Вінницький національний технічний університет
Факультет інтелектуальних інформаційних технологій та автоматизації
Кафедра комп'ютерних систем управління

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:
«ОПТИМІЗАЦІЯ МАРШРУТУ ДОСТАВКИ ЗАМОВЛЕННЯ ЗА ДИНАМІЧНИМИ ДАНИМИ»

Виконав: студент 2 курсу, групи 2АКІТ-21м
спеціальності 151 – Автоматизація
та комп'ютерно-інтегровані технології
Михайло ЩЕРБАНЬ

Керівник:
к.т.н., доцент, доцент кафедри КСУ
Олена НИКИТЕНКО

«12» грудня 2022 р.

Опонент:
д.т.н., професор, професор каф. АІІТ
Роман КВЄТНИЙ

«13» грудня 2022 р.

Допущено до захисту
Зав. кафедри КСУ
В'ячеслав КОВТУН
«14» грудня 2022

Вінниця ВНТУ – 2022 рік

Вінницький національний технічний університет

Факультет інтелектуальних інформаційних технологій та автоматизації

Кафедра комп'ютерних систем управління

Рівень вищої освіти другий (магістерський)

Галузь знань – 15 – Автоматизація та приладобудування

Спеціальність – 151 – Автоматизація та комп'ютерно-інтегровані технології

Освітньо - професійна програма – Інтелектуальні комп'ютерні системи

ЗАТВЕРДЖУЮ
Завідувач кафедри КСУ

В'ячеслав КОВТУН

"03" жовтня 2022 року

ЗАВДАННЯ
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Щербаню Михайлу Олександровичу
(прізвище, ім'я, по батькові)

1. Тема роботи Оптимізація маршруту доставки замовлення за динамічними даними

керівник роботи Никитенко Олена Дмитрівна
затверджені наказом ВНТУ від "14" вересня 2022 року №203

2. Термін подання студентом роботи "12" грудня 2022 року

3. Вихідні дані до роботи: підтримка ОС – Windows, база даних, JSON файл з ключами-атрибутами, JSON файл із сформованим маршрутом з системи Google Maps, файл формату GeoJson створений після обробки маршруту

4. Зміст текстової частини: вступ, аналіз аналогів систем доставки. Огляд сервісів побудови маршрутів, огляд проблеми оптимізації маршруту доставки, аналіз методів отримання динамічних даних, аналіз способів створення веб-додатку з доставки, аналіз методів розробки системи доставки. аналіз альтернативних методів розробки, створення uml діаграми варіантів використання. розробка проекту бази даних, вирішення задачі з оптимізації, обгрунтування методів автоматизації та масштабування системи, реалізація фронт-енду системи, реалізація бек-енду системи, комерційний та технологічний аудит науково-технічної розробки, прогнозування витрат.

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

Вступ, Мета, предмет та об'єкт дослідження, Порівняння аналогів сервісів доставки, Завдання системи оптимізації доставки, Аналіз картографічних систем, Вибір підходу для розробки системи, Вибір алгоритму пошуку короткого шляху, Вибір системи управління базами даних, Вибір системи управління контентом(CMS), Представлення UML діаграми варіантів використання, Результат розробки сервісу,Висновки.

1. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	виконання прийняв
4	Небава М.І., пофесор кафедри ЕПВМ		

2. Дата видачі завдання "03" жовтня 2022 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва та зміст етапу	Термін виконання		Примітка
		початок	закінчення	
1	Аналіз аналогів систем доставки. Аналіз способів створення веб-додатку для сервісу доставки.	*04.10.2022р*		
2	Аналіз методів розробки системи доставки. Аналіз альтернативних методів розробки.	*08.10.2022р*		
3	Огляд технологій побудови маршрутів, аналіз методів отримання динамічних даних.	*13.10.2022р*		
4	Створення UML діаграми варіантів використання, UML діаграма послідовності та Діаграми класів. Розробка проекту бази даних.	*17.10.2022р*		
5	Огляд алгоритмів пошуку найкоротшого маршруту, Огляд методів масштабування системи та відстеження кур'єрів.	*24.10.2022р*		
6	Реалізація системи	*30.10.2022р*		
7	Розрахунок економічної частини	*05.12.2022р*		
8	Графічні матеріали	*21.12.2022р*		

Студент Михайло Щербань
(підпис) (*Ім'я ПРІЗВИЩЕ*)

Керівник роботи Олена Никитенко
(підпис) (*Ім'я ПРІЗВИЩЕ*)

АНОТАЦІЯ

УДК 004.4

Щербань М.О. Оптимізація маршруту доставки замовлення за динамічними даними. Магістерська кваліфікаційна робота зі спеціальності 151 – Автоматизація та комп’ютерно-інтегровані технології, освітня програма – Інтелектуальні комп’ютерні системи. Вінниця: ВНТУ, 2022. 107ст. На укр. мові. Бібліогр.: 61 назв; рис.: 62; табл. 3.

В даній роботі розглянуто основні питання щодо реалізації підсистеми автоматизованого управління доставкою з можливістю відслідковування кур'єра за допомогою трекінгової системи. У оглядово-аналітичній частині роботи проведено порівняльний аналіз інснуючих систем доставки та обрано критерії для побудови власної системи, розглянуто технології для реалізації відстежування замовлення в реальному часі. У теоретично-методичній частині проаналізовано методи проектування й розробки, проведено проектування бази даних. Також було розроблено UML діаграми та проаналізовано методи отримання та обробки динамічних даних та алгоритми побудови коротких шляхів. У практичній частині обгрунтовано інструментальні засоби реалізації системи та розроблено прототип підсистеми автоматизованого управління доставкою. В економічній частині проаналізований технічний рівень і розрахована собівартість реалізації розробки. Ілюстративна частина складається з 15 плакатів із результатами роботи та графічними матеріалами.

Ключові слова: *система*, *управління доставкою, оптимізація маршруту*, *кур’єр, трекінг, карта*.

ABSTRACT

Shcherban M.O. Order delivery route optimization based on dynamic data. Master's thesis on specialty 151 - Automation and computerintegrated technologies, educational program - Intelligent computer systems. Vinnytsia: VNTU, 2022. 107p. In English speech Bibliography: 61; pic.: 62; table 3.

In this work, the main issues regarding the implementation of the subsystem of automated delivery management with the possibility of tracking the courier using a tracking system are considered. In the review and analytical part of the work, a comparative analysis of existing delivery systems was carried out and criteria for building one's own system were selected, technologies for real-time order tracking were considered. In the theoretical and methodological part, the methods of design and development were analyzed, the design of the database was carried out. UML diagrams were also developed and methods of obtaining and processing dynamic data and algorithms for constructing short paths were analyzed. In the practical part, the instrumental means of implementing the system are substantiated and a prototype of the automated delivery management subsystem is developed. In the economic part, the technical level is analyzed and the cost of development implementation is calculated. The illustrative part consists of 15 posters with work results and graphic materials.

Keywords: delivery, *management system, route optimization, courier, tracking, map*.

ВІДГУК
керівника магістерської кваліфікаційної роботи

студента Щербаня Михайла Олександровича група 2АКІТ-21м

на тему: Оптимізація маршруту доставки замовлення за динамічними даними

Представлена магістерська кваліфікаційна робота присвячена розробці підсистеми автоматизованого управління доставкою та повністю відповідає контексту спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». Робота відповідає чинним вимогам щодо повноти розкриття теми та оформлення її вербальної, графічної й бібліографічної частин. Теоретичний і практичний матеріал роботи є достатнім та добре структурованим. На позитивну оцінку заслуговує вміння здобувача творчо підходити до систематизації теоретичної інформації та інтерпретувати й узагальнювати ілюстративний матеріал.

Позитивними рисами дипломної роботи є системність та послідовність викладення матеріалу, а також застосування прогресивного досвіду в сфері керування контентом веб-сайтів та його практичне застосування.

Студентом було проведено аналіз та порівняння можливих методів розв'язання поставленої задачі та обрано оптимальний варіант. Окрім того, було досліджено існуючі реалізації розв'язання подібних задач. Під час виконання магістерської кваліфікаційної роботи студент проявив себе грамотним, кваліфікованим спеціалістом здатним приймати самостійно складні технічні рішення.

Результати, що представлені в роботі отримані студентом самостійно. Результат перевірки на наявність запозичень підтверджує, що роботі властивий високий ступінь унікальності.

Недоліки: Розроблена студентом підсистема оптимізації маршруту виконана в вигляді окремого модуля і має складнощі з інтеграцією у систему доставки. Автор частково опирається на застарілі джерела. Не всі представлені рисунки достатньої якості.

Загалом магістерська кваліфікаційна робота ***відповідає*** спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», заслуговує на оцінку _____B_____, а її автор ***заслуговує*** присудження кваліфікації: ступінь вищої освіти магістр, спеціальність «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні комп'ютерні системи»

Керівник магістерської кваліфікаційної роботи

к.т.н., доцент, доцент кафедри КСУ — Олена НИКИТЕНКО
(посада, науковий ступінь, вчене звання) (підпис) (Ім'я ПРІЗВИЩЕ)

ВІДГУК
опонента на магістерську кваліфікаційну роботу

студента Щербаня Михайла Олександровича група 2АКІТ-21м

на тему: Оптимізація маршруту доставки замовлення за динамічними даними

Представлена магістерська кваліфікаційна робота присвячена розробці підсистеми автоматизованого управління доставкою та повністю відповідає контексту спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології». Робота відповідає чинним вимогам щодо повноти розкриття теми та оформлення її вербальної, графічної й бібліографічної частин.

У оглядово-аналітичній частині магістерської кваліфікаційної роботи проведено порівняльний аналіз існуючих аналогів систем доставки та обрано критерії для побудови власної системи, розглянуто технології для реалізації відстежування замовлення в реальному часі.

У теоретично-методичній частині проаналізовано методи проектування й розробки систем управління доставкою. Студентом було проведено аналіз та порівняння можливих методів розв'язання поставленої задачі та обрано оптимальний варіант.

Реалізація проекту підсистеми проведена у відповідності до представлених методів для вирішення поставленої задачі. Тестування розробленої системи покриває заявлений функціонал в завданні та підтверджує правильність підібраних методів вирішення.

Недоліки: Серед недоліків розробленої системи не достатньо зрозумілий інтерфейс. Не обґрунтовано достатнім чином вибір картографічної системи для відображення отриманого маршруту. Не до кінця реалізоване поєднання модулів оптимізації маршрутів та системи доставки.

Загалом магістерська кваліфікаційна робота ***відповідає*** спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», заслуговує на оцінку ____B_____, а її автор ***заслуговує*** присудження кваліфікації: ступінь вищої освіти магістр, спеціальність «Автоматизація та комп'ютерно-інтегровані технології», освітня програма «Інтелектуальні комп'ютерні системи»

Опонент

д.т.н., проф., зав. каф. АІІТ
(посада, науковий ступінь, вчене звання)

(підпис)

Роман КВЄТНИЙ
(*Ім'я ПРІЗВИЩЕ*)

Печатка установи, організації опонента

Зміст

Вступ

Актуальність. Сервіси доставки з кожним днем все більше ввпливають на життя людей, адже важко відмовитись від зручності, що пропонують мережі доставок. Головна перевага сервісів доставки над їхніми фізичними аналогами дотупними на ринку заключається в більшій швидкості отримання необхідних товарів, що значно збільшує шанси скористатися одним із таких сервісів у майбутньому.

З розвитком та зі збільшенням поширення мережі Інтернет багато сервісів мігрували в онлайн системи та продовжують там активно розвиватись, використовуючи для цього наявні сучасні методи. Все більше і більше користувачів віддають перевагу різним інтернет-послугам в мережі замість звичайних фізичних закладів чи магазинів. Інтернет-сервіси для різних видів діяльності стають все більш популярними через їх зручність і швидкість використання. Багато різних постачальників послуг своєчасно підхопили цю тенденцію відкрили власні сервіси доставки, що дозволило зайняти ринкову нішу, яка набирала популярності ще до пандемії. Яскравим прикладом таких систем є сервіси доставки їжі.

Популярність використання сервісів доставок серед закладів харчування пов’язана, із зручністю та великою економією часу клієнта, що не сам особисто очікує замовлення в закладі, а отримає теж саме замовлення додому. Суттєвий вплив на економію часу матиме наявність великої кількості людей, які працюють на сервіси доставки кур’єрами, а так як кур’єрів на ринку багато, замовлення буде виконуватись кур’єром швидше через більшу конкуренцію.

Розробка ефективних інструментів з оптимізації маршруту для системи доставки має велике значення, так як скорочення часу доставки за рахунок прокладання кращих маршрутів, дозволить зменшити час доставки і збільшити кількість замовлень доставлених кур’єром, що в свою чергу призведе до зменшення витрат та ефективного споживання ресурсів компанії.

Метою дослідження є побудова підсистеми доставки замовлень з використанням динамічних даних, а саме координат кур’єра, які змінюються при русі.

Для досягнення наведеної мети були поставлені та вирішені наступні задачі:

- здійснити аналіз ринку сервісів доставок
- проаналізувати методи оптимізації шляху з використанням динамічних даних
- створити підсистему автоматизованого слідкування за координатами курєра з використанням сучасних технологій
- розробка інформаційної системи оптимізації маршрутів доставки онлайн замовлень.

Об’єктом дослідження є технології побудови підсистем та сервісів доставки, пошук методів оптимізації шляху доставки.

Предметом дослідження є методи та засоби побудови сервісів з управління системою доставки, пошук найкоротшого шляху доставки.

Новизна полягає в розробці сервісу з управління системою доставки з використанням удосконаленого алгоритму пошук найкоротшого шляху доставки з використанням динамічних даних.

1 АНАЛІЗ СИСТЕМ УПРАВЛІННЯ ДОСТАВКОЮ

Система управління доставкою це комплексне рішення для покращення роботи закладу. Правильно реалізована система це точне поєднання модулів створення замовлення клієнтом, отримання кур’єром замовлення, та доставки замовлення до клієнта найкоротшим маршруто в найшвидший час.

Якщо виділити основні завдання, які має виконувати система управління доставкою для збільшення ефективності використання системи, то коротко їх можна охарактеризувати так:

– Можливість швидкого створення замовлення клієнтом;

– Коректне групування замовлень по регіонах міста і призначення їх для виконання оптимальному кур’єру.

– Аналіз даних про місця доставки, отримання динамічних даних з картографічних систем.

– Обробка отриманих даних, пошук короткого маршруту для виконання замовлення.

Реалізація системи управління доставкою, що матиме розв’язок виділених основних завдань, збільшить ефективність та якіст доставки замовлень, що в свою чергу призвуду до збільшення кількості клієнтів.

1.1 Аналіз конкурентних систем доставки

На сьогоднішній день існує велика кількість сервісів доставок доступних на ринку. Кожен із таких сервісів має свої переваги та недоліки. Для порівняння функціональних рішень варто виділити найбільш популярні, а саме: “Glovo”, “Bolt Food”, “Igogo”, “Liki24”, “Raketa”.

Glovo[1] – один із найбільших сервісів доставки який доступний на ринку на сьогоднішній день. Головне призначення сервісу доставки, доставляти товар, будь-якого виду на невеликі відстані, лише з обмеженням щоб товар міг поміститись в рюкзак. Зазвичай цією доставкою замовляють: їжу, документи, ліки. На рисунку 1.1 зображено вигляд головної сторінки сервісу.

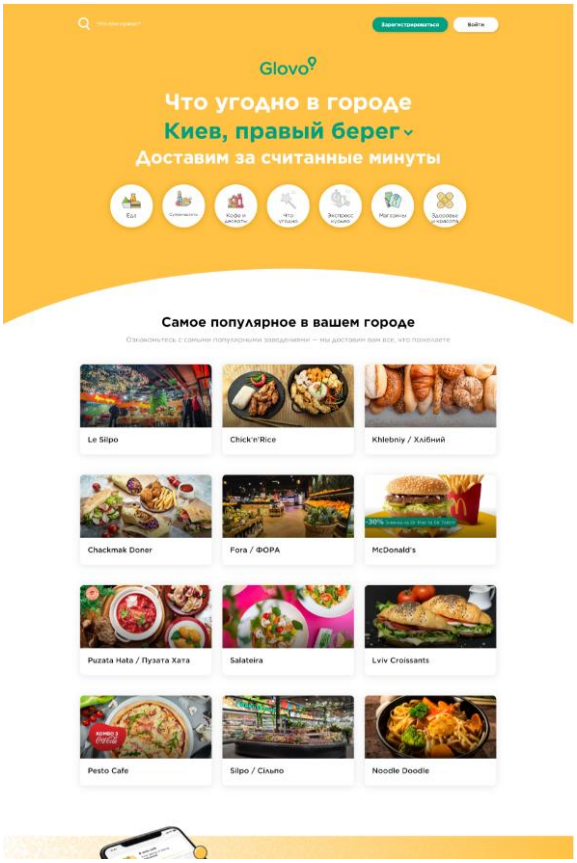


Рис 1.1 Вигляд головної сторінки сервісу

Сервіс за допомогою контрактів і договорів пов’язує заклади, кур’єрів та кінцевих замовників. Що дозволяє всім краще виконувати свої зобов’язання щодо замовлень.На рисунку 1.2 зображений приклад вже сформованого замовлення.

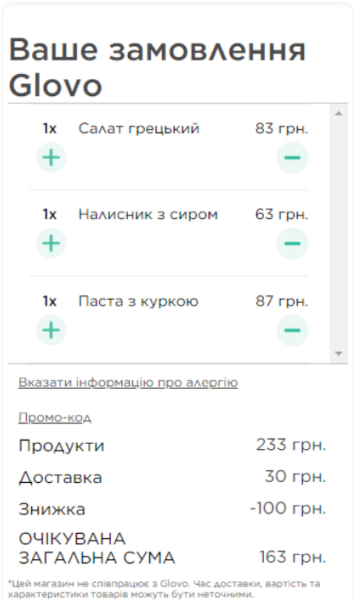


Рисунок 1.2 – Вигляд створеного замовлення

Для створення замовлення в системі, для початку необхідно зареєструватись. Реєстрацію можна пройти з допомогою номеру мобільного телефона або з використанням соціальних мереж для цього таких як “Facebook” або сервіс “Google”.

Після реєстрації користувач отримує змогу переглядати список доступних ресторанів для доставки на даний час доби, меню доступних ресторанів та отримує змогу створювати замовлення.

Після створення замовлення клієнт отримає сповіщення, про обрання кур’єра та етапи замовлення, які змінюються в реальному часі відносно того, на якому етапі зараз кур’єр, а саме:

- Отримання замовлення кур’єром;
- Відвідання закладу кур’єром;
- Очікування приготування замовлення;
- Доставка замовлення до клієнта.

Однією із переваг даної системи є те, що клієнт може відслідковувати на карті в якому місці в даний час перебуває кур’єр із замовленням, що дає змогу краще прогнозувати час доставки і звільняє час очікування.

Сервіс має гнучку систему оплати замовлення, з використанням готівки і сервісів Mastercard та Visa.

В ситуації коли виникають труднощі з доставкою чи отриманням замовлення, наявний чат-бот для підтримки і вирішення найпоширеніших питань, якщо чат-бот не покрває необхідні питання, є можливість зв’язатись з працівниками сервісу та отримати допомогу.

Bolt Food[2] – нова компанія на ринку доставки замовлень й через це вона ще не отримала багато уваги від клієнтів. Сервіс спеціалізується саме на доставці їжі з закладів, що із ним співпрацюють. Через невелике поширення сервіс має дуже обмежену присутність на ринку, і на даний час наявний тільки в декількох найбільших містах.

Система має зручний додаток як для кур’єрів так і для замовників. Переваги у використанні додатку, це краща оптимізація і швидкість відносно таких самих сервісів ,що використовують тільки веб-додатки. Додаток цієї системи доставки доступний як на Android так і на IOS. Сервіс має зручний і комфортний для використання інтерфейс, легко налаштовувані фільтри для зміни товарів відносно вподобань клієнта. Приклад зовнішнього вигляду сервісу зображений на рисунку 1.3.

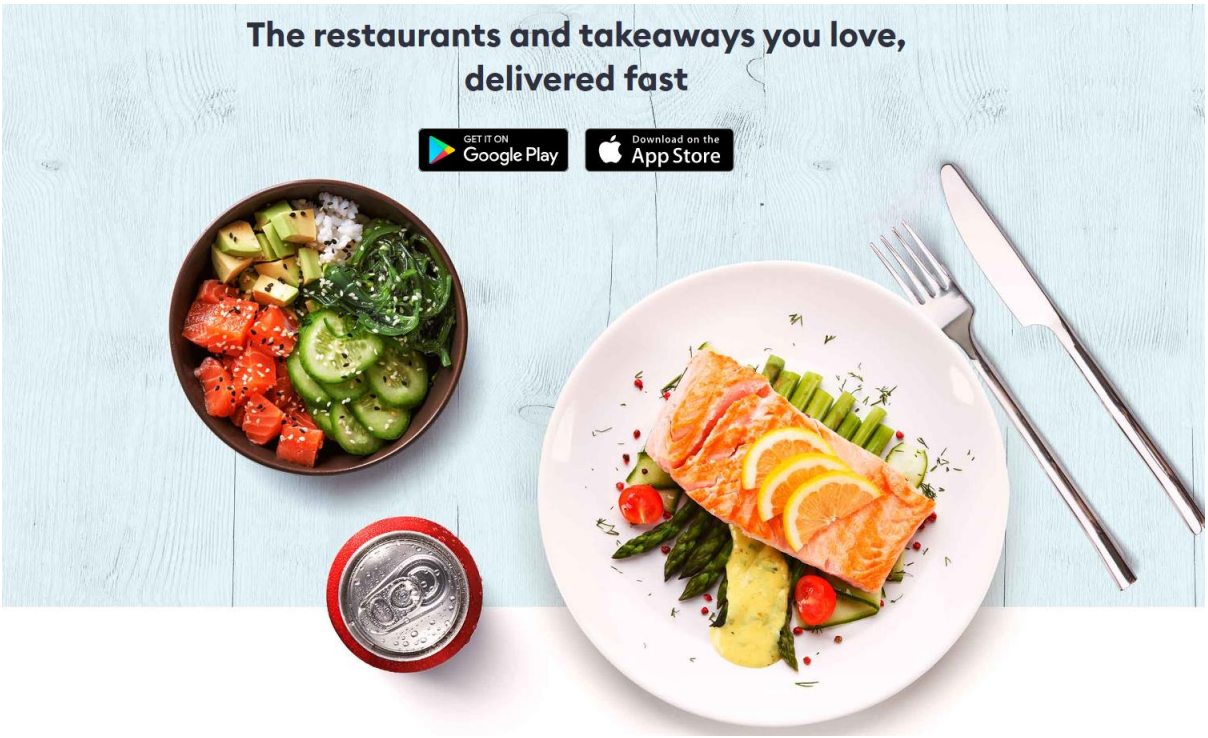


Рисунок 1.3 – Вигляд головної сторінки сервісу Bolt Food

Для створення замовлення необхідно зареєструватись, для цього зпотрібно заповнити стандарну веб-форму реєстрації де вказати електронну пошту, ПІБ та номер телефону, або використати реєстацію з допомогою соціальних мереж.

Ще однією перевагою Bolt Food являється використання кур’єрами більш екологічного транспорту. Сервіс не віддає перевагу автомобілям і кур’єри мають можливість використовувати з такою самою ефективністю велосипеда та електроскутери.

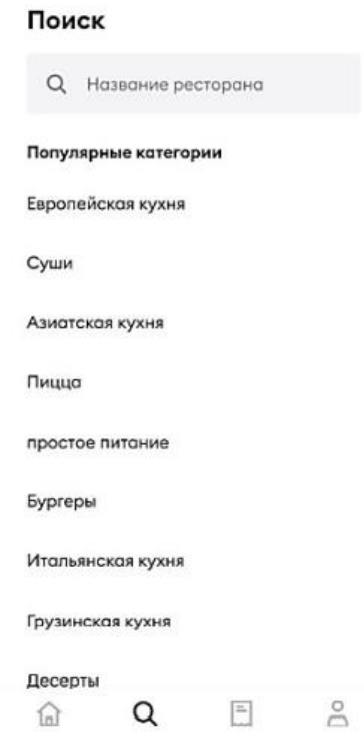


Рисунок 1.4 – Створення замовлення та фільтрація товарів

Створені замовлення зберігаються в системі і за потреби їх можна переглянути. Вікно для перегляду замовлень зображено на рисунку 1.5.

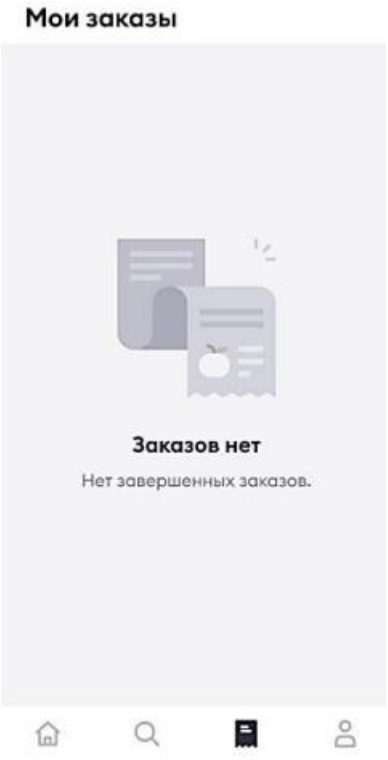


Рисунок 1.5 Вікно створених змовленнь

Igogo[3] – це локальний сервіс доставки їжі, має покриття тільки в одному місці. Через те що червіс не масштабується в нього дуже обмежена кількість закладів, з якими він має можливість співпрацювати. В свою чергу обслуговування меншої кількості закладів, викликає перегляд питання про необхідність використання певної кількості кур'єрів, що в свою чергу через зменшення кількості кур'єрів має нижчу швидкість та ефективність доставки в порівнянні з іншими сервісами доставки доступними на ринку.

Сервіс використовує для роботи веб-додаток, зовнішній вигляд якого зображений на рисунку 1.6. Сервіс має зручну систему навігації, але через відсутність мобільного додатка має трохи нижчу швидкість роботи та меншу ефективність відносно аналогів.

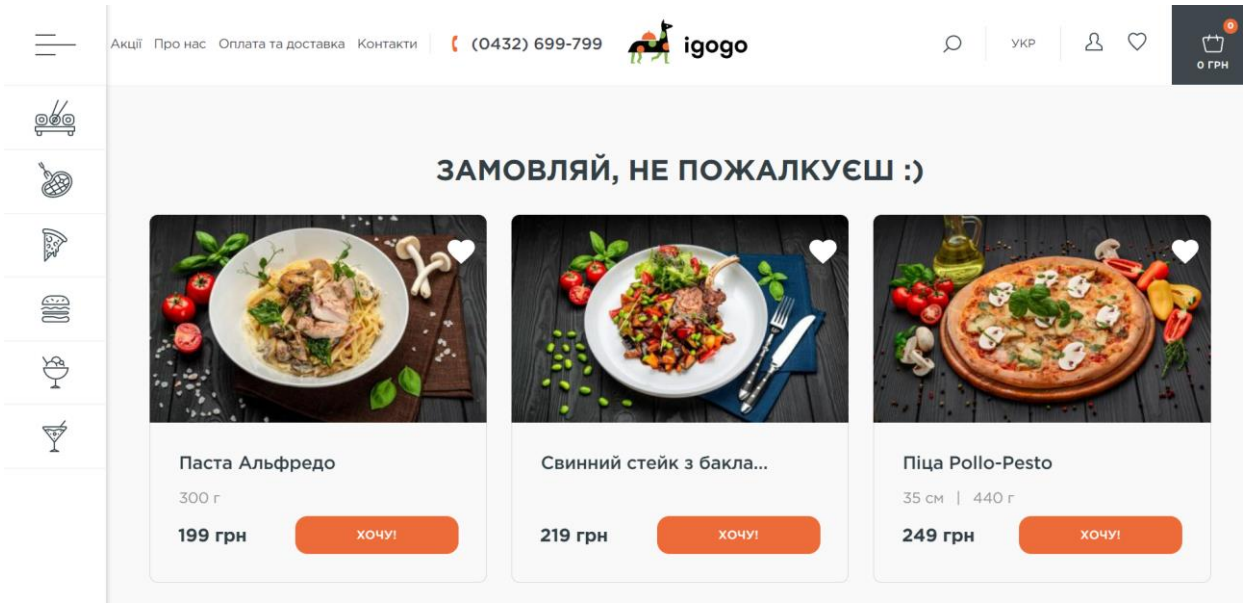


Рисунок 1.6 – Головна сторінка сервісу Igogo

Створення замовлення відбувається без початкової реєстрації, для початку необхідно обрати на сайті товари необхідні для доставки, вибрані товари зберігаються в корзині, після завершення вибору, необхідно перейти до корзини і заповнити необхідні поля для створення за мовлення такі як:

– ПІБ
– Номер телефону

– Адреса
– Час доставки
– Методи оплати

Вигляд створеного замовлення в корзині зображений на рисунку 1.7.

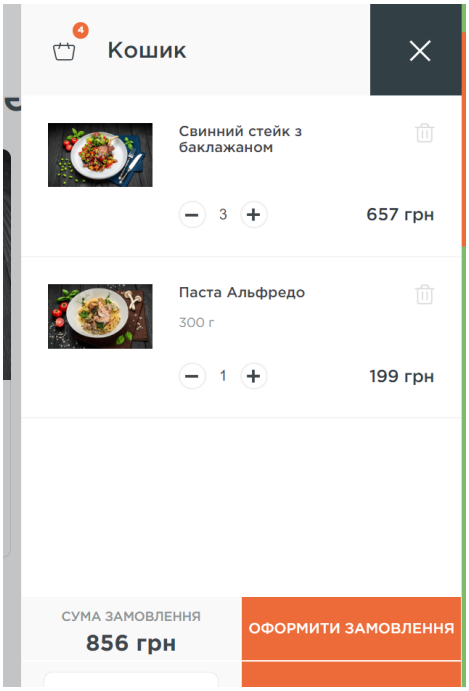


Рисунок 1.7 – Вигляд корзини веб-додатку.

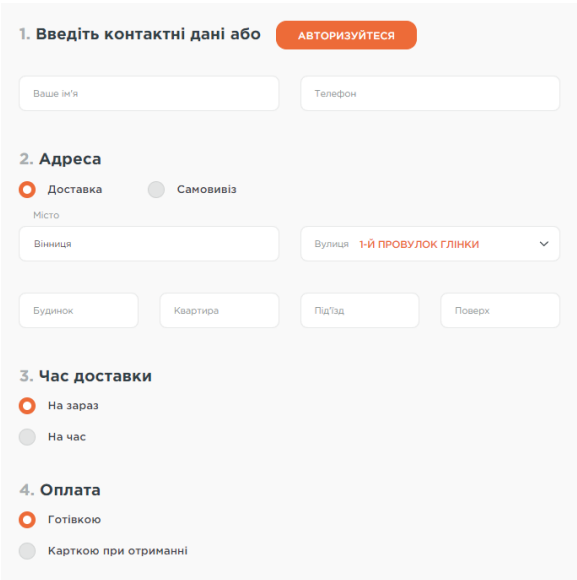


Рисунок 1.8 – Вигляд форми замовлення

Сервіс не має можливості для відстеження замовлення на карті. Та кож не реалізована систем акцій як для кур’єрів так і для клієнтів.

Сервіс “Liki24”[4] – це система що спеціалізується саме на доставці медичних препаратів до клієнта та пошуку найвигідніших цін для замовлення доставки. Основним завданням створення сервісу було зниження вартості отримання медичних препаратів, та зниження кількості людей, що перебувають в черзі в аптеках в будь-який час.

Серед переваг сервісу це зручний та інтуїтивно зрозумілий дизайн. Вигляд основної сторінки наведений у рисунку 1.9. Сервіс, не має розробленого додатку, що впливає на швидкодію та ефективність, через неможливість відвідати сторінку з оптимізованого мобільного додатка.

