайнятий) (доля заявок, що отримали відмову) [6].

$$p_{\text{від}} = \frac{\rho^n}{n!} \quad (2.4)$$

де ρ – інтенсивність навантаження;

n – кількість каналів обслуговування.

Ймовірність обслуговування заявок (ймовірність того, що клієнт буде обслугований):

У системах з відмовами події відмови і обслуговування складають повну групу подій [7], тому: $p_{\text{отк}} + p_{\text{обс}} = 1$

Відносна пропускна здатність: $Q = p_{\text{обс}}$, отже:

$$p_{\text{обс}} = 1 - p_{\text{від}} \quad (2.5)$$

де $p_{\text{від}}$ – ймовірність відмови.

Середнє число каналів, зайнятих обслуговуванням (Середнє число зайнятих каналів) [8]:

$$n_k = \rho \cdot p_{\text{обс}} \quad (2.6)$$

де ρ – інтенсивність навантаження;

$p_{\text{обс}}$ – відносна пропускна здатність.

Середнє число каналів, що перебувають в режимі простою[9]:

$$n_{\text{пр}} = n - n_k \quad (2.7)$$

де n – кількість каналів обслуговування;

n_k – середнє число каналів, зайнятих обслуговуванням.

Абсолютна пропускна здатність [10]:

$$A = p_{\text{обс}} \cdot \lambda \quad (2.8)$$

де $p_{\text{обс}}$ – відносна пропускна здатність;

λ – інтенсивність потоку запитів.

Отже в даному розділі були описані вхідні дані, а також джерела вхідних даних. Описані вихідні дані, що являють собою статистичну і аналітичну інформації щодо ефективності керування потоками заявок в системі масового обслуговування.

В розділі також детально описані таблиці бази даних з описом полів, що зберігаються в системі. Наведена схема бази даних, що показує зв'язки між таблицями, зовнішні і первинні ключі.

Було описано об'єктно-орієнтовану модель з описом архітектури системи, наведенням діаграми класів з описом необхідних для роботи системи методів.

Було описано математичне та алгоритмічне забезпечення з приведенням схеми роботи системи масового обслуговування і наведенням формул для розрахування необхідних характеристик роботи системи.

3 ОПИС ПРИЙНЯТИХ ПРОЕКТНИХ РІШЕНЬ ТА ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

3.1 Засоби розробки

Веб-додаток являє собою веб-сайт, на якому розміщені сторінки з частково або повністю несформованим вмістом. Остаточний вміст формується тільки після того, як відвідувач сайту запросить сторінку з веб-сервера. У зв'язку з тим що остаточний вміст сторінки залежить від запиту, створеного на основі дій користувача, така сторінка називається динамічною [11]. Саме динамічні веб-додатки дозволяють збирати, зберігати і аналізувати дані, отримані від відвідувачів сайту. Вміст сторінок такого додатку зазвичай формується веб-компонентами, наприклад сторінками JavaServer (JSP), Сервлетами і компонентами JavaBean [12], які дозволяють змінювати дані і здійснювати їх тимчасове зберігання, взаємодіяти з базами даних і веб-службами, а також відображати вміст у відповідь на запити клієнтів [13], тому для розробки додатку було обрано мову програмування JAVA 8-ої версії. В якості середовища розробки було обрано IntelliJ IDEA (2020.2). Для спрощення розробки веб-додатку було обрано Spring Framework.

Spring – один з найбільш широко використовуваних фреймворків для розробки веб-додатків, що забезпечує продуману модель програмування і конфігурації[14]. На відміну від інших платформ, Spring фокусується на кількох областях функціонування додатків і надає для них широкий спектр додаткових функцій. Однією з основних особливостей Spring Framework є використання патерну Dependency Injection (DI, впровадження залежності). DI допомагає набагато простіше реалізовувати необхідну для додатків функціональність, а також дозволяє розробляти слабо пов'язані класи, роблячи їх більш універсальними. Для спрощення процесу розгорення веб-додатку було використано Spring Boot. Spring Boot – це корисне доповнення,

метою якого є спрощення створення додатків на основі Spring. Він дозволяє найбільш простим способом створити web-додаток, вимагаючи від розробників мінімум зусиль по його налаштуванню[15].

В якості СУБД було обрано MySQL через високу швидкість і надійність.

3.2 Вимоги до технічного та програмного забезпечення

Для серверної сторони вимоги до технічного та програмного забезпечення наступні:

Обладнання:

- частота процесора 2 Гц;
- оперативна пам'ять 1 Гб;
- дисковий простір 20 Gb.

Програмне забезпечення:

- Java (підтримує специфікацію Servlet 2.5 і вище);
- СУБД MySQL.

Для клієнтської сторони:

- доступ до мережі інтернет;
- браузер з підтримкою CSS, JavaScript.

3.3 Опис програмної реалізації

Додаток являє собою динамічний веб-проект, що містить в собі 6 пакетів. Пакет `com.trucks.truckcontent.configuration` містить в собі конфігурацію веб-додатку для налаштувань доступу різних ролей користувачів в системі.

Діаграму класів пакету `com.trucks.truckcontent.configuration` наведено на рисунку 3.1:

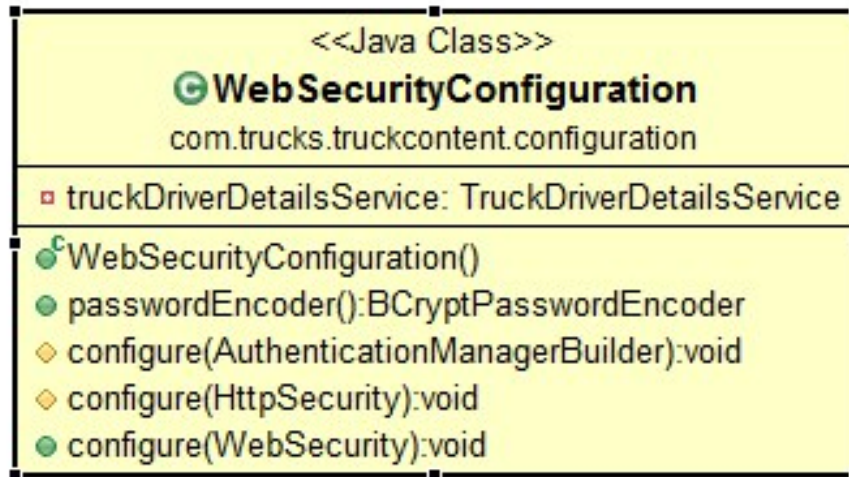


Рисунок 3.1 – Діаграма класів пакету `com.trucks.truckcontent.configuration`

Клас `WebSecurityConfiguration` слугує для конфігурації налаштувань доступу користувачів в системі в залежності від ролі.

Методи `configure` дозволяють гнучко налаштувати доступ в систему для існуючих ролей. Можливості переглядати списки всіх замовлень, списки водіїв, замовлень і аналітичні показники системи, а також редагувати існуючі ресурси в системі мають змогу лише користувачі з роллю ADMIN (адміністратор). Переглядати список поточних замовлень, список водіїв і призначати водія на замовлення має змогу користувач з роллю DISP (диспетчер). Переглядати список своїх замовлень і завершувати їх виконання мають змогу користувачі з роллю DRIVER (водій). Можливість авторизації та створення замовлень є у всіх користувачів.

Метод `passwordEncoder` дозволяє використовувати хешування паролю при використанні його в базі даних. Для кодування використовується потужний алгоритм `BCrypt` що часто використовується для безпечного зберігання паролів.

Пакет `com.trucks.truckcontent.controller.trucks` містить в собі основні класи контролери, що приймають запити в системі і обробляють їх, повертаючи результат обробки. Діаграму класів пакету `com.trucks.truckcontent.controller.trucks` наведено на рисунку 3.2:

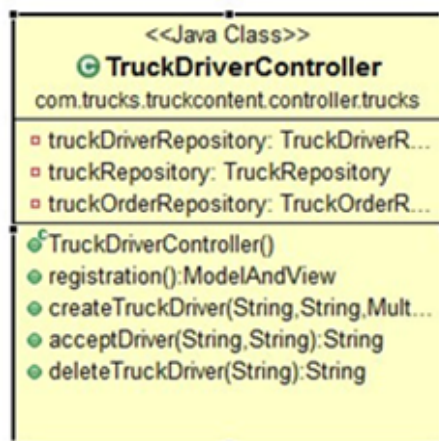
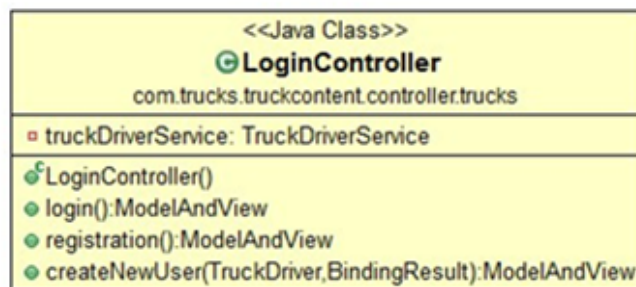
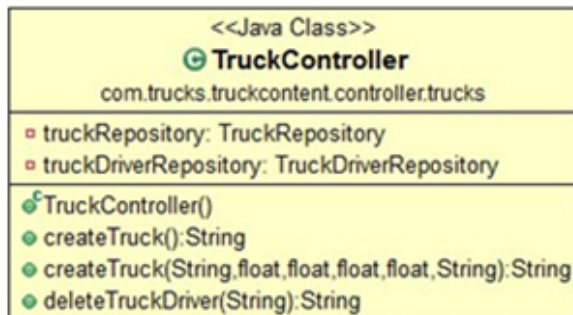
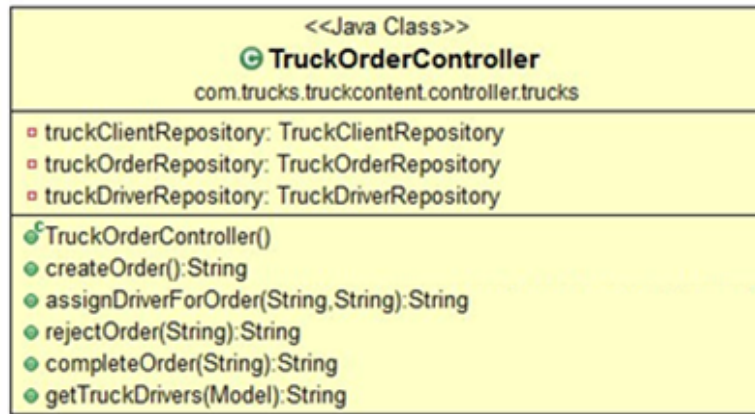


