

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проєкту (роботи)

бакалавр

(ступінь вищої освіти)

на тему ПРОГРАМНА РЕАЛІЗАЦІЯ ІНФОРМАЦІЙНОГО
РЕСУРСУ СЛУЖБИ ТАКСІ
SOFTWARE IMPLEMENTATION OF THE TAXI
SERVICE INFORMATION RESOURCE

Виконала: студентка 4 курсу, групи КНТ-210сп

Спеціальності 122 Комп'ютерні науки

(код і найменування спеціальності)

Освітня програма (спеціалізація)

Комп'ютерні науки

ВАКУЛЕНКО В.В.

(ПРИЗВИЩЕ та ініціали)

Керівник КОЦУР М.І.

(ПРИЗВИЩЕ та ініціали)

Рецензент СКРУПСЬКИЙ С.Ю.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет КНТ
Кафедра програмних засобів
Ступінь вищої освіти бакалавр
Спеціальність 122 Комп'ютерні науки
(код і найменування)
Освітня програма (спеціалізація) Комп'ютерні науки
(назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
« » 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

"ВАКУЛЕНКО Валерії Віталіївни

(ПРИЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Програмна реалізація інформаційного ресурсу служби таксі. Software Implementation of the Taxi Service Information Resource

керівник проєкту (роботи) к.т.н., доцент, КОЦУР Михайло Ігорович,
(науковий ступінь, вчене звання, ПРИЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від " 04 " квітня 2023 року № 87

2. Строк подання студентом проєкту (роботи) 01 червня 2023 року
3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз предметної області. 2. Матеріали і методи. 3. Опис програми. 4. Експлуатація, тестування та експериментальне дослідження програми.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, кількість слайдів, плакатів) Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРИЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	КОЦУР М.І., доцент		
Нормоконтроль	КАМІНСЬКА Ж.К., асистент		

7. Дата видачі завдання «04» квітня 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Вибір мови програмування та інших технологій розробки.	2 тиждень	Розділ 2
4	Розробка структури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-4 тижні	Розділи 3, 4
6	Тестування та експериментальне дослідження програмного забезпечення.	5 тиждень	Розділ 4
7	Оформлення пояснювальної записки та документів до неї.	6 тиждень	Додатки
8	Нормоконтроль та рецензування.	7 тиждень	
9	Захист роботи.	8 тиждень	

Студент(ка)

_____ Валерія ВАКУЛЕНКО
(підпис) (Ім'я ПРИЗВИЩЕ)

Керівник проєкту (роботи)

_____ Михайло КОЦУР
(підпис) (Ім'я ПРИЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи бакалавра:
91 с., 35 рис., 3 табл., 4 дод., 23 джерел.

ІНФОРМАЦІЙНИЙ РЕСУРС, ТАКСІ, ЗАМОВЛЕННЯ АВТІВКИ,
АВТОМОБІЛІ, ТРАНСПОРТ, ВЕБЗАСТОСУНОК, МОВА
ПРОГРАМУВАННЯ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Об'єкт дослідження – процес програмної реалізації інформаційного ресурсу служби таксі.

Предмет дослідження – програмні засоби для реалізації інформаційного ресурсу служби таксі.

Метою роботи є програмна реалізація інформаційного ресурсу служби таксі у вигляді кросплатформного вебзастосунку.

Матеріали, методи та технічні засоби: мова програмування PHP, система керування базами даних MySQL та для спрощення розробки: система керування контентом WordPress.

Результати. Розроблено інформаційний ресурс служби таксі.

Практична цінність роботи полягає у програмній реалізації інформаційного ресурсу служби таксі у вигляді кросплатформного вебзастосунку.

Висновки. Реалізовано інформаційного ресурсу служби таксі у вигляді кросплатформного вебзастосунку.

Галузь використання – пасажери, які віддають перевагу комфорту.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 91 pages, 35 figures, 3 tables, 4 appendixes, 22 sources.

INFORMATION RESOURCE, TAXI, CAR ORDERING, CARS, TRANSPORT, WEB APPLICATION, PROGRAMMING LANGUAGE, SOFTWARE

The object of research is the process of software implementation of the information resource of the taxi service.

The subject of the research is software tools for implementing the information resource of the taxi service.

The aim of the work is to programmatically implement the information resource of the taxi service in the form of a cross-platform web application.

Materials, methods and technical tools: PHP programming language, MySQL database management system to simplify development: WordPress content management system.

Result I. An information resource of the taxi service has been developed.

The practical value of the work lies in the software implementation of the information resource of the taxi service in the form of a cross-platform web application.

Conclusions. Implemented an information resource of the taxi service in the form of a cross-platform web application.

The area of use is passengers who prefer comfort.