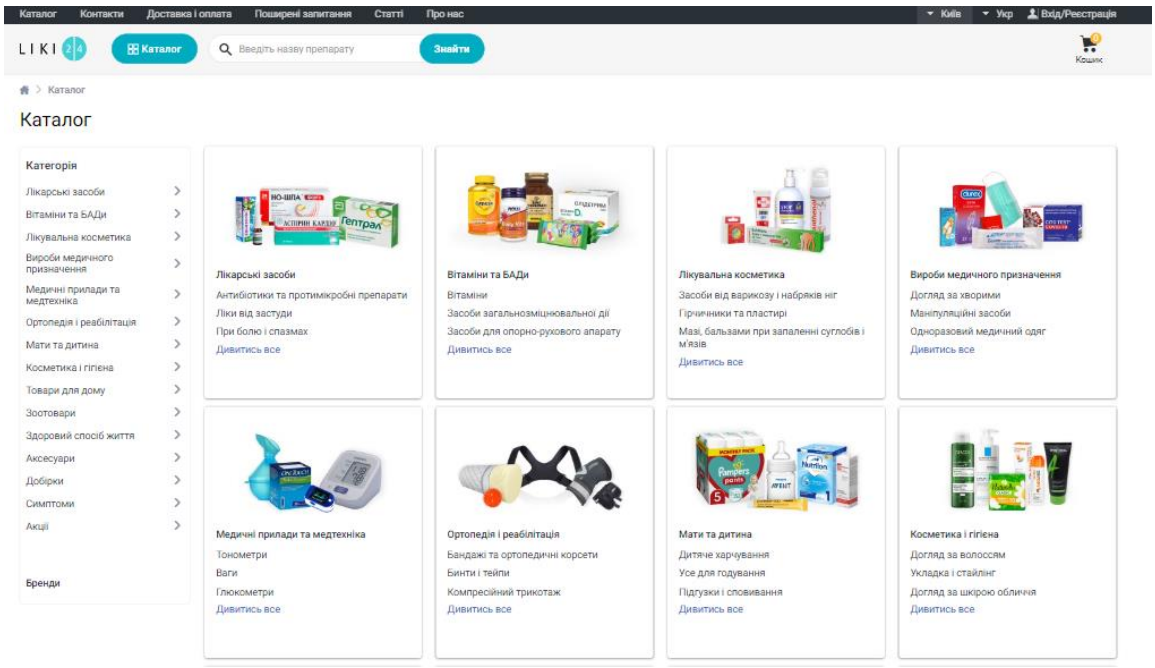


Рисунок 1.9 – Вигляд основної сторінки

Для створення замовлення реєстація в системі не обов’язкова. Для початку Потрібно вибрати необхідну групу ліків. Система робить пошук по базі даних за допомогою за назвою, або за приналежністю до певної групи препаратів. Тому наступним кроком буде шляхом фільтрування обрати необхідні ліки та перейти до

створення замовлення. Форма замовлення зображена на рисунку 1.10. Для заповнення форми необхідно заповнити такі поля як:

– Вибір методів доставки
– Вибір служби доставки
– Вказати адресу
– Вказати ПІБ

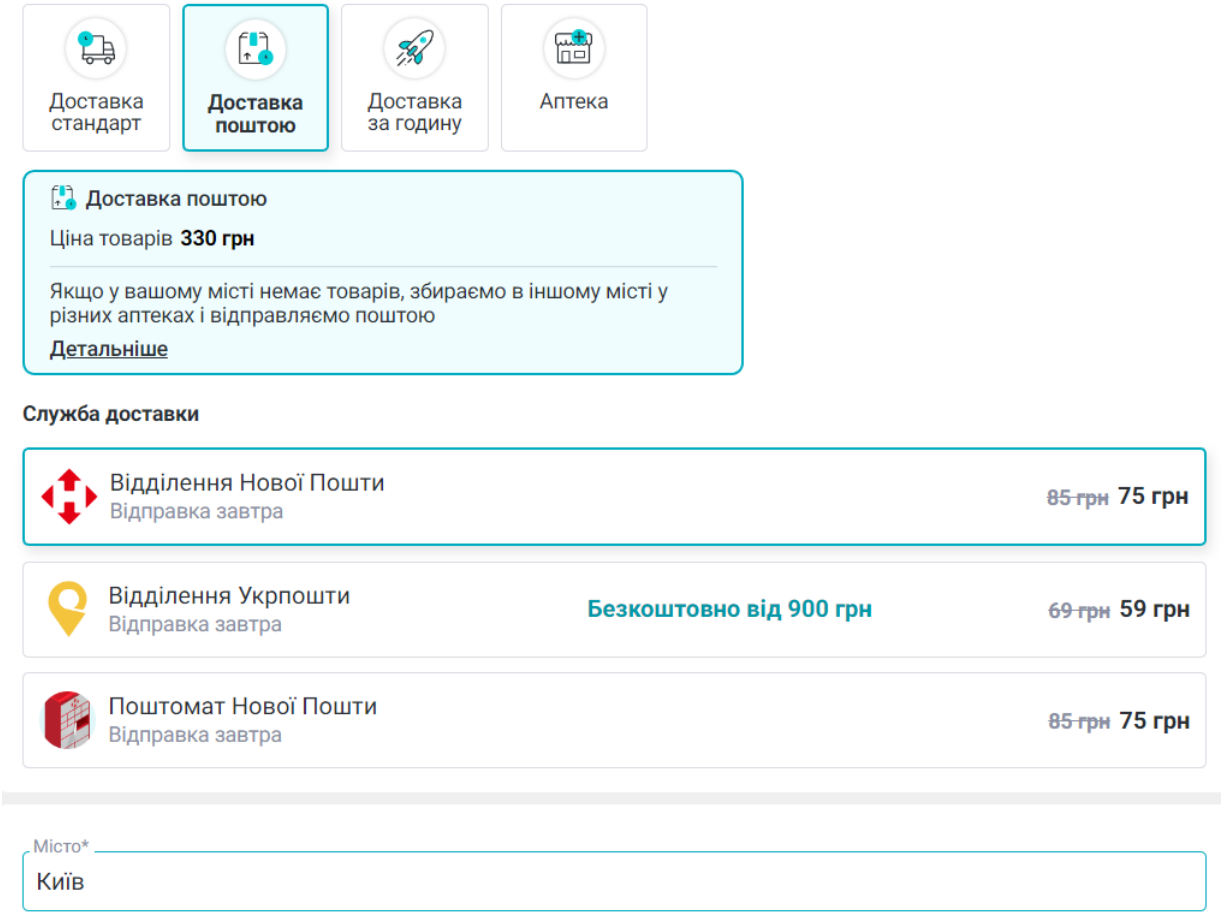


Рисунок 1.10 – Форма для створення замовлення

Raketa[5] – сервіс доставки, що являється одним із найбільших на сьогоднішній день та має широку мережу покриття доставки. Серед компаній, що співпрацюють з цим сервісом є велика кількість ресторанів та супермаркетів.

Оформлення замовлення в сервісі можливе тільки за допомогою мобільного додатка. Для початку замовлення необхідно завантажити додаток та пройти стандартну форму реєстрації, що складається з ПІБ, номеру телефону та

електронниї адреси. Після цього стає доступною можливість створення замовлення. Для початку потрібно відкрити розділ закладів та обрати ресторан. Наступним кроком є вибір товарі для доставки. Далі йде етап перевірки та вибору метів оплати.

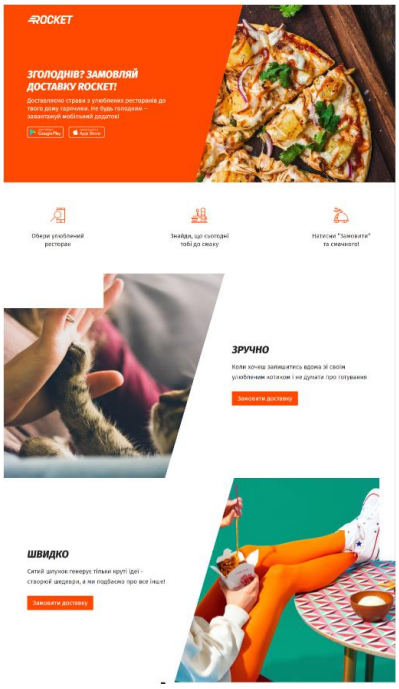


Рисунок 1.11 – Вигляд головної сторінки сервісу.

Рисунок 1.12 – Вигляд мобільного додатку

Після створення замовлення, є можливість відслідковування кур’єра на карті, включно з етапами яке проходить замовлення. В ситуації коли виникають якісь проблеми сервіс має наявний чат підтримки, що дозволяє швидко вирішити будь-які питання.

Для отримання замовлення наявна зручна система оплати з використанням великої кількість методів платежів, основні з яких це:

- Готівка
- Visa
- MasterCard
- AmericanExpress

Для загального порівняння проаналізованих систем доставки можна виділити певні критерії такі як:

- Зручність створення замовлення
- Швидкість роботи системи
- Можливість відстеження кур’єра на карті, та отримати дані на якому етапі доставки замовлення він знаходиться.
- Можливість будувати коротші маршрути
- Якість візуальної частини

Оцінка систем доставки проведена відносно ефективності та швидкості роботи системи, легкості створення замовлення, доступності для людей з обмеженими можливостями, якості візуальної частини.

Таблиця 1.1 – Аналіз доставок представлених на ринку за критеріями

		Назва сервісу			
Функціональні рішення	Glovo	Bolt	Igogo		Raketa
Зручність створення замовлення	+	+	-		+
Швидкість роботи системи	5/5	5/5	3/5	4/5	5/5
Можливість відстеження	+	+	-	+	+

замовлення на карті					
Варіативність способів оплати(к-сть)	4	4	3	3	6
Можливість побудови коротших маршрутів	+	+	-	-	+
Якість візуальної частини	4/5	5/5	3/5	3/5	5/5
Наявність чату підтримки	+	+	-	+	+

Проаналізувавши отримані дані можна дійти висновку, що замовлення в системах розподіляється за таким принципом і на розподілення замовлень впливають такі характеристики, як:

Тип кур’єра – описує який транспорт використовує кур’єр для доставки.

Місцезнаходження – описує в якому місці на карті знаходиться кур’єр і який маршрут він має пройти щоб доставити замовлення.

Рейтинг – описує якість виконання кур’єром своїх обов’язків.

Рейтинг кур’єра це важлива складова проте найлегша для обчислення. Рейтинг формується з кількості виконаних завдань, загального часу доставки та оцінки, яку він отримав від клієнта після доставки.

Тип кур’єра також важливий для призначення замовлення. Для уникнення випадку, коли замовлення призначається кур’єру з невідповідним типом транспорту, якщо кур’єр сам не обрав це замовлення, існує градація для оцінки замовлення з якої відстані необхідно передавати іншій групі кур’єрів.

- Піший кур’єр – може покрвати відстань 1-1.5 кілометри
- Велокур’єр – оптимальна відстань замовлення від 1 до 2.5 кілометри
- Автокур’єр – Всі інші замовлення що мають відстань більше 2.5 кілометри.

Усі проаналізовані компанії використовують правило, що на одне замовлення призначають одного кур’єра, такий підхід може бути ефективний, але він збільшує необхідну кількість кур’єрів, що в свою чергу збільшує кількість витрат на оплату праці кур’єра.

В ситуації коли кількість кур’єрів недостатня для нормального виконання роботи, градація для оцінки замовлень матиме меншу вагу, і

замовлення навіть довші за маршрутом може отримати інший кур’єр, наприклад замовлення, що раніше призначалось велокур’єру переходить до пішого кур’єра, а замовлення автокур’єра переходить до велокур’єра.

Для початку роботи кур’єрові необхідно встановити додаток чи відкрити додаток у веб-версії, зареєструватись заповнивши звичайну стандартну форму з особистою інформацією.

Після попередньої реєстрації кур’єр отримує можливість вибору робочих днів на тиждень, за основу береться 2-4 робочих дні на тиждень, але з вільним графіком, головне завдання цього вибору, повідомити систему скільки орієнтовно вона матиме кур’єрів в кожен день.

Для кращої мотивації співробітників компанії зазвичай використовують дві пов’язані системи:

Система бонусів – працює беручи за основу кількість замовлень, які кур’єр виконує кожного тижня, для кожного сервісу це число своє, але загалом кожне десяте виконане замовлення підвищує відсоткову ставку в певному співвіношені. Також до системи бонусів можна віднести роботу в поганих погодних умовах. Більшість із вищезгаданих сервісів доставки, мають додаткові оцінки оплати роботи кур’єра якщо він працює в погану погоду наприклад дощ чи туман.
Така система заохочує кур’єра виконувати більше замовлень і правцювати швидше, адже він буде знати, що від швидкості виконання його замовлень, власне доставки замовлення, буде залежати його оплата.

Система рейтингу – якщо попередня система працювала загалом на кур’єра і не мала негативних наслідків, система рейтингу, може негативно впливати на роботу кур’єра, шляхом призначення менше замовлень.

Рейтинг кур’єра залежить від кількост, якості та швидкості виконаних замовлень за останній час. Чим більше виконаних замовлень, тим більший рейтинг в кур’єра, отже і замовлень він отримує більше, тому що система передбачає, що він його виконає і пропонує йому. Тоді ж як якщо замовлення не виконуються вчасно, чи взагалі не виконується або кур’єр отримує багато негативних відгуків незадоволени замовленням клієнтів, в такому випадку

система знижує рейтинг кур’єра, щоб він мав більше часу виконати завдання якісніше з максимально можливим результатом. Якщо кур’єр і далі буде продовжувати не виконувати замовлення, система понизить рейтинг кур’єра до «0», що призведе до того що кур’єр якись час не зможе отримувати замовлення.

1.2 Огляд проблеми оптимізації маршруту доставки

Оптимізація маршруту доставки передбачає процес скорочення часу доставки з точки отримання замовлення до точок до доставки замовлення до кінцевого клієнта. Така проблема може мати кілька шляхів вирішення, а саме:

- Побудова коротшого шляху з використанням математичних алгоритмів.
- Побудова шляхів з мінімізацією кількості часу проведення кур’єра в заторах.
- Реалізація кращого розподілу замовлень між кур’єрами.

Найпоширенішими стратегіями маршрутизації, які використовуються в сучасних системах є:

- Динамічна стратегія – керуються щоденно побудованими планами маршрутів створеними як щоденними, так і прогнозованими замовленнями, розробляються для ефективності обслуговування клієнтів.
- Статична стратегія – маршрути доставок будуються на основі зібраних і проаналізованих даних і вподобаннях споживачів на довгий термін.
- Динамічна стратегія в реальному часі – шляхи доставок залежать від даних, які отримуються системою в реальному часі і постійно оновлюють карту, змінюючи траєкторії кур’єрів.

Через складність вирахування усіх параметрів, що впливатимуть на скорочення часу доставки, задача оптимізації вимагає використання різних алгоритмів.

Такі алгоритми спрямовані на вирішення двох найскладніших проблем, класичної «Задачі комівояжера»[6] та проблеми маршрутизації транспортних засобів.

1.2.1 Задача комівояжера

Задача комівояжера (Travelling Salesman Problem, TSP) – це завдання, яке вважається класичним, яке полягає в пошуку найкоротшого, але найефективнішого шляху, що пройде через всі точки позначені на карті.

Очевидно, що можна вибрати багато різних шляхів, але пошук найкращого - того, для якого знадобиться найменша відстань чи найменші витрати - це те, на що математики та інформатики витратили десятки років, намагаючись знайти рішення. TSP привернув стільки уваги, тому що його так легко описати, але так складно вирішити.

Насправді TSP належить до класу комбінаторних задач оптимізації, відомих як NP-повна. Це означає, що TSP класифікується як NP складний, оскільки він не має «швидкого» рішення, і складність розрахунку найкращого маршруту збільшиться, коли ви додасте до задачі більше пунктів призначення. Приклад прокладення маршруту через точки наведено на рисунку 1.13.

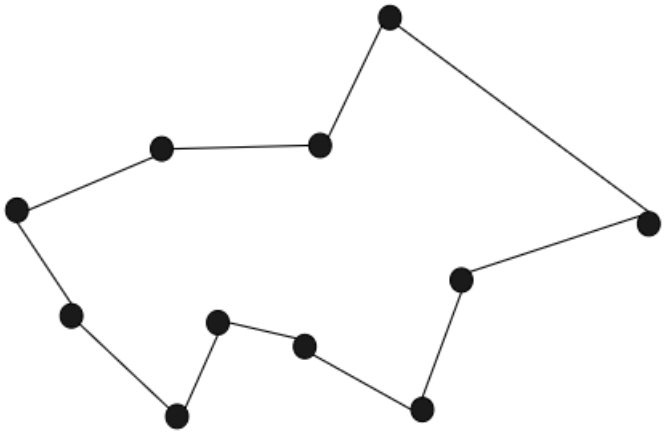


Рис 1.13. Прокладення маршруту через точки

Проблему можна вирішити, проаналізувавши кожен маршрут в обидва кінці, щоб визначити найкоротший. Однак зі збільшенням кількості пунктів призначення, прорахування відповідної кількості пересувань туди і назад перевершує можливості навіть найшвидших комп'ютерів. З 10 пунктів призначення може вийти більше 300 000 перестановок і комбінацій туди і назад.

1.2.2 Проблема маршрутизації транспортних засобів

Проблема маршрутизації транспортних засобів (VRP) - це проблема розробки оптимальних маршрутів від місця початку маршруту до набору пунктів призначення, кожен із яких має конкретні обмеження, такі як обмеження транспортних засобів, контроль витрат, часові вікна.

Це узагальнення проблеми комівояжера. Замість одного кур'єра - ціла команда з них. У такому випадку потрібно вирішити, як розподілити відвідування та оптимально призначити їх своїй команді. Це ще більш складна задача. Приклад відображення проблеми маршрутизації транспортних засобів наведено на рисунку 1.14.

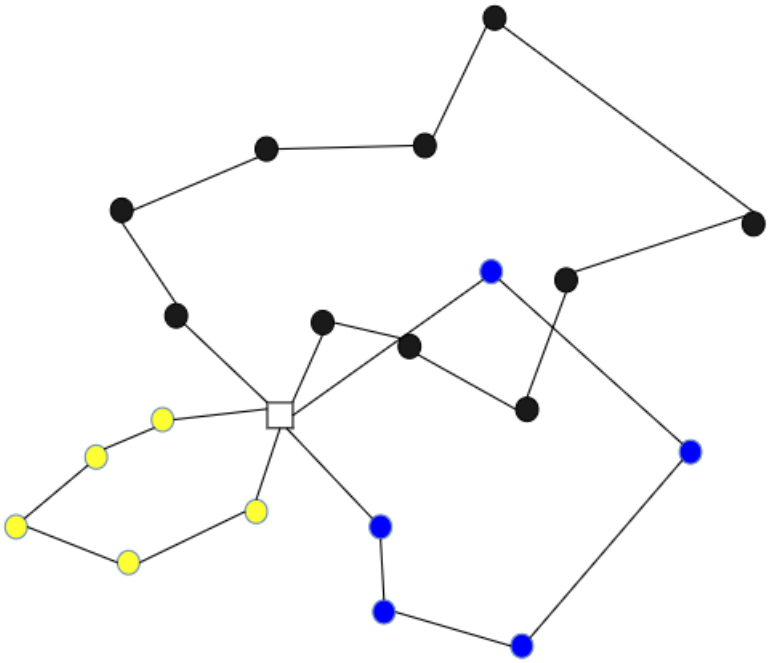


Рис 1.14. Відображення проблеми маршрутизації транспортних засобів.

1.3 Аналіз систем для оптимізації доставки

В сучасних реаліях для оптимізації систем доставки створено велика кількість систем і методів. Для проведення аналізу систем з оптимізації доставки було обрано одні з найкращих систем доступних на ринку а саме: Ant-Logistics і Gurtam.

Такі системи створені для вирішення таких завдань:

Збір усіх замовлень, які надходять в одному зручному місці, в якому легко ними керувати та організовувати, що збільшить швидкість роботи персоналу з такими даними.

Автоматичний розподіл замовлень між кур’єрами.

Оптимізація маршруту доставки

Зручний доступ до маршруту і часу доставки як для кур’єра, що виконує замовлення так і для клієнта, який може в будь-який час перевірити стан замовлення чи на якому етапі зараз знаходиться кур’єр.

Система оновлень замовлень, можливість швидкого підтвердження отримання замовлення клієнтом допоможе краще контролювати роботу кур’єрів.

Можливіть отримання статистики, а саме статистики замовлень, місць де зосереджена найбільша кількість замовлень в певний час доби.

Основні переваги таких систем:

- В перешу чергу це можливість автоматичного прокладання ефективного маршруту за допомогою алгоритмів оптимізації з врахуванням графіку роботи кур’єрів, часових вікон замовлення.
- Оцінка стану завантаженості трафіку, кількості аварійних ситуацій та коригування маршруту кур’єра відносно цієї інформації.
- Відстеження перебування кур’єра в реальному часі з допомогою GPS-трекера чи місцеположення мобільного пристрою з встановленим додатком.

– Ефективна комунікація між кур’єром та диспетчером в разі виникнення проблем із доставкою. Така комунікація реалізується з допомогою чату чи дзвінків.

Враховуючи вище згадані переваги, системи з оптимізації замовлень позитивно впливають на якість та швидкість замовлень, мінімізують кількість невиконаних замовлень та дають можливість для зменшення кількості кур’єрів при такій самій кількості виконаних замовлень.

1.4 Огляд методів відстеження замовлення

Для оптимальної реалізації відстеження замовлення варто використати один із методів відстеження за допомогою GPS.

GPS-система відстеження замовлення – це складне рішення, яке дозволяє визначити місцезнаходження кур’єра завдяки підключенню до глобальної системи позиціонування. Як правило, ця система встановлюється у вигляді пристрою стеження за транспортними засобами, який може використовувати технологію GPS, відстежувати в режимі реального часу та миттєво передавати дані на підключений мобільний додаток.

Крім того, можуть бути додаткові функції, такі як геозонування, моніторинг поведінки водія, планування маршруту та керування паливом.

Класичним прикладом програми GPS-трекінгу є Google Maps: ця система допоможе вам дістатися з точки А в точку Б будь-яким видом транспорту. Це один із найпопулярніших застосувань технології GPS. Серед переваг цієї системи являється:

– Оновлення місцезнаходження в режимі реального часу відображається на карті.
– Точний приблизний прибуток для покращення успішної доставки.
– Автоматичне коригування очікуваного часу прибуття відповідно до змвни обставин, наприклад, раптове завантаження трафіку.

– Прозорість маршруту (кількість зупинок і відстань) для підвищення довіри клієнта.

Задля візуалізації місцезнаходження транспортних засобів варто використати розширення-бібліотеку JavaScript Shipment Tracking Library

Бібліотека дозволяє і цікаві місця, які відстежуються в Fleet Engine. Бібліотека містить компонент карти JavaScript, який є заміною стандартної карти Google Map. Використовуючи бібліотеку JavaScript Shipment Tracking Library, можна запропонувати налаштовуваний анімований метод відстеження замовленнь у підсистемі автоматизованого управління доставкою.

Альтернативним варіантом для розробки підсистеми відстеження замовлення може стати Azure Maps. Це набір геопросторових служб, які використовують свіжі картографічні дані для надання даних про географічне положення та інших даних щодо прокладання маршрутів та візуалізації карт. Для цього слід використати модуль Spatial service, що досволить швидко проаналізувати інформацію про місцезнаходження та допоможе проінформувати клієнта про події що відбуваються із замовленням.

1.5 Висновки

В даному розділі було проаналізовано доступний ринок сервісів систем доставок. Проведено порівняльний аналіз інснуючих систем доставки та обрано критерії для побудови власної системи, Створено таблицю для порівняння обраних доставок за цими критеріями. Розглянуто можливі технології для реалізації відстежування замовлення в реальному часі. Для побудови ефективної системи доставки необхідно розв’язати задачу оптимізації швидкості доставки кур’єром за рахунок пошуку найкоротшого шляху з урахуванням ситуації на шляху, в ситуації, коли наявні кілька замовлень.

2. АНАЛІЗ ТА ОБГРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ СИСТЕМИ ДОСТАВКИ

2.1 Аналіз методів розробки системи

Через зручність створення замовлень через мережу, вони почали користуватись надзвичайною зручністю, з’явилась ціла ніша для різних сервісів і систем, які будуть реалізовувати доставку будь-яких товарів. І така ситуація з розвитком онлайн покупок чи інтернет замовлень, з часом буде тільки збільшувати кількість систем що доставляють замовлення. Таку тенденцію можна добре побачити на графіку 2.1 – що зображує рівень зацікавленості системами доставки, які отримують замовлення через мережу і доставляють кур’єрами до клієнта. Оглянувши цей графік можна також зробити прогноз, що така ситуація з сервісами доставок не запиниться на місці, а буде і далі розвиватись такими ж темпами.

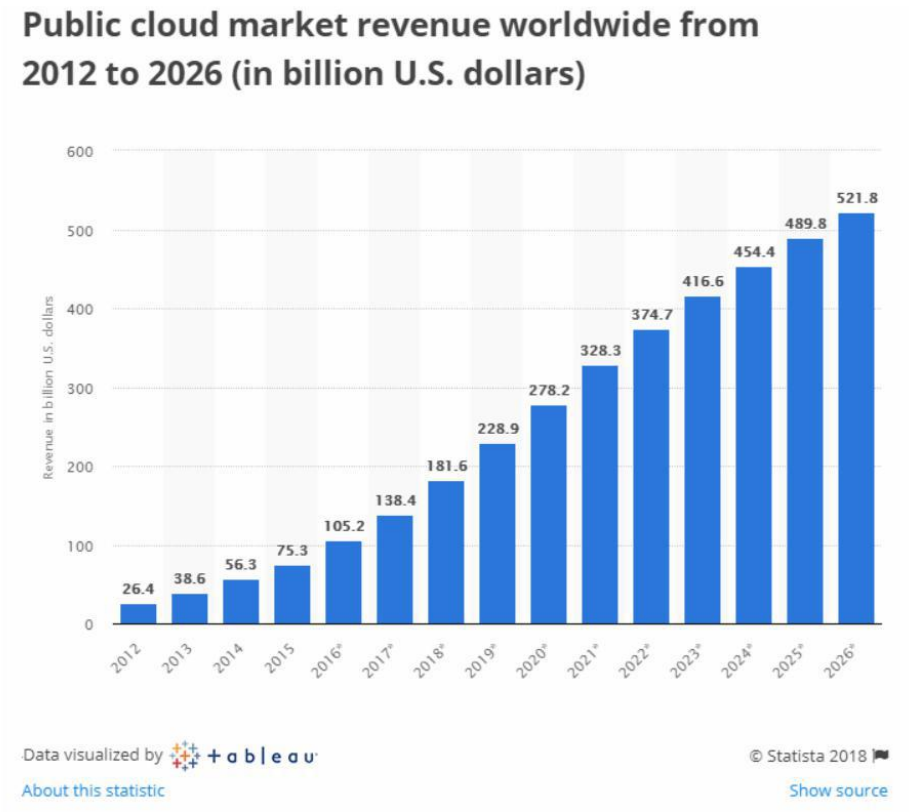


Рисунок 2.1 – Прогнозований рівень систем доставки

Для оптимальної роботи сервісу управління доставкою він має окрім правильного прийому замовлень і прокладання найкращих маршрутів ще бути зручним у використанні, швидким у роботі, і максимально зрозумілим і доступним для клієнта. Для реалізації таких завдань варто використати один із сучасних методів проектування і розробки веб-додатків. Такими методами можуть бути:

- Односторінковий додаток(SPA)[7]
- Багатосторінковий додаток(MPA)[8]
- Мобільний додаток[9]
- PWA – Progressive Web Application

Вищезгадані підходи мають за мету реалізацію додатка, що матиме можливість відкривати сервіс доставки з будь яких платформ, без втрати швидкості. Кожна з цих систем має і переваги і недоліки, що будуть впливати на подальшу роботу розроблюваної системи.

SPA – односторінкова програма – це реалізація веб-програми, яка завантажує лише один веб-документ у браузері, а потім динамічно переписує основний вміст цього єдиного документа за допомогою JavaScript API, таких як XML і Fetch, коли користувач взаємодіє із додатком і має бути показано інший вміст [10].

Таким чином, така система дозволяє користувачам використовувати веб-сайти без завантаження цілих нових сторінок із сервера, що може призвести до підвищення продуктивності та більш динамічного досвіду, з деякими компромісними недоліками, такими як SEO, більше зусиль, необхідних для підтримки стану, реалізації навігації та досягнення значущої продуктивності моніторинг. Це дає можливість не завантажувати на кожній ітерації такіж самі елементи системи, які були завантажені до попереднього, перезавантаження сторінки.

Односторінковий додаток(SPA) це додаток що працює у веб-браузері і призначений для створення в користувача досвіду близького від користування

нативною програмою. Принцип роботи SPA наведений у рисунку 2.2. Характеризується такий додаток зміною частини контенту без перезавантаження веб-сторінки [10]. Таким чином додаток не має необхідності завантажувати кожного разу одні й ті самі елементи, достатньо це зробити один раз. Якщо коротко описати принцип роботи, то загалом сервер отримує AJAX запит, обробляє його та відправляє інформацію на клієнт у JSON форматі [11].

Серед переваг можна виділити:

– Висока швидкість роботи.
– Збільшена швидкість розробки і масштабуваття, є можливість адаптації з мобільними телефонами, як нативна програма.
– Кешування елементів, що дозволяє не завантажувати їх на кожній ітерації.
– Має спільну базу даних як для мобільних, так і для веб-додатків, що збільшить швидкість при бажанні масштабування системи.
– Краща захищеність від зламів системи.

Попри всі свої переваги система також має кілька суттєвих недоліків таких як:

– Менша захищеність від підміни пакетів і шкідливих програм, що впливає на захищеність системи в цілому.

– Великі вимоги до техніки на яких цей додаток запускається, тому що необхідно локально зберігати в кеші великі об’єми даних.

– Великий розмір файлів формату JS, що при перших ітераціях сильно впливає на швидкість завантаження сторінки.

– Адаптування додатку до інших умов, зміна налаштувань, зміна мови, напрямку виведення тексту.

Вище перераховані недоліки мають серйозний вплив на сповільнення системи, але розглядати його все таки варто, тому що, за нормальних умов швидкість роботи збільшується до великих значень.

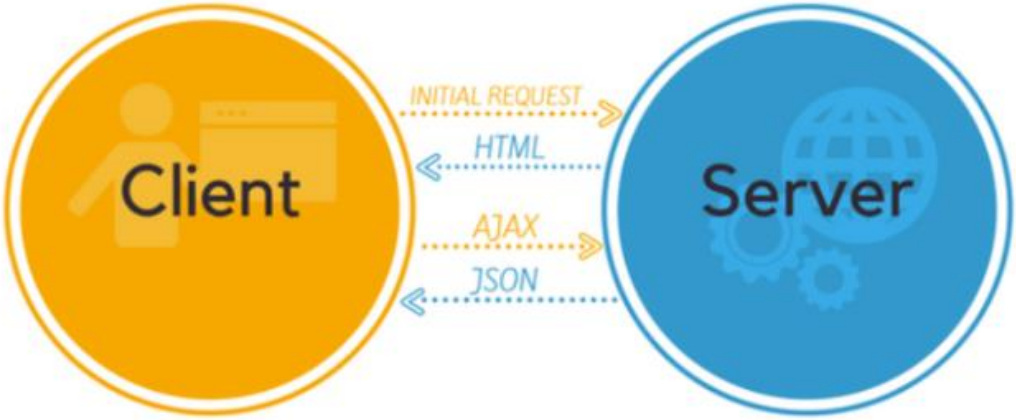


Рисунок 2.2 – Взаємодія клієнт-сервер при SPA.

MPA – Багатосторінковий додаток – працює за «традиційним» способом. Такий додаток зазвичай складається з кількох сторінок і кожна зміна чи дія, яку робить користувач, вимагає надсилання даних назад на сервер, чи отримування запитів від сервера. Таким чином для того щоб перейти на іншу сторінку, кожного разу необхідно передавати великий об’єм інформації. Сам принцип роботи наведений у рисунку 2.3.

Через велику кількість вмісту ці програми мають багато рівнів інтерфейсу користувача. На щастя, це більше не проблема. Завдяки AJAX більше не потрібно турбуватися про те, що важкі та складні програми повинні передавати багато даних між сервером і браузером. Це вирішення покращує і дозволяє оновлювати лише окремі частини програми. Хоча це вирішення з іншого боку теж не ідеальне, так як це додає більше об’єму роботи, і його складніше розробити, ніж односторінковий додаток.

Такий додаток це ідеальний підхід для користувачів, яким потрібна інтуїтивна карта того, як взаємодіяти з додатку. Надійна кількарівнева навігація меню є важливою частиною традиційної багатосторінкової програми.

Така реалізація дуже добре оптимізовується та легко змінюється для належного керування SEO. Це дає кращі шанси для рейтингу відображення сторінок в Google за різними ключовими словами, оскільки додаток можна оптимізувати для одного ключового слова на сторінці. Що призведе до кращої

індексації сторінки, що в свою чергу вплине на порядок відображення в пошуковій системі.

Серед переваг MPA варто зазначити:

– Менша вартість кінцевого продукту, це пов’язано з тим що необхідно менший досвід роботи для реалізації такої системи.

– Краща оптимізація системи, через відсутність великих файлів що завантажуються локально.

Недоліками ж такої системи є

– Неможливість кешування великих файлів, що призводить до вртати швидкості при кожному наступному завантаженню додатка.

– Збільшення кількості коду за рахунок кількох версій додатка

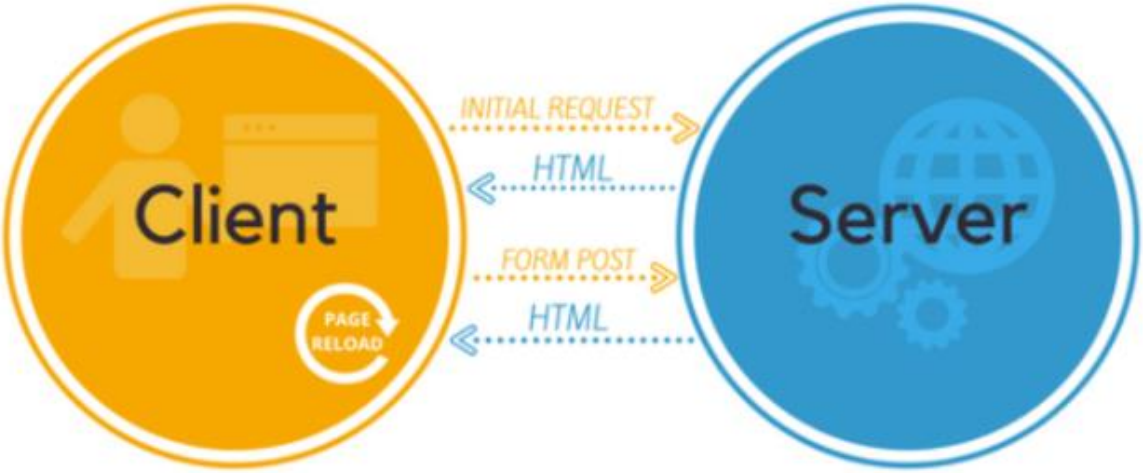


Рисунок 2.3 – Взаємодія клієнт-сервер при MPA.

Розуміння різниці між односторінковими та багатосторінковими програмами може допомогти вирішити, який тип структури додатка потрібний для реалізації системи доставки. Якщо порівняти ці дві методології написання додатків можна дійти таких висновків:

Багатосторінкові програми краще оптимізовані для SEO, щоб користувач міг легко знайти свою аудиторію, тоді як односторінкові програми не зовсім оптимізовані для цього.

Односторінкові програми легко розробляти та підтримувати, тоді як багатосторінкові програми можуть бути складними для розробки та обслуговування.

Багатосторінкові програми відповідають стандартам високого рівня безпеки, тоді як односторінкові програми менш безпечні.