Рисунок 3.2 – Діаграма класів пакету
com.trucks.truckcontent.controller.trucks

Клас `TruckOrderController` обробляє запити пов'язані з замовленнями в системі. Має наступний набір методів:

- `createOrder` – метод для створення замовлень в системі. Приймає набір параметрів, необхідних для створення замовлення, а саме номер телефону замовника, пункт відправки, пункт призначення, дистанція, ціна, додаткова інформація. Цей набір даних повертається з форми, яку заповнює користувач при створенні замовлення. Час створення замовлення призначається при створенні запису в базі даних, статус для нових замовлень стає `PENDING` (очікує в черзі);

- `assignDriverForOrder` – метод для призначення водія на конкретне замовлення. Метод приймає унікальний ідентифікатор замовлення і унікальний ідентифікатор водія;

- `rejectOrder` – метод для переведення замовлення в статус «відхилено». При цьому в системі розраховується кількість витраченого на очікування часу на основі дати створення замовлення і дати його відхилення;

- `completeOrder` – метод для переведення замовлення в статус «завершено». При цьому в системі розраховується кількість витраченого на замовлення часу на основі дати створення замовлення і дати його завершення водієм;

- `getTruckDriverOrders` – метод для отримання списку поточних замовлень конкретного водія. Список замовлень містить лише замовлення зі статусом «в процесі виконання».

- `getTruckDriversOrdersHistory` – метод для отримання списку всіх замовлень конкретного водія. Список замовлень містить всі виконані або відхилені замовлення конкретного водія.

Клас `TruckController` обробляє запити пов'язані з транспортом в системі. Має наступний набір методів:

- `createTruck` – метод для створення об'єктів транспорту і занесення їх до бази даних. Для створення запису надається необхідний набір

параметрів, а саме довжина грузового відсіку, висота грузового відсіку, об'єм грузового відсіку, максимальна вага, модель і номер авто;

– `deleteTruck` – метод для видалення транспорту з системи. Для видалення запису в метод надається унікальний ідентифікатор авто в базі даних.

Клас `TruckDriverController` обробляє запити пов'язані з користувачами системи. Має наступний набір методів:

– `registration` – метод для реєстрації водіїв в системі. Для занесення водія в базу даних, необхідно надати набір параметрів, а саме ім'я, прізвище, телефон, email, пароль. Роль `DRIVER` користувача призначається для запису.

– `assignAuto` – метод для призначення транспорту водіям. В метод надається унікальний ідентифікатор водія та унікальний ідентифікатор транспорту.

Клас `LoginController` обробляє запити пов'язані з користувачами системи. Має наступний набір методів:

- `login` – метод для авторизації користувачів в системі;
- `createNewUser` – метод для створення користувачів в системі.

Пакет `com.trucks.truckcontent.controller` містить в собі клас `AdminController`. Цей клас призначений для обробки запитів адміністратора щодо отримання даних о системі, а також запитів на аналіз роботи системи і оцінки ефективності керування потоками заявок в системі. Діаграму класу пакету наведено на рисунку 3.3:

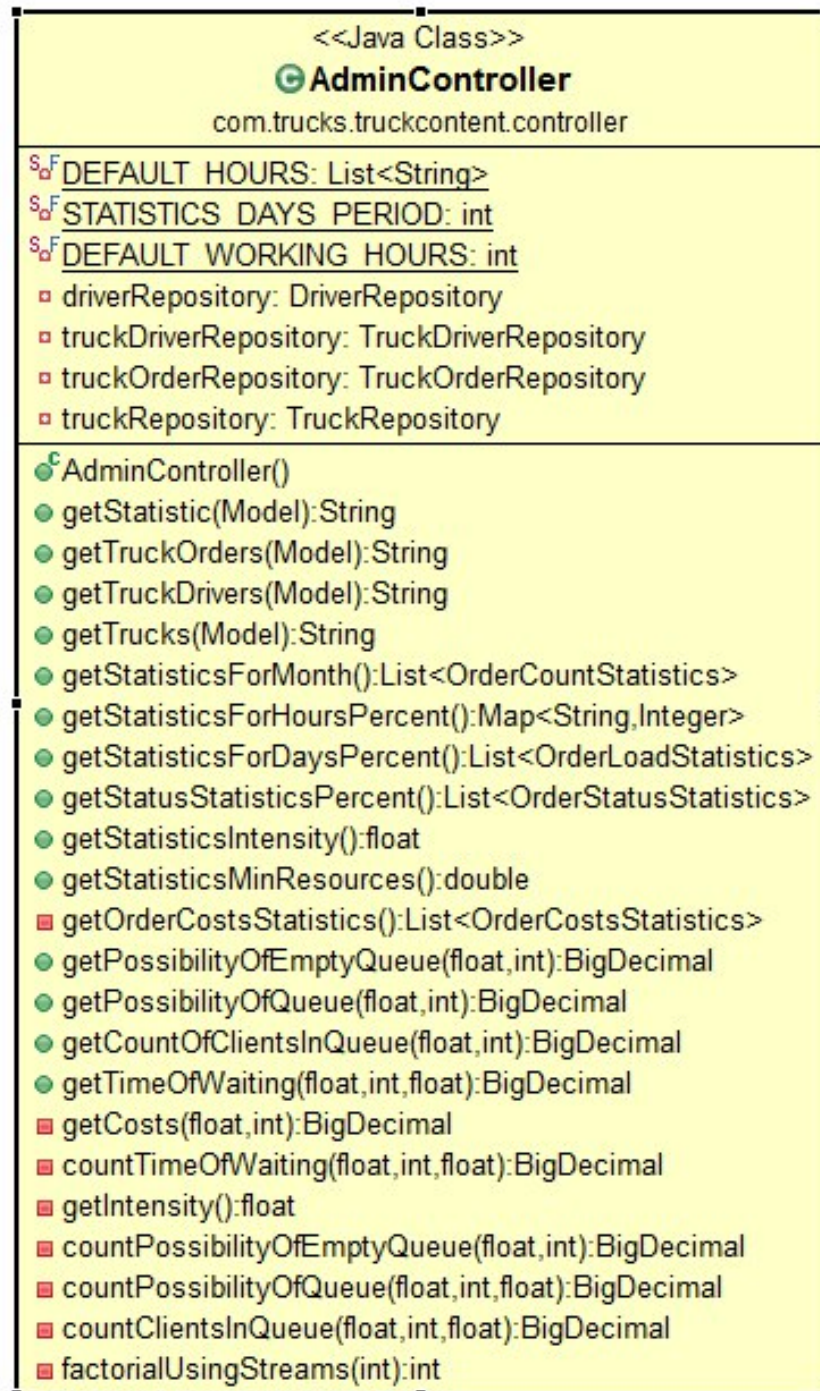


Рисунок 3.3 – Діаграма класів пакету `com.trucks.truckcontent.controller`

Клас `AdminController` має наступний набір методів:

– `getStatistic` – метод для отримання даних щодо функціонування системи, а саме кількості ресурсів (водіїв і авто), кількості замовлень. Також надає інформацію щодо інтенсивності навантаження системи масового

обслуговування, середній час обслуговування, розрахований на основі часу виконання завершених замовлень, ймовірність пустої черги, ймовірність відмови, абсолютну пропускну здатність і відносну величину витрат;

- `getTruckOrders` – метод для отримання списку замовлень;

- `getTruckDrivers` – метод для отримання списку зареєстрованих в системі водіїв;

- `getTrucks` – метод для отримання списку зареєстрованих в системі авто;

- `getStatisticsForMonth` – метод для отримання статистики надходження кількості замовлень на день. На аналітичній сторінці візуалізується у вигляді лінійного графу;

- `getStatisticsForHoursPercent` – метод для отримання статистики надходження замовлень по годинам в процентному співвідношенні. На аналітичній сторінці візуалізується за допомогою гісторграми;

- `getStatisticsForDaysPercent` – метод для отримання статистики завантаженості ресурсів у відсотках. Завантаженість розраховується на основі стандартного робочого часу водіїв, що складає 10 годин на день. На аналітичній сторінці візуалізується за допомогою гісторграми;

- `getStatusStatisticsPercent` – метод для отримання відсоткового співвідношення виконаних і відхилених замовлень. На аналітичній сторінці візуалізується за допомогою секторної діаграми;

- `getStatisticsIntensity` – метод для розрахунку інтенсивності навантаження системи;

- `getStatisticsMinResources` – метод для розрахунку необхідної кількості каналів для функціонування підприємства як системи масового обслуговування;

- `getOrderCostsStatistics` – метод для розрахунку відносного показника витрат;

- `getPossibilityOfEmptyQueue` – метод для розрахунку ймовірності пустої черги;

- `getPossibilityOfQueue` – метод для розрахунку ймовірності наявної черги;
- `getCountOfClientsInQueue` – метод для розрахунку середньої кількості клієнтів в черзі;
- `getTimeOfWaiting` – метод для розрахунку середньої кількості часу, необхідного для виконання замовлення;
- `factorialUsingStreams` – метод для розрахунку факторіалів.

Пакет `com.trucks.truckcontent.dao.trucks` містить в собі класи для взаємодії з базою даних. Діаграма класів пакету наведена на рисунку 3.4.