ЗМІСТ

	С.
Перелік скорочень та умовних позначок	8
Вступ.....	9
1 Аналіз предметної області	13
1.1 Аналіз важливості галузі транспортних пасажирських перевезень	13
1.2 Популярність та необхідність застосунків для організації пасажирських перевезень.....	17
1.3 Порівняння існуючих аналогів.....	19
1.3.1 Застосунок «Uber».....	22
1.3.2 Сервіс «Bolt».....	25
1.3.3 Сервіс «Uklon».....	27
1.3.4 Порівняння існуючих аналогів	29
1.4 Постановка завдання	30
1.4.1 Функціональність системи	31
1.4.2 Вимоги до інтерфейсу.....	31
1.4.3 Вимоги до продуктивності	31
1.4.4 Вимоги до безпеки	32
1.4.5 Експлуатаційні вимоги	32
1.4.6 Не функціональні вимоги до продукту	32
1.5 Висновки за розділом 1	33
2 Матеріали і методи	34
2.1 Вибір мови програмування	34
2.2 Вибір та обґрунтування CMS	38
2.3 Структурна схема розробки.....	43
2.4 Висновки за розділом 2	44
3 Основні рішення щодо реалізації компонентів системи	45
3.1 Проектування моделей системи	45
3.2 Проектування інтерфейсу	54
3.3 Висновки за розділом 3	54

4	Експлуатація, тестування та експериментальне дослідження програми.....	55
4.1	Призначення й умови застосування програми	55
4.2	Вимоги до інтерфейсу	55
4.3	Умови виконання програми.....	55
4.4	Розгортання застосунку.....	56
4.4.1	Звернення до програми.....	58
4.4.2	Вхідні й вихідні дані	58
4.5	Методика та результати тестування	58
	Висновки	61
	Перелік джерел посилання	62
	Додаток А Технічне завдання	65
	Додаток Б Опис програми	69
	Додаток В Текст програми	73
	Додаток Д Слайди презентації	85

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

ОС – операційна система;

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

GUI – графічний інтерфейс користувача;

MVC – model-view-controller;

SEO – пошукова оптимізація;

UML – Unified Modeling Language.

ВСТУП

В останні роки транспортна галузь зазнала значних перетворень з появою застосунків для виклику таксі. Такі програми, як Uber, Lyft та Grab, зробили революцію в тому, як люди викликають таксі та користуються ними [1], [2]. Популярність застосунків для виклику таксі можна пояснити поєднанням факторів, включаючи зручність, прийнятність за ціною, доступність та інновації. У цій статті ми розглянемо ці фактори і з'ясуємо, чому застосунки для виклику таксі стали настільки популярними [1], [2].

Зручність і простота використання. Однією з основних причин популярності застосунків для виклику таксі є зручність, яку вони пропонують. Кількома натисканнями на свої смартфони користувачі можуть замовити поїздку, і водій прибуде до них протягом декількох хвилин. Відпадає необхідність зупиняти таксі на вулиці або чекати на спеціально відведених стоянках. Ця зручність особливо вигідна в переповнених або незнайомих районах, де знайти традиційне таксі може бути складно [1], [2].

Крім того, програми таксі надають користувачам оновлену інформацію про місцезнаходження водія в режимі реального часу, передбачуваний час прибуття та розрахунок вартості проїзду. Така прозорість дає пасажирів відчуття контролю та впевненості, сприяючи їх загальному задоволенню [1], [2].

Розширений доступ. Застосунки для виклику таксі демократизували транспорт, зробивши його доступним для більш широкого кола людей. На відміну від традиційних служб таксі, які часто вимагають від пасажирів телефонувати та спілкуватися усно, програми таксі пропонують зручний інтерфейс, до якого може отримати доступ кожен, хто має смартфон. Ця доступність була особливо вигідною для людей з обмеженою рухливістю або обмеженими можливостями, яким може бути важко користуватися традиційними таксі. Крім того, Програми для виклику таксі зазвичай

пропонують багатомовну підтримку, що робить їх більш доступними для мандрівників та туристів [1], [2].

Економічна ефективність та прозорість. Доступність за ціною є ключовим фактором, що сприяє популярності застосунків для виклику таксі. Ці платформи часто пропонують конкурентоспроможні ціни та гнучку структуру тарифів, включаючи такі варіанти, як спільні поїздки або послуги об'єднання, які допомагають зменшити витрати пасажирів. Крім того, застосунки для таксі надають попередню оцінку вартості проїзду, усуваючи несподіванки або розбіжності, зазвичай пов'язані з традиційними таксі. Така прозорість підвищує довіру і гарантує, що пасажирів будуть обізнані про витрати до підтвердження своїх поїздок [1], [2].

Рейтинг водія та характеристики безпеки. Програми таксі надають пріоритет безпеці, впроваджуючи такі функції, як системи оцінки водіїв, GPS-відстеження та цифрові платежі. Пасажирів мають можливість оцінювати своїх водіїв і залишати відгуки, що стимулює водіїв підтримувати високий рівень обслуговування. Крім того, GPS-трекінг дозволяє як пасажирів, так і їхнім близьким стежити за поїздкою в режимі реального часу, підвищуючи безпеку і душевний спокій [1], [2].

Більш того, Цифрова платіжна система, що використовується в застосунках таксі, усуває необхідність в операціях з готівкою, знижуючи ризик пограбування або крадіжки. Цей аспект у поєднанні з процесами перевірки, реалізованими програмами, створює більш безпечне середовище як для пасажирів, так і для водіїв [1], [2].