Одна з переваг односторінкових додатків полягає в тому, що вони не здійснюють зворотний перехід, коли користувач взаємодіє з системою, тоді як багатосторінкові додатки здійснюють запит до сервера для кожного запиту клієнта.

Односторінкові програми сумісні з офлайн-сервісом, коли з'єднання з мережею переривається, тоді ж як багатосторінкові програми потребують з'єднання для виконання запитів користувача.

Незважаючи на те, що SPA сьогодні є поширеним явищем, проблеми з пошуковою оптимізацією (SEO) сильно впливають на якість відображення в мережі, керуванням історією веб-переглядача та вирішенням питань безпеки, пов'язаних із виконанням JavaScript. Як результат існує багато випадків, коли MPA стає кращим вибором в таких умовах.

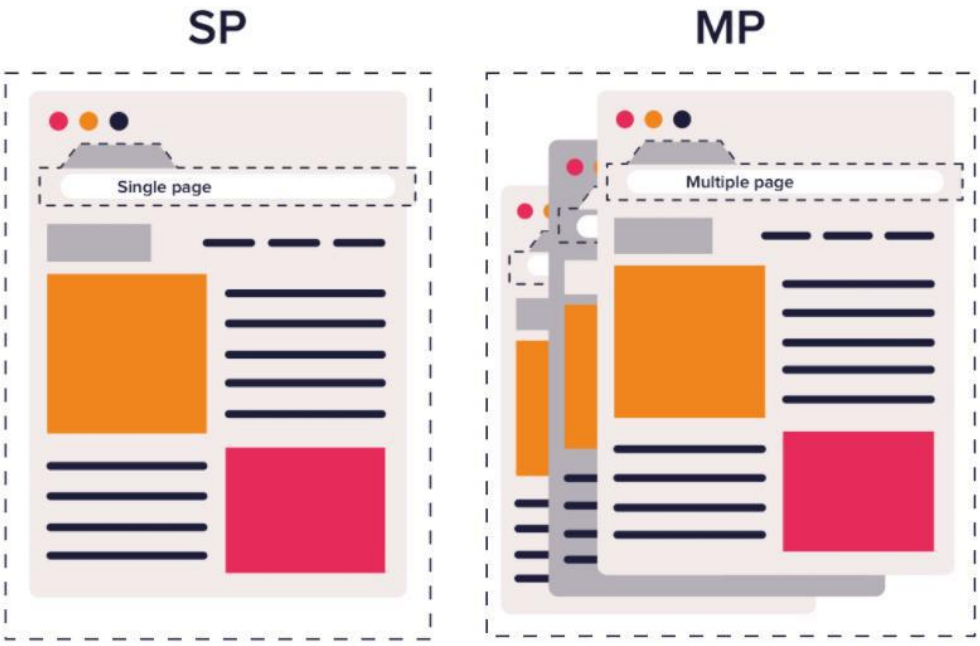


Рисунок 2.4 – Візуалізація різниці між SPA та MPA

PWA – Прогресивний веб-додаток – являється перехідним етапом, між односторінковим і багатосторінковим додатком. Реалізований такий додаток має структуру середню між нативною програмою і веб-сайтом. Для запуска, користувачу спочатку необхідно встановити цей додаток в систему, подальший досту відбувається через взаємодію із значком на робочому столі. Після відкривання сайт розгортається і відкривається без використання браузера.

Серед переваг такого додатку:

- Кросплатформенність – сервіс буде працювати на будь-яких операційних системах
- Сервіс швидко встановлюється і розгортається, немає потреби переходити вв маркети додатків.
- Доступ в оффлайн режимі
- Легка можливість налаштування push-повідомленнь

Серед недоліків, такіж як і в одно сторінкового додатка гірша SEO оптимізація і важче налаштування, хоча загалом система працює швидше.

Мобільний додаток – це додаток, який створений для використання на смартфонах. Стандартний мобільний додаток використовує мережеве з’єднання для роботи з віддаленими обчислювальними ресурсами. Таким чином, процес розробки для мобільних пристроїв передбачає створення пакетів програмного забезпечення, які можна встановити, впровадження серверних служб, таких як доступ до даних за допомогою API, і тестування програми на цільових пристроях.

У перші роки розвитку мобільних додатків єдиним способом гарантувати оптимальну роботу додатка на будь-якому пристрої була його власна розробка. Це означало, що новий код потрібно було написати спеціально для конкретного процесора кожного пристрою. Сьогодні більшість розроблених мобільних додатків не залежать від пристрою.

Як і розробка веб-додатків, розробка мобільних додатків бере свій початок у більш традиційній розробці програмного забезпечення. Але є одна велика відмінність що полягає в тому, що мобільні програми зазвичай пишуться

спеціально під використання переваг унікальних функцій конкретного мобільного пристрою. Наприклад, ігровий додаток може бути написаний, щоб використовувати переваги акселерометра iPhone, або мобільний додаток для здоров'я може бути написаний, щоб використовувати переваги датчика температури розумного годинника.

Основна проблема в реалізації системи доставки з використанням мобільного додатку, це відмінність двох найбільших мобільних систем Android та IOS. Для такої реалізації буде необхідно дів версії додатка написаних з використанням різних інструментів, через те що вони не взаємозамінні, і програми написані для однієї системи не будуть працювати в іншій. Для написання сервісу доставки на андроїд необхідно буде використати Java[49] чи Kotlin[14]. Для написання мобільного додатку на IOS, необхідно буде використати Swift[15].

Перевагами написання сервісу з використанням технологій мобільного додатку буде:

–Можливість працювати в оффлайн, за відсутності мережі

–Краща захищеність ніж в одно сторінкового чи багатосторінкового додатку, це пов'язано з неможливістю довантажувати пакети і файли в систему.

Основними недоліками такої розробки виступають:

–Більша ціна розробки, що пов'язано з різницею мобільних систем.

–Для використання додатку, його спочатку потрібно встановити на пристрій користувача.

Порівняння вигляду головних сторінок веб-додатку та мобільного додатку наведено на рисунку 2.5.

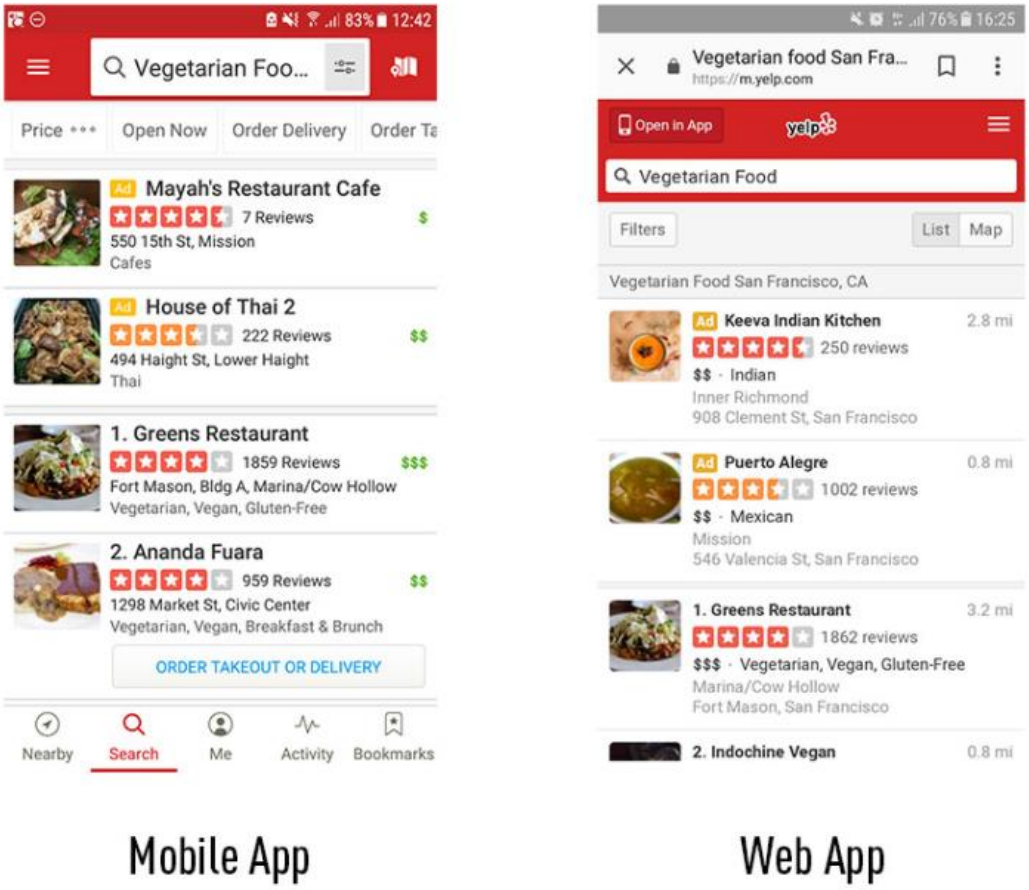


Рисунок 2.5 – Порівняння веб-додатку та мобільного додатку

2.2 Вибір методів й інструментів розв’язання задачі

Провівши аналіз систем розробки сервісу, було порівняно кожен з методів, найоптимальнішим виявився метод створення системи Багато Сторінкового Додатка(MPA). Так як вона однаково добре відображається і на мобільних системах і на десктопних системах. Також MPA має перевагу в швидкості, через те що намає необхідності зберігати локально великі об’єми файлів, система працюватиме швидше і матиме кращу SEO оптимізацію. Така система добре показує себе у написанні інтернет магазинів чи схожими за структурою сервіси. Для прискорення швидкодії рекомендується окремі модулі розробляти з використанням технології SPA.

Для побудови інтерфейсу веб-додатку в першу чергу, необхідна реалізація структури. Для цього потрібно розподілити частини екрану відносно до їхнього

функціоналу за допомогою мови тегів HTML. Ця мова створена для розмічування сторінки веб=документа відносно контенту, створення логічних блоків, створення компонентів що будуть використовуватись в майбутньому, створення елементів сторінки. Після виконання цих процесів браузер відображає графічні і текстові матеріали. HTML створений, щоб передавати логічно та семантично розмічені дані з бази даних сервера або локальної пам’яті до сторінки браузера, де цей код буде оброблятись

Серед основних призначень цієї мови є:

- Логічне розділення інформації на секції
- Створення елементів та компонентів
- Форматування тексту, а саме робота з заголовками, текстом та списками.
- Додавання графічних та відео-матеріалів до веб-сторінки.
- Створення форм

Створена структура додатку це лише початкова стадія, наступна стадія стилізація створеної розмітки і надання коректного вигляду. Для реалізації цього завдання необхідно використати каскадні таблиці стилів, CSS. Такі таблиці створені для надання стилізації готової розмітки, шляхом надання їй кольору, форм, зміни розміщення елементів на сторінці, зміна використовуваних шрифтів.

Серед переваг що дає така стилізація є:

- Кращий вигляд документу;
- Краща структурованість контенту;
- Кращий користувацький досвід;
- Збільшення універсальності щодо інших пристроїв;
- Швидше завантаження веб-історінки;
- Краща індексація веб-сторінки;
- Краща адаптація для людей з обмеженими можливостями[18];

Для оптимізації роботи зі стилями слід використати одну із існуючих технологій обробки таблиці стиців пропроцесор Sass[19]. Sass – скриптова

метамова, що інтерпретується в каскадні таблиці стилів, призначена для підвищення рівня абстракції коду та спрощення файлів CSS.

Серед переваг такої технології є :

–Вкладеність – Sass дозволить вкладати селектори CSS у спосіб, який слідує тій же візуальній ієрархії що і HTML.

–Міксини – Міксин дозволяє створювати групи оголошень CSS, які ви можна повторно використовувати на своєму сайті. Це допомагає зберегти Sass дуже чистим.

–Наслідування – наслідування допомагає ділитися набором властивостей CSS від одного селектора до іншого.

–Змінні – змінні використовуються, як про спосіб зберігання інформації, яку можна повторно використовувати у таблиці стилів. Можна зберігати такі речі, як кольори, стеки шрифтів або будь-які значення CSS.

Для обробки та інтерпретації коду отриманого з препроцесора варто використовувати один із бандлерів або таск-ранерів, наприклад Webpack[23] або Gulp[25] або Grunt[26].

Webpack це сервіс створений для обробки коду і збору його з окремих модулів в один файл, що отримує браузер для виконання в кінці обробки. Він створює граф залежностей, що допомагає веб-розробникам використовувати модульний підхід при створенні додатків. Має можливість для запуску і роботи з командного рядка або налаштуватись за допомогою конфігураційного файлу, що використовується для визначення плагінів і файлів для проекту [24], та запускатися з графічного інтерфейсу.

Для роботи Webpack, необхідний один із менеджерів пакунків. В проекті використаний Yarn [28], необхідний для користування модулями JavaScript. Альтернативний менеджер – NPM [29]. Порівняння цих пакетних менеджерів наведено в таблиці 2.1.

Таблиця 2.1 – Порівняння Yarn і NPM

	Yarn	NPM
Продуктивність	29 секнд необхідні для збору пакетів	61 секнда необхідна для збору пакетів
Безпечність	Оцінює кожен пакат і під час передачі періодично аналізує цілісність пакетів	Має захист лише від стандартних відомих вразливостей
Оновлення	Для кращої роботи з оновленнями необхідно використовувати yarn.lock	Є аналог lock файла, може використовувати для оновлень package.json

Порівнявши менеджери для управління модулями було виявлени, що різниця між ними невелика, і для проекту не принципова. Проте Yarn показує кращі результати при порівнянні, тому варто використати його.

Отриманий веб-додаток це ще не закінчення роботи, на даному етапі він тільки створений і стилізований, але функціоналу і анімацій в нього ще намає. Для написання логіки для цієї системи варто використати мову JavaScript[30], це стандартна та універсальна мова для написання скриптів, що можна запустити на будь-якому пристрої. JS виконуючи скрипти дозволяє реалізувати на веб-сторінці будь-який сценарій, наприклад можливість клієнта взаємодіяти з браузером, можливість обмінюватися даними з сервера, можливість зміни структури та зовнішнього вигляду[31]. Альтернативними варіантами є використання фреймворків Angular [32, 33] і React[34].

Для виконання взаємодій з сервером для цього проекту цілком вистачить звичайного нативного JS, і не має сенсу використовувати сторонні бібліотеки для цього.

До переваг використання JS варто віднести малий об’єм та дуже легку конфігурацію, а так як таке використання зменшить об’єм даних, що дозволить збільшини швидкодію. Для роботи з DOM [35] і використання AJAX [36] JS має повністю відповідати потребам. JavaScript працює із сторінкою визначеною HTML і пов’язує форму замовлення з моделлю бази даних.

Серед головних переваг Javascript є:

Універсальність – система, написана з його допомогою може запуститись на будь-якому девайсі, без жодних проблем.

Модульність – дозволяє створювати цілі сценарії із скриптів та масштабувати систему.

Швидкість – зменшення об’єму файлів допоможе збільшити швидкодію.

2.3 Огляд сервісів побудови маршрутів

В сучасних реаліях у великому місті неможливо орієнтуватись без карт в незнайомих місцях. А в ситуації коли потрібно відвідати різні місця й продумати маршрут, без карт зовсім не обійтись. В такому випадку варто обрати сервіси що надають картографічні послуги. Для огляду і аналізу в цій роботі обрано такі сервіси як Google Maps[39] і Bing Maps[40]. Картографічні служби – ефективний інструмент, що дозволяє створити позитивну репутацію в мережі та збільшити лояльність клієнтів. Якому сервісу віддати перевагу залежить від цілей інтеграції. Наприклад, якщо ви хочете потрапити на перші сторінки в пошуковій видачі на локальному рівні, виберіть Bing Maps або Google Maps.

2.3.1 Google Maps

Google Maps – безкоштовний картографічний сервіс, а також набір додатків що створений компанією Google і набув широкого використання завдяки. Сервіс являє собою географічну карту і надає можливість для прокладання маршрутів, аналізу дорожнього трафіку в реальному часі.

Для розробників Google Maps являються зручним і ефективним інструментом через можливість інтегрування карт на будь-який веб-сайт з використанням CDN скрипта або спеціальним API, яке легко використати і підтримувати через добре описану документацію.

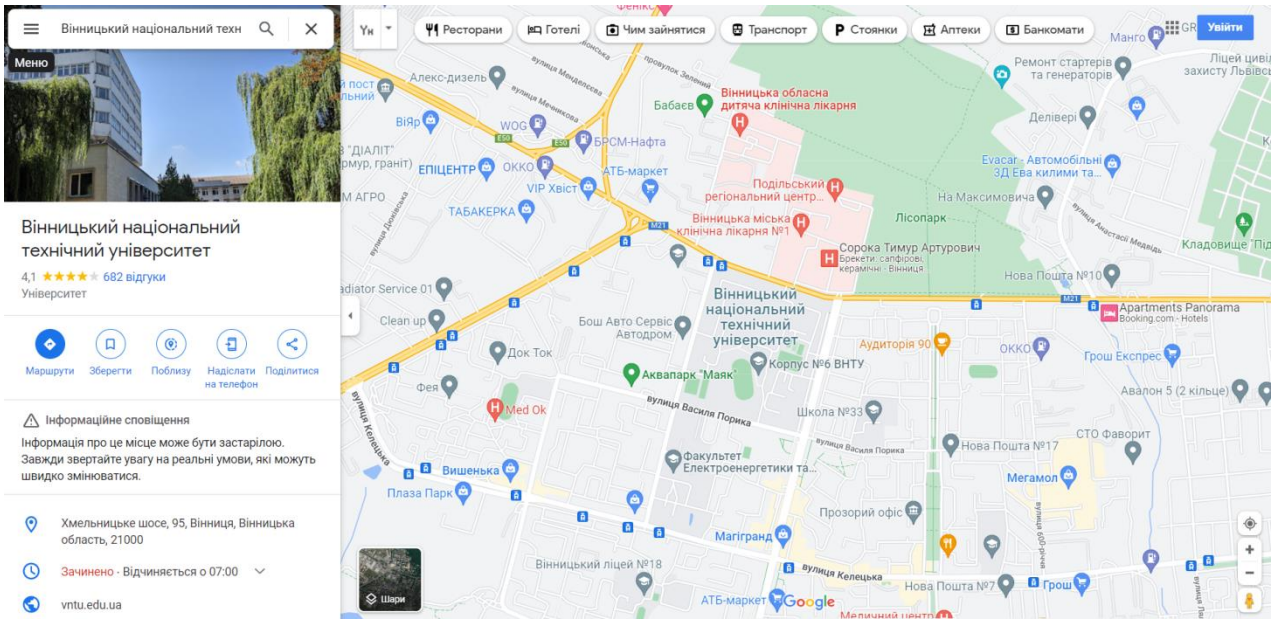


Рисунок 2.6 – Вигляд сервісу Google Maps

Серед основних переваг використання сервісу Google Maps є:

– Поширеність – на сьогоднішній день це найбільш поширений і підтримуваний картографічний сервіс.

– Простота у використанні

– Зрозумілість як і для користувача так і для розробника

– Можливість розширення. Сервіс можна стилізувати під будь-якиу тематику веб-сторінки. Також є можлививість розширити функціонал карти.

Недоліком є те що використання Google з допомогою api не є безкоштовним, для цього спочатку треба пройти реєстрацію і замовити спеціальний ключ.

2.3.2 Bing Maps

Bing Maps – картографічний сервіс створений компанією Microsoft [], і являється частиною порталу Bing. Компанія Microsoft намагалась зайняти нішу найкращого картографічного сервісу, але витримати конкуренцію з Google Maps не вдалось, тому що карти від Google набули більшої популярності через

доступніше і зручніше API для інтеграції створених ними карт на різноманітні сайти.

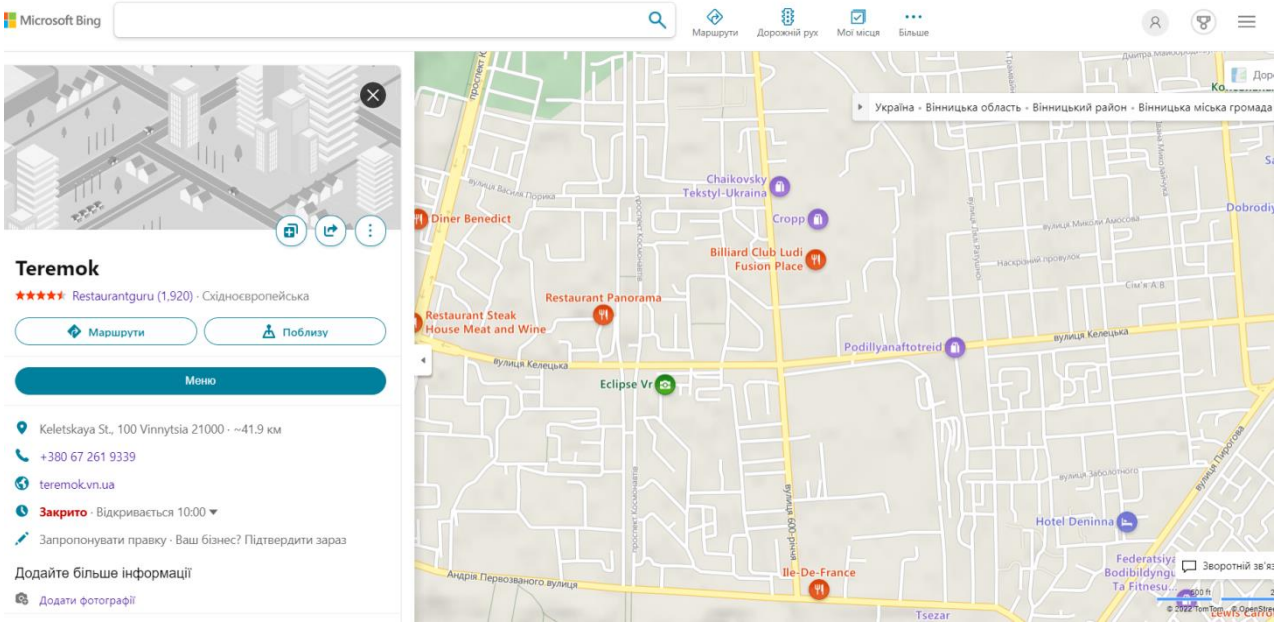


Рисунок 2.7 – Вигляд сервісу Bing Maps

Серед переваг сервісу Bing Maps є:

– Висока деталізація карти
– Швидкодія
– Вільне розповсюдження Api ключів

Для використання Bing Maps на будь-якому сервісі недхідно використати спеціальні ключі саме необхідної платформи.

Проаналізувавши картографічні сервіси, можна припустити, що Google Maps є більш вигідним для інтеграції в систему оптимізації маршруту замовлення, через більш широке поширення, кращу підтримку, та краще написану документацію, що допоможе при розробці даної системи.

2.4 Аналіз методів отримання динамічних даних

Під визначенням динамічні дані в цій роботі мається на увазі:

– шлях доставки, альтернативні шляхи доставки;

- координати місць доставки;
- стан завантаженості дорожнього трафіку;
- кількість аварій на прогнозованому шляху кур’єра;
- час доставки.

Для отримання динамічних про шлях і координати в першу чергу слід використати сервіс Google Maps. Так як це найпоширеніший сервіс для створення й перегляду карт на даний момент.

Для пошуку найкоротшого шляху доставки одним із алгоритмів необхідно отримати дані про координати на певній визначеній території. Для обробки таких даних необхідно їх виразити в числах можливих для побудови графів. Це можливо зробити, експортувавши дані з Google Maps в форматі JSON. Один з найкращих сервісів для вирішення цієї задачі це GeoJson[41] – відкритий і загальнодоступний стандартний формат для представлення простих географічних об’єктів з їх непросторовими атрибутами. Він заснований на форматі JSON.

GeoJson передає інформацію за допомогою функцій . Функції включають в себе точки, а саме адреси та місцезнаходження, полілінії, такі як: дороги, шосе та кордони, полігони (тобто країни, провінції, ділянки землі) і багатокомпонентні колекції цих типів. Функції GeoJSON не обов’язково представляють функції лише у фізичному світі; Наприклад, мобільні додатки для маршрутизації та навігації можуть описувати охоплення своїх послуг за допомогою GeoJSON.

Формат карт GeoJson різняється від інших стандартів Геоінформаційних технологій[42], тому що він написаний і підтримується за допомогою групи звичайних розробників, ніяк не пов’язаний з компаніями і не має підтримки організації з стандартизації.

Для прикладу приведено група координат і шлях між ними побудованих з використанням GeoJson (Рисунок 2.8) і відображення цих даних в форматі JSON(Рисунок 2.9).

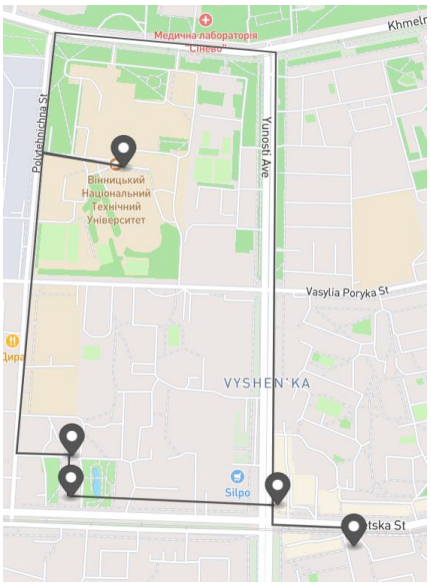


Рисунок 2.8 – Виділена група координат та прокладений шлях

Для побудови графа для оптимізації маршруту використовується таке представлення. Мережа доріг подається графом з завжди позитивними вагами. Вершини виступають дорожніми перехрестями, а дуги – виконують функцію доріг, що їх з'єднують. Вага дуг відповідає протяжності даної ділянки, часу необхідного для його подолання або вартості подорожі нею. Для позначення односторонніх вулиць можна використати орієнтовані ребра. У такому графі можна ввести характеристику, яка вказує на те, що одні дороги важливіші за інші для тривалих подорожей.

Існує багато алгоритмів для реалізації підходу, коли одні вулиці важливіші за інші. Вони вирішують завдання пошуку найкоротшого шляху набагато швидше, ніж аналогічні на звичайних графах. Подібні алгоритми складаються з двох етапів:

Етап опрацювання даних. Граф попередньо обробляється без урахування початкової та кінцевої вершин (може зайняти до кількох днів при роботі з реальними даними). Зазвичай він виконується один раз, а потім використовуються отримані дані великий проміжок часу.

Етап побудови. Виконується запит і пошук найкоротшого шляху, поки відомі початковий і кінцевий вузли.

Для оптимізації графів такого виду рекомендується використовувати такі алгоритми:

– Ієрархії скорочень [43]
– Reach based Pruning [44]
– Transit node routing [45]
– ALT алгоритм (A* алгоритм)


```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {},
      "geometry": {
        "coordinates": [
          28.407307620278885,
          49.227569579937466
        ],
        "type": "Point"
      }
    },
    {
      "type": "Feature",
      "properties": {
        "1": 3
      },
      "geometry": {
        "coordinates": [
          28.412539088202067,
          49.22596469396092
        ],
        "type": "Point"
      }
    },
    {
      "type": "Feature",
```


Рисунок 2.9 – Візуалізаця вищенаведених даних в форматі GeoJson

2.5 Вирішення задачі з оптимізації доставки

За умови, коли кур’єрові для виконання замовлення необхідно відвідати кілька точок протягом свого маршруту, з’являється потреба в оптимізації шляху доставки.

Для побудови найкращих методів пошуку найкоротшого шляху зазвичай використовується не один алгоритм оптимізації графів, а певна комбінація таких алгоритмів.

Одним із можливих алгоритмів для вирішення цієї задачі є алгоритм ієрархії скорочень. Це алгоритм для прискорення пошуку найкоротшого шляху в графі. Зазвичай він використовується в автомобільних навігаційних системах. Його можна описати таким чином: Кур'єр має добратись із точки А до точки Б за найкоротшим можливим шляхом. В даному випадку перехрестя представлені вершинами, а дороги що їх з'єднують виступають дугами. Маршрут від А до Б це послідовність дуг і мінімальний шлях це сума дуг що мають найменшу вагу.

Прискорення що дає алгоритм отримується за рахунок створення поміток на етапі попередньої обробки карти, які потім використовуються під час запиту найкоротшого шляху. Це пояснюється тим що дороги це майже не змінні вхідні дані які мають їєрархічніть. Наприклад деякі перехрестя є користуються більшою популярністю і є більш важливими. Створені помітки можна використовувати для збереження попередньо обчисленої відстані між двома переходами, що дає змогу алгоритмові не вираховувати шлях між цими вершинами кожного разу під час запиту. Алгоритм ієрархії скорочень не визначає, які дороги люди вважають «важливими», але він обробляє вхідні дані таким чином що вони можуть надавати значення вершинам за допомогою евристики.

Алгоритм ієрархії скорочень складається з двох етапів:

- Етап попередньої обробки даних
- Етап запиту

Етап попередньої обробки існує по тій причині, що транспортні мережі майже не змінюються і має сенс для початку провести обчислення. Такі попередні обчислення дадуть змогу в майбутньому будувати шляхи в найкоротший термін, так як всі дані будуть відомі. Під час проведення попередньої обробки алгоритм виключає певні зони і використовує ярлик замість нього, це призводить до зменшення вершин, які необхідно обробити алгоритму. На рисунку 2.10

представлено візуалізацію такого скорочення. Шлях uv виключається і замість нього створюється ярлик з вагою 4.

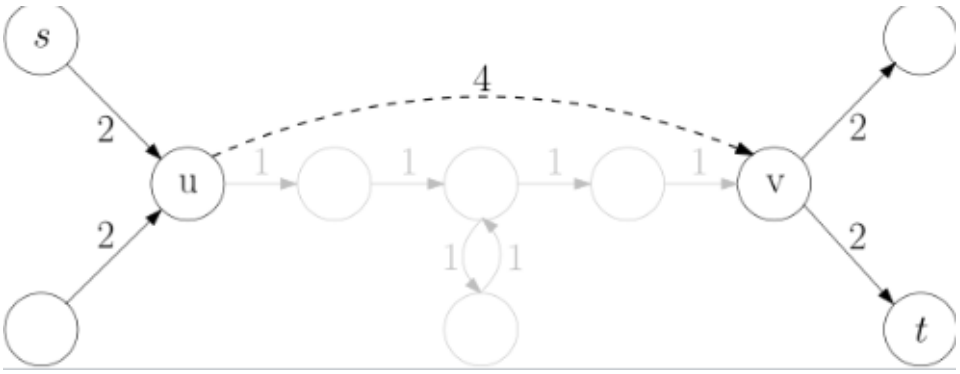


Рисунок 2.10 – Приклад створення помітки-ярлика

Якщо взяти для прикладу зважений граф який має вигляд як на рисунку 2.11

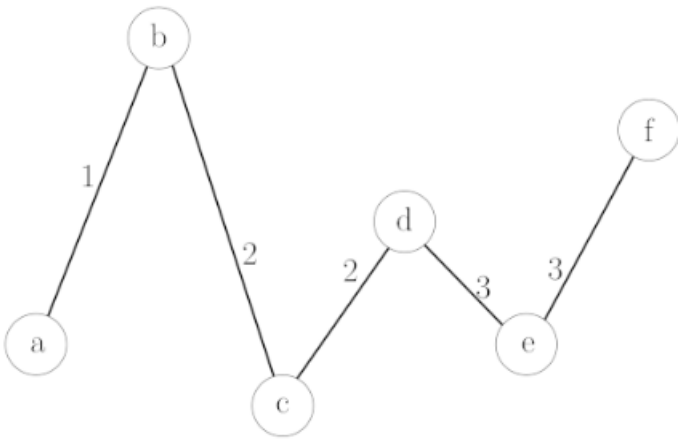


Рисунок 2.11 – Початок роботи алгоритму

Наступним кроком створимо ярлики для проміжних вершин, це зображено на рисунку 2.12

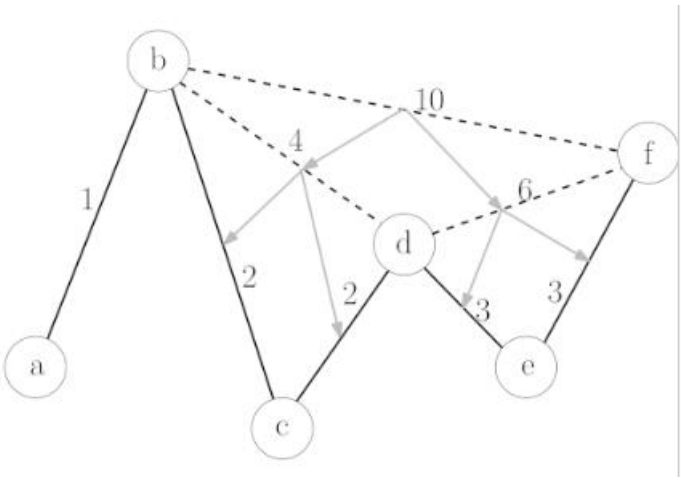


Рисунок 2.12 – Створення ярликів

При пошуку найкоротшого маршруту від s до t, пошуки вперед і назад потрібно виконувати лише по дугах, які йдуть вгору в ієрархії. Знайдений шлях позначений червоним кольором і використовує один ярлик(Рисунок 2.13)

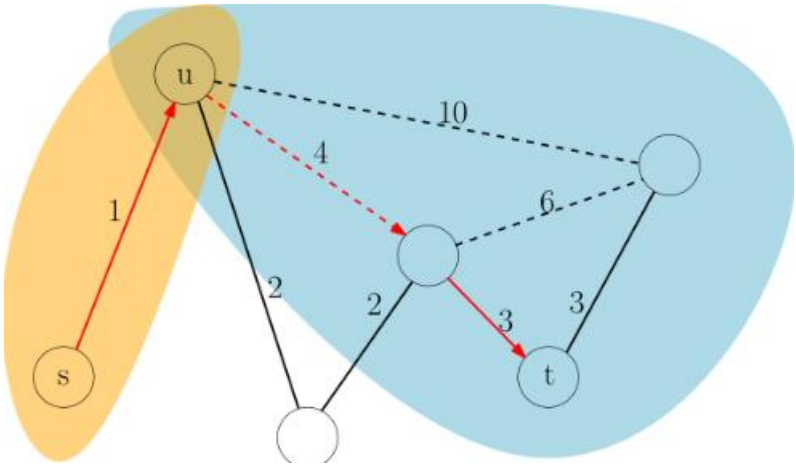


Рисунок 2.13 – Пошук найкоротшого маршруту

Ще одним із варіантів вирішення цієї задачі, використання алгоритму Дейкстри. Алгоритм зображений на рисунку 2.11 [46].