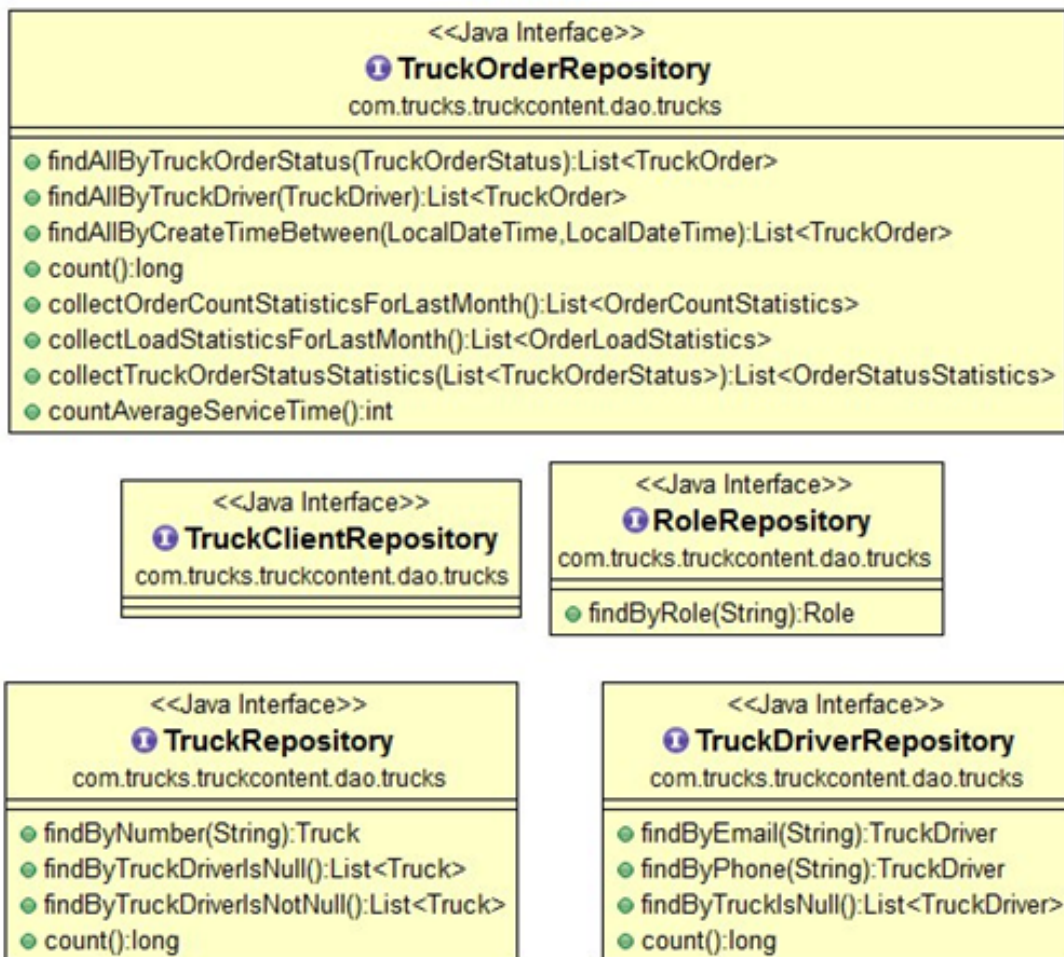


Рисунок 3.4 – Діаграма класів пакету `com.trucks.truckcontent.dao.trucks`

Клас `TruckOrderRepository`, окрім стандартних методів для роботи з базою даних (отримання списку всіх замовлень, збереження, видалення, оновлення, створення записів), що були успадковані завдяки можливостям `Spring Framework`, має наступний набір додаткових методів:

- `findAllByTruckOrderStatus` – метод для отримання всіх замовлень за конкретним статусом, що передається в якості параметра для методу;
- `findAllByTruckDriver` – метод для отримання списку замовлень за конкретним водієм;
- `findAllByCreateTimeBetween` – метод для отримання списку замовлень між заданими датами;
- `count` – метод для отримання кількості замовлень;
- `collectOrderCountStatisticsForLastMonth` – метод для отримання статистики надходження кількості замовлень на день за останній місяць
- `collectLoadStatisticsForLastMonth` – метод для отримання статистики завантаженості ресурсів;
- `collectTruckOrderStatusStatistics` – метод для отримання кількості замовлень за кожним статусом;
- `countAverageServiceTime` – метод для отримання середнього часу виконання замовлень.

Клас `TruckDriverRepository`, окрім стандартних методів для роботи з базою даних (отримання списку всіх користувачів, збереження, видалення, оновлення, створення записів), що були успадковані завдяки можливостям `Spring Framework`, має наступний набір додаткових методів:

- `findByEmail` – метод для отримання користувача за поштою;
- `findByPhone` – метод для отримання користувача за номером телефону;
- `findById` – метод для отримання транспорту за його ідентифікатором;
- `findByTruckIsNull` – метод для отримання списку водіїв без призначеного транспорту;

- `findByTruckIsNotNull` – метод для отримання списку водіїв з призначеним транспортом;

- `count` – метод для отримання кількості користувачів.

Клас `TruckClientRepository` має набір стандартних методів для роботи з базою даних (отримання списку всіх клієнтів, збереження, видалення, оновлення, створення записів), що були успадковані завдяки можливостям `Spring Framework`.

Клас `RoleRepository` має набір стандартних методів для роботи з базою даних (отримання списку всіх адміністраторів, збереження, видалення, оновлення, створення записів), що були успадковані завдяки можливостям `Spring Framework`. Також має додатковий метод `findByRole` для отримання списку користувачів за конкретною роллю.

Клас `TruckRepository`, окрім стандартних методів для роботи з базою даних (отримання списку всіх авто, збереження, видалення, оновлення, створення записів), що були успадковані завдяки можливостям `Spring Framework`, має наступний набір додаткових методів:

- `findByNumber` – метод для отримання транспорту за його номером;

- `findById` – метод для отримання транспорту за його ідентифікатором;

- `findByTruckDriverIsNull` – метод для отримання списку авто без призначеного водія;

- `findByTruckDriverIsNotNull` – метод для отримання списку авто з призначеним водієм;

- `count` – метод для отримання кількості зареєстрованих в системі авто.

Пакет `com.trucks.truckcontent.model.trucks` містить в собі класи для взаємодії з базою даних. Діаграма класів пакету наведена на рисунку 3.5:

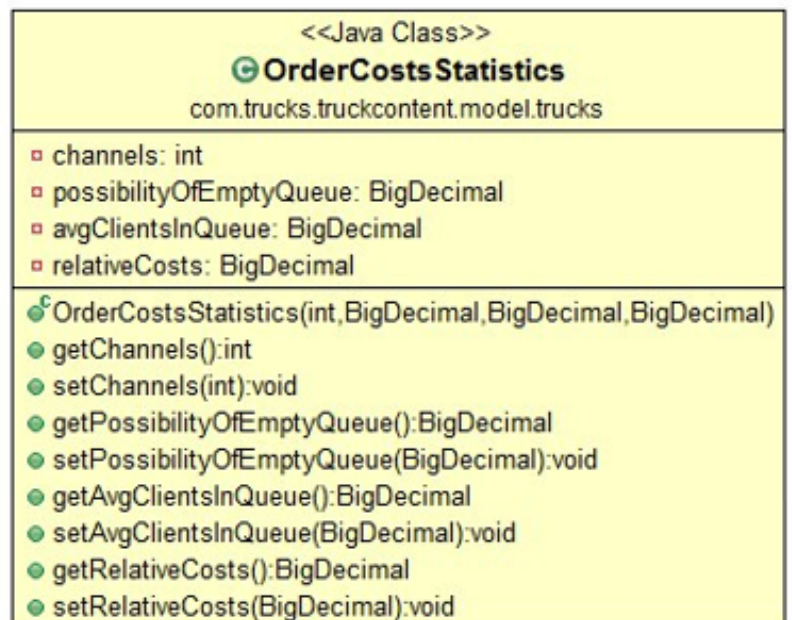
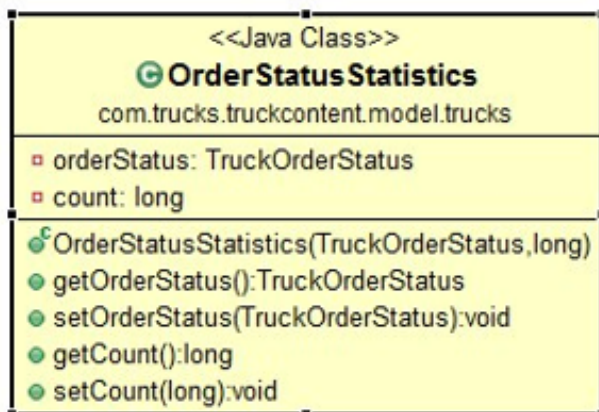
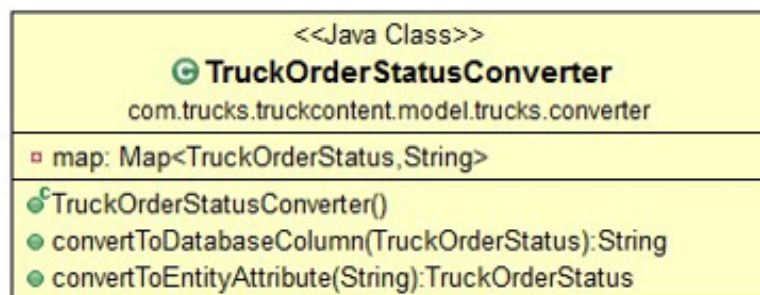
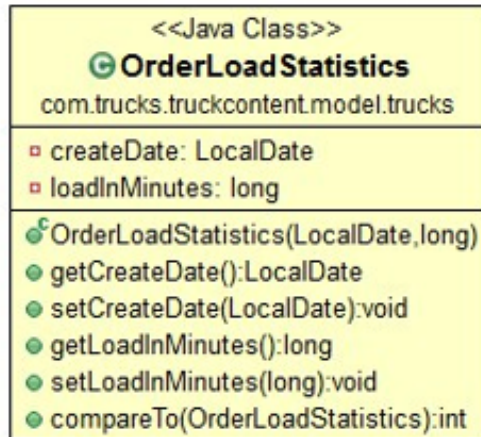
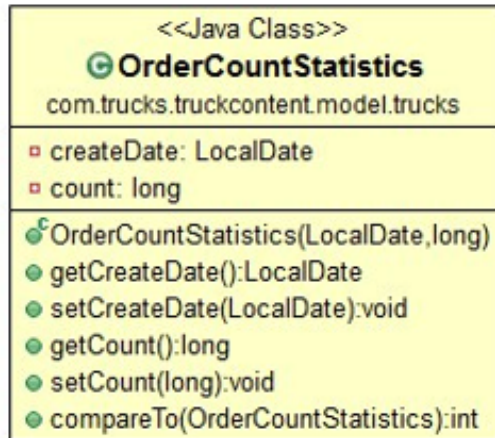


Рисунок 3.5 – Діаграма класів пакету ua.nure.levytska.web.listener

Клас `TruckOrderStatusConverter` слугує для конвертації об'єктів статусу замовлення в записи для бази даних.