Технологічні досягнення та інновації. Програми таксі використовували технологічні досягнення, щоб забезпечити більш ефективне та приємне транспортування. Такі функції, як інтелектуальні алгоритми маршрутизації, допомагають водіям орієнтуватися на перевантажених дорогах і вибирати найбільш оптимальні маршрути, скорочуючи час у дорозі. Алгоритми машинного навчання та штучний інтелект дозволяють програмам

персоналізувати користувальницький інтерфейс, запам'ятовуючи бажані місця, маршрути та способи оплати [1], [2].

Крім того, Програми для виклику таксі еволюціонували і включають застосункові послуги, які виходять за рамки традиційних поїздок на таксі. Зараз деякі платформи пропонують доставку їжі, посилок та обмін велосипедами чи скутерами, розширюючи сферу своєї діяльності та надаючи користувачам більше застосункових послуг [1], [2].

Руйнування традиційної індустрії таксі. Зростання застосунків для таксі знищило традиційну індустрію таксі, яку часто критикували за неефективність, ненадійний сервіс та відсутність інновацій. Впровадження застосунків для таксі змусило традиційні служби таксі адаптуватися та вдосконалювати свою діяльність, щоб залишатися конкурентоспроможними. Отже, ця конкуренція стимулювала інновації та покращувала загальну якість послуг таксі в усіх напрямках [1], [2].

На закінчення, застосунки для таксі завоювали величезну популярність завдяки своїй зручності, прийнятності за ціною, доступності, функціям безпеки, технологічним досягненням і руйнуванням, які вони викликали в традиційній індустрії таксі. Ці програми змінили те, як люди сприймають та користуються послугами таксі, роблячи транспортування більш плавним та приємним. Оскільки технології продовжують розвиватися, цілком ймовірно, що застосунки для таксі будуть розвиватися і далі, пропонуючи користувачам по всьому світу ще більш зручний і всебічний сервіс [1], [2].

Отже актуальною задачею залишається програмна реалізація інформаційного ресурсу служби таксі.

Для виконання такої задачі необхідно виконати наступні завдання:

- проаналізувати сучасний стан області транспортних застосунків;
- дослідити існуючі реалізації застосунків для замовлення перевезень;
- з дослідження виділити переваги та недоліки таких рішень;
- сформулювати список вимог до нового рішення;

- визначивши засоби реалізації, виконати проєктування нового застосунку;
- виконати розробку та тестування застосунку на основі проєктування;
- скласти необхідну програмну документацію за результатами усіх етапів: аналітика вимог, вибір технічних засобів реалізації, проєктування реалізації, розробка та тестування.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

В першому розділі основну увагу буде приділено дослідженню проблемної області транспортних пасажирських перевезень. Спочатку буде розглянуто сучасний стан проблемної області. Після цього буде досліджено можливості часткової автоматизації процесу виконання замовлення пасажирського транспортного перевезення, через використання програмних засобів. На основі попереднього дослідження буде виділено основних лідерів галузі та детально проаналізовано їх переваги та недоліки.

З рештою, це все дозволить виконати формування переліку функціональних та нефункціональних вимог до майбутнього нового рішення, що і є головною метою цього розділу.

1.1 Аналіз важливості галузі транспортних пасажирських перевезень

Галузь пасажирських перевезень відіграє ключову роль у суспільстві, надаючи життєво важливі послуги, які об'єднують людей, громади та підприємства. Від місцевих систем громадського транспорту до варіантів подорожей на великі відстані, ця галузь служить джерелом мобільності та взаємозв'язку. У цій статті ми розглянемо важливість галузі пасажирських перевезень з різних точок зору, включаючи економічні, соціальні, екологічні та особисті аспекти [1]–[5].

Економічний вплив. Галузь пасажирських перевезень має значний економічний вплив як на місцевому, так і на глобальному рівнях. Це сприяє створенню робочих місць і економічному зростанню, надаючи можливості працевлаштування в різних секторах, таких як агентства громадського транспорту, компанії таксі і райдшерінга, авіакомпанії, залізничні оператори та автобусні служби [3]–[5]. Ці посади варіюються від водіїв та пілотів до

представників служби обслуговування клієнтів, обслуговуючого персоналу та адміністративного персоналу [3]–[5].

Більш того, галузь генерує дохід за рахунок продажу квитків, тарифів і транспортних зборів. Цей дохід не тільки підтримує транспортні компанії, але і підживлює економіку за рахунок грошового обігу. Галузь пасажирських перевезень також стимулює економічну активність в інших секторах, таких як туризм, гостинність та роздрібна торгівля, оскільки люди покладаються на транспортні послуги для доступу до цих галузей [3]–[5].

Крім того, ефективні та надійні транспортні системи відіграють вирішальну роль у залученні бізнесу та інвестицій. Галузі промисловості потребують ефективної логістичної мережі для транспортування товарів та послуг, а добре розвинена транспортна інфраструктура сприяє загальній конкурентоспроможності регіону чи країни [3]–[5].