Алгоритм Дейкстри створений для вирішення задачі з пошуку найкоротшого шляху. Цей алгоритм має змогу працювати лише за умови коли в графі всі дуги відомі і мають невід'ємні значення. Додатні значення дуг необхідні тому що, алгоритму під час виконання роботи з пошуку найкоротшого шляху необхідно додати вагу зважених дуг. Якщо відштовхуватись від завдання, то матимемо граф у якому його вершини мають відповідну точку на карті. Дуги ж відповідають з'єднанням цих точок на карті, тому їхня вага відома, і граф можна визначити, як зважений.

Початок алгоритму можна охарактеризувати так:

Кур'єр отримує замовлення і починає свій маршрут з тієї точки координат де він знаходиться в дений час. Наступним етапом буде початок руху кур'єра до відомих точок, відстань до яких менша за початкову вагу. Процес відбувається до того часу, поки кур'єр не завершить виконання всіх замовлень і на прибуде до фінальної точки координат, яка являється його місцем призначення.

Отже завдання, це типова задача комівояжера, для вирішення якої можна обрати один із альтернативних алгоритмів пошуку найкоротшого шляху.

На початковому етапі алгоритму маємо деяку кількість вершин, що мають найкоротші маршрути, що пролягають з тоочки початку алгоритму до позначених точок.

Наступним етапом буде початок виконання алгоритму, а саме:

Необхідно розглянути усі ребра, що проходять з позначених вершин в одну непозначену. Усі ребра що підпадають під це опис, являються останніми на шляху від точки початку алгоритму до останньої непозначеної координати

В першу чергу необхідно позначити всі вершини графа не відвіданими. Початкова точка встановлена, з неї й починається робота алгоритму. Далі точка, що суміжна до початкової і є найближчою до неї буде наступною очікуваною вершиною.

Визначення початкової вершини представлено на рисунку 2.14.

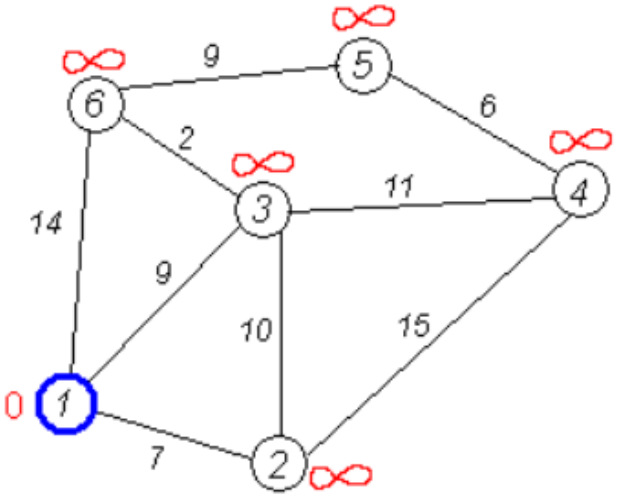


Рисунок 2.14 – Визначення точки початку роботи алгоритму

За умовою, ми мємо зважений граф, тому і маємо представлення про ваги всіх ребер до інших наближчих точок. Наступним етапом буде вибір з відомих шляхів накоротшого, перехід до наступної точки та позначення щойно відвіданої точки.

Визначення найкоротшого шляху представлено на рисунку 2.15.

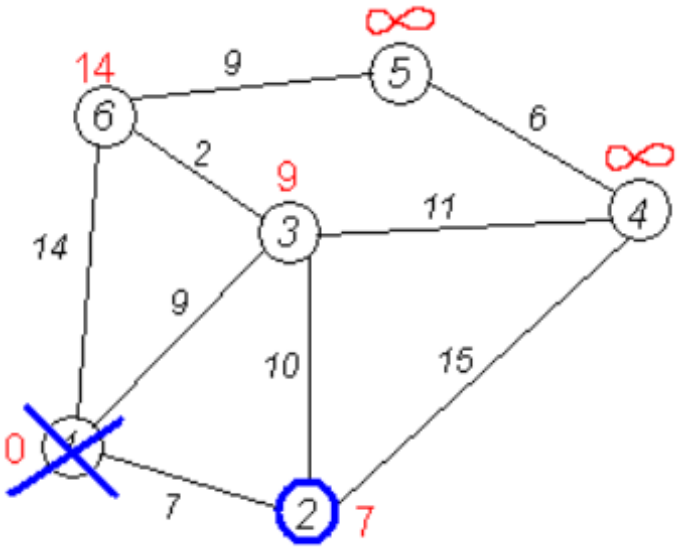


Рисунок 2.15 – Визначення найкоротшого шляху алгоритму

Після виконання цього етапу, циклічно повторюємо останню ітерацію з вибором точки з найкоротшим шляхом і переходим до неї. З часом число непозначених вершин буде зменшуватись

Побудова повного маршруту алгоритму представлена на рисунку 2.16.

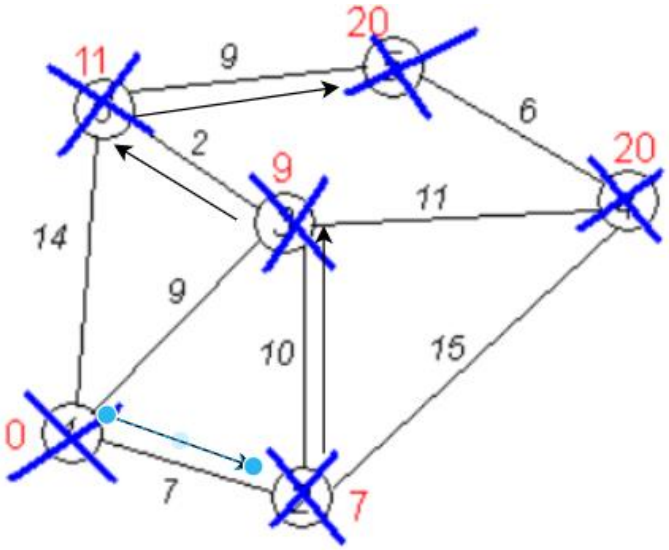


Рисунок 2.16 – Побудова найкоротшого маршруту алгоритму

Описаний алгоритм зможе завершитися в тій ситуації, коли всі вершини будуть позначеними. В результаті виконання цього алгоритму отримаємо дерево найкоротших шляхів, що й було необхідно.

Ще одним варіантом алгоритму, що може допомогти вирішити поставлену задачу є алгоритм A*[47].

A* алгоритм – це простий але ефективний алгоритм пошуку, що можна використати для пошуку оптимального шляху між двома вузлами на графі. Він буде використовуватися для пошуку найкоротшого шляху. Цей алгоритм являється розширенням алгоритму пошуку найкоротшого шляху Дейкстри. Розширення полягає в тому, що замість використання черги пріоритетів для зберігання всіх елементів, ми використовуємо бінарні дерева для їх зберігання. Алгоритм пошуку A* також використовує евристичну функцію, що надає додаткову інформацію про те, наскільки далеко ми знаходимося від цільового вузла. Ця функція використовується в поєднанні зі структурою даних f-heap, щоб зробити пошук більш ефективним. Дуже часто практичне використання сього алгоритму можна зустріти в додатках карт чи навігаторів.

В навігаторах цей алгоритм використовують для визначення найдальшої відстані між точкою початку алгоритму і кінцем. Такий алгоритм * має 3 параметри:

- G: являє собою вартість переходу від комірки початку роботи алгоритму до поточної комірки. В основному це сума всіх відвіданих вершин з моменту виходу першої осередку.
- H: виступає евристичним значенням, це приблизна очікувана ціна переходу від поточної комірки до останньої комірки. Фактичні витрати не можна розрахувати, поки не буде досягнуто останню комірку.
- F: - сума двох вищезгаданних зимінних, g і h.

Для опису алгоритму обрано граф, який зображено на рисунку 2.17.

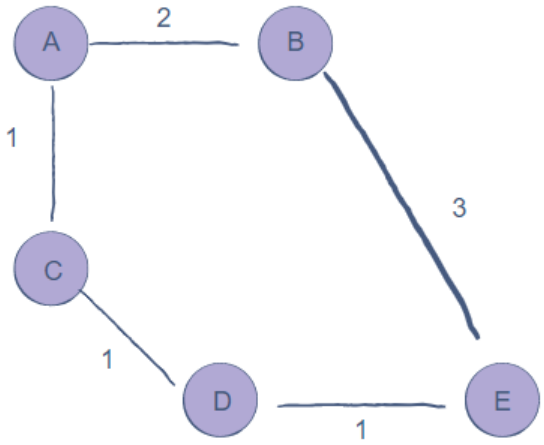


Рисунок 2.17 – Граф для розв’язку алгоритму A*

Евристика — це в основному освічені припущення. Важливо розуміти, що ми не знаємо відстань до кінцевої точки, доки не знайдемо маршрут, оскільки є дуже багато речей, які можуть стати на заваді. Алгоритм приймає рішення, беручи за основу значення f. Наступним етапом є вибір алгоритмом комірки з найменшим значенням f і просування до цієї комірки. Цей процес триває до того часу, поки алгоритм не досягне фінальної точки, а отже не виконає своє завдання.

Якщо розгорнути цей граф у вигляді дерева, то він матиме вигляд який зображено на рисунку 2.18:

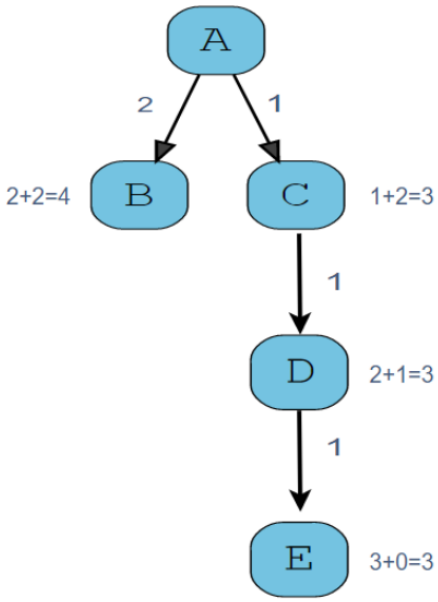


Рисунок 2.18 – Предсталення графа алгоритму A*

Отримані значення найменшої суми g і h розраховується кожного разу на кожному новому етапі алгоритму, отримуючи значення f. Мінімальний отриманий вузол f-значення вибирається для досягнення цільового стану.

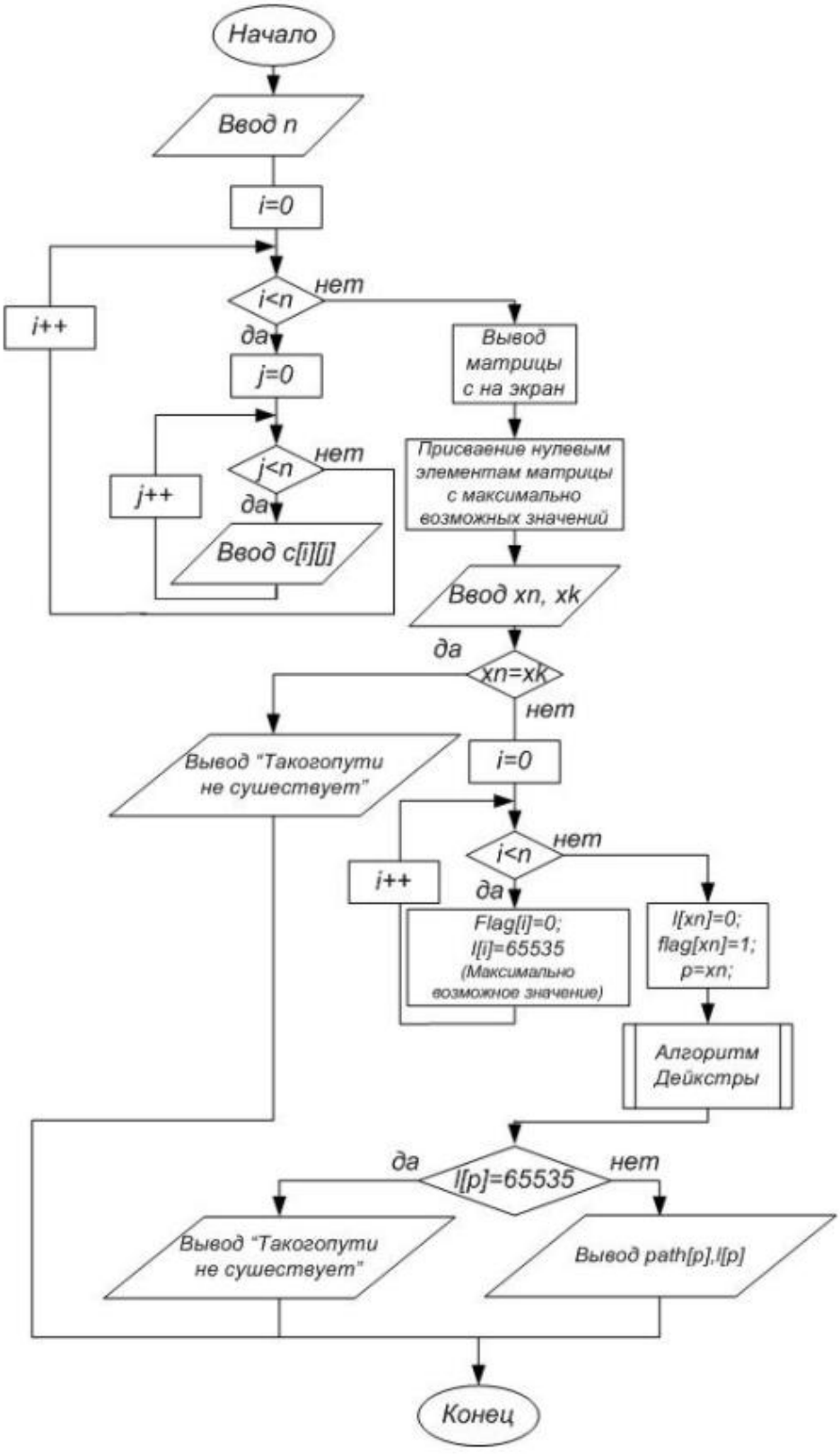


Рисунок 2.19 – Блок-схема алгоритму Дейкстри з пошуку шляху

2.6 Реалізація UML діаграм

UML діаграми варіантів використання моделюють поведінку, що може дізйснювати система та допомагають охопити вимоги системи.

Діаграми варіантів використання описують функції високого рівня та область застосування системи. Ці діаграми також ідентифікують взаємодію між системою та її акторами. Варіанти використання та актори на діаграмах варіантів використання описують, що робить система та як актори її використовують, але не те, як система працює всередині.

Діаграма варіантів використання ілюструє і визначає контекст і вимоги до всієї системи або важливих частин системи. Ви можете змоделювати складну систему за допомогою однієї діаграми варіантів використання або створити багато діаграм варіантів використання для моделювання компонентів системи. Зазвичай розробляється діаграма варіантів використання на ранніх етапах проекту та посилається на них протягом усього процесу розробки.

2.6.1 UML діаграма варіантів використання

В даному випадку система, що проектується представлена як сукупність сутностей, що пов’язані та взаємодіють із собою через прецеденти. В такому випадку суб’єктом виступає кожен, що взаємодіє із системою ззовні. Тобто кожна ситуація що призводить до використання, визначає послідовність подій, які система виконуватиме в діалозі з актором.

На зображенії системі можна представити чотири актори:

- заклад.
- сервіс;
- замовник;
- кур’єр;

Процес створення і обробки замовлення починається із етапу, створення клієнтом замовлення в системі доставки. Наступним етапом є передача створеного замовлення сервісу доставки, для подальшої обробки, а саме: запис

створеного замовлення, розподілення замовлення кур’єрові, та надання інформації про замовлення ресторану.

На наступному етапі, заклад отримує замовлення з сервісу доставки, виготовляє його та передає готове замовлення кур’єрові. Для кур’єра процес починається із відвідання закладу та отримання замовлення. Наступним кроком буде доставка замовлення до клієнта.

Діаграма представлена на рисунку 2.20.

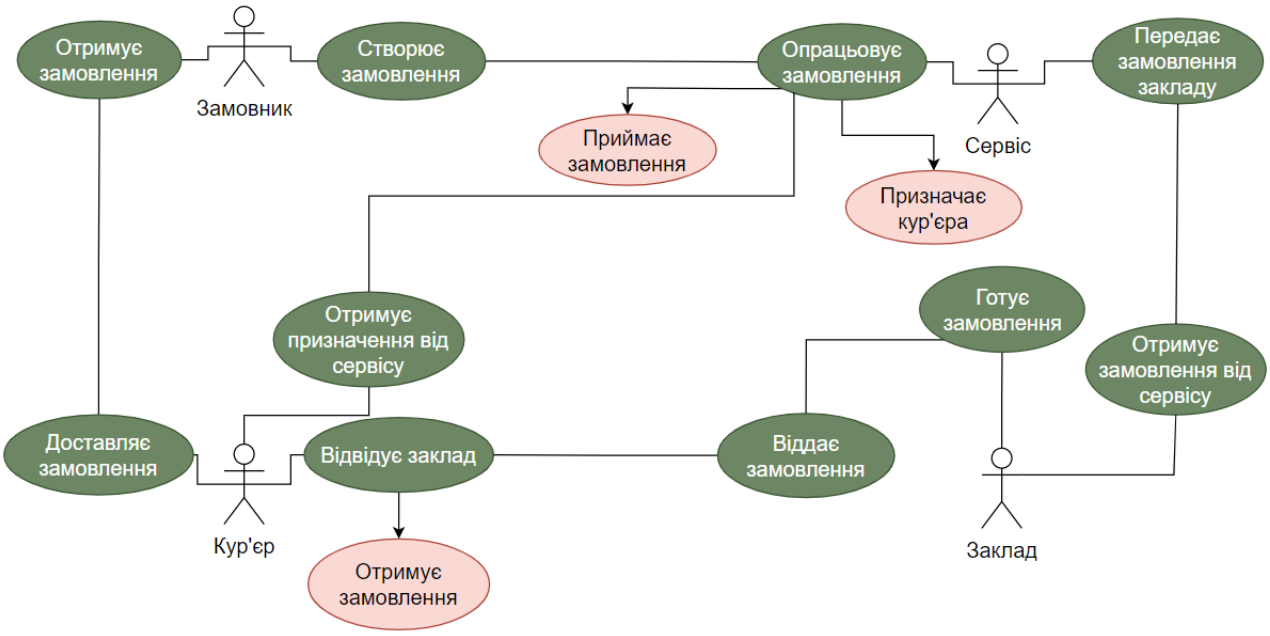


Рисунок 2.20 – Діаграма варіантів використання

Процес оптимізації маршруту замовлення починається з отримання системою даних з про координати адрес. Отримані дані перетворюються в потрібний формат для обробки системою. Перетворені дані обробляються одним із алгоритмів пошуку найкоротших шляхів. Наступний крок передача результату обробки на сервер. Побудова найкоротшого шляху з отриманих даних. Передача найкоротшого шляху системі для розподілу замовлень. Діаграма представлена на рисунку 2.21.

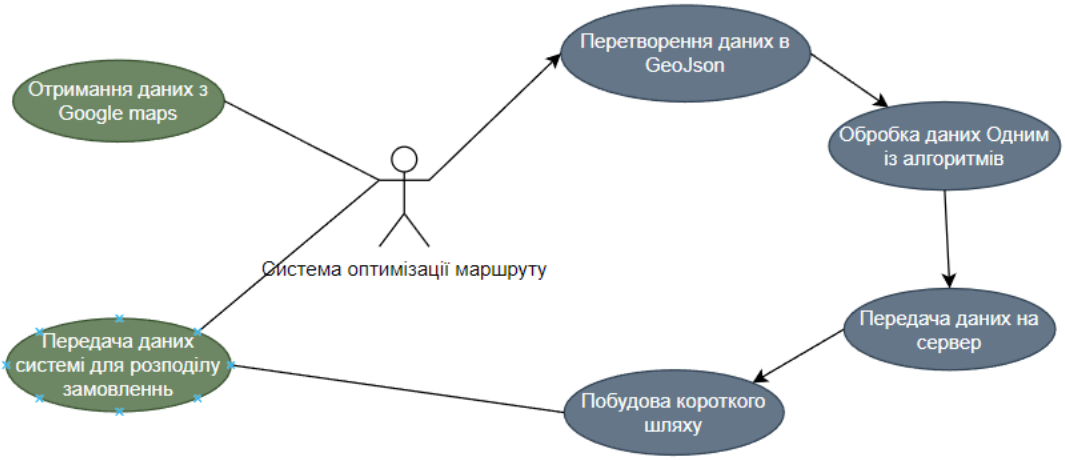


Рисунок 2.21 – UML діаграма варіантів використання

2.6.2 UML діаграма послідовності

Діаграми послідовності UML — це діаграми взаємодії, які детально описують, як виконуються операції. Вони фіксують взаємодію між об’єктами в контексті співпраці. Діаграми послідовності зосереджені на часі, і вони візуально показують порядок взаємодії за допомогою вертикальної осі діаграми для представлення часу, які повідомлення надсилаються та коли.

Основна мета діаграми послідовності — визначити послідовність подій, які призводять до певного бажаного результату. Основна увага приділяється не самим повідомленням, а більше порядку їх появи; незважаючи на це, більшість діаграм послідовності повідомлятимуть, які повідомлення надсилаються між об’єктами системи, а також порядок, у якому вони відбуваються. Діаграма передає цю інформацію вздовж горизонтального та вертикального вимірів: вертикальний вимір показує зверху вниз часову послідовність повідомлень-викликів, а горизонтальний вимір показує зліва направо екземпляри об’єктів, яким надсилаються повідомлення.

В даному випадку, на діаграмі зображено послідовність подій, що необхідно виконати користувачеві для створення замовлення на сервісі доставки. Тодів користувач виступає актором цієї діаграми, і для нього можна виділити кілька

основних дій таких як: створення замовлення в системі та отримання доставленого замовлення кур’єром. Сам процес стоврення замовлення в системі проходить за таким алгоритмом: клієнт обирає необхідні йому товари та додає до корзини, після додавання товарів до замовлення, клієнт заповнює форму з особистою інформацією.

Діаграма відображення створення замовлення представлена на рисунку 2.22.

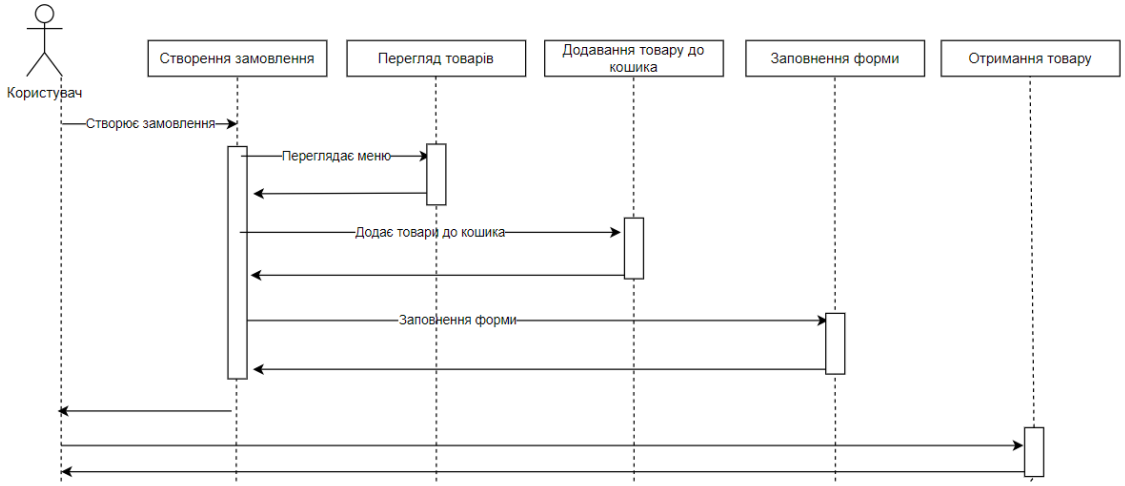


Рисунок 2.22 – Діаграма послідовності створення замовлення

Процес пошуку найкоротшого шляху системою відбувається в кілька етапів. В даному випадку користувачем виступає система з пошуку найкоротших шляхів. Першим етапом є отримання даних з про координати з відкритого доступу. Наступним – перетворення системою отриманих даних в зручний для обробки формат. Далі система використовує один із алгоритмів пошуку найкоротшого шляху, передає дані на сервер де будує найкоротший шлях. В кінці система передає побудований шлях для розподілення замовлень. Діаграма представлена на рисунку 2.23.

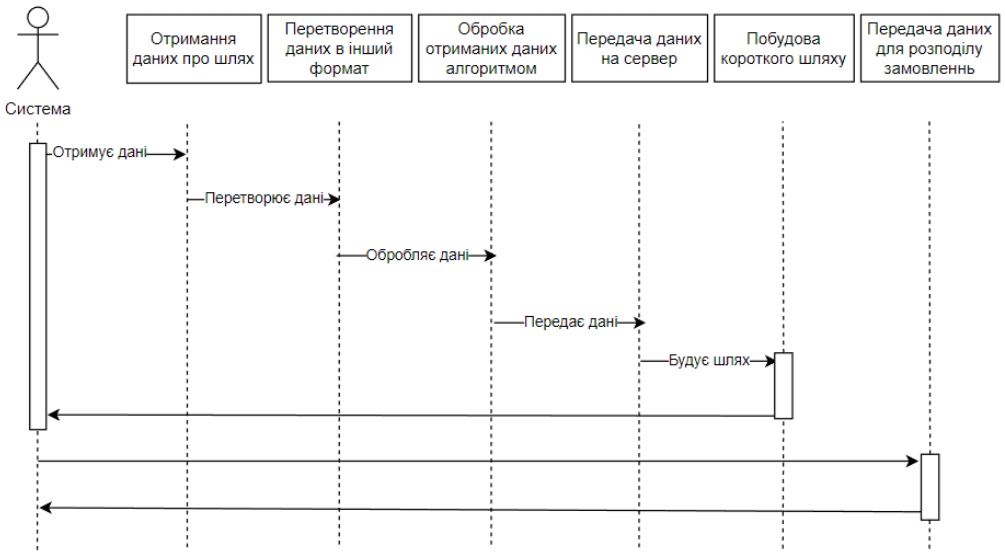


Рисунок 2.23 – UML діаграма послідовності

2.6.3 UML діаграма класів

Діаграма класів – це тип діаграми статичної структури, який описує структуру системи, показуючи класи системи, їхні атрибути, операції (або методи) і зв’язки між об’єктами.

Діаграма класів є основною для орієнтації об'єктно-орієнтованого моделювання. Така діаграма використовується, як для загального концептуального моделювання структури програми, так і для детального моделювання, переведення моделей у програмний код. Діаграми класів також можна використовувати для моделювання даних. Класи на діаграмі класів представляють як основні елементи, взаємодії в додатку, так і класи, які потрібно запрограмувати.

На запропонованій діаграмі покзані класи що описують реалізацію створення замовлення та показані зв’язки між ними:

– товар;
– форма.
– головна сторінка;
– корзина;

В діаграмі класів наведені атрибути, які охарактеризовують клас за якостями чи діями. Діаграма представлена на рисунку 2.24.

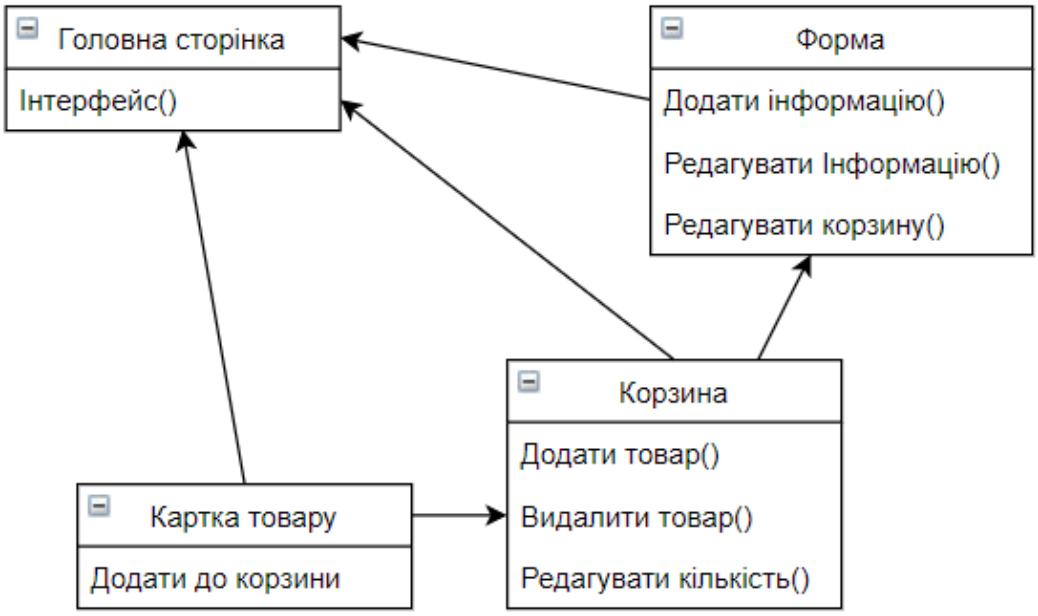


Рисунок 2.24 – Діаграма класів

Якщо описувати наведену діаграму класів, то можна сформувати такі визначення:

Корзина відповідає за можливість додавання та видалення товару із списку клієнта. Основною складовою корзини виступає картку замовлення, так як саме вона необхідна для роботи системи і це вона рендериться головною сторінкою.

Головна сторінка обробляє і відображає інтерфейс системи, і пов’язана з усіма такими класами як картка тофару, корзина та форма.

Форма приймає створений список замовлення клієнта, обробляє ці дані і має можливість використати одну і функцій, наприклад редагування чи додавання інформації.

Картка замовлень відповідає лише за додавання замовлення до корзини, саме з нею проходять основні обчислення.

2.7 Розробка проекту бази даних

Після створення замовлення всі дані що отримала система від введеня клієнтом чи обробки готових даних заносяться в базу даних системи, для подальшої ефективної роботи.

Діаграма взаємозв'язку розшифровується як – схема концептуальної моделі даних високого рівня. Модель допомагає систематично аналізувати вимоги до даних для створення добре спроектованої бази даних. Модель представляє сутності реального світу та зв’язки між ними. Створення такої моделі в СУБД вважається найкращою практикою перед впровадженням бази даних.

Набір сутностей — це сукупність сутностей подібних типів. Набір сутностей може містити сутності з атрибутами, що мають подібні значення. Наприклад, набір Сервіс може містити всіх Кур’єрів. Набори сутностей не обов’язково роз’єднуються.

Діаграма взаємозв'язку допомагає систематично аналізувати вимоги до даних для створення добре спроектованої бази даних. Отже, вважається найкращою практикою завершити моделювання взаємозв’язку перед впровадженням вашої бази даних.

Така діаграма будує логічну структуру бази даних використовуючи сутності та їх взаємозв’язки.

Сутність – за визначенням, це певна абстракція реального об’єкта, що описує особу чи об’єкт, яка виконує дії чи може взаємодіяти з іншими об’єктами системи. В розроблюваній системі можна виділити наступні сутності:

- сервіс;
- кур’єр;
- заклад.
- клієнт;

Для опису взаємодій можна використати модель «сутність-зв’язок» [50, 51], яка визначає значення даних в контексті взаємозв’язку з іншими даними. Характеристика зв’язків наведена в таблиці 2.2.

Таблиця 2.2 – Характеристика зв’язків

Сутність 1	Сутність 2	Зв’язок	Ім’я зв’язку
Клієнт	Сервіс доставки	N : 1	Робить замовлення
Сервіс доставки	Кур’єр	1 : N	Взаємодіє
Кур’єр	Заклад	1: 1	Взаємодіє
Клієнт	Кур’єр	1: 1	Отримує замовлення

Опираючись на таблицю створено ER-модель [52, 53] предметної області, яка наведена в рисунку 2.25.