Клас `OrderCountStatistics` є моделлю статистичних даних щодо кількості замовлень на день. Містить в собі наступні поля:

- `createDate` – дата;
- `count` – кількість замовлень, що поступили в систему.

Клас `OrderStatusStatistics` є моделлю статистичних даних кількості замовлень в системі з конкретним статусом. Містить в собі наступні поля:

- `orderStatus` – статус замовлення;
- `count` – кількість замовлень, що поступили в систему.

Клас `OrderLoadStatistics` є моделлю статистичних даних навантаження ресурсів системи. Містить в собі наступні поля:

- `createDate` – дата;
- `loadInMinutes` – сума витрачених хвилин на виконання замовлень всіх каналів обслуговування.

Клас `OrderCostsStatistics` є моделлю аналітичних даних відносної величини витрат за конкретної кількості каналів обслуговування. Містить в собі наступні поля:

- `channels` – кількість каналів обслуговування;
- `possibilityOfEmptyQueue` – ймовірність пустої черги;
- `avgClientsInQueue` – середня кількість клієнтів в черзі;
- `relativeCosts` – відносна величина витрат.

Пакет `com.trucks.truckcontent.model.trucks` містить в собі основні моделі системи, що являють собою такі сутності як водій/користувач, замовлення, клієнт, транспорт і роль. Діаграма класів пакету наведена на рисунку 3.6.

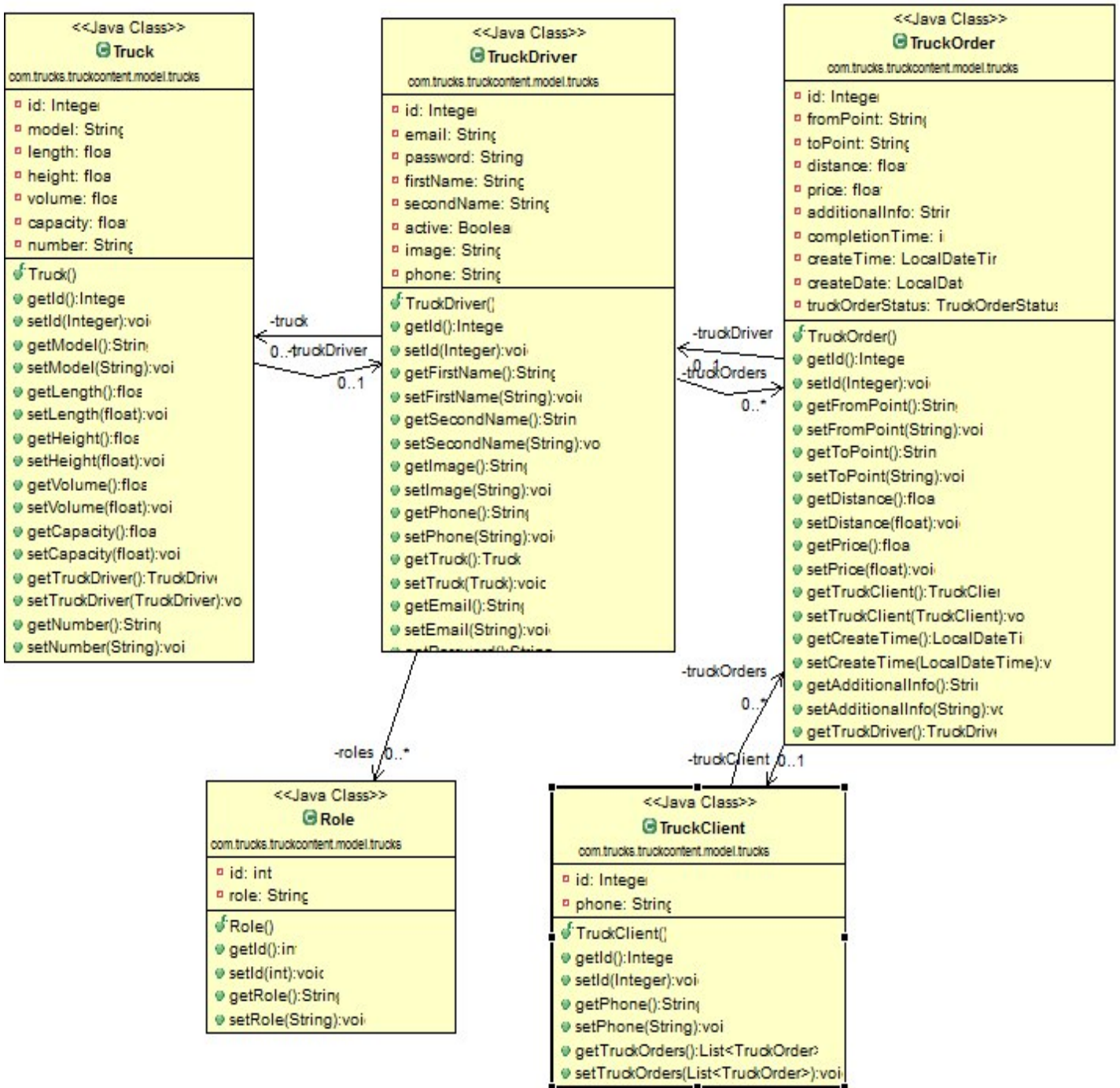


Рисунок 3.6 – Діаграма класів пакету com.trucks.truckcontent.model.trucks

Клас Truck є моделлю транспорту в системі в системі і має наступні поля:

- id – унікальний ідентифікатор;
- model – назва моделі транспорту;

- length – довжина багажного відсіку авто;
- height – висота багажного відсіку авто;
- volume – об'єм багажного відсіку авто;
- capacity – максимальна вага для транспортування;
- number – номер авто.

Клас `TruckDriver` є моделлю водія в системі і має наступні поля:

- id – унікальний ідентифікатор;
- email – пошта водія;
- password – пароль для особового запису;
- firstName – ім'я;
- secondName – прізвище;
- active – флаг, що визначає чи є запис діючим в системі;
- image – фото водія;
- phone – номер телефону;
- truck – призначене авто;
- truckOrders – призначені замовлення.

Клас `TruckOrder` є моделлю замовлень в системі і має наступні поля:

- id – унікальний ідентифікатор;
- fromPoint – пункт відправлення;
- toPoint – пункт призначення;
- price – розрахована ціна;
- additionalInfo – додаткова інформація;
- completionTime – час виконання замовлення;
- createDate – дата і час створення замовлення;
- truckDriver – водій;
- truckClient – клієнт;
- truckOrderStatus – поточний статус замовлення.

Клас `TruckClient` є моделлю клієнта в системі і має наступні поля:

- id – унікальний ідентифікатор;
- phone – номер телефону.

Клас `Role` є моделлю ролі в системі і має наступні поля:

- `id` – унікальний ідентифікатор;
- `role` – назва ролі.

Пакет `com.trucks.truckcontent.service` містить в собі класи-сервіси для коректної авторизації. Діаграма класів пакету наведена на рисунку 3.7:

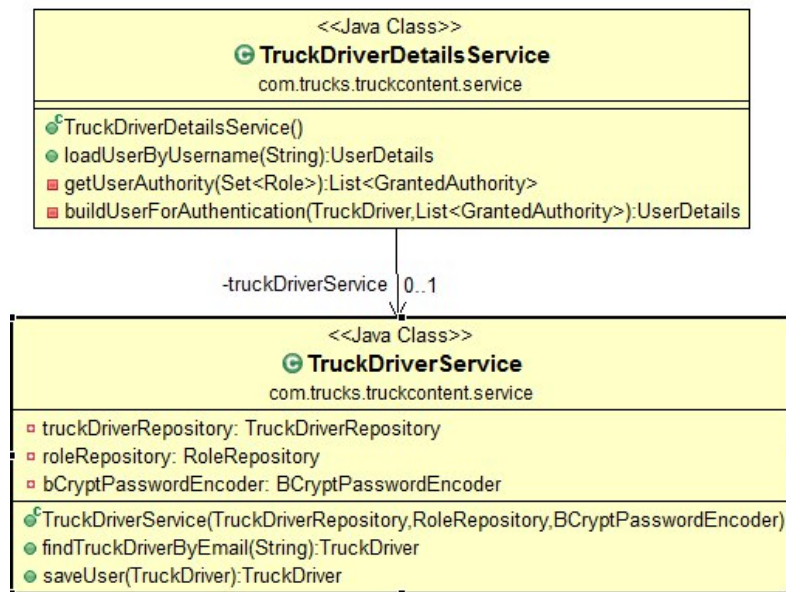


Рисунок 3.7 – Діаграма класів пакету `com.trucks.truckcontent.service`

Пакет `com.trucks.truckcontent` містить в собі клас для ініціації веб-додатку. Діаграма класів пакету наведена на рисунку 3.8:

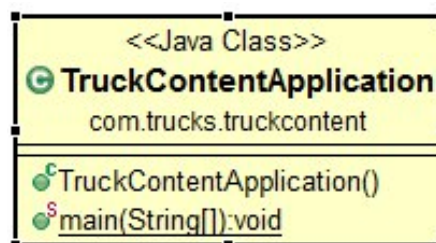
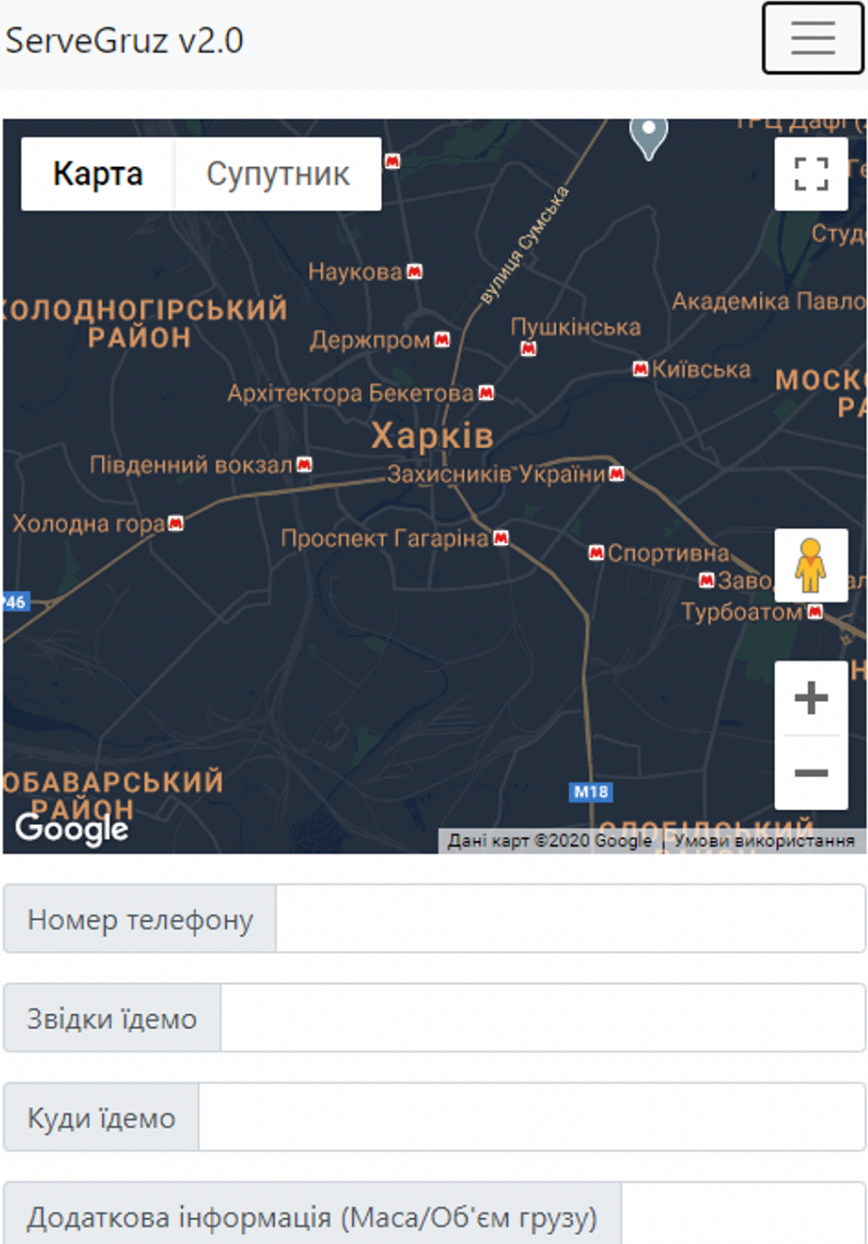


Рисунок 3.8 – Діаграма класів пакету `com.trucks.truckcontent`

3.4 Розробка інтерфейсу додатку

Для початку роботи з веб-додатком, він має бути розгорнутий в мережі, використовуючи хостинг, або розгорнутий локально, використовуючи можливості SpringBoot.

Взаємодія з додатком, з точки зору користувача, починається з основної сторінки для створення замовлень (рисунок 3.9):



ServeGruz v2.0

Карта Супутник

Харків

Номер телефону

Звідки їдемо

Куди їдемо

Додаткова інформація (Маса/Об'єм грузу)

Рисунок 3.9 – Основна сторінка веб-додатку

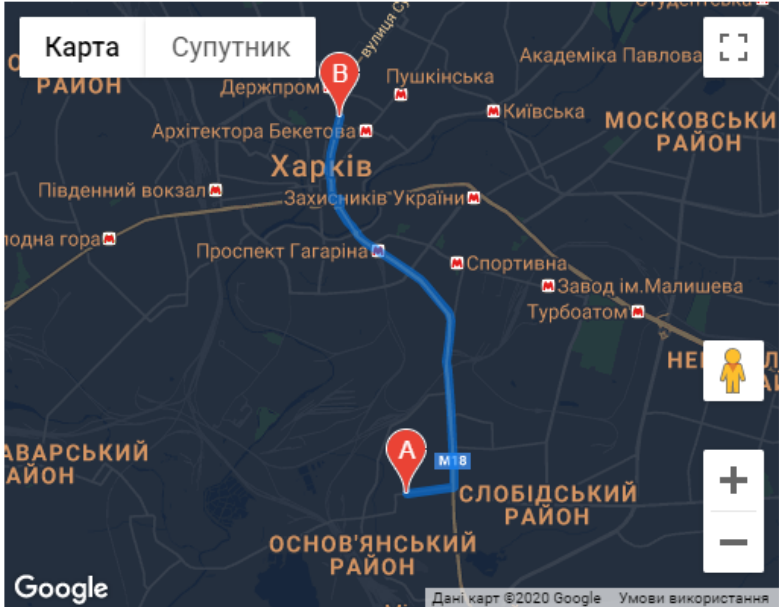
З цієї сторінки користувач має змогу здійснити замовлення. Для замовлення необхідно заповнити необхідні поля в формі, після чого система розрахує додаткові дані, такі як дистанція шляху, ціна обслуговування і побудує приблизний маршрут (рисунок 3.10):

ServeGruz v2.0
☰

Карта

Супутник

📏



Номер телефону +38066918777

Звідки їдемо вулиця Одеська, Харків, Харківська область, Украї

Куди їдемо Харківський національний університет будівництва

Додаткова інформація (Маса/Об'єм грузу)

Маршрут побудовано

Відстань: 7,7 км

Ціна: 92 грн

Замовити транспорт

localhost:8083/#

Рисунок 3.10 – Заповнення форми для замовлення

Після відправлення форми, користувач побачить повідомлення про успішно створене замовлення (рисунок 3.11):

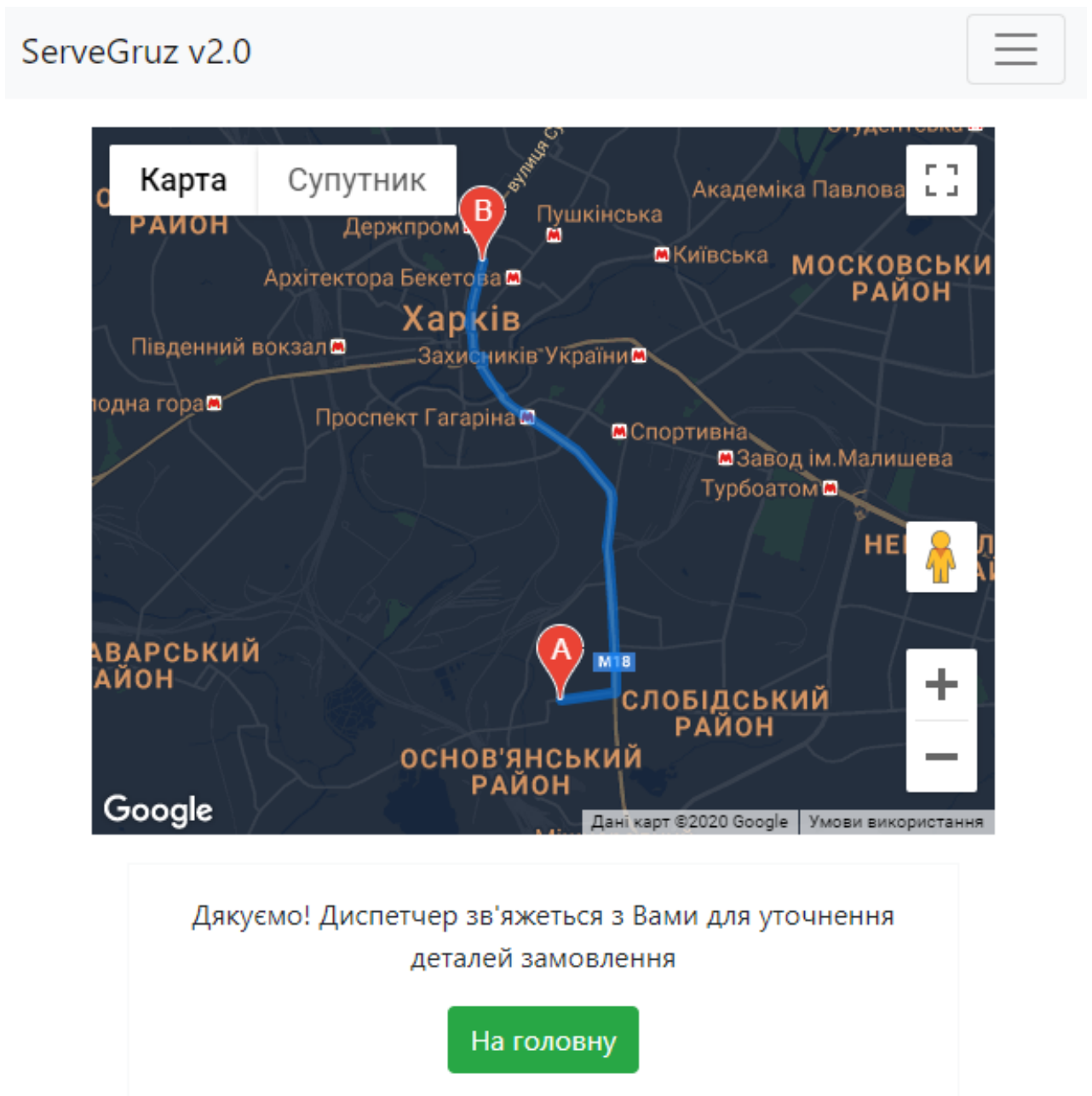


Рисунок 3.11 – Повідомлення про успішно створене замовлення

Розглянемо взаємодію з веб-додатком з точки зору диспетчера. Для початку роботи диспетчеру необхідна авторизація в системі. Для авторизації в системі необхідно перейти на сторінку для входу (рисунок 3.12):

ServeGruz v2.0 ☰

Авторизація

Рисунок 3.12 – Форма для входу в систему на сторінці авторизації

Після входу в систему диспетчер бачить свою версію веб-додатку. В можливості диспетчера входить перегляд сторінки з замовленнями, на якій він має змогу переглянути інформацію щодо кожного замовлення, а також призначити водія на виконання (рисунок 3.13):

Домашня [Замовлення](#) Поточний користувач: disp@gmail.com

Замовлення

Відобразити результатів Пошук:

Замовник	Маршрут	Відстань	Ціна	Додаткова інформація	Час виконання	Виконавець	Статус	Дії
123	вулиця Нетеченська, 25, Харків, Харківська область, Україна - вулиця Державінська, 38, Харків, Харківська область, Україна	2.555 км	31.0 грн	Пісок 4т	14.11.20 15:48 - 80 хв.	Іван Фоменко	Відхилено	<input type="button" value="Замовлення фіналізовано"/>
232323	вулиця Гвардійців-Широнінців, 33, Харків, Харківська область, Україна - вулиця Літературна, 5, Харків, Харківська область, Україна	9.77 км	117.0 грн	Телевізори 50дм 10шт	01.12.20 22:21 - 122 хв.	Юрій Качанов	В процесі виконання	<input type="button" value="Призначити"/> <input type="button" value="Відхилити"/>
232323	вулиця Гвардійців-Широнінців, 33, Харків, Харківська область, Україна - вулиця Літературна, 5, Харків, Харківська область, Україна	9.77 км	117.0 грн	Телевізори 50дм 10шт	01.12.20 22:21 - 122 хв.	Іван Фоменко	Виконано	<input type="button" value="Замовлення фіналізовано"/>
+38066918777	вулиця Одеська, Харків, Харківська область, Україна - Харківський національний університет будівництва та архітектури ЦЕНТР ЗАОЧНОГО НАВЧАННЯ, вулиця Сумська, Харків, Харківська область, Україна	7.703 км	92.0 грн		12.12.20 15:35 - Незавершений	Не встановлено	Очікує в черзі	<input type="button" value="Призначити"/> <input type="button" value="Відхилити"/>
+380669143453	вулиця Нетеченська, 25, Харків, Харківська область, Україна - магазин обуви Rieker, вулиця Пушкінська, 43, Харків, Харківська область, Україна	2.325 км	28.0 грн	Перевезення столів 100x40	08.12.20 17:18 - 100 хв.	Іван Фоменко	Виконано	<input type="button" value="Замовлення фіналізовано"/>

Рисунок 3.13 – Сторінка відображення замовлень в системі для роботи диспетчера

За допомогою можливостей пошуку, диспетчер має змогу отримати список замовлень за статусом. За допомогою елементів інтерфейсу на панелі «Дії», диспетчер має змогу відхилити замовлення в разі, якщо всі водії на даний момент перевантажені, або в автопарку нема транспорту з необхідними характеристиками грузового відсіку. Також диспетчер має змогу призначити на виконання водія зі списку (рисунок 3.14-3.15):

Домашня Замовлення Поточний користувач: disp@gr

Замовлення

Відобразити 10 результатів Пошук: Очік

Замовник	Маршрут	Відстань	Ціна	Додаткова інформація	Час виконання	Виконавець	Статус	Дії
+38066918777	вулиця Одеська, Харків, Харківська область, Україна - Харківський національний університет будівництва та архітектури ЦЕНТР ЗАОЧНОГО НАВЧАННЯ, вулиця Сумська, Харків, Харківська область, Україна	7.703 км	92.0 грн		12.12.20 15:35 - Незавершений	Не встановлено	Очікує в черзі	Призначити Відхилити
+380669143453	вулиця Нетеченська, 25, Харків, Харківська область, Україна - вулиця Воробйова, 4, Харків, Харківська область, Україна	1.712 км	21.0 грн	Пісок 4т	08.12.20 20:44 - Незавершений	Не встановлено	Очікує в черзі	Призначити Відхилити

Відображено записи з 1 до 2 з 2 (пошук проведено серед 35 записів)

Назад 1 Далі

Рисунок 3.14 – Фільтрація замовлень

Оберіть транспорт

Відобразити 10 результатів Пошук:

Модель	Довжина грузового відсіку	Висота грузового відсіку	Об'єм грузового відсіку	Максимальна вага	Номер	Активні замовлення	Дії
Hyundai Mighty	12.0 м	2.0 м	43.0 м ³	8.0 т	AX 2323 CY	2	Призначити
ISUZU ELF	6.0 м	2.0 м	21.6 м ³	5.0 т	AX 2333 CY	0	Призначити
MAN TG510A	8.0 м	2.4 м	34.0 м ³	6.0 т	AX 2323 CЦ	2	Призначити
Scania 5-Series	5.0 м	2.0 м	18.0 м ³	6.0 т	AX 0001 CY	2	Призначити
Volvo FH	9.0 м	1.8 м	29.6 м ³	8.0 т	AX 2222 CB	4	Призначити

Відображено записи з 1 до 5 з 5

Назад 1 Далі

Назад

Рисунок 3.15 – Призначення водія на виконання замовлення

Під час призначення водія, диспетчер має змогу бачити повну інформацію щодо транспорту, а також кількість активних замовлень кожного водія.

Розглянемо взаємодію з веб-додатком з точки зору водія. Для початку роботи водію також необхідна авторизація в системі.

Після входу в систему водій бачить свою версію веб-додатку. В можливості водія входить перегляд сторінки з призначеними йому поточними замовленнями, що очікують обслуговування, та перегляд сторінки з його фіналізованими замовленнями. Активні замовлення водій має змогу завершити за допомогою кнопки «Завершити», що переведе замовлення в статус «виконано» (рисунки 3.16-3.17):

Домашня Мої замовлення Мої історія Поточний користувач: 2@gmail.com

Замовлення

Відобразити 10 результатів Пошук:

Замовник	Маршрут	Відстань	Ціна	Додаткова інформація	Виконавець	Дії
380669143453	Московський проспект, 118, Харків, Харківська область, Україна - вулиця Пушкінська, 5, Харків, Харківська область, Україна	4.482 км	54.0 грн	Пісок 4т	Іван Фоменко	Завершити

Відображено записи з 1 до 1 з 1 [Назад](#) [Далі](#)

Рисунок 3.16 – Активні замовлення водія

Домашня Мої замовлення Мої історія Поточний користувач: 2@gmail.com

Замовлення

Відобразити 10 результатів Пошук:

Замовник	Маршрут	Відстань	Ціна	Додаткова інформація	Виконавець	Дії
123	вулиця Нетеченська, 25, Харків, Харківська область, Україна - вулиця Державінська, 38, Харків, Харківська область, Україна	2.555 км	31.0 грн	Пісок 4т	Іван Фоменко	Відхилено
232323	вулиця Гвардійців-Широнінців, 33, Харків, Харківська область, Україна - вулиця Літературна, 5, Харків, Харківська область, Україна	9.77 км	117.0 грн	Телевізори 50дм 10шт	Іван Фоменко	Виконано
+380669143453	вулиця Нетеченська, 25, Харків, Харківська область, Україна - магазин обуви Rleker, вулиця Пушкінська, 43, Харків, Харківська область, Україна	2.325 км	28.0 грн	Перевезення столів 100x40	Іван Фоменко	Виконано
+380669143453	вулиця Державінська, 38, Харків, Харківська область, Україна - вулиця Пушкінська, 5, Харків, Харківська область, Україна	3.553 км	43.0 грн	Перевезення столів 80x20	Іван Фоменко	Виконано
+380669143453	Московський проспект, 118, Харків, Харківська область, Україна - вулиця Державінська, 38, Харків, Харківська область, Україна	2.155 км	26.0 грн	Товар 3т	Іван Фоменко	Виконано

Рисунок 3.17 – Історія фіналізованих замовлень водія

Розглянемо взаємодію з веб-додатком з точки зору адміністратора. Для початку роботи адміністратору також необхідна авторизація в системі.

Водій	Телефон	Авто	Дії
Василь Іванов	+380669143453	Volvo FH	Призначити транспорт, Видалити
Владислав Семенчук	+380669143451	Hyundai Mighty	Призначити транспорт, Видалити
Сергій Іванченко	+380951402488	Scania S-Series	Призначити транспорт, Видалити
Юрій Качанов	+380669142222	MAN TG510A	Призначити транспорт, Видалити
Іван Фоменко	+380951402489	ISUZU ELF	Призначити транспорт, Видалити

Рисунок 3.18 – Перегляд списку зареєстрованих водіїв

Після авторизації адміністратор має можливості перегляду списку зареєстрованих в системі ресурсів, а саме водіїв та авто (рисунок 3.18-3.19):

Модель	Довжина грузового відсіку	Висота грузового відсіку	Об'єм грузового відсіку	Максимальна вага	Номер	Водій	Дії
Foton Auman	12.0 м	2.0 м	43.0 м ³	10.0 т	AX 1111 СУ	Не встановлено	Видалити
Hyundai Mighty	12.0 м	2.0 м	43.0 м ³	8.0 т	AX 2323 СУ	Владислав Семенчук	Видалити
ISUZU ELF	6.0 м	2.0 м	21.6 м ³	5.0 т	AX 2333 СУ	Іван Фоменко	Видалити
MAN TG510A	8.0 м	2.4 м	34.0 м ³	6.0 т	AX 2323 СЦ	Юрій Качанов	Видалити

Рисунок 3.19 – Перегляд адміністратором списку авто

На сторінці для адміністрування водіїв, адміністратор має змогу призначити водію транспорт зі списку доступних авто (рисунок 3.20-3.21):