Соціальні зв'язки та доступність. Пасажирські перевезення необхідні для зміцнення соціальних зв'язків і забезпечення доступності для всіх людей, незалежно від їх соціально-економічного статусу або фізичних можливостей. Надійні транспортні послуги дозволяють людям отримати доступ до можливостей працевлаштування, навчальних закладів, медичних закладів та місць відпочинку. Вони дозволяють людям спілкуватися з друзями та родиною, відвідувати громадські збори та брати участь у соціальних заходах [1]–[5].

Громадський транспорт, зокрема, відіграє найважливішу роль у забезпеченні рівного доступу до транспорту для всіх. Вона надає доступні та зручні варіанти для людей, які, можливо, не мають доступу до приватних транспортних засобів, сприяючи соціальній інтеграції та скорочуючи соціально-економічну нерівність. Громадський транспорт також відіграє життєво важливу роль у зменшенні заторів, забруднення навколишнього середовища та загального впливу транспорту на навколишнє середовище [1]–[5].

Екологічна стійкість. Не можна ігнорувати вплив галузі пасажирських перевезень на навколишнє середовище. Цей сектор несе відповідальність за значну частку глобальних викидів парникових газів, головним чином в результаті автомобільних і повітряних перевезень. Однак галузь також має потенціал для сприяння екологічній стійкості за допомогою різних заходів [3]–[5].

Зусилля щодо скорочення викидів вуглекислого газу та сприяння сталому транспорту набирають обертів. Багато транспортних компаній переходять на електромобілі або вивчають альтернативні види палива, щоб зменшити свій вплив на навколишнє середовище. Уряди та організації також інвестують в інфраструктуру громадського транспорту, щоб заохотити людей переходити з особистих транспортних засобів на більш екологічні види транспорту [3]–[5].

Крім того, набирають популярність варіанти спільного пересування, такі як спільне використання поїздок і велосипедів, що призводить до скорочення кількості індивідуальних транспортних засобів на дорогах. Ця тенденція не тільки зменшує затори на дорогах, але й знижує викиди шкідливих речовин і сприяє більш ефективному використанню транспортних ресурсів [3]–[5].

Особиста зручність і якість життя. Пасажирські перевезення значно підвищують особисту зручність і покращують загальну якість життя людей. Це дає людям свободу та гнучкість подорожувати куди і коли їм потрібно, зменшуючи обмеження, пов'язані з відстанню та часом [3]–[5].

Ефективні транспортні системи, такі як Високошвидкісні залізничні мережі або аеропорти з хорошим сполученням, дозволяють людям швидко та комфортно дістатися до місця призначення, полегшуючи ділові поїздки, відпустки та сімейні візити. Аналогічним чином, надійні системи громадського транспорту в міських районах полегшують необхідність володіння автомобілем, знижуючи фінансовий тягар і стрес, пов'язані з паркуванням, дорожнім рухом і технічним обслуговуванням [3]–[5].

Більше того, такі варіанти транспортування, як програми для спільного користування поїздками, змінили спосіб доступу людей до транспортних послуг. Ці платформи забезпечують зручність, доступність і простоту використання, дозволяючи користувачам замовляти поїздки кількома натисканнями на свої смартфони. Такі інновації зробили транспорт більш доступним і персоналізованим, підвищивши зручність і комфорт для окремих осіб [3]–[5].

Реагування на надзвичайні ситуації та ліквідація наслідків стихійних лих. Під час надзвичайних ситуацій або стихійних лих галузь пасажирських перевезень відіграє вирішальну роль у сприянні своєчасній евакуації, реагування на надзвичайні ситуації та ліквідації наслідків стихійних лих. Транспортні мережі, включаючи автобуси, поїзди та авіакомпанії, мають вирішальне значення для транспортування постраждалих людей у безпечніші райони та надання допомоги під час криз [3]–[5].

Під час стихійних лих, таких як урагани, землетруси або повені, транспортні послуги необхідні для доставки запасів, матеріалів надзвичайної допомоги та медичної допомоги в постраждалі райони. Здатність швидко мобілізувати ресурси та евакуювати людей у таких ситуаціях може врятувати життя та пом'якшити наслідки стихійних лих [3]–[5].

Галузь пасажирських перевезень має величезне значення в економічному, соціальному, екологічному та особистому контекстах. Це стимулює економічне зростання, об'єднує громади, забезпечує доступність та підвищує особисту зручність. Впроваджуючи стійкі методи та використовуючи технологічні досягнення, галузь може ще більше посилити свій вплив на суспільство, сприяючи економічному розвитку, сприяючи соціальній зв'язності та сприяючи більш стійкому майбутньому [3]–[5].

1.2 Популярність та необхідність застосунків для організації пасажирських перевезень

Поява застосунків для виклику таксі зробила революцію в тому, як люди отримують доступ до транспортних послуг і користуються ними. Такі платформи, як Uber, Lyft та Grab, швидко набули популярності та перетворили традиційну індустрію таксі. У цій статті ми розглянемо причини швидкого зростання та значущості застосунків для таксі, досліджуючи їх вплив на зручність, доступність, ефективність та загальний транспортний ландшафт [6]–[8].