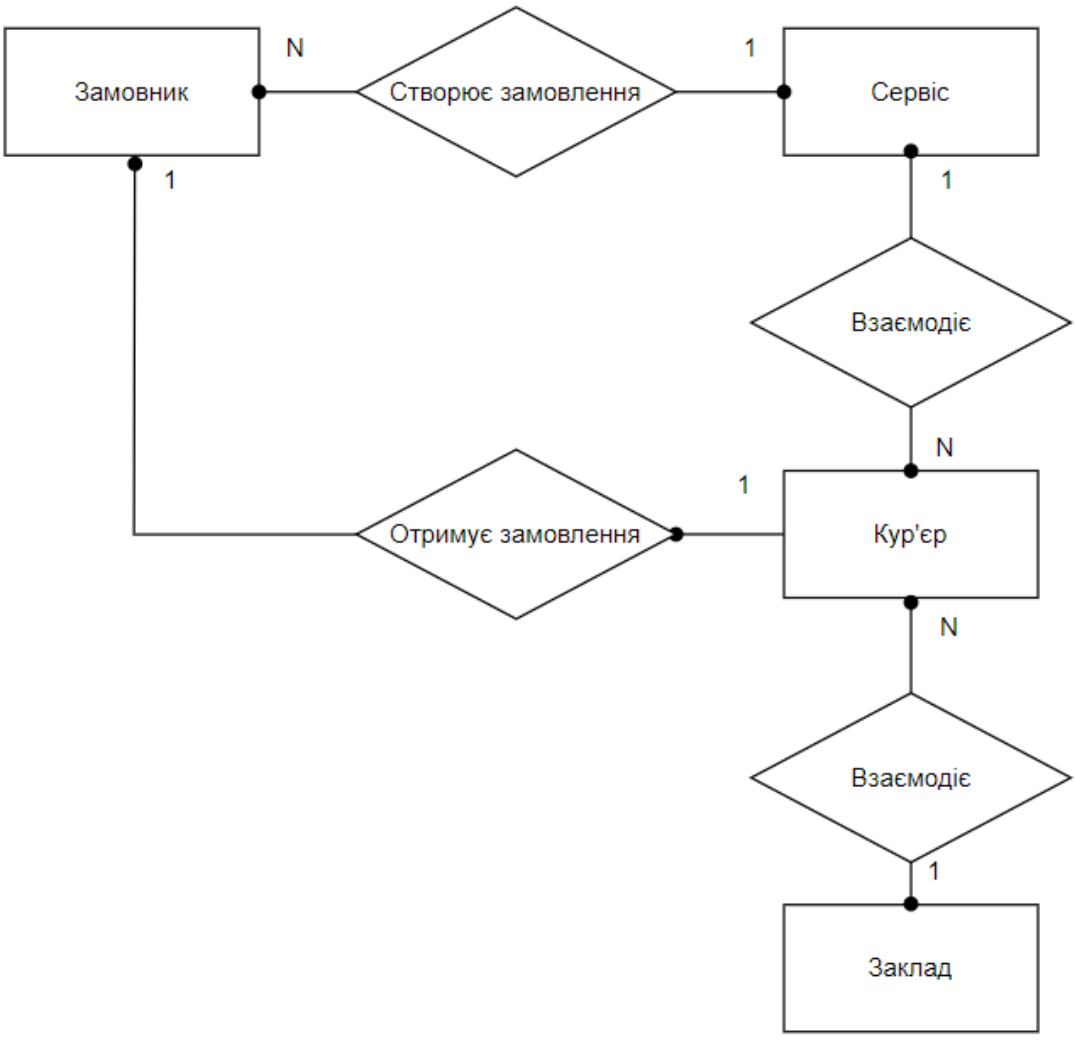


Рисунок 2.25 – ER-модель предметної області

Для розробки ER-моделі в першу чергу необхідно виділити сутності. Сутності можна позначити за допомогою таблиці, в якій міститься унікальне ім’я.

Зв’язок між сутностями позначається у вигляді лінії з виділеними ступенями залежності на вході і виході.

Схема бази даних наведена в рисунку 2.26.

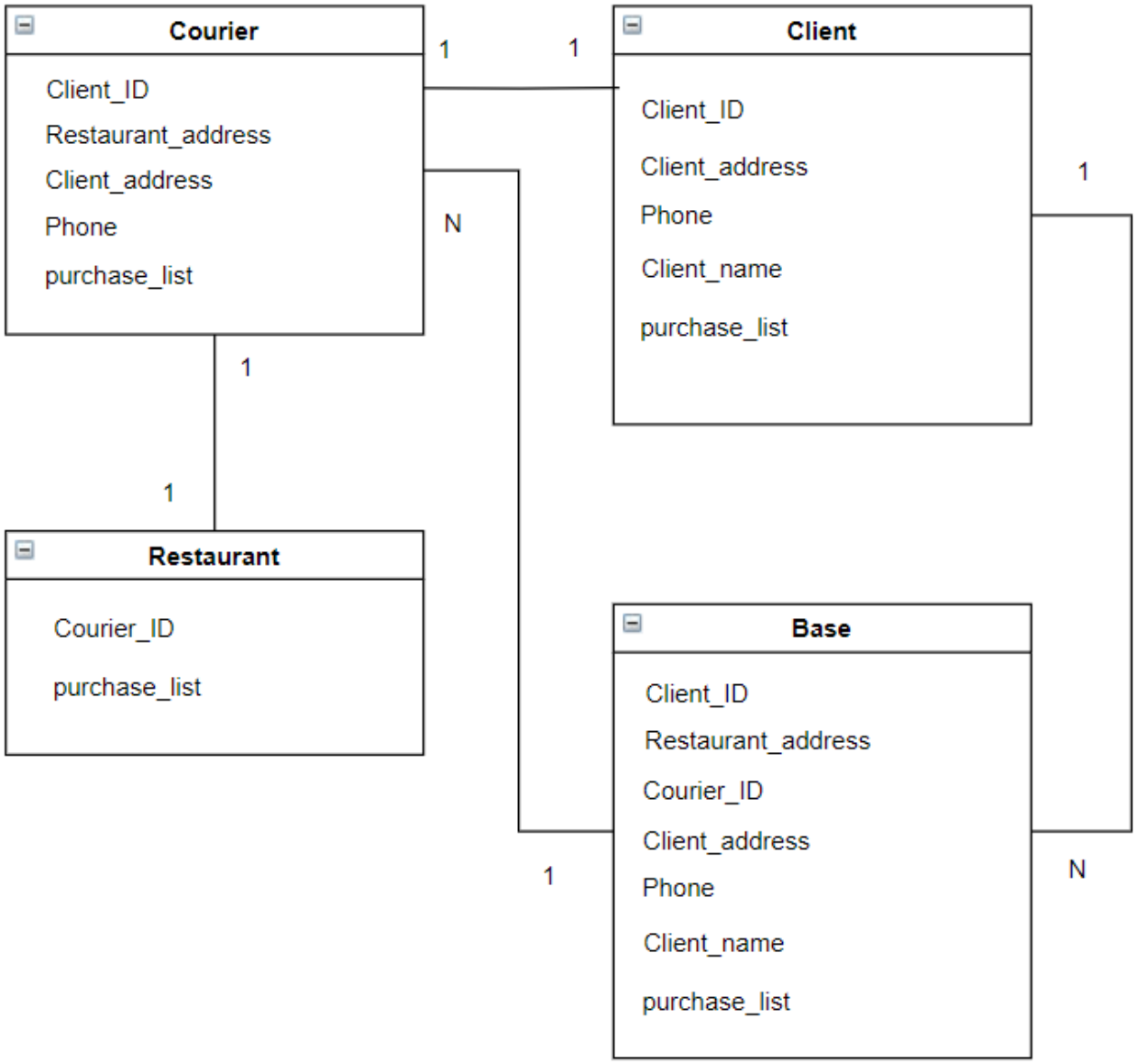


Рисунок 2.26 – Схема бази даних

2.8 Висновки

В даному розділі було проаналізовано методи проектування й розробки, що допоможуть при створенні підсистеми автоматизованого управління доставкою.

Розглянуто, методи написання веб-додатків зі збереженням оптимальної швидкості. Також було розглянуто можливість написання сервісу з допомогою використання JavaScript. Для кращої постановки задачі було розроблено UML таблиці, що допомогли краще описати дії які проходять на кожному етапі, починаючи від створення клієнтом замовлення, закінчуючи доставкою замовлення кур’єром. Було розроблено проект бази даних, та визначення основних сутностей із зв’язками між ними. Також було проаналізовано методи отримання та обробки динамічних даних для подальшої побудови найкоротших шляхів.

3. РЕАЛІЗАЦІЯ СИСТЕМИ УПРАВЛІННЯ ДОСТАВКОЮ НА ПРАКТИЦІ

3.1 Вибір технологій для реалізації сервісу.

Для реалізації функціоналу сервісу в першу чергу необхідно виділити базові завдання сервісу. Одними з таких завдань є: можливість вибору товару клієнтом, Можливість додавання вибрних товарі до корзини замовлення, створення та редагування з отриманих товарів форми замовлення, за умови якщо клієнт продовжить виконання роботи.

Для розробки частини інтерфейсу, що відповідає за можливість вибору товару клієнтом варто розглянути кілька моделей зберігання карток товарів.

Зберігання даних про товар у форматі JSON[54]. Таке зберігання вимагає записування інформації про кожен товар у таблицю з ключами але дозволяє зручно організовувати роботу з цими товарами. Приклад таких ключів зображено на рисунку 3.1.

```
[
    {
        "id": "1",
        "place" : "McDonalds",
        "name": "French fries",
        "price": "40",
        "img": "img/goods/fries-2.jpg"
    },
    {
        "id": "2",
        "place" : "KFC",
        "name": "Burger",
        "price": "90",
        "img": "img/goods/kfc-burger.jpg"
    }
]
```

Рисунок 3.1 – збереження інформації в форматі JSON

Для кожного товару виблано п’ять ключів, що дозволяє записати всю необхідну інформацію про кількість, місце замовлення, назву, ціну, та зображення зовнішнього вигляду товару.

- Img – Для збереження зображення, що відповідає зовнішньому вигляду товару.
- Name – Зберігає назву товару.
- Place – Створенний для збереження інформації про місце та назву закладу.
- Id – Створений для призначення унікальних номерів кожному товарові, для полегшення пошуку в таблиці.
- Price – Створений для запису ціни, яка відповідає товару.

Ще однією можливою системою реалізації товарів є створення таблиці в одній із баз даних, таких як MySQL [55] і MongoDB [56]. Основна різниця між запропонованими базами даних полягає в структурі зберігання даних. MySQL – реляційна система і має значну перевагу в швидкості роботи при великих об'ємах товарів у базі, а MongoDB – документо-орієнтована система керування і має набагато більшу ефективність коли необхідно створити менше об'єктів але з більшою кількість атрибутів, така система є дуже ефективною але ефективність зменшується пропорційно при великій кількості об'єктів.

SQL – це безкоштовна база даних із відкритим кодом, якою ви можете користуватися. Ви можете ефективно керувати своїми даними за допомогою цієї програми, оскільки це дуже потужне, надійне та стабільне рішення.

Серед переваг цієї бази даних є:

Захист даних – Це одна з найпопулярніших систем керування базами даних з точки зору безпеки та надійності. Він використовується в багатьох веб-додатках, включаючи Twitter, Facebook, Joomla, Drupal і WordPress. Для обробки транзакцій підтримка даних і безпека можуть бути корисними для вашої організації, особливо якщо ви працюєте в галузі електронної комерції.

Масштабованість на вимогу – Неймовірна масштабованість цієї платформи може допомогти з керуванням вбудованими програмами. Це вірно, навіть якщо є велике сховище з величезними обсягами даних. Насправді, одна з головних характеристик системи — адаптація за вимогою.

Перевага цієї платформи з відкритим вихідним кодом полягає в тому, що вона дозволяє повністю налаштувати, що дуже корисно, якщо у вас є магазин електронної комерції.

Висока ефективність – My SQL містить унікальний механізм зберігання, який полегшує адміністрування вашої системи. Крім того, ви можете налаштувати сервер бази даних MySQL для досягнення максимальної продуктивності. Іншими словами, навіть якщо у вас є високошвидкісна система обробки даних або веб-сайт, який отримує мільйони звернень щодня, ви можете скористатися цією технологією.

Безвідмовна робота – Система MySQL унікальна тим, що забезпечує постійну безвідмовну роботу. Окрім цього, він надає безліч опцій, таких як параметри реплікації підлеглий/головний і унікальні кластерні сервери.

Відмінна підтримка транзакцій – MySQL займає перше місце в списку доступних сьогодні рішень для швидких транзакційних баз даних. Він має кілька функцій, наприклад послідовну й автоматичну підтримку транзакцій. Крім того, це один із найкращих варіантів повної цілісності даних.

Відмінний контроль робочого процесу – Вам не потрібно витрачати багато часу на налаштування цієї програми, оскільки завантаження та встановлення займають менше 30 хвилин. Ви можете скористатися перевагами цього фантастичного рішення незалежно від вашої платформи. Його можна використовувати для автоматизації широкого діапазону процесів, включаючи керування базами даних, проектування даних і збільшення простору.

Нижча загальна вартість володіння – Ви можете заощадити багато грошей, якщо конвертуєте існуючі програми баз даних на цю платформу. Простота управління та надійність можуть допомогти у вирішенні проблем. У результаті у вас не виникне жодних простоїв або проблем із продуктивністю.

SQL – мова програмування, що створена для взаємодії клієнта з базами даних, які застосовуютьсяться для створення вікон запитів, зміною та управлінням реляційними БД, що представлені в форматі таблиці, створення схеми бази даних, система контролю за доступом до бази даних. Сам по собі SQL

не є ні системою управління базами даних, ні окремим програмним продуктом, крім того, вміщує функції для визначення зміни, перевірки і захисту даних.

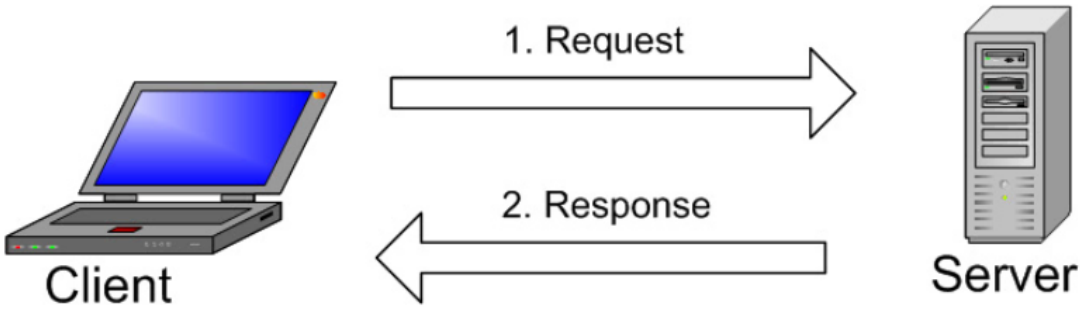


Рис 3.1 – Зображення клієнт-серверної структури запиту

Зображення пояснює основну структуру клієнт-серверної структури. Один або кілька пристроїв (клієнтів) підключаються до сервера через певну мережу. Кожен клієнт може зробити запит із графічного інтерфейсу користувача (GUI) на своїх екранах, і сервер видасть бажаний результат, якщо обидві сторони розуміють інструкцію. Не заглиблюючись у технічні аспекти, основні процеси, що відбуваються в середовищі MySQL, однакові, а саме:

–MySQL створює базу даних для зберігання та обробки даних, визначаючи зв’язок кожної таблиці.

–Клієнти можуть робити запити, вводячи певні оператори SQL на MySQL.

–Серверна програма відповість запитуваною інформацією, і вона з’явиться на стороні клієнтів.

Так як проект системи управління доставкою передбачає велику кількість вхідних даних, з нафеликою можливістю до розширення, варто для розробки такої системи обрати реляційну базу даних. Перед початком роботи із SQL необхідно створити та встановити сервер на одному із сервісів для хостингу веб ресурсів. Після виконання такого встановлення необхідно відвідати сторіку керування перейшовши за посиланням до “phpMyAdmit”. Це додаток розроблений для управління і адміністрування будь яких СУБД, такий сервіс дозволяє через звичайне вікно браузера переглядати зміст таблиць, запускати скрипти та запити,

здійснювати управління сервісом. Отже для початку необхідно використати веб-інтерфейс для керування сервером та створити базу даних. Вище згаданий сервіс для управління маж можливість імпорту даних, але даному випадку, такий підхід не підійде для виконання завдання. Після створення бази даних, з'явиться можливість для створення таблиць та профілів користувачів системою в середині бази.

Сервіс phpMyAdmin має велику популярність серед веб-розробників, через те що дозволяє швидко і зручно управляти будь-яким СУБД, без безпосереднього введення SQL команд. Саме він буде використовуватися для роботи з базою даних в сайті, що створюється.

	#	Ім'я	Тип	Зіставлення	Атрибути	Нуль	За замовчуванням	Коментарі	Додатково	Дія
☐	1	id	int(20)			Ні	*Немає*			Змінити Знищити Більше
☐	2	place	varchar(128)	utf8mb4_general_ci		Ні	*Немає*			Змінити Знищити Більше
☐	3	name	varchar(128)	utf8mb4_general_ci		Ні	*Немає*			Змінити Знищити Більше
☐	4	price	int(64)			Ні	*Немає*			Змінити Знищити Більше
☐	5	img	varchar(128)	utf8mb4_general_ci		Ні	*Немає*			Змінити Знищити Більше

Рисунок 3.2 – Створення таблиці базі даних MySQL

Розроблена таблиця, матиме ключі-значення, про які було згадано на етапі отримання значень з файлу формату JSON, очікувана матиме такий вигляд:

id	place	name	price	img
1	mcdonalds	french fries	80	img.jpg
1	mcdonalds	french fries	80	img.jpg
2	kfc	burger	90	img1.jpg

Рисунок 3.3 – Вигляд даних в базі даних MySQL

Для запуску сервера використовувався сервер Apache та база даних MySQ.

Apache – це безкоштовний веб-сервер із відкритим вихідним кодом, який передає веб-вміст через Інтернет. Його зазвичай називають Apache, і після розробки він швидко став найпопулярнішим HTTP-клієнтом в Інтернеті.

Серед переваг серверу Apache є його можливість обробляти великі обсяги трафіку з мінімальною конфігурацією та використаним часом. Він легко масштабується, а завдяки його модульній функціональності в основі ви можете налаштувати Apache так, щоб він робив те, що ви хочете, як ви хочете. Ви також можете видалити непотрібні модулі, щоб зробити Apache більш легким і ефективним.

Деякі з найпопулярніших модулів, які можна додати, це SSL, підтримка серверного програмування (PHP) і конфігурації балансування навантаження для обробки великих обсягів трафіку. Серевер Apache також можна розгорнути в більшості існуючих операційних систем, таких як: Linux, MacOS і Windows.

Apache функціонує як спосіб обміну даними через мережі від клієнта до сервера за допомогою протоколу TCP/IP. Apache можна використовувати для багатьох протоколів, але найпоширенішим є HTTP/S. HTTP/S або Hyper Text Transfer Protocol (S означає Secure) є одним із основних протоколів в Інтернеті та єдиним протоколом, за яким Apache найбільше відомий.

HTTP/S протоколи зазвичай використовуються для визначення того, як повідомлення форматуються та передаються через Інтернет, з інструкціями для браузерів і серверів, як відповідати на різні запити та команди. Захищений протокол передавання гіпертексту зазвичай здійснюється через порт 443, а незахищений протокол – через порт 80.

Сервер Apache налаштовується за допомогою конфігураційних файлів, у яких модулі використовуються для керування його поведінкою. За замовчуванням Apache прослуховує IP-адреси, налаштовані у файлах конфігурації, які запитуються.

За допомогою директиви прослуховування Apache може приймати та направляти певний трафік на певні порти та домени на основі конкретних запитів комбінації адреса-порт. За замовчуванням прослуховування працює на незахищеному порту 80, але Apache може бути прив’язаний до різних портів для різних доменів, що дозволяє розміщувати багато різних веб-сайтів і доменів на одному сервері.

Коли повідомлення досягає місця призначення або одержувача, воно надсилає сповіщення або повідомлення ACK, фактично надаючи підтвердження початковому відправнику, що його дані успішно надійшли. Якщо сталася помилка під час отримання даних або деякі пакети були втрачені під час передавання, хост або клієнт надсилає повідомлення Not Acknowledged або NAK, щоб повідомити відправника про те, що дані потрібно передати повторно.

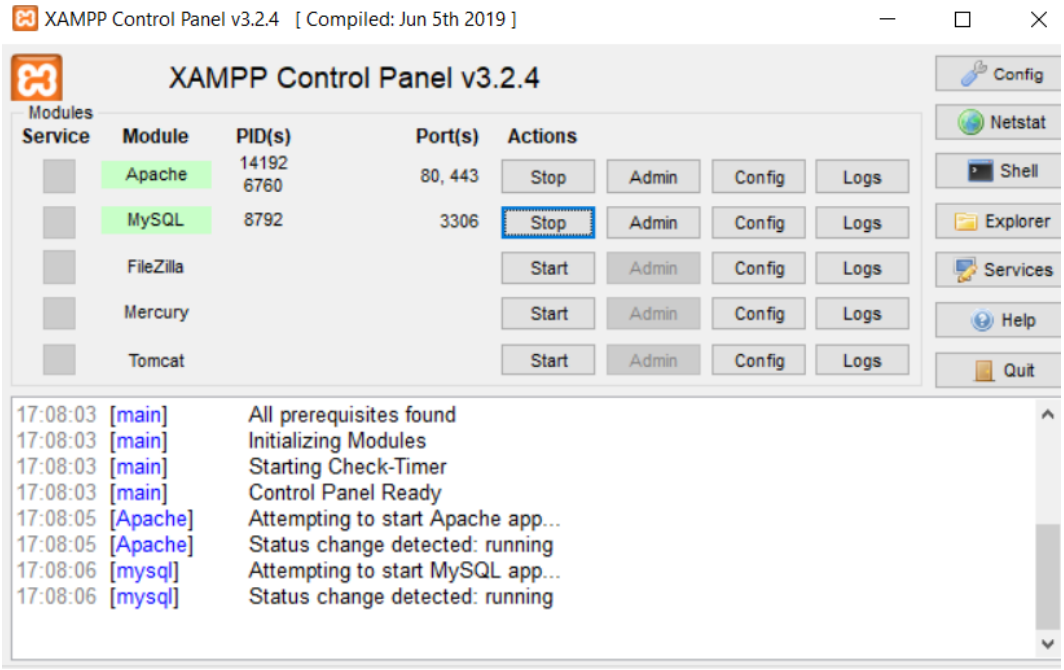


Рисунок 3.4 – Зображення контрольної панелі для управління сервером

Для формування карток товарів, дані отримуються з збережених реляційних бач даних, передається підсистемі управління картками товарів, і в кінці рендериться картка товару на головній сторінці, чи сторінці “Меню”

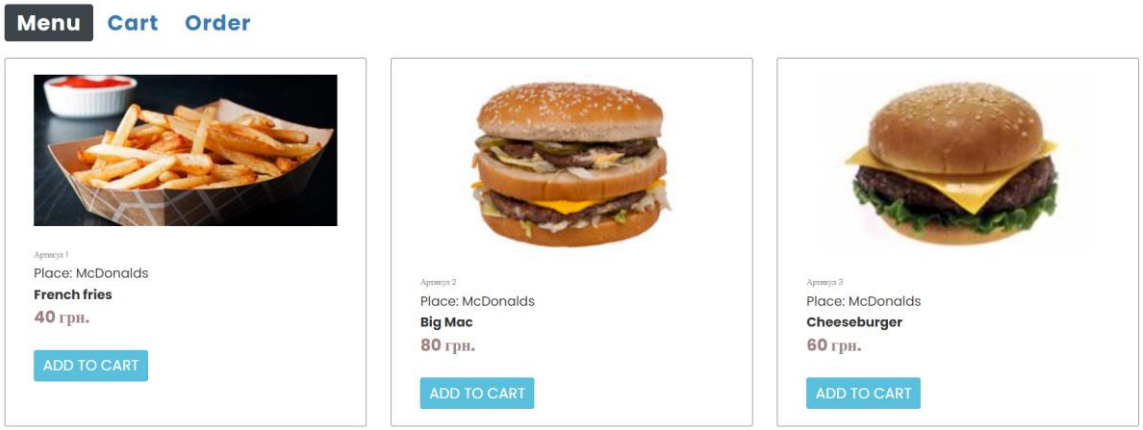


Рисунок 3.5 – Вигляд товарів отриманих з бази даних

Розроблена система для корзини спрацьовує наступним чином. Клієнт переглядає картки товарів представлених на веб-сторінці та обирає які йому набільше підходять, натискаючи кнопку, “Додати до корзини”, Сервіс відстежує які товари обрано клієнтом, тому коли клієнт переходить до фомування замовлення, усі вибрані ним товари, відображаються у вікні корзини. Для полегшення роботи серверу, формування і збереження проходить на стороні клієнта, використовуючи принцип одно сторінкового додатку. Отримана інформація локально зберігається на пристрої клієнта, і таким чином захищена від очищення, коли клієнт перейде на іншу сторінку, не зберігши свій вибір. Такий підхід дозволить меншу кількість разів звертатись до бази даних, що збільшить швидкість роботи як і системи в цілому на стороні сервера, так і додатку на стороні клієнта. Для забезпечення безпеки системи, кожне створене і відправлене замовлення порівнюється із замовленням, що зберігається в локальній пам’яті клієнта, що дає змогу ідентифікувати помилку, і не виконувати замовлення в разі загрози системі.

На рисунку рисунку 3.6. зображено вигляд замовлення що знаходиться в корзині.

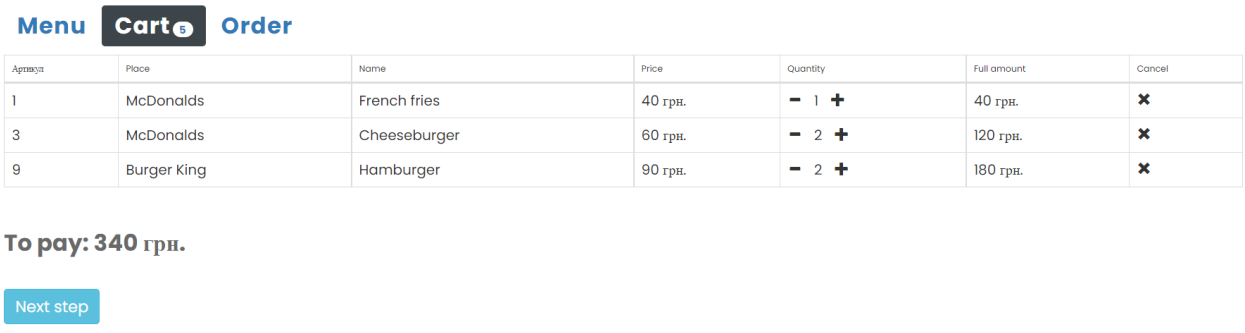


Рисунок 3.6 – Вигляд замовлення що знаходиться в корзині

Після отримання корзиною інформації про обрані товари і її перевірки, процес створення замовлення переходить до наступного етапу. Етап заповнення форми замовлення працює за наступним принципом. Усі необхідні дані отримуються з бази даних, що відповідає картці товару. Для відображення ціни і переліку товарів форма отримує дані з корзини. Після передачі ціна вставляється

у відповідне місце у формі замовлення. Наступним етапом є введення особистих даних клієнтом. Після цього замовлення має усю необхідну інформацію, щоб бути створеним, Далі сформоване замовлення переходить до бази даних для перевірки та опрацювання. На рисунку 3.7. наведено вигляд форми замовлення

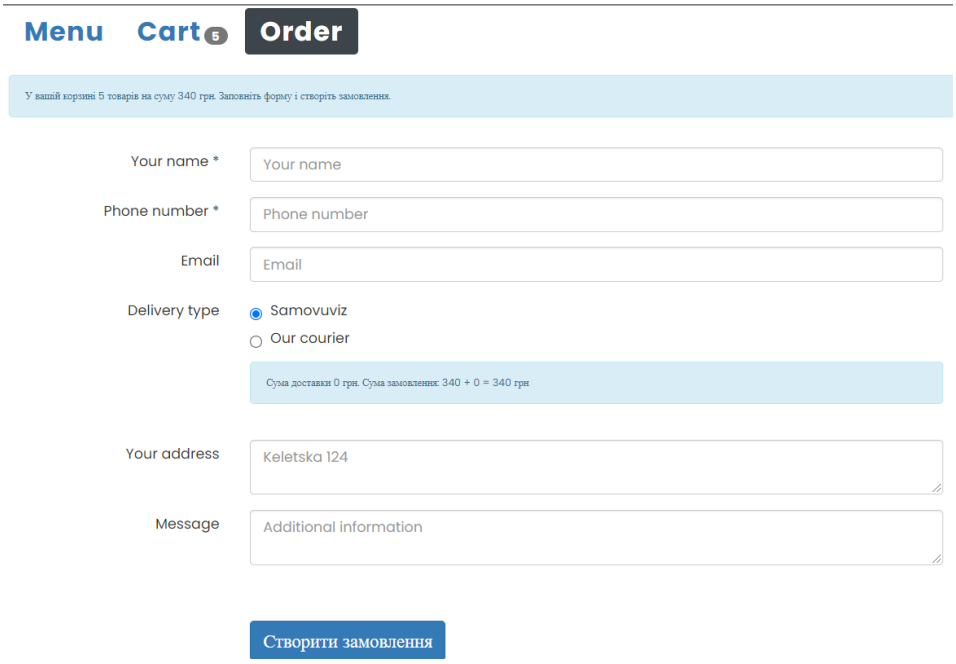


Рисунок 3.7 – Вигляд форми-замовлення на стороні клієнта

Наступним модулем системи є управління доставкою, для цього необхідно реалізувати частину сервісу, що відповідає за інформування то контроль кур’єрів. Першим етапом роботи кур’єра є реєстрація та вхід до системи управління доставкою. За умови якщо це перший вхід, кур’єрові потрібно заповнити невелику стандартну форму з особистою інформацією, такою як ПІБ, електронна пошта та номер телефону. Форма входу зображена на рисунку 3.8

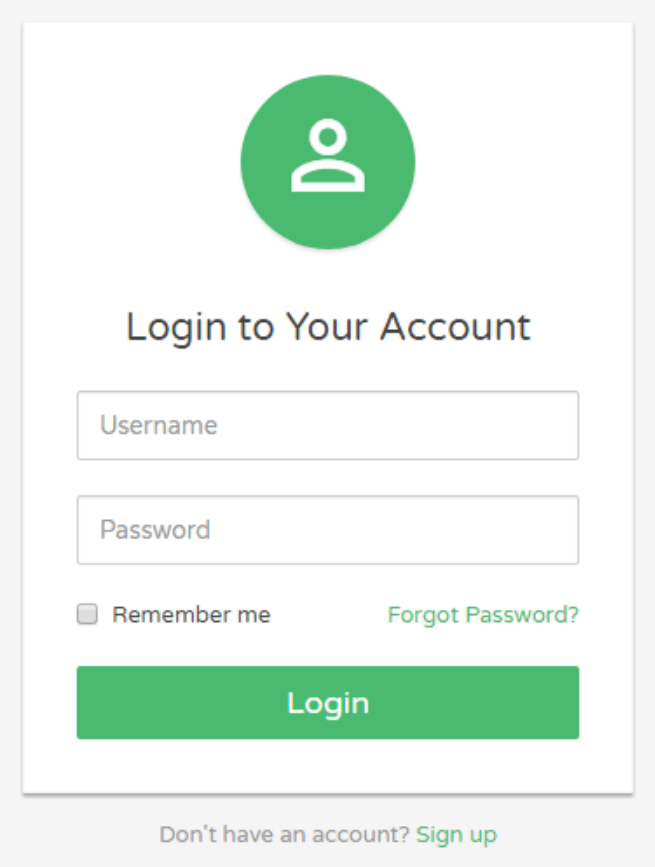


Рисунок 3.8 – Форма входу до системи для кур’єра

Після викнання входу до системи кур’єр отримує статус «активний», та має можливість отримувати замовлення. Замовлення для кур’єра передаються наступним чином, поля які підтягуються з бази даних, формуються в сповіщення, чи push повідомлення. Поля отримуються із форми замовлення яку заповнив клієнт і мають такий вигляд:

Після виконання входу кур’єр має можливість отримувати замовлення. Із замовленнь з форми, яку заповнює клієнт формується список із такими полями:

– Name – Поле, що створене для отримання даних про ім’я замовника.

– Phone – Поле, що створене для отримання даних про телефон замовника.

– Place – Поле, що створене для отримання даних про назву закладу який необхідно відвідати кур’єру.

– Title – Поле, що створене для отримання даних про назви необхідних товарів.

– Price – Поле визначення ціни, яка відповідає товару.

– Road – Повідомлення, яке містить в собі рекомендації що до найкоротшого шляху, та дії для того щоб слідувати сим маршрутом.

–Id – Поле, що створене для отримання даних про унікальний номер товару, для того що бкаще розрізняти замовлення.

Вигляд полів бази даних для кур’єра представлений на рисунку 3.9.

```
[
    {
        "id": "1",
        "place" : "McDonalds",
        "name": "French fries",
        "price": "40",
        "img": "img/goods/fries-2.jpg"6,
        "personName": "1111",
        "personPhone": "123451",
        "road": "",
    },
    {
        "id": "2",
        "place" : "McDonalds",
        "name": "Big Mac",
        "price": "80",
        "img": "img/goods/big-mac.jpg",
        "personName": "11234",
        "personPhone": "874574",
        "road": "",
    }
]
```


Рисунок 3.9 – Вигляд полів бази даних для кур’єра

Замовлення розподіляються між кур’єрами за допомогою сканування карти, і вибору найближчих від ресторану кур’єрів для передачу замовлення. Наступний етап це аналіз можливих маршрутів для кур’єра та пошуку найкоротшого шляху, відносно положення кур’єра на карті. Після цього система рекомендує повідомлення круєрові про оптимальний час і опис з порядком руху, щоб дотримуватись маршруту. Отрмане завдання завершується після того як клієнт отримує своє замовлення

3.2 Обгрунтування вибору системи керування вмістом для сервісу

Для реалізації управлення вмістом додатку необхідно використати одну із систем керування вмістом. Такі системи зазвичай назвають CMS[57], і вони

створені для адміністрування веб-сторінрк, організації та оптимізації роботи додатку.

CMS – програма, що дозволяє багатьом учасникам створювати, редагувати та публікувати певний контент. Вміст у CMS зазвичай зберігається в базі даних і відображається на рівні презентації на основі набору шаблонів, як веб-сайт.

Основні функції таких систем це:

–Створення вмісту – дозволяє користувачам легко створювати та форматувати вміст

–Зберігання вмісту – зберігає вміст в одному місці, узгоджено

–Оптимізація робочих процесів – призначає дозволи для керування вмістом на основі ролей, таких як автори, редактори та адміністратори

–Публікація, систематизація та розміщення вмісту в реальному часі

Однією з основних переваг CMS є її можливість працювати з кількома комп'ютерами одночасно. Кілька користувачів можуть увійти в систему та робити внески, планувати або керувати вмістом для публікації. Оскільки інтерфейс зазвичай базується на браузері, до CMS може отримати доступ будь-яка кількість користувачів з будь-якого місця.