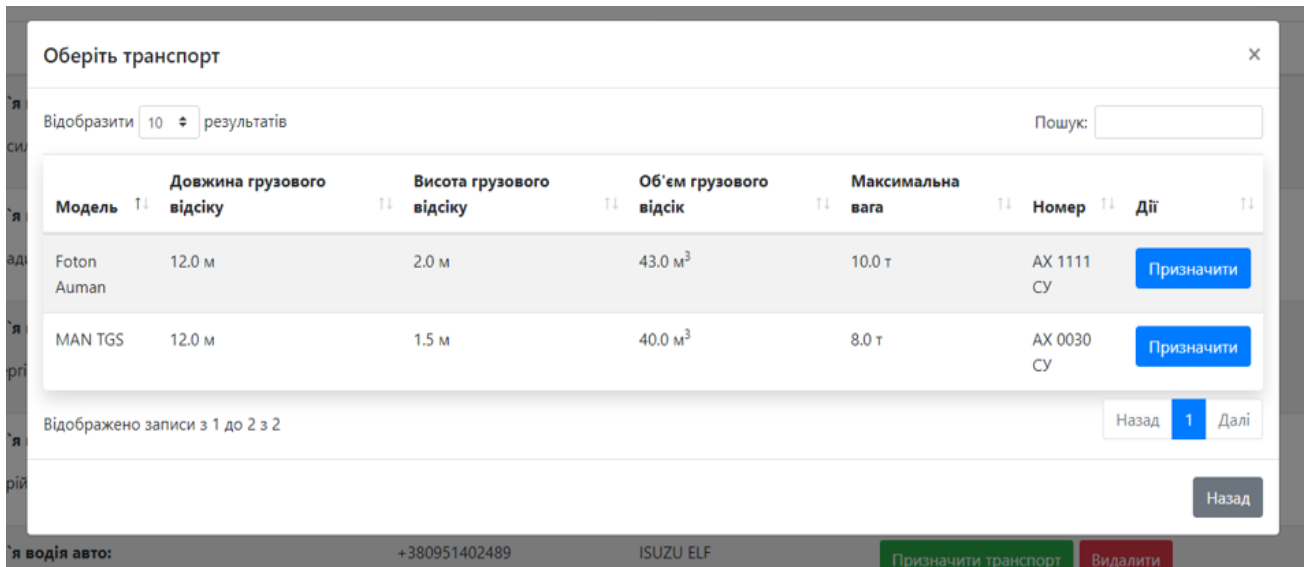


Рисунок 3.20 – Список транспорту для призначення

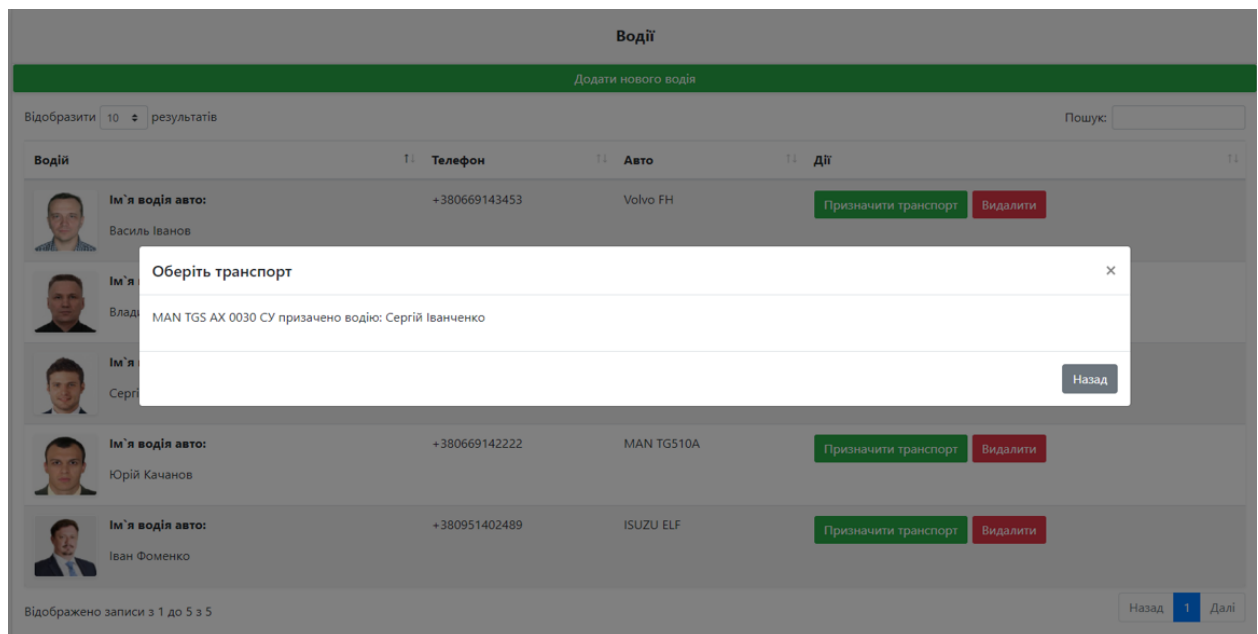
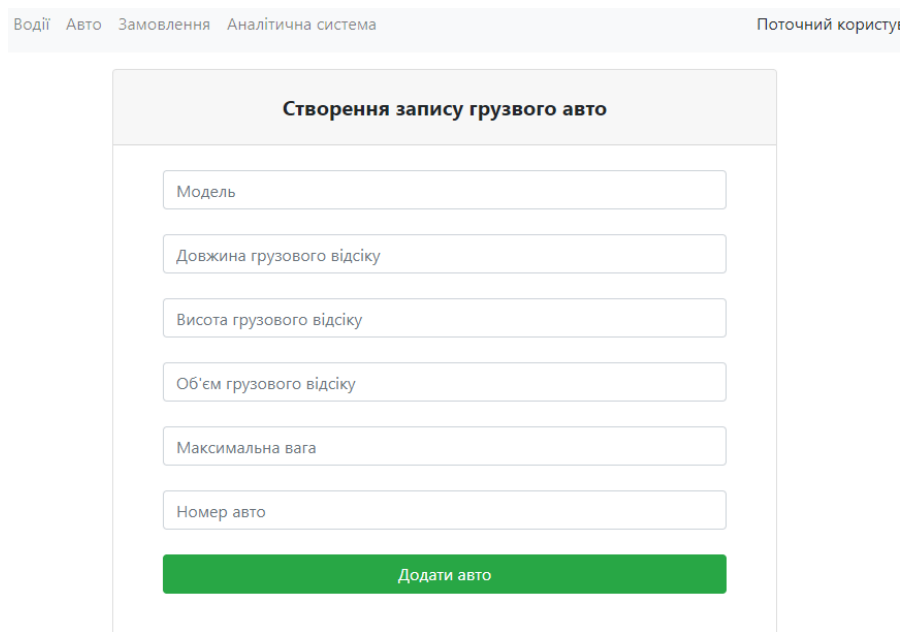


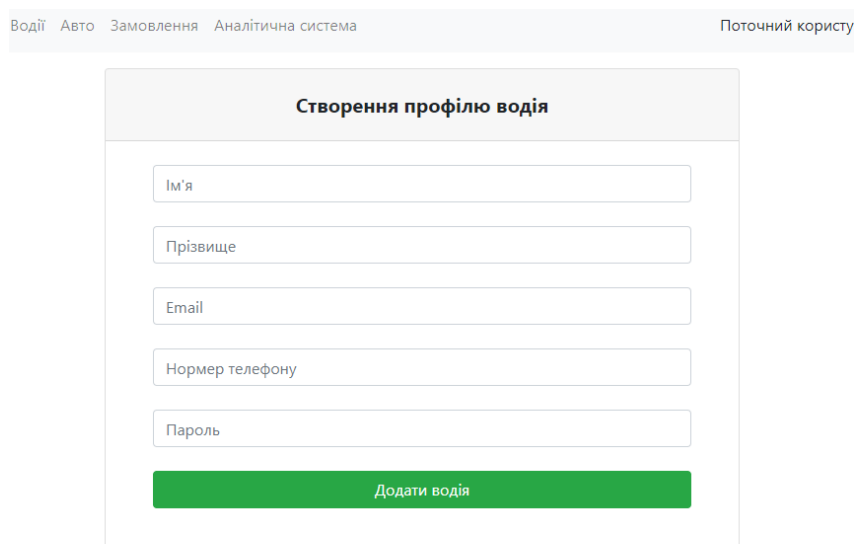
Рисунок 3.21 – Повідомлення про успішне призначення транспорту для водія

Окрім видалення з системи зареєстрованих водіїв і авто, адміністратор також має можливість додавання нових записів за допомогою спеціальних форм (рисунок 3.22-3.23):



The screenshot shows a web application interface with a navigation bar at the top containing the links: 'Водії', 'Авто', 'Замовлення', 'Аналітична система', and 'Поточний користувач'. The main content area is titled 'Створення запису грузового авто'. It contains a vertical stack of six text input fields with the following labels: 'Модель', 'Довжина грузового відсіку', 'Висота грузового відсіку', 'Об'єм грузового відсіку', 'Максимальна вага', and 'Номер авто'. At the bottom of the form is a prominent green button labeled 'Додати авто'.

Рисунок 3.22 – Форма для створення авто



The screenshot shows a web application interface with a navigation bar at the top containing the links: 'Водії', 'Авто', 'Замовлення', 'Аналітична система', and 'Поточний користувач'. The main content area is titled 'Створення профілю водія'. It contains a vertical stack of six text input fields with the following labels: 'Ім'я', 'Прізвище', 'Email', 'Номер телефону', and 'Пароль'. At the bottom of the form is a prominent green button labeled 'Додати водія'.

Рисунок 3.23 – Форма для створення профілю водія

Адміністратор має змогу передивлятися всі замовлення в системі (рисунок 3.24):

2.0 Домашня Водії Авто Замовлення Аналітична система Поточний користувач: test@gmail.com

Замовлення

Відобразити 10 результатів Пошук:

Замовник	Маршрут	Відстань	Ціна	Додаткова інформація	Час виконання	Виконавець	Статус
123	вулиця Нетеченська, 25, Харків, Харківська область, Україна - вулиця Державінська, 38, Харків, Харківська область, Україна	2.555 км	31.0 грн	Пісок 4т	14.11.20 15:48 - 80 хв.	Іван Фоменко	Відхилено
232323	вулиця Гвардійців-Широнінців, 33, Харків, Харківська область, Україна - вулиця Літературна, 5, Харків, Харківська область, Україна	9.77 км	117.0 грн	Телевізори 50дм 10шт	01.12.20 22:21 - 122 хв.	Юрій Качанов	В процесі виконання
232323	вулиця Гвардійців-Широнінців, 33, Харків, Харківська область, Україна - вулиця Літературна, 5, Харків, Харківська область, Україна	9.77 км	117.0 грн	Телевізори 50дм 10шт	01.12.20 22:21 - 122 хв.	Іван Фоменко	Виконано
+38066918777	вулиця Одеська, Харків, Харківська область,	7.703 км	92.0 грн		12.12.20 15:35 -	Не встановлено	Очікує в черзі

Рисунок 3.24 – Список замовлень в системі

Головною можливістю адміністратора є перегляд сторінки з аналітичною і статистичною інформацією щодо роботи підприємства як системи масового обслуговування. На цій сторінці адміністратор має змогу побачити кількість замовлень, кількість ресурсів в системі, кількість активних водіїв, статистику щодо замовлень в системі, а також аналітичну інформацію щодо ефективності керування потоками замовлень в вигляді графів, діаграм і таблиць. Є можливість переглянути відносну величину витрат за поточною кількістю ресурсів, а також прогнозну величину витрат за альтернативної кількості ресурсів (рисунок 3.25):