Руйнування традиційної індустрії таксі. Одним з основних факторів, що сприяють популярності та важливості застосунків для таксі, є їх руйнівний характер. Традиційні служби таксі часто стикалися з критикою через такі проблеми, як недоступність у години пік, невідповідні ціни та погане обслуговування клієнтів. Програми таксі виявили ці больові точки та запропонували інноваційні рішення для їх усунення, що призвело до значної зміни переваг споживачів [6]–[8].

Впроваджуючи такі функції, як відстеження місцезнаходження в режимі реального часу, попереднє ціноутворення та системи оцінки водіїв, програми таксі забезпечили прозорість та підзвітність галузі. Цей збій змусив традиційні служби таксі адаптуватися та вдосконалювати свою діяльність, щоб залишатися конкурентоспроможними, що в кінцевому підсумку підвищило загальні стандарти галузі [6]–[8].

Зручність і простота використання. Додатки для виклику таксі зробили замовлення поїздки більш зручним, ніж будь-коли раніше. Кількома натисканнями на смартфон користувачі можуть замовити поїздку, і водій прибуде до них протягом декількох хвилин. Усунення необхідності фізично викликати таксі або чекати на спеціально відведених стоянках зробило транспорт більш доступним і ефективним [6]–[8].

Крім того, Програми таксі надають користувачам важливу інформацію, таку як приблизний час прибуття водія, дані про водія та попередній розрахунок вартості проїзду. Цей рівень зручності та прозорості підвищує загальний досвід користувачів та забезпечує відчуття контролю та безпеки [6]–[8].

Доступність та інклюзивність. Програми для виклику таксі значно покращили доступність, зробивши транспортні послуги доступними для широкого кола осіб. На відміну від традиційних служб таксі, які вимагали телефонних дзвінків або дзвінків на вулицю, програми для виклику таксі пропонують зручний інтерфейс, до якого може отримати доступ кожен, хто має смартфон. Така доступність була особливо вигідною для людей з обмеженою рухливістю, обмеженими можливостями або мовними бар'єрами [6]–[8].

Крім того, програми таксі часто надають багатомовну підтримку, дозволяючи мандрівникам та туристам зручно користуватися послугами в незнайомих місцях. Інклюзивність та доступність, пропоновані програмами таксі, значно покращили транспортні можливості для людей, які раніше стикалися з труднощами при доступі до традиційних таксі [6]–[8].

Технологічні досягнення та інновації. Успіх і важливість застосунків для таксі можна пояснити їх ефективним використанням технологічних досягнень і безперервними інноваціями. Ці платформи використовують GPS-відстеження, дані в режимі реального часу та інтелектуальні алгоритми маршрутизації для оптимізації ефективності своїх послуг. Аналізуючи структуру дорожнього руху та пропонуючи найбільш оптимальні маршрути, програми таксі заощаджують час як водіїв, так і пасажирів, що призводить до швидших та ефективніших поїздок [6]–[8].

Крім того, Програми таксі використовують алгоритми машинного навчання та штучний інтелект для персоналізації користувацького досвіду. Ці технології запам'ятовують переваги користувачів, такі як бажані місця,

маршрути та способи оплати, забезпечуючи більш індивідуальний та зручний інтерфейс [6]–[8].

Крім того, Програми для виклику таксі вийшли за рамки традиційних поїздок на таксі, включивши застосункові послуги. Деякі платформи тепер пропонують доставку їжі, посилок і навіть можливість спільного використання велосипедів або скутерів. Така диверсифікація та постійні інновації допомогли застосункам таксі залишатися актуальними та цінними у мінливому транспортному ландшафті [6]–[8].

Безпека та довіра. Програми таксі надають пріоритет безпеці, створюючи функції, які підвищують довіру пасажирів. Системи оцінки водіїв дозволяють пасажиром залишати відгуки та оцінювати свій досвід, сприяючи підзвітності та стимулюючи водіїв підтримувати високу якість обслуговування. Ці рейтингові системи також дозволяють користувачам приймати обґрунтовані рішення при виборі водія [6]–[8].

Крім того, програми таксі реалізують функції безпеки, такі як GPS-відстеження в режимі реального часу, цифрові платежі та цілодобова підтримка клієнтів 7 днів на тиждень. Можливість відстежувати поїздку в режимі реального часу забезпечує пасажиром відчуття безпеки, в той час як цифрові платежі знижують ризик, пов'язаний з перевезенням готівки. Суворі процедури перевірки водіїв, що проводяться застосунками таксі [6]–[8].

1.3 Порівняння існуючих аналогів

Популярні програми для таксі змінили спосіб доступу людей до транспортних послуг. Ці платформи пропонують зручність, прозорість та доступність, але вони також мають свою частку переваг та недоліків. У цій статті ми розглянемо основні плюси і мінуси популярних застосунків для виклику таксі, щоб провести всебічний аналіз їх впливу на користувачів і транспортну галузь.

Розглянемо загальні переваги таких застосунків.

Зручність і простота використання. Однією з головних переваг застосунків для виклику таксі є зручність, яку вони пропонують. Кількома натисканнями на смартфон користувачі можуть замовити поїздку, і водій прибуде до них протягом декількох хвилин. Це позбавляє від необхідності фізично підкликати таксі або чекати на спеціально відведених стоянках, роблячи транспорт більш доступним і ефективним.