Друга головна перевага CMS полягає в тому, що вона дозволяє нетехнічним людям, які не знають мов програмування, легко створювати та керувати власним веб-вмістом. Редактори типової платформи керування вмістом із функцією перетягування дозволяють користувачам вводити текст і завантажувати зображення без знання HTML чи CSS.

Коли компанія використовує CMS для публікації своїх веб-сторінок, вона зменшує свою залежність від інтерфейсних розробників для внесення змін до веб-сайту, що робить публікацію нових веб-сторінок швидшою та легшою.

Для використання системи керування вмістом в вистемі управління доставвкою, було проаналізовано ринок такх доступних систем і обрано такі системи для порівняння як: Wordpress [58], Joomla [59] та OpenCart

WordPress — це платформа з відкритим кодом. Що означає, що будь-хто може змінювати вихідний код і розповсюджувати програмне забезпечення. Він призначений для того, щоб люди могли вносити свої ідеї в покращення платформи.

Для роботи із системою WordPress не потрібні особливі знання з комп'ютерної інженерії. Цю систему створено для полегшення роботи людей, що в першу чергу не дотичні до розробки, і можуть отримати інструменти для реалізації, своїх бодатків самостійно використовуючи графічний інтерфейс.

Інтерфейс системи інтуїтивно зрозумілий і дружній до користувача. Для того щоб додати дизайн для додатка необхідно, або створити тему додатка самостійно з використанням HTML та CSS, або обрати тему із запропонованих сервісом.

Сервіс має велику спроможність для розширення за допомогою плагінів, Якщо розроблюваний додаток потребує інтеграцію форм чи будь-яких інших сервісів, є можливість просто обрати плагін, який виконає всю роботу, єдине що залишиться це налаштувати проект за допомогою графічного інтерфейсу.

Сервіс також має багато можливостей для покращення SEO оптимізації чи збільшення оцінки індексації веб-додатку. Для покращення SEO оптимізації варто використати один із плагінів SEO, таких як Yoast SEO. Також є можливість збільшити рейтинг оптимізації за допомогою внутрішніх посилань і шаблонів метаописів. Після виконання цих процесів, оцінка індексації від Google збільшиться на певний відсоток, що призведе до того, що веб-додаток буде показуватись в пошуковій системі на кращих позиціях.

WordPress не обмежує жодних основних типів медіа. Отже, за допомогою цієї системи є можливість використовувати будь-яку комбінацію тексту, зображень і відео на створеному веб-додатку. Єдиним винятком є обмеження використання векторних зображень формату SVG, через їхню меншу захищеність, Так як це формат зображення що передається кодом, він більш вразливий до використання в хакерських атаках. Для запобігання цьому, зображення формату svg імпортують до проекту не звичайними зображеннями як зазвичай, а кодом.

З огляд на захист безпеки системи WordPress є однією з найбезпечніших і надійних платформ CMS. Платформа пропонує кілька функцій для захисту вашого веб-сайту, як-от вихід із системи неактивних користувачів і додавання двофакторної автентифікації. В ситуації коли за система має постійні оновлення плагінів і паролів, зламати її буде важким завданням.

Серед основних переваг системи управління вмісто WordPress можна виділити:

–Легкість у використанні.

–Можливість працювати із системою із будь-якого місця.

–Велика кількість розширень, що дозволять покращити управління системою.

–Можливість редагування сторінок із графічного інтерфейсу.

–Можливість створення шаблонів сторінок або використання шаблонів, що пропонує система.

–Можливість краще провести SEO оптимізацію ніж в аналогічних сервісах.

Зовнішній вигляд інтерфейсу системи керування вмістом WordPress зображений на рисунку 3.10.

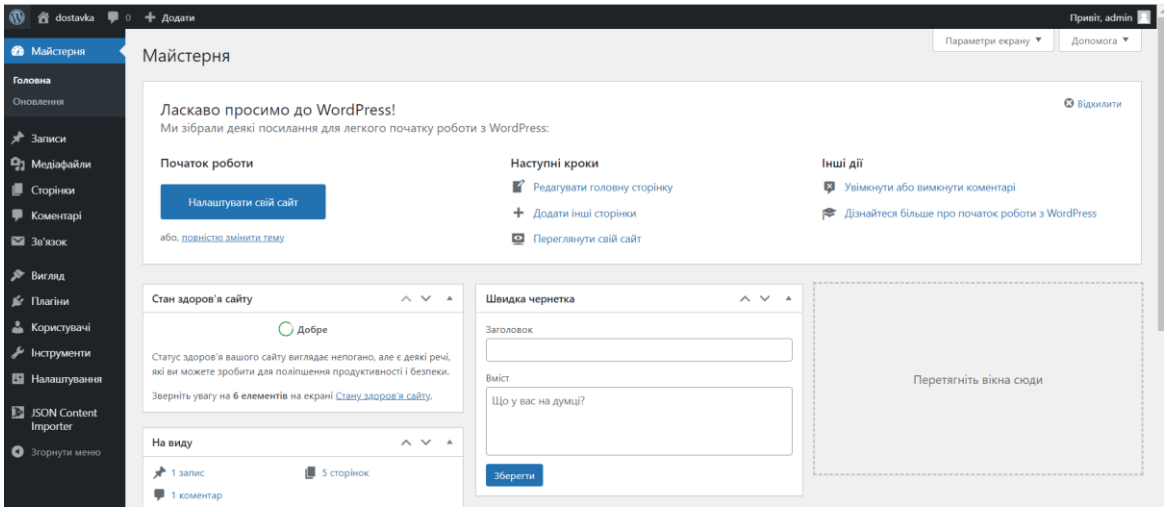


Рисунок 3.10 – Інтерфейс CMS Wordpress

Ще одна із переваг у використанні даної системи управління контентом це те що WordPress абсолютно безкоштовний. Використання програмного

забезпечення нічого не коштуватиме. Але варто зайважити, що більша частина плагінів, що значно полегшують роботу з даною системою платні, і для їх використання необхідно буде витратити первну суму. Ще однією витратою буде оплата послуг хостинг провайдера, для розміщення серверу, бази даних і власне додатка, в мережі, для того щоб з’явилась можливість його використання.

Для роботи з текстом, є можливість зручного оправління створенняв чи редагуванням за допомогою візуальних редакторів, наприклад Wysiwyg або Guttenberg. На рисунку 3.11 зображено зовнішній вигляд редактора тексту.

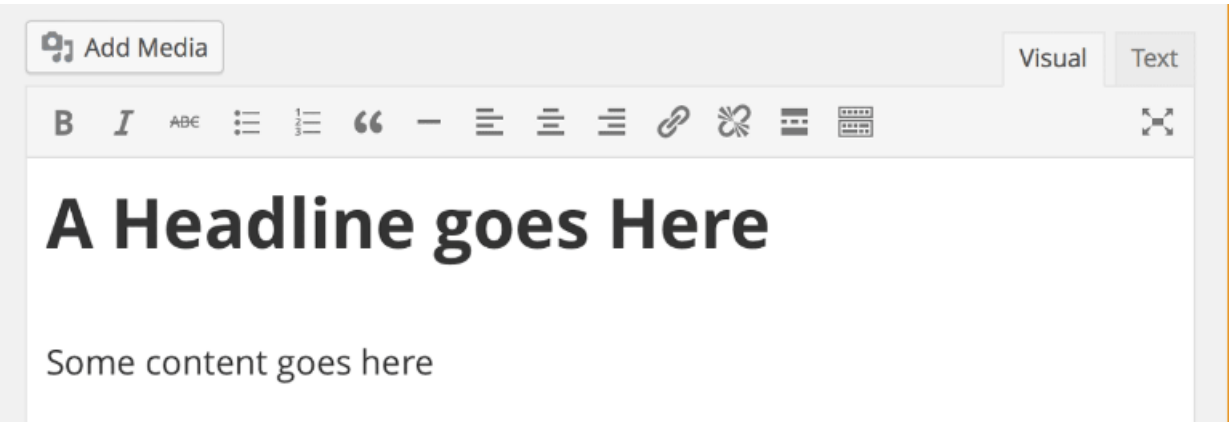


Рисунок 3.11 – Зображення візуального редактору Wysiwig

Робота з редактором Wysiwig приносить дуже велику економію в часі, так як не треба створювати кожен параграф і заголовок окреми, а можна використи поле “текст”.

Система управління контентом Joomla — це потужний вибір для створення веб-сайту незалежно від його розміру. Основними його характеристиками є:

Багатомовність – сервіс має можлвівсть підтримувати велику кількість мов у використанні додатку.

Адаптивність – на сьогоднішні час наявність адаптивного веб-сайту сьогодні єстандартом. Тому що кожна друга людина користується мережею через смартфон. Таким чином, необхідно мати веб-сайт, який ідеально працює на будь-якому пристрої будь-якого розміру. І сервіс може допомогти в реалізації цього.

Шаблони сторінок повністю адаптивні. Таким чином, клієнт що використовує смартфон, також отримає свою версію додатку, і зможе здійснити щамовлення.

Легкість у використанні – сервіс являється системою з відкритим вихідним кодом і повністю безкоштовним для використання. Але безкоштовність не означає відсутність функцій. Фактично, повністю зручний інтерфейс вразить вас своєю функцією Wysiwyg, яка дає точно такі ж результати. Іншим фактором, який викликає задоволення, є часті оновлення. Joomla приносить нові оновлення у вигляді нових можливостей і функцій. З кожним новим оновленням працювати стає легше.

Безпека – під час створення веб-додатку безпека інформації є важливим фактором. Сервіс керування вмістом Joomla надає можливість використання двофакторної автентифікації, щоб уникнути ймовірності зламу. Тож додаток буде захищений від зламу, але якщо станеться ситуація коли, втрата даних таки відбулась є можливість швидко відновити свії логін і пароль для входу в систему, що дозволить швидко повернути сервіс управління доставкою до нормальної роботи.

Перевагою використання цієї системи є велика кількість шаблонів, для створення веб-сторінки. Шаблон описує, як виглядатиме додаток. Якщо ви бажаєте, ви можете зберегти вигляд за замовчуванням або налаштувати його відповідно до ваших вимог. З останніми оновленнями системи якість шаблонів збільшилася і вони отримали адаптивність, тепер є можливість використовувати шаблон сторінки на будь-якому пристрої.

Вигляд інтерфейсу Joomla зображений на рисунку 3.12.

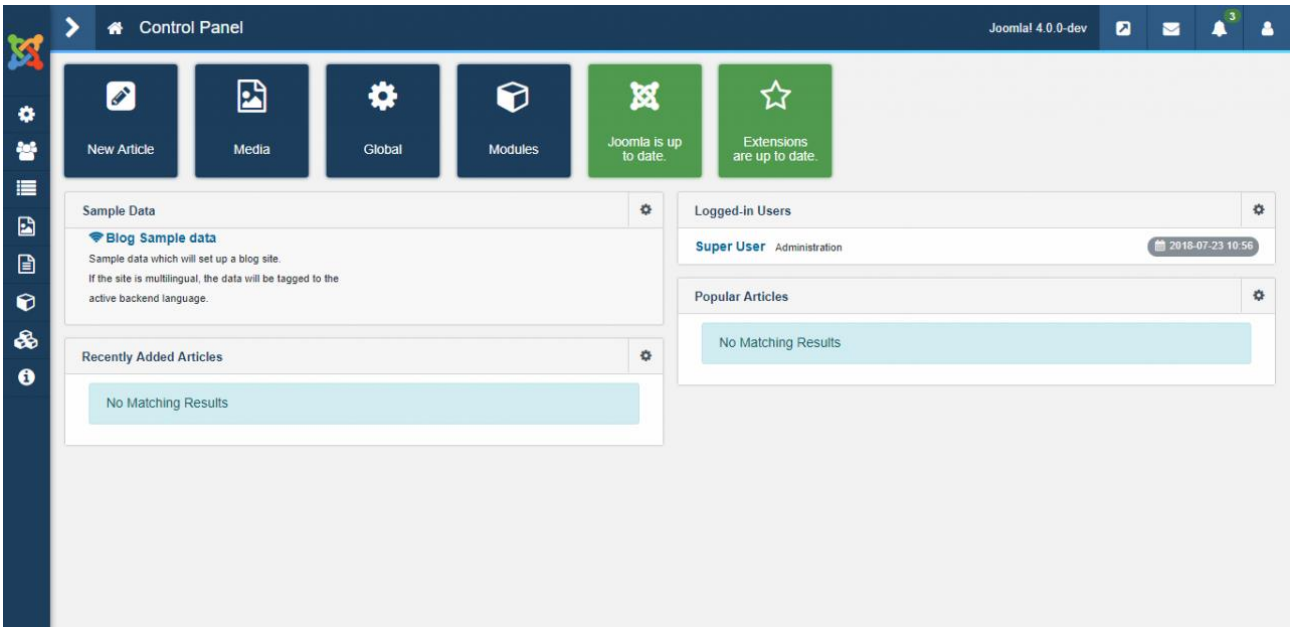


Рисунок 3.12 – Інтерфейс CMS Joomla

Joomla має потужні інструменти для оптимізації пошукових систем, створені та готові одразу після виходу з коробки такі як:

–Метадані та ключові слова.

–Підтримка Mod_rewrite для URL-адрес SEF.

–Створення меню з можливістю створення чітких і послідовних карт сайту.

Серед недоліків цієї системи є те що:

– Сервіс може повільно завантажуватися, особливо на великих веб-сайтах.

– Обмеженість функцій, в порівнянні з іншими сервісами уплвління вмістом

– Несумісність із певними браузерами: Joomla не сумісна з деякими найпопулярнішими браузерами, такими як Chrome і Firefox.

– Сервіс має обмежені можливості щодо налаштування, в порівнянні з іншими системами.

– Ціна плагінів та шаблонів, зависока

– Сервіс не має стільки варіантів підтримки серед сторонніх розробників, в порівнянні з іншими системами.

Сервіс має велику кількість розширень, що допомагає полегшити розробку додатка. Велика кількість плагінів і різних модулів доволить реалізувати інтеграцію форм чи будь-яких інших сервісів, наявна можливість просто обрати пагін та налаштувати його для виконання задачі.

Велика кількість матеріалів для навчання таких як: статті, книжки, уроки допоможуть краще розібратись з встановленням і налаштуванням системи, допоможуть краще вибрати методі реалізації певних задач.

Вигляд інтерфейсу рошириннь Joomla зображений на рисунку 3.13.

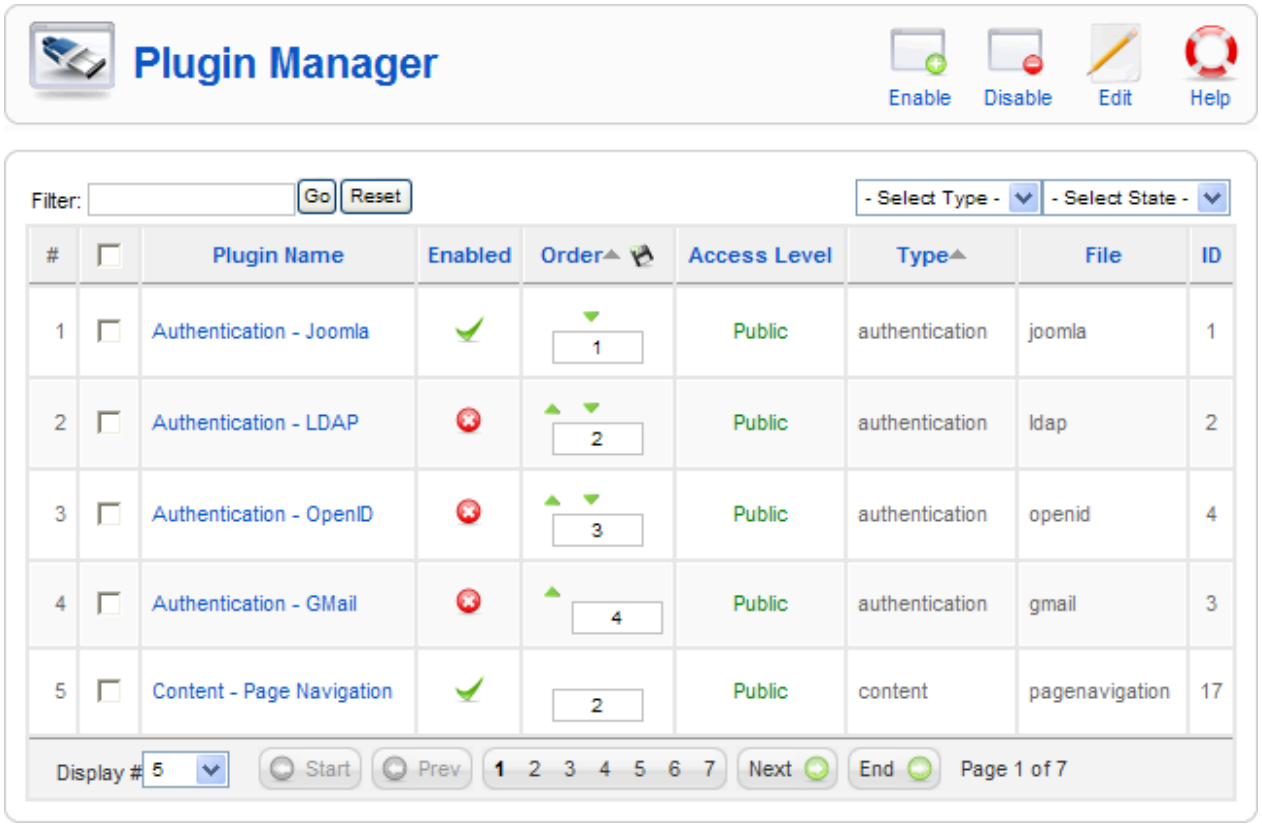


Рисунок 3.13 – Шаблони і плагіни в CMS Joomla

Також серед недоліків цієї системи можна виділити, велика кількість зайвого коду що створюється самою системою, при розробці додатка. Актуальність цього створеного коду під питанням, так як краще і швидше для нових функцій написати, новий код, а завелика кількість само генерованого коду, може впливати на зрозумілість системи дла розробника.

Велика кількість зайвого коду, впливає на вагу і об’єм веб-сторінок, що в свою чергу призводить до зниження швидкості завантаження додатка. Також це призводить до більшого навантаження на сервер, що також сповільнює роботу.

Важливою проблемою є відсутність технічної підтримки. Якщо буде ситуація з проблемами в реалізації сервісу, просто так знайти відповідь буде важко, і необхідно буде звертатись безпосередньо до розробників цієї системи керування вмістом.

Ще проблеми з навігацію створюють незручність у використаннях сервісів, розроблених з використанням цієї системи керування вмістом. Це пов’язано з направильною обробкою і перетворенням url посилань.

Альтернативним сервісом для використання для контролю вмісту може стати CMS Drupal. Drupal є потужною платформою для розробки веб-сайтів. Drupal дотримується сучасних моделей об’єктно-орієнтованого програмування, найкращих практик PHP, стандартів HTML5 і YAML. Він також містить інші веб-технології, зокрема CKEditor, Symfony2, Twig, jQuery, Backbone.js і Guzzle. Розширення функціональних можливостей і отримання повного контролю над дизайном досягається завдяки широкому асортименту доповнень у формі модулів, шаблонів тиа розширеннь.

Серед переваг цієї системи є:

Безкоштовне використання та відкритий код. Drupal був би все ще популярний, якби він був комерційним. Але той факт, що ця CMS абсолютно безкоштовна, дозволяє їй конкурувати з іншими подібними продуктами - такими як Joomla і WordPress. Крім того, системний код відкритий, що дозволяє користувачеві перебудувати його відповідно до своїх потреб.

Зосередження на професійних веб-майстрах, це дає перевагу в швидкості розробки системи. Drupal початково був розроблений з розрахунком, що ним будуть користуватися переважно професіонали. Це головна перевага системи, яка виділяє її серед основних конкурентів.

Багато простору для творчості. За допомогою Drupal веб-розробники мають можливість створювати ресурси, нехтуючи різницею між складним і оригінальним дизайном. Це істотно відрізняє Drupal від основної безкоштовної конкуренції з CMS, яка не може похвалитися такою здатністю створювати складні дизайни.

Висока швидкість. У порівнянні з іншими CMS Drupal, за рахунок вбудованого кешування має найкращу швидкість.

Багатий набір модулів. Оскільки вихідний код відкритий, як і в будь-якій безкоштовній CMS, Drupal має можливість покращувати та розширювати свою функціональність шляхом додавання додаткових модулів.

Наявність додаткових функцій. Завдяки наявності вбудованих запитів, форумів і мультиблогів будь-який проект, створений за допомогою Drupal, отримує хороший старт.

Безпека. На відміну від своїх безкоштовних конкурентів, ця система має підвищений захист, тому не так легко виходить з ладу.

Професійна онлайн-спільнота. Незважаючи на те, що фанатів Joomla та WordPress стало більше, в інтернет-спільноті Drupal є велика кількість професійних програмістів.

Вигляд інтерфейсу системи управління вмістом Drupal зображений на рисунку 3.14.

Рисунок 3.14 – інтерфейсу CMS Drupal

Серед недоліків цієї системи можна виділити:

Важкість у ознайомленні і вивченні системи. Користувачі, які раніше н стикалися з подібними системами, матимуть проблеми з реалізацією їхніх додатків.

Розширений інтерфейс. Через велику кількість доступних інструментів ускладнене швидке освоєння системи.

Високі системні вимоги. Система займає багато місця на хостингу, тому сильно збільшує вартість оплати та підтримки.

В порівнянні з іншими шаблонами, доволі мало шаблонів для створення сторінок і більшість з них доволі низької якості, для реалізації шаблона що матиме всю необхідну структуру, його спочатку потрібно буде купити.

Складність до розширення і масштабування проектів, тому що для створення і оновлення файлів необхідно буде використовувати FTP, що в першу чергу збільшує час роботи, необхідний для реалізації та додає складності в реалізації.

Якщо порівнювати вищезгадані системи керуванням вмістом то з них найкраще підходять для виконання задачі wordpress та drupal. Порівнюючи їх варти в першу чергу виділити такі критерії як:

- Легкість в розробці
- Можливість масштабування системи
- Можливість розширення й покращення роботи системи шляхом додавання плагінів.
- Швидкість реалізованого додатка для кінцевого користувача.

В такому випадку для розробки системи управління доставкою варто обрати сервіс керування вмістом wordpress, так як він переважає свого опонента по всіх виділених критеріях.

Представлення вигляду основної сторінки наведено в рисунках 3.15 та 3.16.

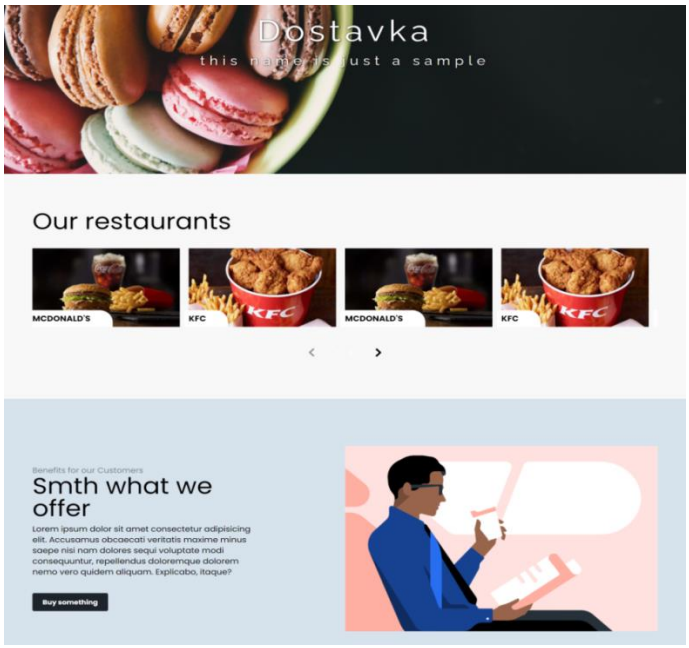


Рисунок 3.15 – Вигляд частини головної сторінки сервісу доставки

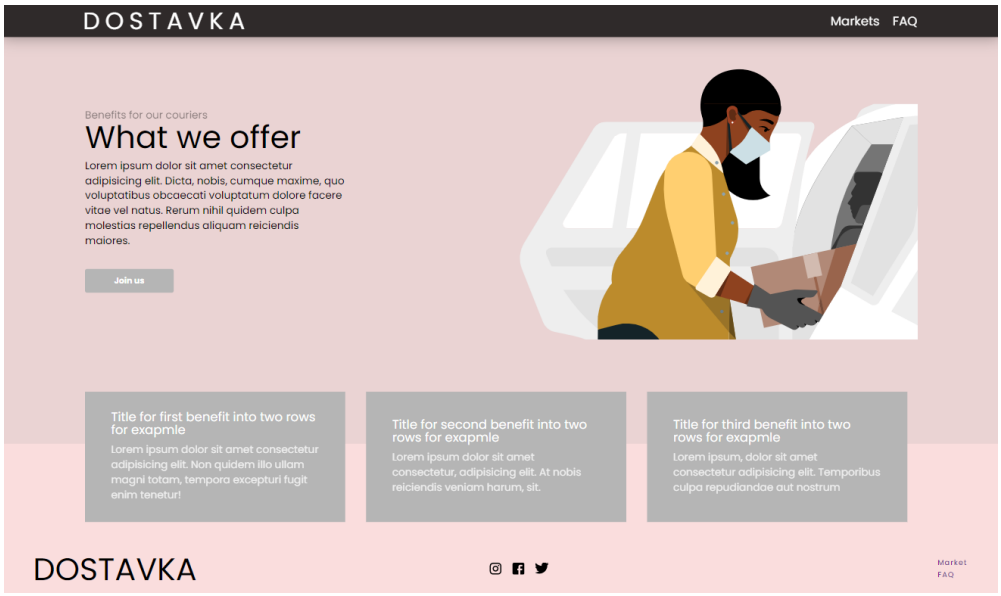


Рисунок 3.16 – Вигляд частини головної сторінки сервісу доставки

Для візуалізації організації файлової структури системи управління доставкою, було застосовано одну із можливих діаграм, а саме діаграму компонентів[60, 61].

Діаграма компонентів використовується для розбиття великої об’єктно-орієнтованої системи на менші компоненти, щоб зробити їх більш легкими для організації. Діаграма моделює фізичний вигляд системи, такий як файли, бібліотеки тощо, які знаходяться всередині вузла.

Діаграма візуалізує зв’язки, а також організацію між вузлами, присутніми в системі. Це допомагає у формуванні виконуваної системи. Вузол - це окремий елемент системи, який можна замінити та виконати. Деталі реалізації вузла приховані, і для виконання функції потрібен інтерфейс. Це як чорний ящик, поведінка якого пояснюється наданими та необхідними інтерфейсами.

Для реалізації діаграми компонентів обрано головну сторінку системи, що має назву Main.php, далі в свою чергу від основоної сторінки за ієрархією додаються інші модулі системи та їхні залежності між собою.

Представлення діаграми компонентів основної сторінки на рисунку 3.17

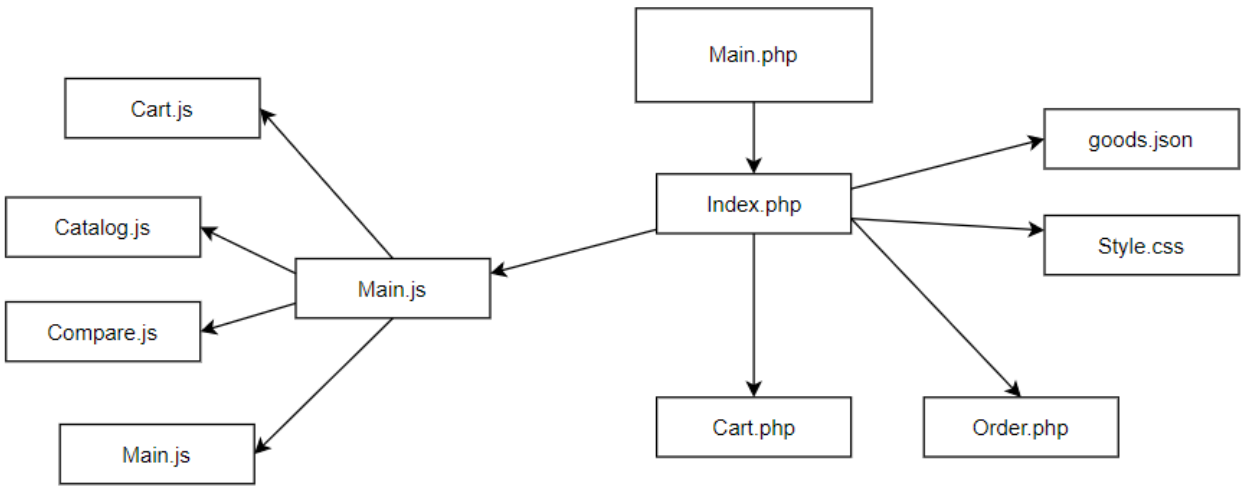


Рисунок 3.17 – Представлення діаграми компонентів основної сторінки

3.3 Реалізація системи побудови найкоротших шляхів

Перша частина системи, що відповідає за отримання даних реалізована за допомогою картографічного сервісу Google Maps, а саме з допомогою веб-сервісу Distance Matrix API що повертає інформацію про рекомендований маршрут між почтковими і кінцевими точками(Рисунок 3.18).

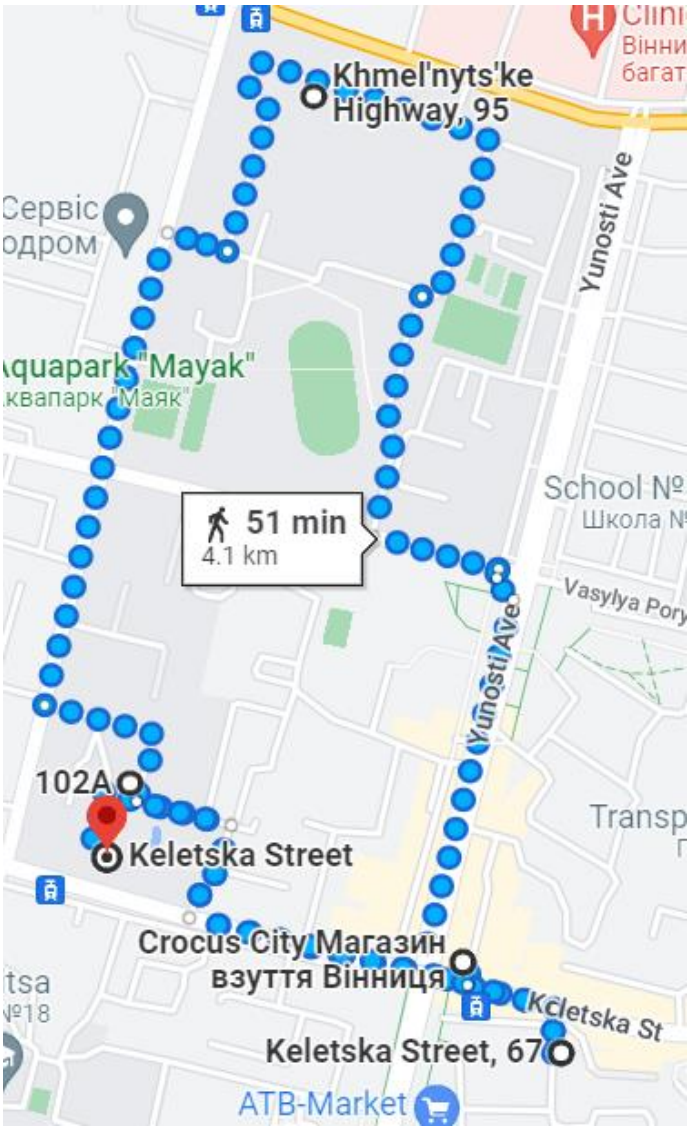


Рисунок 3.18 – Маршрут прокладений за допомогою Google Maps

Отримані дані відправляються до сервісу GeoJson, де з них формується колекція з додаванням до вже існуючих даних інформацію про кількість перехресть та довжину доріг що їх з’єднують.

З отриманих даних система формує граф і за допомогою алгоритму A* обчислюється найкоротший шлях.

Для відображення обчисленого найкоротшого маршруту використовується бібліотека GeoJson-Pathfinder, яка з допомогою розширеного API імпортує отриманий маршрут в систему Open Source Routing Machine, що використовує для відображення картографічну систему OpenStreetsMap (Рисунок 3.19).

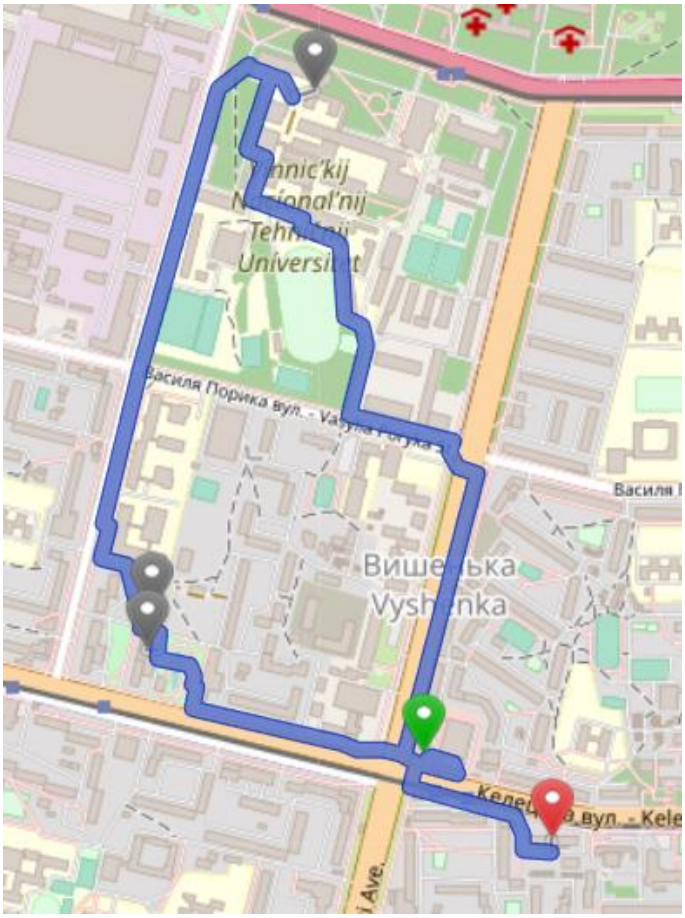


Рисунок 3.19 – Оптимізований маршрут

В кінці система OSRM додає навігатор до вже створеного маршруту, що допоможе кур’єрові краще взаємодіяти з картою (Рисунок 3.20).