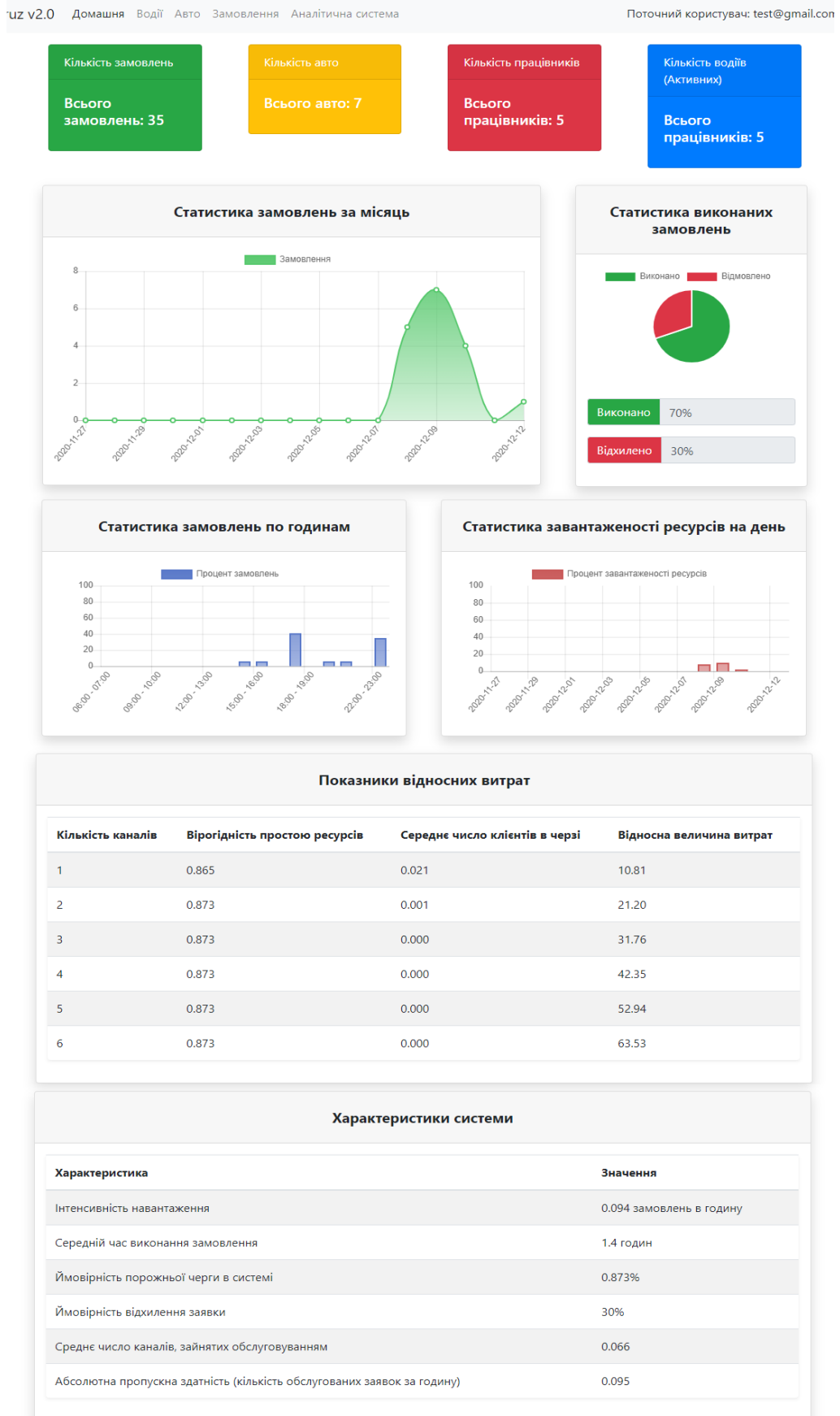


Рисунок 3.25 – Сторінка «Аналітична система»

Для виходу з системи користувач має змогу скористатися кнопкою «Вихід» на панелі керування веб-додатком (поруч з поштою поточного авторизованого користувача (рисунок 3.26):

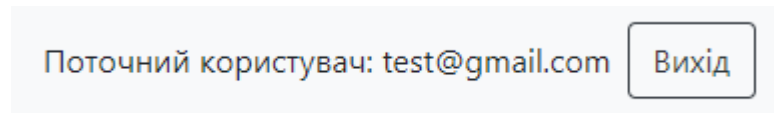


Рисунок 3.26 – Кнопка виходу з системи

Отже в даному розділі були описані засоби розробки додатку, а саме – Spring та його розширення Spring Boot, у якості СУБД було використано MySQL.

Була описана програмна реалізація додатку, описані реалізовані класи та методи кожного класу.

Було описано об'єктно-орієнтовану модель з описом архітектури системи, наведенням діаграми класів з описом необхідних для роботи системи методів.

Був розроблений, реалізований і описаний інтерфейс додатку.

ВИСНОВКИ

У результаті виконання кваліфікаційної роботи було реалізовано веб-сервіс для підприємства з вантажоперевезень, а також аналітичну систему оцінки ефективності, яка надає змогу генерувати детальний аналіз ефективності керування потоками в системі у вигляді показових графів, таблиць і кількісних показників.

В результаті реалізації веб-додатку було виконано аналіз предметного середовища і розглянуто та досліджено переваги та недоліки відомих систем оцінки ефективності. Розглянуті аналогові програмні засоби. Також була описана постановка завдання на розробку програмного засобу. Описані вхідні дані, джерела вхідних даних та вихідні дані.

Були проаналізовані існуючі програмні методи, фреймворки, види побудови архітектури веб-додатків. У результаті аналізу були вибрані технології для створення веб-сервісу, а саме: мова програмування Java, фреймворк Spring, та його розширення Spring Boot, як ORM – Hibernate, у якості засобу управління базою даних – MySQL. Також, була реалізована інтеграція з Google Maps API.

Для визначення та уточнення вимог до розроблюваного програмного продукту було проведено моделювання даних, створено діаграму варіантів використання (Use Case Diagram) та діаграми діяльності (Activity Diagram).

Було проведено моделювання бази даних, описані таблиці бази даних та колонки таблиць записом даних, що зберігаються. Наведена схема бази даних, що показує зв'язки між таблицями, зовнішні і первинні ключі.

Було сформовано системні вимоги для серверної сторони і для клієнтської. В описі програмної реалізації були описані класи, що формують додаток, разом із наведенням діаграм класів для кожного пакету проекту.

З використанням вибраних програмних компонентів та відповідно до створених моделей була розроблена серверна і клієнтська частини додатку. Було реалізовано виконання бізнес-функцій для ролей у системі – для

користувача, диспетчера, адміністратора, водія. Була реалізована аналітична система, яка відображає статистику замовлень за певний проміжок часу (години, місяці), статистику завантаженості ресурсів за день, показники відносних витрат, середній час виконання замовлення, інтенсивність навантаження, ймовірність відхилення заявки, абсолютна пропускна здатність.

Розроблений веб-сервіс задовольняє всім вимогам, поставленим на етапі постановки завдання.

Як подальше вдосконалення створеного веб-додатку представляється можливим допрацювання інтерфейсу, а також – створення мобільного додатку, адже це значно спростить використання системи, додання нової функціональності для замовника – персональний кабінет з замовленнями, можливість спостерігати за замовленням у реальному часі, а також зміцнення системи безпеки додатку, а саме – додання засобів аутентифікації користувача з використанням капчі або двохфакторної аутентифікації.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки до організації виконання та захисту кваліфікаційної роботи на здобуття другого (магістерського) рівня вищої освіти для студентів усіх форм навчання спеціальності 122 Комп'ютерні науки за ОНП (ОПП) Системи штучного інтелекту. Харків: ХНУРЕ, 2021
2. Положення про проведення практики студентів вищих навчальних закладів України; Наказ Міністерства освіти України від 8 квітня 1993р. №93.
3. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура та правила оформлювання. [Чинний з 2017-07-01]. Вид. офіц. Київ: ДП «УкрНДНЦ», 2016. 26 с.
4. Наказ Державного комітету України з питань нагляду за охороною праці №15 від 26.01.2005 р.
5. Коннолли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение: Теория и практика / за ред. Коннолли Т., Бегг К., Страчан А. Москва: Издательский дом «Вильямс», 2003. 1436 с.
6. Дейт К. Введение в системы баз данных. Москва: Издательский дом «Вильямс», 2001. 1072 с.
7. Пасічник В.В. Організація баз даних та знань. Київ: Видавнича група ВНУ, 2006. 384 с.
8. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов. Москва: ДМК издательство, 2002. 169 с.
9. Основы Web-технологий: учеб. / под ред. П.Б.Храмцова: учеб. Москва: ИНТУИТ.РУ «Интернет универсальных информационных технологий», 2003. 512 с.
10. Hibernate Everything data. URL: <https://hibernate.org> (дата звернення 11.11.2021).

11. Козмина Ю., Харроп Д., Шефер К., Хо К. Spring 5 для профессионалов Pro Spring 5: An In-Depth Guide to the Spring Framework and Its Tools. Москва: «Вильямс», 2019. 1120 с.

12. Раджпут Д. Spring. Все паттерны проектирования = Spring 5 Design Patterns: Master efficient application development with patterns such as proxy, singleton, the template method, and more. Санкт-Петербург: «Питер», 2019. 320 с.

13. Лонг Д., Бастани К. Java в облаке. Spring Boot, Spring Cloud, Cloud Foundry = Cloud Native Java: Designing Resilient Systems with Spring Boot, Spring Cloud, and Cloud Foundry. Санкт-Петербург: «Питер», 2019. 624 с.

14. Климов Г.П. Теория массового обслуживания. М.: Изд-во Московского университета, 2011. 312 с.

15. Шилдт Г. Java 8. Руководство для начинающих 6-е издание. Харків: Право, 2005. 599 с.

16. Буч Г. Язык UML. Руководство пользователя / за ред. Буч Г., Рамбо Дж., Джекобсон А. СПб.: Питер, 2004. 432 с.

17. Блох Дж. Java. Эффективное программирование 3-е издание 2014. 310 с.

18. Экономико-математическое моделирование: Учебник для студ. вузов, обуч. по спец. "Финансы и кредит", "Бух. учет, анализ и аудит", "Мировая экономика" / Л.В. Абланская, Л.О. Бабешко, Л.И. Баусов и др. / Под общ. ред. И.Н. Дрогобыцкого. М.: Экзамен, 2004. 800 с.

19. Вентцель Е. С. Исследование операций. Задачи, принципы, методология: Учебное пособие для студентов вузов. М.: Высшая шк., 2001. 206 с.

20. Клейнрок Л. Теория массового обслуживания. Пер. с англ. / Пер. И.И. Грушко; ред. В.И. Нейман. М.: Машиностроение, 1979. 432 с.