Прозоре та попереднє ціноутворення. Програми таксі надають попередню оцінку цін на основі таких факторів, як відстань та час, забезпечуючи прозорість для пасажирів. Користувачі дізнаються передбачуваний тариф до підтвердження своєї поїздки, що дозволяє уникнути несподіванок або суперечок про тарифи в кінці поїздки. Така прозорість зміцнює довіру та покращує загальний досвід користувачів.

Функції безпеки. Програми таксі надають пріоритет безпеці як пасажирів, так і водіїв. Вони реалізують функції безпеки, такі як системи оцінки водіїв, GPS-відстеження та цифрові платежі. Пасажири можуть оцінювати водіїв і залишати відгуки, що стимулює водіїв підтримувати високу якість обслуговування. GPS-відстеження дозволяє пасажирам та їхнім близьким стежити за поїздкою в режимі реального часу, додаючи застосунковий рівень безпеки. Цифрові платежі знижують ризик крадіжки та забезпечують безпечніший спосіб здійснення транзакцій порівняно з передачею готівки.

Підвищена доступність. Програми для виклику таксі значно покращили доступність, зробивши транспортні послуги доступними для широкого кола осіб на відміну від традиційних служб таксі, які часто вимагають телефонних дзвінків або дзвінків на вулицю, до програм таксі може отримати доступ кожен, хто має смартфон. Така доступність вигідна людям з обмеженою рухливістю, обмеженими можливостями або мовними бар'єрами.

Підвищена підзвітність водіїв. Завдяки системам оцінки водіїв та зворотного зв'язку з клієнтами програми таксі сприяють підзвітності водіїв та підтримують високі стандарти обслуговування. Водії отримують стимул

забезпечувати чудове обслуговування клієнтів, що призводить до загального поліпшення якості поїздок.

Розглянемо загальні недоліки таких застосунків.

Різке підвищення цін. Одним з головних критичних зауважень до застосунків таксі є різке зростання цін, який виникає в періоди високого попиту. У години пік, святкові дні або спеціальні заходи Тарифи можуть значно підвищуватися завдяки алгоритмам динамічного ціноутворення. Різке зростання цін може застати пасажирів зненацька, що призведе до несподівано високих тарифів. Однак деякі програми пропонують сповіщення про підвищення цін, що дозволяє користувачам приймати обґрунтовані рішення.

Відсутність пільг для співробітників. Водії, які працюють на таксі, зазвичай класифікуються як незалежні підрядники, а це означає, що вони не отримують традиційних пільг для працівників, таких як медичне страхування, пенсійні плани або оплачувані відпустки. Ця відсутність пільг стала предметом суперечок, були висловлені побоювання з приводу справедливої компенсації і прав працівників.

Можливість перебоїв в обслуговуванні. Оскільки програми для таксі покладаються на мережу незалежних водіїв, існує ймовірність перебоїв в обслуговуванні в періоди високого попиту або нестачі водіїв. Це може призвести до збільшення часу очікування або недоступності поїздок, особливо у віддалених районах або в періоди підвищеної завантаженості.

Міркування безпеки. Незважаючи на те, що в застосунках таксі реалізовані функції безпеки, надходили повідомлення про інциденти, пов'язані з безпекою як пасажирів, так і водіїв. Хоча ці інциденти відносно рідкісні, зберігаються побоювання щодо ефективності перевірок біографічних даних і протоколів безпеки. Програми таксі продовжують інвестувати в заходи безпеки, щоб вирішити ці проблеми та підвищити загальну безпеку своїх платформ.

Вплив на традиційну індустрію таксі. Зростання застосунків для таксі знищило традиційну індустрію таксі, що призвело до проблем для

традиційних таксистів та компаній. Швидке зростання застосунків для таксі призвело до посилення конкуренції, змусивши традиційні служби таксі адаптуватися та знаходити способи залишатися конкурентоспроможними. Цей перехід створив проблеми для деяких водіїв таксі, яким, можливо, буде важко конкурувати з зручністю та доступністю, пропонованими програмами для виклику таксі.

Програми таксі пропонують багато переваг, таких як зручність, прозоре ціноутворення, функції безпеки та підвищена доступність. Вони змінили спосіб доступу людей до транспортних послуг і стали невід'ємною частиною міського ландшафту мобільності. Однак слід враховувати такі проблеми, як різке зростання цін, відсутність пільг для працівників, потенційні перебої в обслуговуванні, проблеми безпеки та їх вплив на традиційну індустрію таксі. Оскільки галузь продовжує розвиватися, вирішення цих проблем буде мати вирішальне значення для забезпечення балансу між інноваціями, зручністю та добробутом як пасажирів, так і водіїв.

1.3.1 Застосунок «Uber»

Історія Uber сягає 2009 року, коли її заснували Тревіс Каланік та Гарретт Кемп. Ідея Uber народилася з розчарування в традиційній індустрії таксі і бажання створити більш ефективне і зручне транспортне рішення [9], [10].