Після обробки результатів маршруту наведеного в прикладі, можна побачити, що звичайний маршрут має протяжність 4.1 км, та орієнтовний час доставки 51 хвилина. Побудований системою маршрут має трохи меншу протяжність, а саме 3.2 км, та орієнтовний час доставки 44 хвилини.

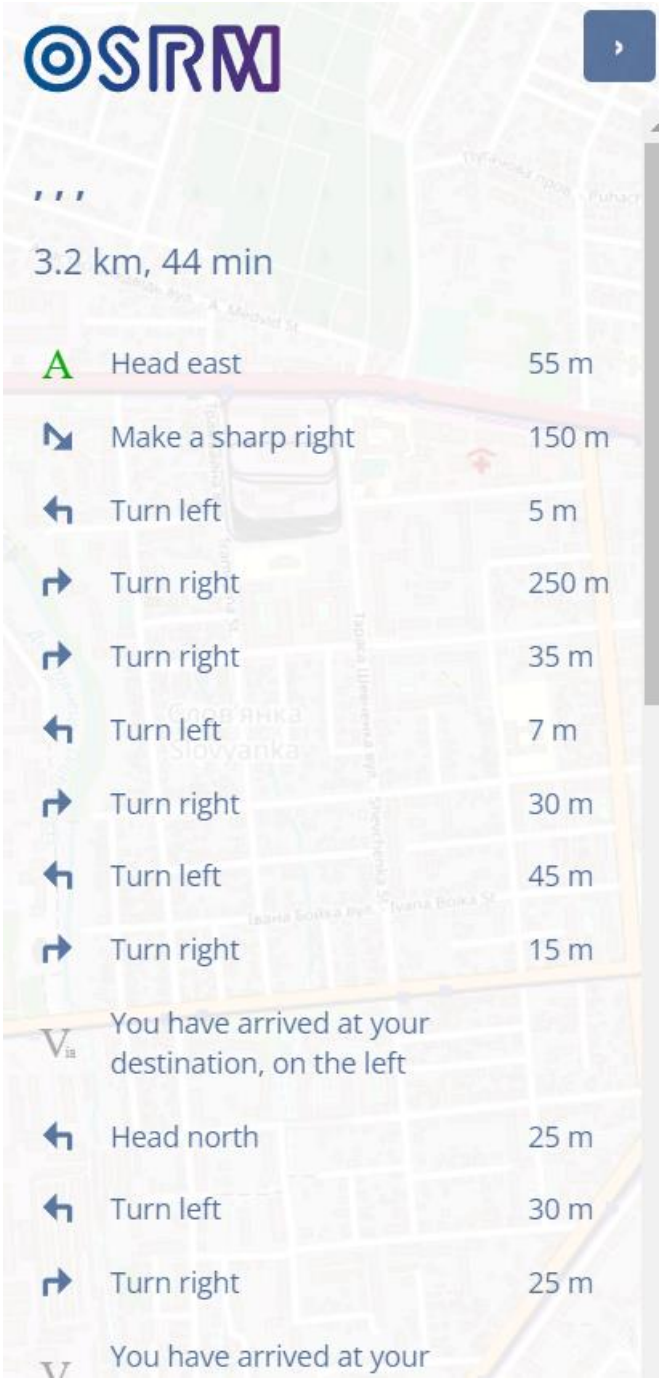


Рисунок 3.20 – Вигляд побудованого маршруту

3.4 Обгрунтування методів масштабування та автоматизації системи

Для покращення клієнтського досвіду з роботою із системою, а також кращого збору і контролю інформації запропоновано використовувати систему трекінгу.

Для реалізації відстеження використано внутрішню функцію API Google Maps для відображення місцезнаходження девайса – Location Sharing. Для

стилізації позначок кур’єрів використано сервіс Snazzy Maps, що дозволяє змінювати базові стилі стандартних карт. Зображення отриманого результату показано на рисунку 3.21.

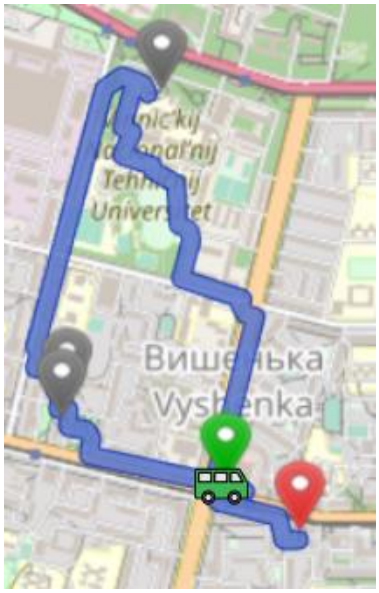


Рисунок 3.21 – Позначка кур’єра на карті

В ситуації коли кур’єрів більше ніж один система будує інші шляхи і зображує їх на дисплеї менеджера в такому вигляді. Отриманий результат зображений на рисунку 3.22.

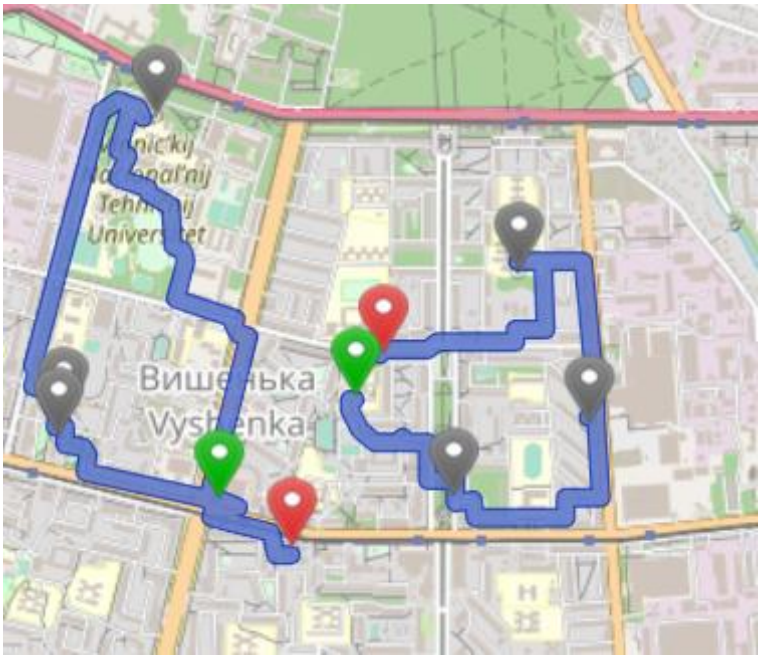


Рисунок 3.22 – Відображення шляхів на карті

3.5 Інструкція програмісту

Підсистема автоматизованого управління доставкою розроблена з використанням мов програмування HTML для розмітки самої системи, CSS для додавання стилів до вже розміченої системи, Javascript для реалізації логіки та анімування всієї системи, PHP для створення програми в сервісі керування вмістом. Як середовище управління контентом використовується сервіс WordPress.

Необхідні вимоги:

–- WINDOWS 7/8/10

– CMS

– Середовище для створення програмного забезпечення

Для запуску реалізованого додатку локально необхідно отримати доступ до одного із стандартних портів портів, наприклад localhost:3000, підключитись до нього та запустити систему.

3.6 Інструкція користувачу

В ситуації коли додаток запускається локально, його потрібно запусукати з допомогою серверу, для цього потрібно встановити один із доступних серверів. Для запуску реалізованого додатку локально необхідно отримати доступ до одного із стандартних портів портів, наприклад localhost:3000, підключитись до нього та запустити систему. Після цього програма цілком готова до роботи. Для створення замовлення необхідно відкрити оду із вкладок що з’яляться при запуску додатку, а потім необхідно слідувати алгоритму. Для перегляду додатку з ролі кур’єра, необхідно відкрити вкладку кур’єр, пройти реєстрацію корткої форми і увійти до свого акаутну. Після цього стане доступний функціонал кур’єрської частини. Для завершення роботи системи потрібно вийти з аканту, закрити вкладку та вимкнути сервер.

4. ЕКОНОМІЧНА ЧАСТИНА

4.1 Комерційний та технологічний аудит науково-технічної розробки

Метою даного розділу є проведення технологічного аудиту, в даному випадку розробки системи з пошуку оптимального маршруту доставки за динамічними даними, що отримуються з відкритих джерел. Особливістю програми є те, що дана технологія є веб-сервісом, який отримує дані про можливі шляхи та стан дорожнього трафіку і на їх основі будує маршрут.

Аналогом може бути: Gurtam – 1680 грн/рік; ANT-Logistics ® – 6000 грн/рік; BAS Управління автотранспортом Стандарт – 12 300 грн; My Route Online – 20400 грн/рік.

Для проведення комерційного та технологічного аудиту залучають не менше 3-х незалежних експертів. Оцінювання науково-технічного рівня розробки та її комерційного потенціалу рекомендується здійснювати із застосуванням п’ятибальної системи оцінювання за 12-ма критеріями, у відповідності із табл. 4.1.

Таблиця 4.1 – Рекомендовані критерії оцінювання комерційного потенціалу розробки та їх можлива бальна оцінка

Бали (за 5-ти бальною шкалою)					
Крите-рій	0	1	2	3	4
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздат-ність продукту в реальних умовах

Продовження табл. 4.1

Ринкові переваги					
2	Багато аналогів на малому ринку	Ринкові п Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші, ніж в аналогів	Технічні та споживчі властивості продукту трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі, ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатаційних витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий, але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практик на здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти та час на навчання наявних фахівців	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так із комерційної реалізації ідеї
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використовуються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використовуються у виробництві

Продовження табл. 4.1

11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більше 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менше 3-х років. Термін окупності інвестицій менше 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхідно отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що вимагає значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту вимагає незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Усі дані по кожному параметру занесено в таблиці 4.2

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії оцінювання	ПІБ експертів		
	Експерт 1	Експерт 2	Експерт 3
	Бали		
Технічна здійсненність концепції	4	4	4
Наявність аналогів на ринку	3	4	4
Цінова політика	4	4	4
Технічні та споживчі властивості виробу	4	3	4
Експлуатаційні витрати	3	3	4
Ринок збуту	4	4	4
Конкурентоспроможність	3	3	2
Фахівці з технічної і комерційної реалізації	3	3	3
Фінансування	4	3	4
Матеріально-технічна база	3	4	4
Термін реалізації ідеї	4	4	4
Супровідна документація	3	4	3
Сума	42	43	44
Середньоарифметична сума балів	(42+43+44) / 3 = 43		

За даними таблиці 4.2 можна зробити висновок щодо рівня комерційного потенціалу даної розробки. Для цього доцільно скористатись рекомендаціями, наведеними в таблиці 4.3.

Таблиця 4.3 - Рівні комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
31-40	Вище середнього
41-48	Високий

Як видно з таблиці, рівень комерційного потенціалу розроблюваного нового програмного продукту є високим, що досягається за рахунок прокладання кращих маршрутів, дозволить зменшити час доставки і збільшити кількість замовлень доставлених кур'єром. Особливістю програми є те, що дана технологія є веб-сервісом, який отримує дані про можливі шляхи та стан дорожнього трафіку і на їх основі будує оптимальний маршрут доставки за динамічними даними, що отримуються з відкритих джерел.

4.2 Прогнозування витрат на виконання науково-дослідної (дослідно-конструкторської) роботи

4.2.1 Основна заробітна плата розробників, яка розраховується за формулою:

$$З_о = \frac{M}{T_р} \cdot t, \qquad (4.1)$$

де М – місячний посадовий оклад конкретного розробника (дослідника), грн.;

$T_р$ – число робочих днів в місяці, 21 днів;

t – число днів роботи розробника (дослідника).

Результати розрахунків зведемо до таблиці 4.1.

Таблиця 4.1 – Основна заробітна плата розробників

Найменування посади	Місячний посадовий оклад, грн.	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	32500	1547,62	33	51071,429
Програміст	30000	1428,57	33	47142,857
Всього				98214,29

Так як в даному випадку розробляється програмний продукт, то розробник виступає одночасно і основним робітником, і тестувальником розроблюваного програмного продукту.

4.2.2 Додаткова заробітна плата розробників, які приймали участь в розробці обладнання.

Додаткова заробітна плата прийнято розраховувати як 12 % від основної заробітної плати розробників та робітників:

$$З_д = З_о \cdot 12\ \% / 100\ \% \qquad (4.2)$$

$$З_д = (98214{,}29 \cdot 12\ \% / 100\ \%) = 11785{,}71 \ \text{(грн.)}$$

4.2.3 Нарахування на заробітну плату розробників.

Згідно діючого законодавства нарахування на заробітну плату складають 22 % від суми основної та додаткової заробітної плати.

$$Н_з = (З_о + З_д) \cdot 22\ \% / 100\% \qquad (4.3)$$

$$Н_з = (98214{,}29 + 11785{,}71) \cdot 22\ \% / 100\ \% = 24200{,}00 \ \text{(грн.)}$$

Оскільки для розроблювального пристрою не потрібно витрачати матеріали та комплектуючи, то витрати на матеріали і комплектуючі дорівнюють нулю.

4.2.4 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалось для розробки в спрощеному вигляді амортизація обладнання, що використовувалась для розробки розраховується за формулою:

$$A = \frac{Ц}{Тв} \cdot \frac{t_{вик}}{12} \text{ [грн.].} \qquad (4.4)$$

де Ц – балансова вартість обладнання, грн.; Т – термін корисного використання обладнання згідно податкового законодавства, років, $t_{вик}$ – термін використання під час розробки, місяців

Розрахуємо, для прикладу, амортизаційні витрати на комп'ютер балансова вартість якого становить 30000 грн., термін його корисного використання згідно податкового законодавства – 2 роки, а термін його фактичного використання – 1,57 міс.

$$A_{обл} = \frac{30000}{2} \times \frac{1{,}57}{12} = 1964{,}29 \ грн.$$

Аналогічно визначаємо амортизаційні витрати на інше обладнання та приміщення. Розрахунки заносимо до таблиці 4.2. Для розрахунку амортизації нематеріальних ресурсів використовується формула:

$$A_{н.р.} = Ц_{н.р.} * Н_а * \frac{t_{вик.}}{12} \qquad (4.5)$$

Але, так як вартість ліцензійної ОС та спеціалізованих ліцензійних нематеріальних ресурсів менше 20000 грн, то даний нематеріальний актив не амортизується, а його вартість включається у вартість розробки повністю, $B_{нем.ак.}$

= 8100 грн.

Таблиця 4.2 – Амортизаційні відрахування матеріальних і нематеріальних ресурсів для розробників

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортиза-ційні відрахуван-ня, грн.
Комп'ютер та комп'ютерна переферія	30000	2	1,57	1964,286
Офісне обладнення	23000	4	1,57	752,976
Приміщення	1100000	20	1,57	7202,381
Всього				9919,64

Тарифи на електроенергію для непобутових споживачів (промислових підприємств) відрізняються від тарифів на електроенергію для населення. При цьому тарифи на розподіл електроенергії у різних постачальників (енергорозподільних компаній), будуть різними. Крім того, розмір тарифу залежить від класу напруги (1-й або 2-й клас). Тарифи на розподіл електроенергії для всіх енергорозподільних компаній встановлює Національна комісія з регулювання енергетики і комунальних послуг (НКРЕКП). Витрати на силову електроенергію розраховуються за формулою:

$$В_е = В \cdot П \cdot Ф \cdot К_п, \qquad (4.6)$$

де В – вартість 1 кВт-години електроенергії для 1 класу підприємства, В = 6,2 грн./кВт;

П – встановлена потужність обладнання, кВт. П = 0,5 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

$К_п$ – коефіцієнт використання потужності, $К_п$ = 0,9.

$$В_е = 0,9 \cdot 0,5 \cdot 8 \cdot 33 \cdot 6,2 = 736,56 \text{ (грн.)}$$

Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» належать витрати, які не знайшли відображення у зазначених статтях витрат і можуть бути віднесені безпосередньо на собівартість досліджень за прямими ознаками. Витрати за статтею «Інші витрати» розраховуються як 50…100% від суми основної заробітної плати дослідників:

$$I_в = (З_о + З_р) \cdot \frac{Н_{ів}}{100\%}, \quad (4.7)$$

де $Н_{ів}$ – норма нарахування за статтею «Інші витрати».

$$I_в = 98214,29 * 70\% / 100\% = 68750 \text{ (грн.)}$$

До статті «Накладні (загальновиробничі) витрати» належать: витрати, пов'язані з управлінням організацією; витрати на винахідництво та раціоналізацію; витрати на підготовку (перепідготовку) та навчання кадрів; витрати, пов'язані з набором робочої сили; витрати на оплату послуг банків; витрати, пов'язані з освоєнням виробництва продукції; витрати на науково-технічну інформацію та рекламу та ін. Витрати за статтею «Накладні (загальновиробничі) витрати» розраховуються як 100…150% від суми основної заробітної плати дослідників:

$$Н_{нзв} = (З_о + З_р) \cdot \frac{Н_{нзв}}{100\%}, \quad (4.8)$$

де $Н_{нзв}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати».

$$Н_{нзв} = 98214,29 * 140 \% / 100 \% = 137500 \text{ (грн.)}$$

Витрати на проведення науково-дослідної роботи.

Сума всіх попередніх статей витрат дає загальні витрати на проведення науково-дослідної роботи:

$$B_{заг} = 98214,29+11785,71+24200,00+9919,64+8100+736,56+68750+$$
$$+137500 = 359206,20 \text{ грн.}$$

Розрахунок загальних витрат на науково-дослідну (науково-технічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються *ЗВ*, визначається за формулою:

$$ЗВ = \frac{B_{заг}}{\eta} \text{ (грн)}, \qquad (5.9)$$

де η – коефіцієнт, який характеризує етап (стадію) виконання науково-дослідної роботи.

Так, якщо науково-технічна розробка знаходиться на стадії: науково-дослідних робіт, то η=0,1; технічного проектування, то η=0,2; розробки конструкторської документації, то η=0,3; розробки технологій, то η=0,4; розробки дослідного зразка, то η=0,5; розробки промислового зразка, то η=0,7; впровадження, то η=0,9. Оберемо η = 0,5, так як розробка, на даний момент, знаходиться на стадії дослідного зразка:

$$ЗВ = 359206,20 / 0,5 = 718412 \text{ грн.}$$

4.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

В ринкових умовах узагальнювальним позитивним результатом, що його може отримати потенційний інвестор від можливого впровадження результатів тієї чи іншої науково-технічної розробки, є збільшення у потенційного інвестора величини чистого прибутку. Саме зростання чистого прибутку забезпечить потенційному інвестору надходження додаткових коштів, дозволить покращити фінансові результати його діяльності, підвищить конкурентоспроможність та може позитивно вплинути на ухвалення рішення щодо комерціалізації цієї розробки.

Для того, щоб розрахувати можливе зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно:

а) вказати, з якого часу можуть бути впроваджені результати науково-технічної розробки;

б) зазначити, протягом скількох років після впровадження цієї науково-технічної розробки очікуються основні позитивні результати для потенційного інвестора (наприклад, протягом 3-х років після її впровадження);

в) кількісно оцінити величину існуючого та майбутнього попиту на цю або аналогічні чи подібні науково-технічні розробки та назвати основних суб'єктів (зацікавлених осіб) цього попиту;

г) визначити ціну реалізації на ринку науково-технічних розробок з аналогічними чи подібними функціями.

При розрахунку економічної ефективності потрібно обов'язково враховувати зміну вартості грошей у часі, оскільки від вкладення інвестицій до отримання прибутку минає чимало часу. При оцінюванні ефективності інноваційних проектів передбачається розрахунок таких важливих показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки проведення науково-технічних розробок, розрахунок економічної ефективності науково-технічної розробки за її можливої

комерціалізації потенційним інвестором можна об’єднати, враховуючи визначені ситуації з відповідними умовами.

4.3.1 Розробка чи суттєве вдосконалення програмного засобу (програмного забезпечення, програмного продукту) для використання масовим споживачем.

В цьому випадку майбутній економічний ефект буде формуватися на основі таких даних:

$$\Delta П_i = (\pm \Delta Ц_0 \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}), \qquad (4.10)$$

де $\pm\Delta Ц_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_o$ – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки, $Ц_o = Ц_б \pm \Delta Ц_o$;

$Цб$ – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт $\lambda = 0{,}8333$.

p – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2022 році ϑ =18%.

Припустимо, що при прогнозованій ціні 750 грн. за одиницю виробу, термін збільшення прибутку складе 3 роки. Після завершення розробки і її вдосконалення, можна буде підняти її ціну на *50* грн. Кількість одиниць реалізованої продукції також збільшиться: протягом першого року – на 15000 шт., протягом другого року – на 12000 шт., протягом третього року на 8000 шт. До

моменту впровадження результатів наукової розробки реалізації продукту не було:

$$\Delta П_1 = (0*50 + (750 + 50)*15000)* 0{,}8333* 0{,}38) * (1 - 0{,}18) = 2921249{,}883 \text{ грн.}$$

$$\Delta П_2 = (0*50 + (750 + 50)*(15000+12000)* 0{,}8333* 0{,}38) * (1 - 0{,}18) = 5608799{,}776 \text{ грн.}$$

$$\Delta П_3 = (0*50 + (750 + 50)*(15000+12000+8000)* 0{,}8333* 0{,}38) * (1 - 0{,}18) = 7270666{,}376 \text{ грн.}$$

Отже, комерційний ефект від реалізації результатів розробки за три роки складе 15800716,03 грн.

4.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Розраховуємо приведену вартість збільшення всіх чистих прибутків *ПП*, що їх може отримати потенційний інвестор від можливого впровадження та комерціалізації науково-технічної розробки:

$$ПП = \sum_{1}^{T} \frac{\Delta П_i}{(1+\tau)^t}, \qquad (5.11)$$

де $\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, $\tau = 0{,}05 \ldots 0{,}15$;

t – період часу (в роках).

Збільшення прибутку ми отримаємо починаючи з першого року:

$$ПП= (2921249,883/(1+0,1)^1)+(5608799,776/(1+0,1)^2)+(7270666,376/(1+0,1)^3) =$$
$$2655681,71 + 4635371,715 + 5462559,261 = 12753612,69 \text{ грн.}$$

Далі розраховують величину початкових інвестицій *PV*, які потенційний інвестор має вкласти для впровадження і комерціалізації науково-технічної розробки. Для цього можна використати формулу:

$$PV = k_{інв} * ЗВ, \quad (4.12)$$

де $k_{інв}$ – коефіцієнт, що враховує витрати інвестора на впровадження науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку приміщень, розробку технологій, навчання персоналу, маркетингові заходи тощо; зазвичай $k_{інв}=2...5$, але може бути і більшим;

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн.

$$PV = 2 * 718412 = 1436824,81 \text{ грн.}$$

Тоді абсолютний економічний ефект $E_{абс}$ або чистий приведений дохід *(NPV, Net Present Value)* для потенційного інвестора від можливого впровадження та комерціалізації науково-технічної розробки становитиме:

$$E_{абс} = ПП – PV, \quad (4.13)$$

$$E_{абс} = 12753612,69 – 1436824,81 = 11316787,88 \text{ грн.}$$

Оскільки $E_{абс} > 0$ то вкладання коштів на виконання та впровадження результатів даної науково-дослідної (науково-технічної) роботи може бути доцільним.

Для остаточного прийняття рішення з цього питання необхідно розрахувати внутрішню економічну дохідність або показник внутрішньої но-рми дохідності *(IRR, Internal Rate of Return)* вкладених інвестицій та порівняти її з так званою бар'єрною ставкою дисконтування, яка визначає ту мінімальну внутрішню економічну дохідність, нижче якої інвестиції в будь-яку науково-технічну розробку вкладати буде економічно недоцільно.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій $E_в$. Для цього використаємо формулу:

$$E_в = \sqrt[T_ж]{1 + \frac{E_{абс}}{PV}} - 1, \qquad (4.14)$$

$T_ж$ – життєвий цикл наукової розробки, роки.

$$E_в = \sqrt{з\ (1 + 11316787{,}88/1436824{,}81} - 1 = 1{,}071$$

Визначимо мінімальну ставку дисконтування, яка у загальному вигляді визначається за формулою:

$$\tau = d + f, \qquad (4.15)$$

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2022 році в Україні d = (0,09…0,14);

f – показник, що характеризує ризикованість вкладень; зазвичай, величина f = (0,05...0,5).

$$\tau_{\min} = 0{,}14 + 0{,}05 = 0{,}19.$$

Так як $E_в > \tau_{min}$, то інвестор може бути зацікавлений у фінансуванні даної наукової розробки.

Розрахуємо термін окупності вкладених у реалізацію наукового проекту інвестицій за формулою:

$$T_{ок} = \frac{1}{E_в}, \tag{4.16}$$

$$T_{ок} = 1 / 1{,}071 = 0{,}93 \text{ р.}$$

Оскільки $T_{ок}$ < 3-х років, а саме термін окупності рівний 0,93 роки, то фінансування даної наукової розробки є доцільним.

Висновки до розділу: економічна частина даної роботи містить розрахунок витрат на розробку нового програмного продукту, сума яких складає 718412 гривень. Було спрогнозовано орієнтовану величину витрат по кожній з статей витрат. Також розраховано чистий прибуток, який може отримати виробник від реалізації нового технічного рішення, розраховано період окупності витрат для інвестора та економічний ефект при використанні даної розробки. В результаті аналізу розрахунків можна зробити висновок, що розроблений програмний продукт за ціною дешевший за аналог і є висококонкурентоспроможним. Період окупності складе близько 0,93 роки.

Висновки

В першому розділі було проаналізовано ринок альтернативних систем доставок. Проведено порівняльний аналіз інснуючих систем доставки та обрано критерії для побудови власної системи. Розглянуто можливі технології для реалізації відстежування замовлення в реальному часі. Поставлена задача для оптимізації швидкості доставки кур’єром за рахунок пошуку найкоротшого шляху з урахуванням стану шляху, в ситуації, коли наявні кілька замовлень.

В другому розділі було проаналізовано методи проектування й розробки підсистеми автоматизованого управління доставкою з можливістю відслідковування кур'єра за допомогою трекінгової системи з використанням сучасних технологій. Для отримання початкових данних про координати, шляхи, час доставки було проаналізовано картографічні сервіси доступні на ринку та використано один іх них. Для обробки і побудови оптимізованого маршруту використовувались сервіси, що дають найкращий результат при виконані даних задач.

Для вирішення задачі з оптимізації маршруту було проведено аналіз алгоритмів пошуку шляху на графах та запропоновано комплексний метод пошуку найкоротшого шляху.

В результаті виконання роботи було створено прототип підсистеми автоматизованого управління доставкою з можливістю відслідковування кур'єра за допомогою трекінгової системи з використанням сучасних технологій. Перевагою розробленої підсистеми виступає удосконалений алгоритм пошуку найкоротшого шляху доставки з використанням динамічних даних та можливості відстеження кур’єра при доставлені замовлення на карті.

Список джерел

1. Glovo [Електронний ресурс] //Режим доступу до матеріалу: https://glovoapp.com/ua/uk/
2. Bolt Food [Електронний ресурс] //Режим доступу до матеріалу: https://food.bolt.eu/uk-UA
3. Igogo [Електронний ресурс] //Режим доступу до матеріалу: https://igogo.vn.ua/ua/
4. Liki24 [Електронний ресурс] //Режим доступу до матеріалу: https://liki24.com/uk/
5. Raketa [Електронний ресурс] //Режим доступу до матеріалу: https://raketaapp.com/
6. Задача комівояжера [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Travelling_salesman_problem
7. Single-page application [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Single-page_application
8. Static web page [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Static_web_page
9. Mobile app [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Mobile_app
10. Стивен Хольцнер. Ajax Bible. 2009. — С. 553.
11. Flanagan, David, "JavaScript - The Definitive Guide", 5th ed., O'Reilly, Sebastopol, CA, 2006, p.497
12. Melendez, Steven (10 August 2018). "The Difference Between Dynamic & Static Web Pages"
13. Single-page applications vs. multiple-page applications: pros, cons, pitfalls - BLAKIT - IT Solutions". blak-it.com. BLAKIT - IT Solutions. October 17, 2017.
14. Kotlin [Електронний ресурс] //Режим доступу до матеріалу: https://ru.wikipedia.org/wiki/Kotlin

15. Swift [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Swift_(programming_language)
16. HTML [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/HTML
17. CSS [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/CSS
18. Flanagan, David. JavaScript - The definitive guide (вид. 6). с. 1. «JavaScript is part of the triad of technologies that all Web developers must learn: HTML to specify the content of web pages, CSS to specify the presentation of web pages, and JavaScript to specify the behaviour of web pages
19. SASS [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/Sass
20. Media Mark (3.2.12). Sass - Syntactically Awesome Stylesheets.
21. SASS [Електронний ресурс] //Режим доступу до матеріалу: https://sass-lang.com/documentation
22. LESS [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/Less_(%D0%BC%D0%BE%D0%B2%D0%B0_%D1%81%D1%82%D0%B8%D0%BB%D1%96%D0%B2)
23. Webpack [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/Webpack
24. Web Performance Optimization with webpack. Google Developers. 16 Oct 2018.
25. Gulp.js [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/Gulp.js
26. Grunt [Електронний ресурс] //Режим доступу до матеріалу: https://ru.wikipedia.org/wiki/Grunt_(%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5)
27. Pillora, Jamie. Getting Started with Grunt: The JavaScript Task Runner. — 2014. — ISBN 978-1-78398-062-8.

28. Yarn [Електронний ресурс] //Режим доступу до матеріалу: https://yarnpkg.com/getting-started
29. Npm [Електронний ресурс] //Режим доступу до матеріалу: https://ru.wikipedia.org/wiki/Npm
30. JavaScript [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/JavaScript
31. JavaScript basics [Електронний ресурс] //Режим доступу до матеріалу: https://developer.mozilla.org/en-US/docs/Learn/Getting_started_with_the_web/JavaScript_basics?retiredLocale=uk
32. AngularJS [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/AngularJS
33. What Is Angular? 2013-02-12.
34. React [Електронний ресурс] //Режим доступу до матеріалу: https://ru.wikipedia.org/wiki/React
35. Document Object Model [Електронний ресурс] //Режим доступу до матеріалу: https://ru.wikipedia.org/wiki/Document_Object_Model
36. Ajax [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Ajax_(programming)
37. Model-View-Controller [Електронний ресурс] //Режим доступу до матеріалу: https://ru.wikipedia.org/wiki/Model-View-Controller
38. UI [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/User_interface
39. Google Maps [Електронний ресурс] //Режим доступу до матеріалу: https://developers.google.com/maps/documentation
40. Bing Maps [Електронний ресурс] //Режим доступу до матеріалу: https://www.bingmapsportal.com/
41. GeoJSON [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/GeoJSON

42. Геоінформаційна система [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Geographic_information_system
43. Contraction hierarchies [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikihre.ru/wiki/Contraction_hierarchies
44. Russell, Stuart J. (2018). Artificial intelligence a modern approach. Norvig, Peter (4th ed.). Boston: Pearson. ISBN 978-0134610993.
45. Transit node routing [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Transit_node_routing
46. Алгоритм Дейкстри [Електронний ресурс] //Режим доступу до матеріалу: https://studwood.ru/1244874/matematika_himiya_fizika/algoritm_deykstry_poiska_kratchayshego_puti
47. A* алгоритм [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/A*_search_algorithm
48. API [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/API
49. Java [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Java_(programming_language)
50. Модель «сутність — зв'язок» [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%C2%AB%D1%81%D1%83%D1%82%D0%BD%D1%96%D1%81%D1%82%D1%8C_%E2%80%94_%D0%B7%D0%B2%27%D1%8F%D0%B7%D0%BE%D0%BA%C2%BB
51. G. Everest, «BASIC DATA STRUCTURE MODELS EXPLAINED WITH A COMMON EXAMPLE», in Computing Systems 1976, Proceedings Fifth Texas Conference on Computing Systems, Austin, TX, 1976 October 18-19, pages 39-46. (Long Beach, CA: IEEE Computer Society Publications Office).
52. Entity–relationship model [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

53. Chen, Peter (2002). "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned". Software pioneers. Springer-Verlag. pp. 296–310.
54. JSON [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/JSON
55. MySQL [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/MySQL
56. Mongo DB [Електронний ресурс] //Режим доступу до матеріалу: https://www.mongodb.com/
57. CMS [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Content_management_system
58. WordPress [Електронний ресурс] //Режим доступу до матеріалу: https://uk.wordpress.org/
59. Joomla [Електронний ресурс] //Режим доступу до матеріалу: https://www.joomla.org/
60. Діаграма компонентів [Електронний ресурс] //Режим доступу до матеріалу: https://en.wikipedia.org/wiki/Component_diagram
61. Bell, Donald (December 15, 2004). "UML basics: The component diagram". IBM Developer. Retrieved June 15, 2019.

ДОДАТКИ

ПРОТОКОЛ
ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ
НА НАЯВНІСТЬ ТЕКСТОВИХ ЗАПОЗИЧЕНЬ

Назва роботи: «Оптимізація маршруту доставки замовлення за динамічними даними»

Тип роботи: Магістерська кваліфікаційна робота
(БДР, МКР)

Підрозділ КСУ, ФІТА
(кафедра, факультет)

Показники звіту подібності Unicheck

Оригінальність 95,8% Схожість 4,2%

Аналіз звіту подібності (відмітити потрібне)

☐ Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на розгляд експертної комісії кафедри.

☐ Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Особа, відповідальна за перевірку _____ Галущак А.В.
(підпис) (прізвище, ініціали)

Ознайомлені з повним звітом подібності, який був згенерований системою Unichek щодо роботи.

Автор роботи _____ Щербань М.О.
(підпис) (прізвище, ініціали)

Керівник роботи _____ Никитенко О.Д.
(підпис) (прізвище, ініціали)

Додаток А

ВНТУ

ЗАТВЕРДЖЕНО
Зав. кафедри КСУ ВНТУ,
д.т.н., доцент
_____В'ячеслав КОВТУН

" 14 " ___грудня___ 2022 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

«ОПТИМІЗАЦІЯ МАРШРУТУ ДОСТАВКИ ЗАМОВЛЕННЯ ЗА ДИНАМІЧНИМИ ДАНИМИ»

08-33.МКР.011.00.000 ТЗ

Студент групи ___2АКІТ-21м_____

Підпис ___Михайло ЩЕРБАНЬ___

Керівник к.т.н., доцент, доцент кафедри КСУ

Підпис ___Олена НИКИТЕНКО___

Вінниця 2022

1. Назва та галузь застосування

1.1. Назва – Оптимізація маршруту доставки замовлення за динамічними даними.

1.2. Галузь застосування – Інформаційно-управляючі системи.

2. Підстава для проведення розробки.
Тема магістерської кваліфікаційної роботи затверджена наказом по ВНТУ від “14” вересня 2022 року №203

3. Мета та призначення розробки.

Метою магістерської кваліфікаційної роботи є збільшення швидкості доставки замовленнь,

4. Джерела розробки.

Магістерська кваліфікаційна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

1. Flanagan, David, "JavaScript - The Definitive Guide", 5th ed., O'Reilly, Sebastopol, CA, 2006, p.497

2. Melendez, Steven (10 August 2018). "The Difference Between Dynamic & Static Web Pages"

3. Single-page applications vs. multiple-page applications: pros, cons, pitfalls - BLAKIT - IT Solutions". blak-it.com. BLAKIT - IT Solutions. October 17, 2017.

4. Web Performance Optimization with webpack. Google Developers. 16 Oct 2018.

5. G. Everest, «BASIC DATA STRUCTURE MODELS EXPLAINED WITH A COMMON EXAMPLE», in Computing Systems 1976, Proceedings Fifth Texas Conference on Computing Systems, Austin, TX, 1976 October 18-19, pages 39-46. (Long Beach, CA: IEEE Computer Society Publications Office).

6. Chen, Peter (2002). "Entity-Relationship Modeling: Historical Events, Future Trends, and Lessons Learned". Software pioneers. Springer-Verlag. pp. 296–310.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- можливість отримання даних про місцезнаходження з відкритих сервісів;
- побудова найкоротших шляхів, опираючись на отримані дані;
- можливість визначення координат кожного кур’єра
- представлення в зручному для користувача форматі даних

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

– WINDOWS 7/8/10
– CMS
– Середовище для створення програмного забезпечення

5.2.2. Умови експлуатації системи:

- робота на веб-додатках;
- можливість цілодобового функціонування системи;
- дані оновлюються і є актуальними.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

1. Аналіз існуючих систем управління доставкою. Аналіз можливиз технології для реалізації відстеження замовлення в реальному часі. Постановка задач дослідження «03»_09__ 2022 р.
2. Визначення найкращих алгоритмів пошуку короткого шляху для реалізації поставленої задачі. «11»__11__ 2022 р.
3. Проектування системи управління доставкою «21»__11__ 2022 р.
4. Розробка програмного забезпечення системи «28»__11__ 2022р.

6.2 Графічні матеріали:

1. Розробка UML-діаграм системи «11»__10_ 2022 р.
2. Розробка моделі бази даних системи «24»___10__ 2022р.
3. Тестування програмного забезпечення «16»__11___ 2022 р.

7. Порядок контролю і приймання.

7.1. Хід виконання роботи контролюється керівником роботи. Рубіжний контроль провести до «28»__11_ 2022 р.

7.2. Атестація МКР здійснюється на попередньому захисті. Попередній захист магістерської кваліфікаційної роботи провести до «16»__12_ 2022 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ЕК. Захист магістерської кваліфікаційної роботи провести до «23»__12_ 2022 р.

Додаток В

(обов'язковий)

ІЛЮСТРАТИВНА ЧАСТИНА

«ОПТИМІЗАЦІЯ МАРШРУТУ ДОСТАВКИ ЗАМОВЛЕННЯ ЗА ДИНАМІЧНИМИ ДАНИМИ»

Студент групи 2АКІТ-21м

Підпис Михайло ЩЕРБАНЬ

Керівник к.т.н., доцент, доцент кафедри КСУ

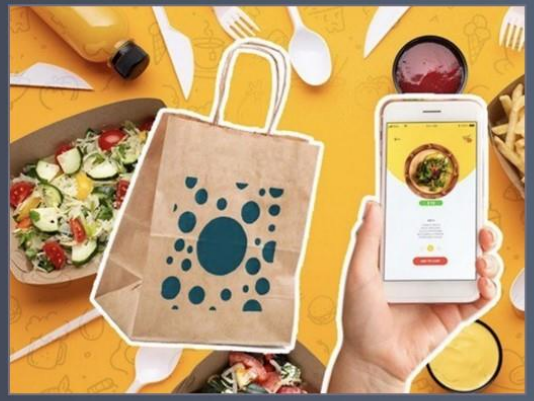
Підпис Олена НИКИТЕНКО

Оптимізація маршруту доставки замовлення за динамічними даними

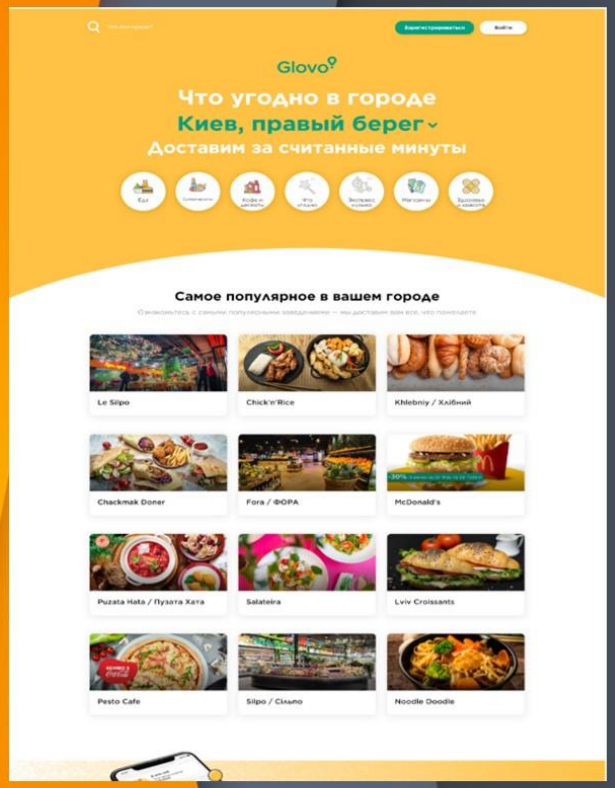
Дипломна робота
(освітньо-кваліфікаційний рівень магістр)
Студента групи 2АКІТ-21м
Щербаня Михайла Олександровича

Науковий керівник
к.т.н., доцент, доцент кафедри КСУ
Никитенко Олена Дмитрівна

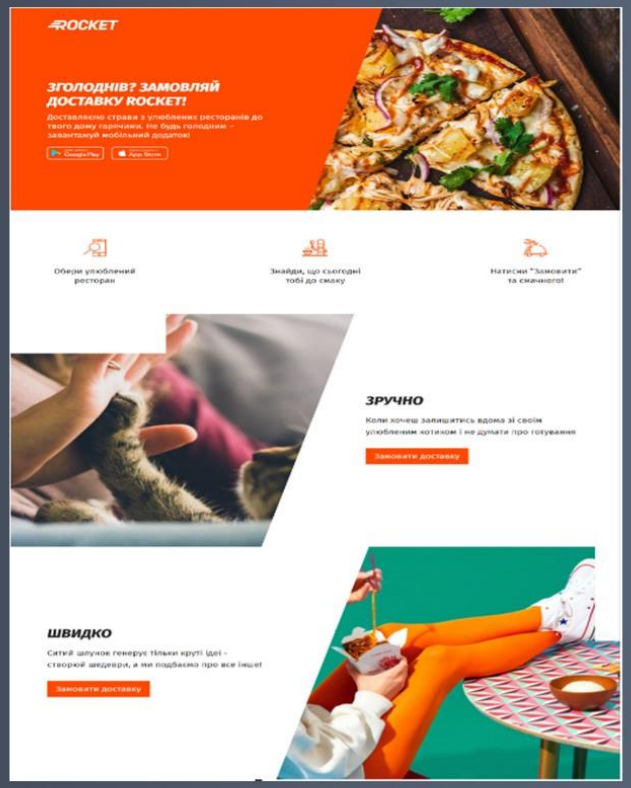
Актуальність теми



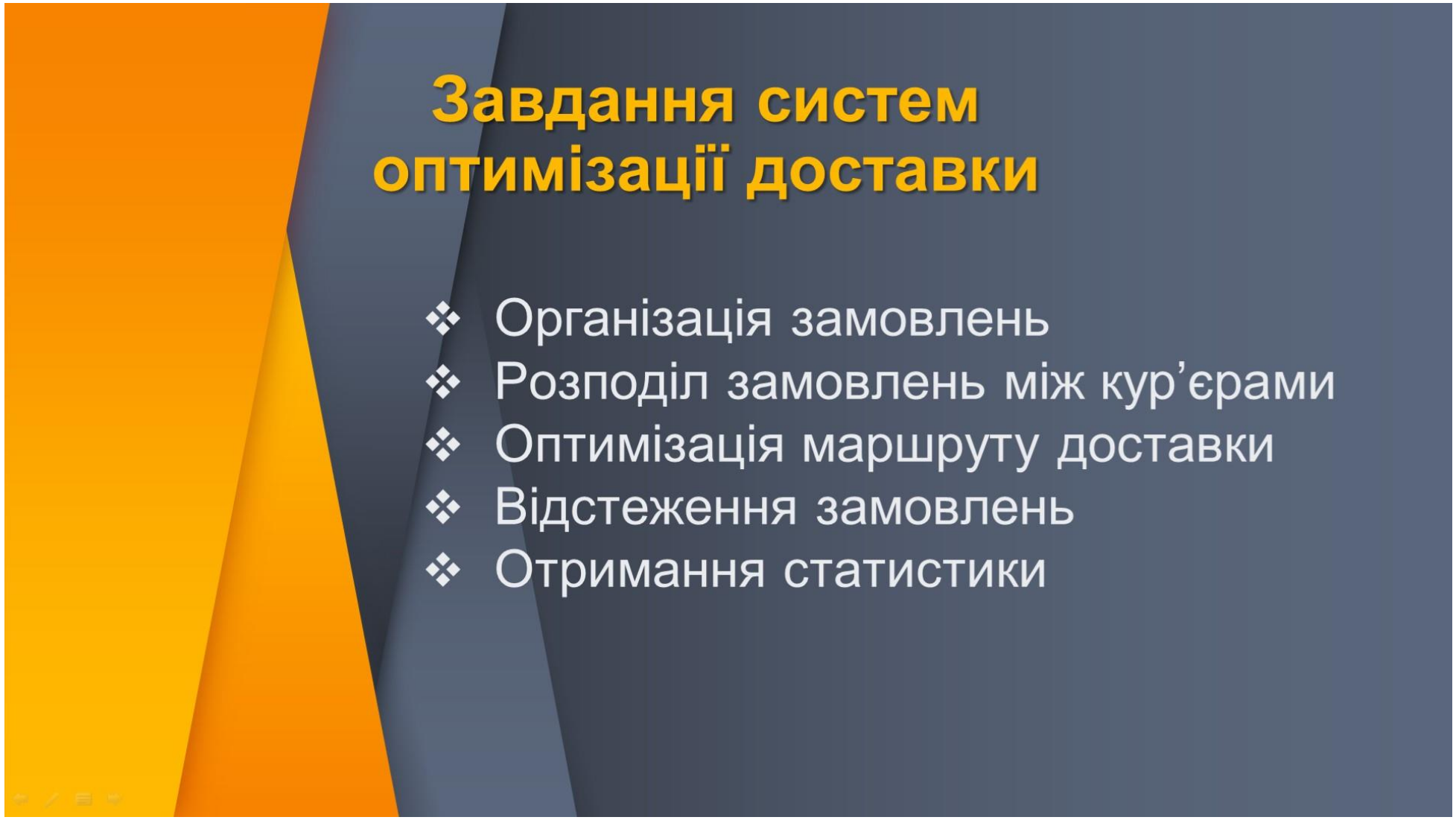
Огляд аналогів сервісів доставки



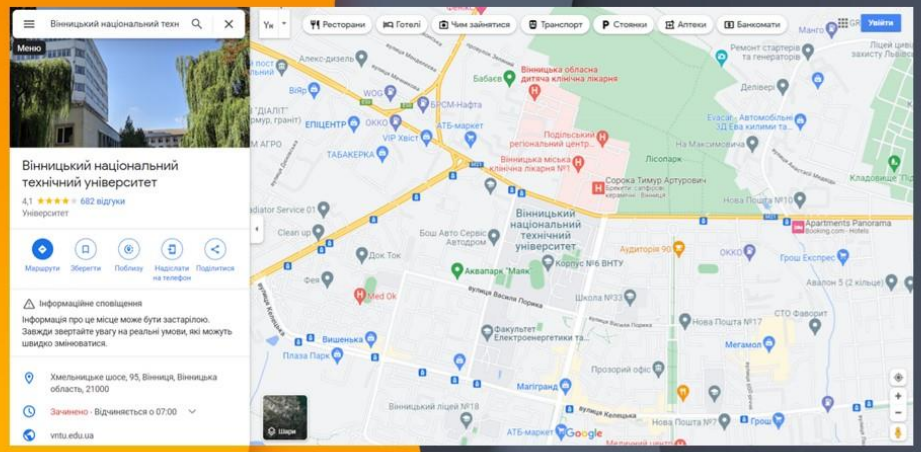
Доставка “Glovo”



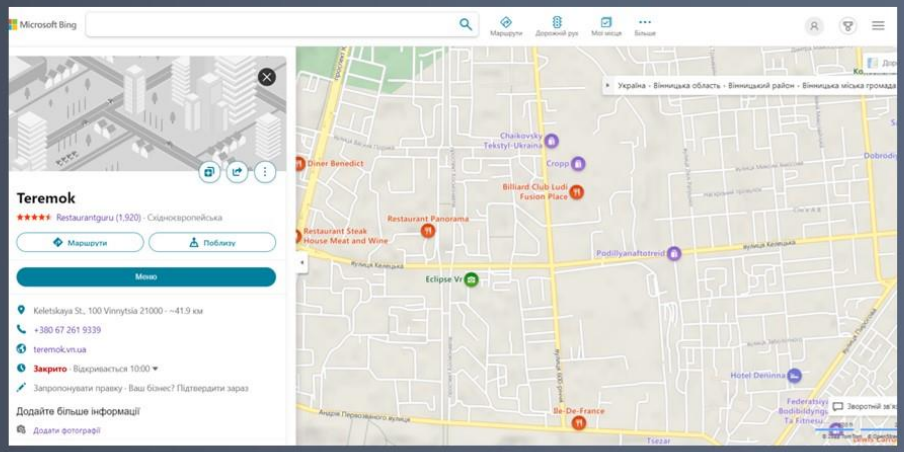
Доставка “Raketa”


Завдання систем оптимізації доставки
Організація замовлень
Розподіл замовлень між кур’єрами
Оптимізація маршруту доставки
Відстеження замовлень
Отримання статистики

Огляд картографічних систем

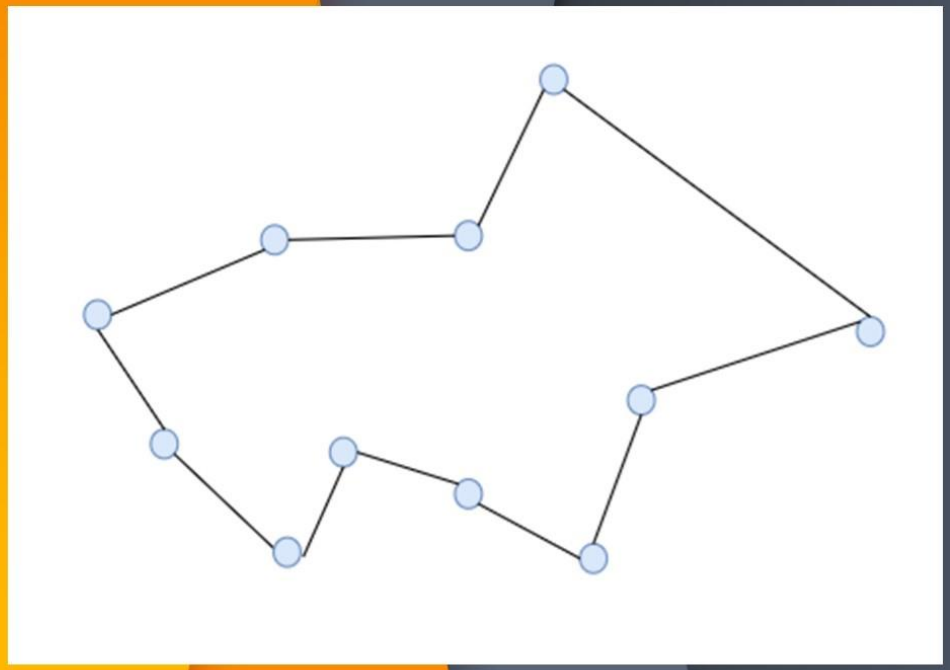


Google Maps



Bing Maps

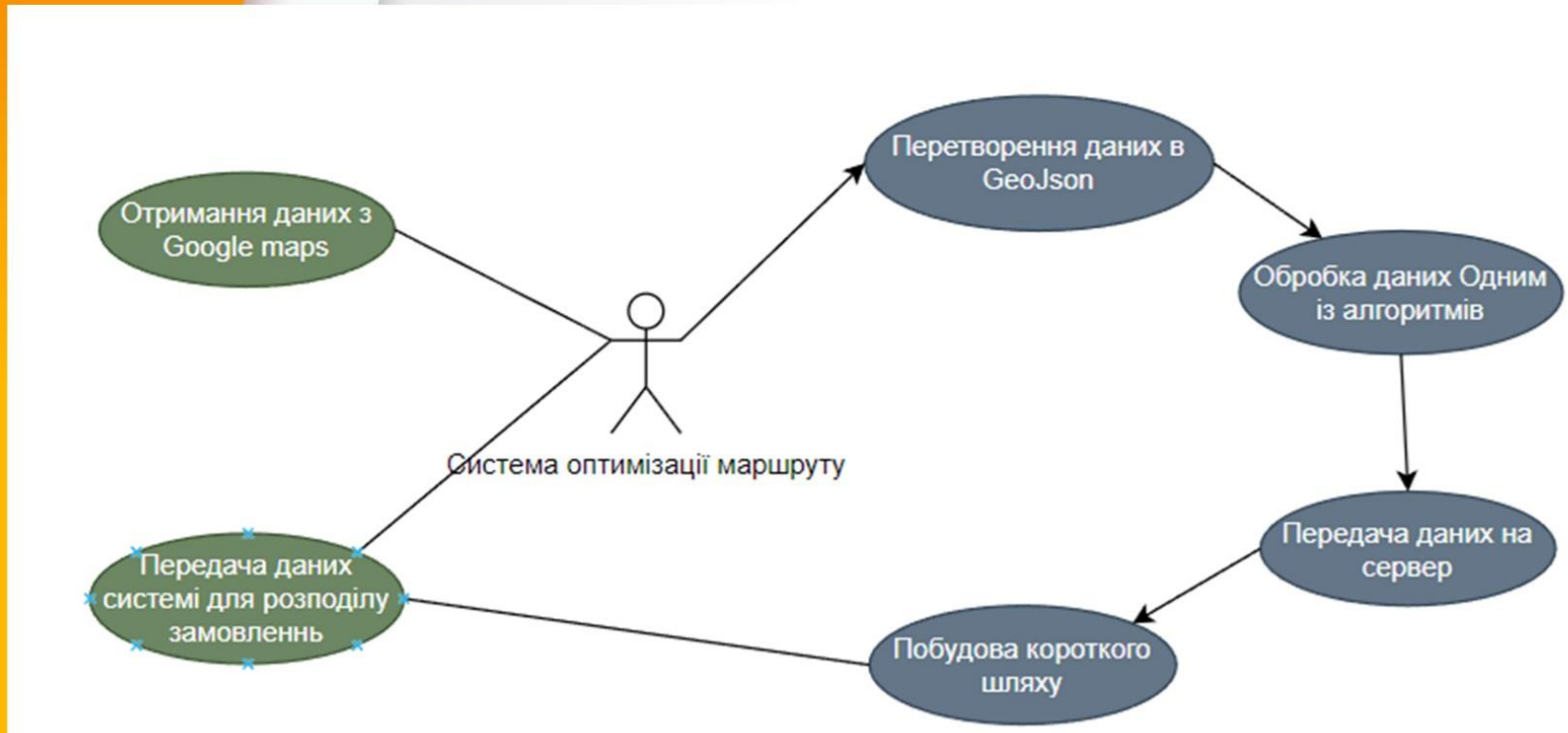
Огляд проблеми оптимізації маршруту доставки



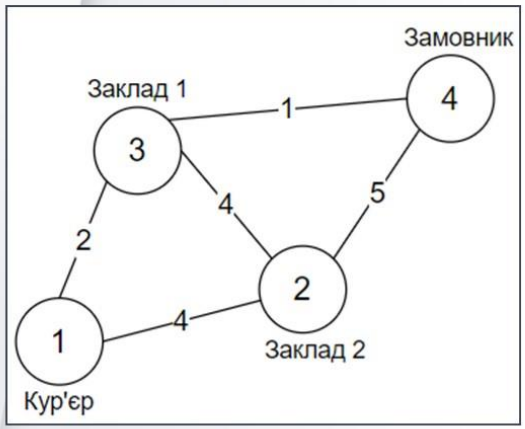
Візуалізація прокладення маршруту через точки

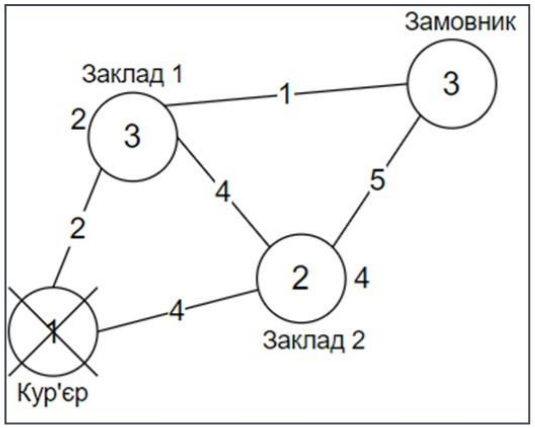
Задача комівояжера

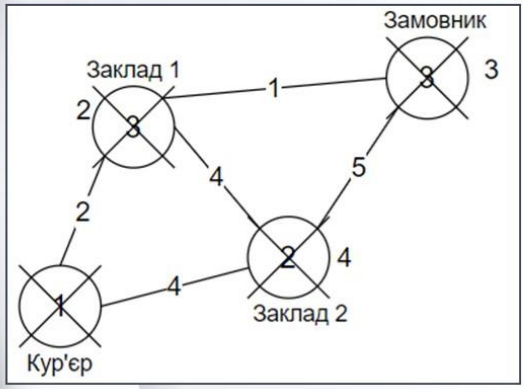
Представлення UML діаграми варіантів використання

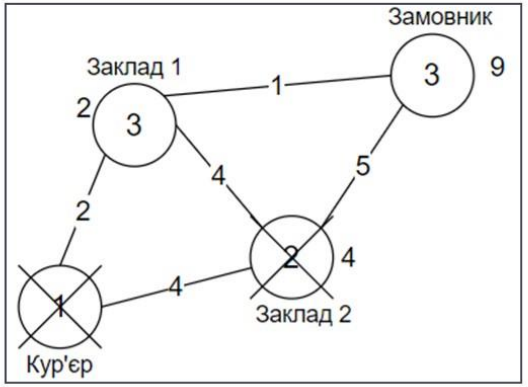


Вибір алгоритму пошуку короткого шляху

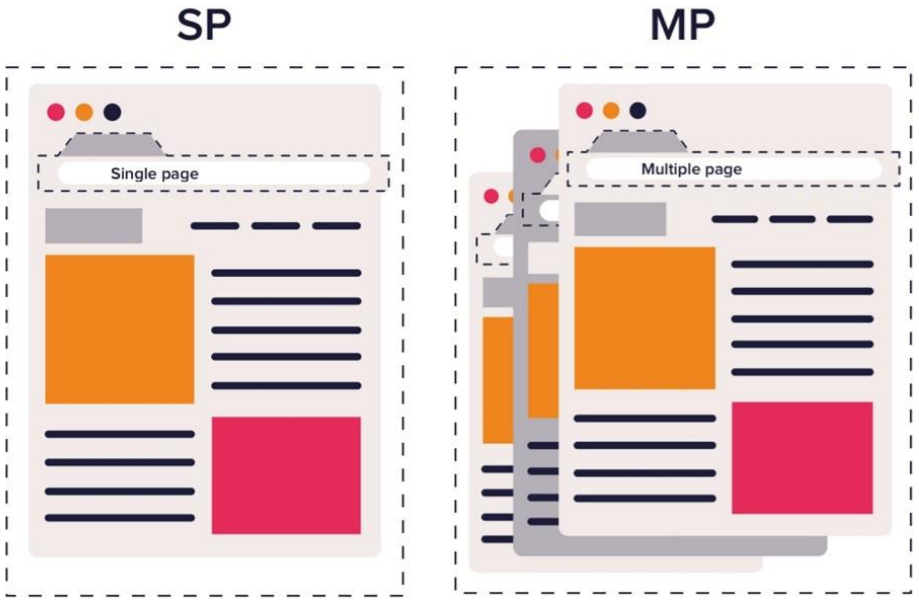








Вибір підходу для розробки системи



Вибір системи управління контентом(CMS)

Переваги Joomla:

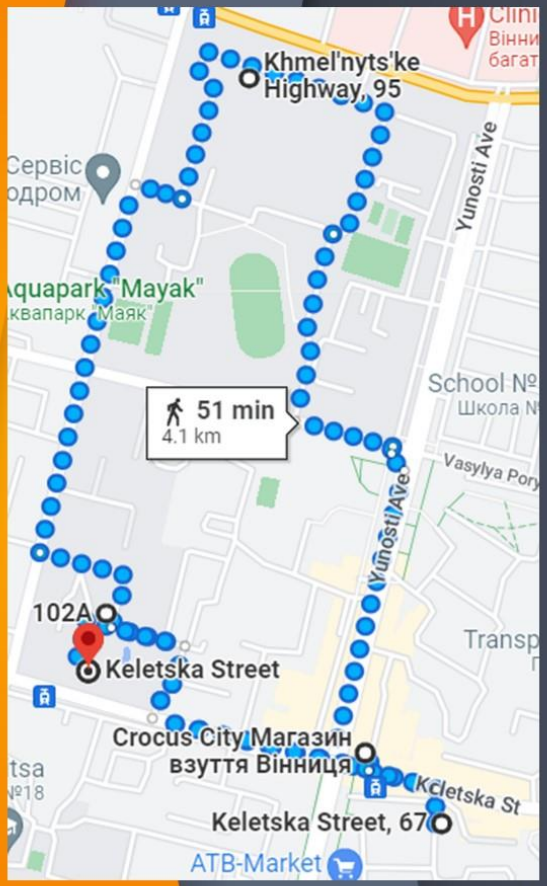
- Велика кількість сторонніх розширенньь
- Велика кількість матеріалів для навчання
- Безкоштовність розповсюдження

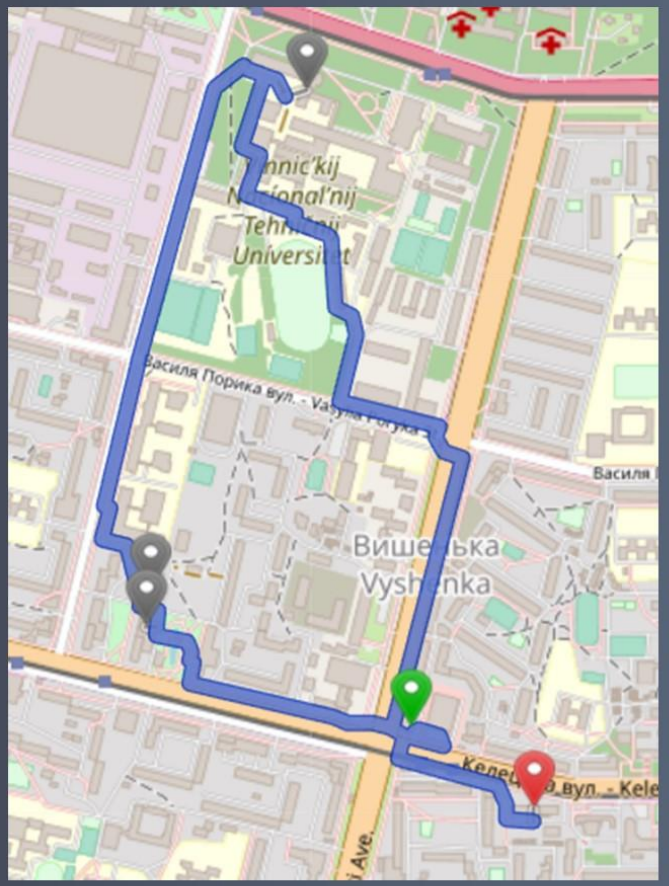
Переваги Wordpress:

- Велика кількість сторонніх розширень і шаблонів сторінок
- Швидке встановлення і оновлення системи.
- Зручніший інтерфейс серед альтернативних варіантів.
- Краща SEO оптимізація серед альтернативних варіантів.

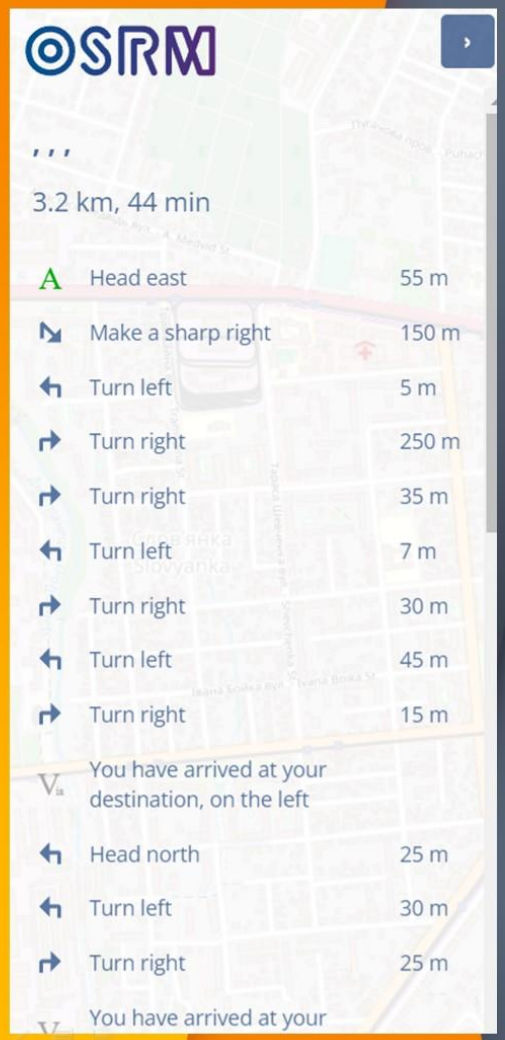
Порівняння маршрутів

Google Maps





Оптимізований маршрут



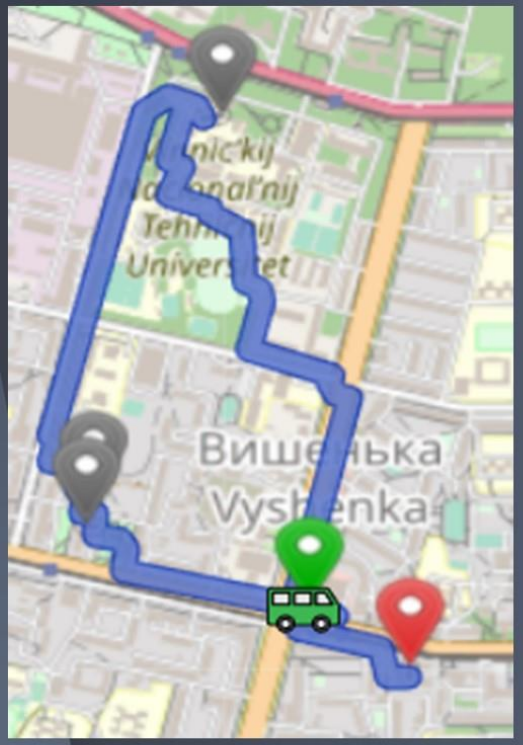
Додавання навігатора

Результати роботи:

Початковий маршрут
Протяжність – 4.1км
Орієнтований час доставки 51 хвилина

Оптимізований маршрут
Протяжність – 3.2км
Орієнтований час доставки 44 хвилини

Вигляд функції відстеження кур’єра



Висновки

В магістерській роботі було проаналізовано ринок альтернативних систем доставок. Проведено порівняльний аналіз інснуючих систем доставки та обрано критерії для побудови власної системи. Розглянуто можливі технології для реалізації відстежування замовлення в реальному часі.

Під час виконання роботи було проаналізовано методи вирішення задачі з пошуку найкоротшого шляху та обробки даних отриманих з картокрафічних сервісів

В результаті виконання роботи було розроблено підсистему з пошуку найкоротшого шляху доставки замовлення, орієнтуючись на дані отримані в реальному часі