Перші роки роботи Uber. Оригінальна концепція Uber була натхненна труднощами Кемпа з пошуком таксі в сніжний вечір у Парижі. Цей досвід спонукав його уявити собі послугу, яка дозволила б людям легко забронювати поїздки за допомогою своїх смартфонів. Кемп поділився своєю ідеєю з Каланіком, і вони разом розробили UberCab, оригінальну назву компанії [9], [10].

UberCab розпочав свою діяльність як послуга розкішних чорних автомобілів у Сан-Франциско. Засновники запустили бета-версію мобільного

застосунка в травні 2010 року, що дозволяє користувачам замовляти поїздки у професійних водіїв. Послуга швидко завоювала популярність завдяки своїй зручності та високій якості обслуговування, яке вона пропонувала в порівнянні з традиційними таксі [9], [10].

Розширення та ребрендинг. У міру того, як Uber набирав обертів у Сан-Франциско, компанія поширювалася і на інші міста, як всередині Сполучених Штатів, так і за кордоном. У 2011 році Uber офіційно вилучив слово "таксі" зі своєї назви і став відомим як Uber [9], [10].

Розширення Uber зіткнулося з регуляторними перешкодами та опором традиційної індустрії таксі у багатьох містах. Тим не менш, компанія продовжувала зростати, залучаючи мільйони користувачів і розширюючи свої послуги, включаючи різні типи транспортних засобів, такі як UberX (недорогі поїздки на звичайних автомобілях) і UberPool (послуги з об'єднання автомобілів) [9], [10].

Застосунок Uber і як він працює. В основі успіху Uber лежить його зручний мобільний застосунок, доступний на пристроях iOS та Android. Додаток дозволяє користувачам запитувати поїздки, відстежувати місцезнаходження призначеного їм водія та безперешкодно здійснювати безготівкові платежі. Ось як працює застосунок Uber [9], [10].

Реєстрація: користувачі починають із завантаження програми Uber та створення облікового запису. Вони надають свою особисту інформацію, включаючи ім'я, адресу електронної пошти та платіжні реквізити [9], [10].

Запит на поїздку: користувачі відкривають застосунок і встановлюють місце, де їх заберуть. Вони також можуть ввести пункт призначення або ввести кілька зупинок, якщо це необхідно. Потім застосунок надає приблизну вартість поїздки [9], [10].

Підбір водія: як тільки запит на поїздку відправлений, застосунок підключається до найближчих водіям. Найближчий доступний водій, який прийме запит, буде призначений для того, щоб забрати пасажирів [9], [10].

Інформація про водія та час прибуття: користувачі отримують інформацію про призначеного водія, включаючи його ім'я, фотографію та дані про транспортний засіб. Вони можуть відстежувати місцезнаходження водія та Розрахунковий час прибуття (ETA) у режимі реального часу [9], [10].

Враження від поїздки: користувачі насолоджуються поїздкою, а застосунок забезпечує водієві покрокову навігацію. При необхідності пасажери можуть зв'язатися з водієм через застосунок [9], [10].

Безготівкова оплата: після прибуття в пункт призначення застосунок автоматично розраховує вартість проїзду в залежності від відстані і часу. Платежі обробляються в електронному вигляді за допомогою попередньо зареєстрованого способу оплати користувача (кредитна /дебетова картка, цифрові гаманці або Uber Cash). Поради також можна додавати через застосунок [9], [10].

Рейтинг та відгуки: користувачі мають можливість оцінити свого водія та залишити відгук про свій досвід. Такий зворотний зв'язок допомагає підтримувати якість обслуговування і забезпечує підзвітність водіїв [9], [10].

Бізнес-модель Uber. Uber працює як технологічна платформа, яка з'єднує пасажирів з водіями. Компанія не володіє транспортними засобами і не наймає водіїв безпосередньо. Натомість він надає платформу, яка дозволяє незалежним водіям пропонувати свої послуги пасажиром. Uber стягує комісію або плату за обслуговування з тарифу за кожну завершену поїздку [9], [10].

Uber розширив свої послуги за рамки традиційного райдшерингу, включаючи доставку їжі за допомогою Uber Eats і Вантажні перевезення за допомогою Uber Freight. Ці розширення дозволили компанії диверсифікувати свої пропозиції та охопити ширшу клієнтську базу [9], [10].

Приклад інтерфейсу Uber наведено на рис. 1.1.

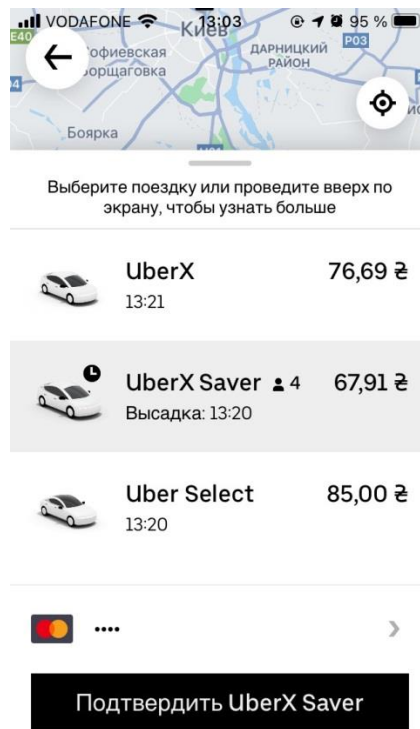


Рисунок 1.1 – Приклад інтерфейсу Uber [9]

1.3.2 Сервіс «Bolt»

Bolt (раніше Taxify) - компанія з подорожей на автомобілі та мікромобільних перевезень, заснована в 2013 році Маркусом Віллігом, 19-річним учнем середньої школи в Естонії. Компанія розпочала свою діяльність як застосунок для виклику таксі в Таллінні, Естонія, і з тих пір поширилася на понад 45 країн Європи та Африки. Bolt пропонує безліч варіантів транспорту, включаючи прокат автомобілів, автомобілі загального користування, скутери та електронні велосипеди [11], [12].

Bolt працює, з'єднуючи гонщиків з водіями через мобільний застосунок. Водії можуть вимагати поїздки за допомогою програми, а водії, які знаходяться поблизу, будуть повідомлені про цей запит. Як тільки водій прийме запит, гонщик зможе відстежувати прогрес водія на карті. Коли водій прибуде, пасажир зможе оплатити поїздку за допомогою програми [11], [12].

Bolt отримав високу оцінку за його зручність і доступність за ціною. Компанію також похвалили за функції безпеки, які включають відстеження в

режимі реального часу та цілодобову підтримку клієнтів. Bolt також прагне бути екологічним, і компанія пропонує безліч екологічно чистих варіантів, таких як електромобілі та скутери [11], [12].

Ось деякі з ключових особливостей Bolt.

Зручність: за допомогою Bolt легко пересуватися по місту. Користувач може замовити поїздку лише кількома натисканнями на свій телефон [11], [12].

Доступність: Bolt є більш доступним варіантом, ніж традиційні таксі. Користувач може заощадити гроші на своїх поїздках, використовуючи Bolt.

Безпека: компанія Bolt піклується про вашу безпеку. Всі водії пройшли перевірку і застраховані. Користувач також може відстежувати свою поїздку в режимі реального часу [11], [12].

Екологічно чистий: Bolt пропонує безліч екологічно чистих варіантів, таких як електромобілі та скутери. Користувач може допомогти навколишньому середовищу, вибравши екологічний варіант [11], [12].

Якщо користувач шукає зручний, доступний і безпечний спосіб пересування по місту, то Bolt - відмінний варіант. За допомогою Bolt користувач може швидко і легко дістатися туди, куди потрібно.

Приклад інтерфейсу Bolt наведено на рис. 1.2.

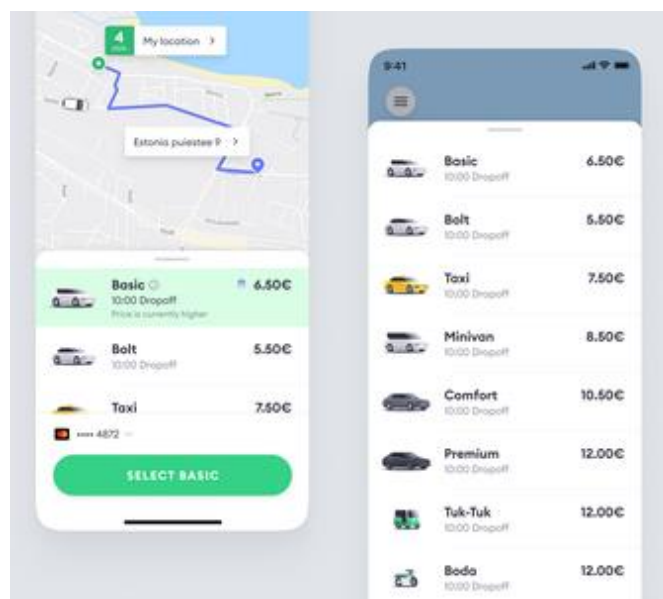


Рисунок 1.2 – Приклад інтерфейсу Bolt [11]

1.3.3 Сервіс «Uklon»

Uklon – український сервіс за викликом таксі, заснований в 2009 році. Штаб - квартира компанії знаходиться в Києві, Україна, і здійснює свою діяльність у більш ніж 28 містах України, а також у кількох містах Польщі, Румунії та Грузії. Uklon пропонує безліч варіантів транспортування, включаючи виклик таксі, прокат автомобілів і доставку [13], [14].

Компанію заснували Дмитро Дубровський, Вікторія Дубровська та Сергій Смас. У 2006 році Дмитро Дубровський заснував Evos, традиційну службу таксі в Україні. У 2009 році на базі Evos з'явилася компанія, яка була типовим прикладом традиційної служби таксі в Україні. Розробкою продукту Uklon займався випускник Києво - Могилянської академії-21-річний математик Віталій Дятленко. У 2008 році Сергій і Віталій стали співвласниками Evos. Uklon розпочав свою діяльність у 2010 році [13], [14].

Компанія Uklon швидко зростала з моменту свого заснування. У 2016 році компанія залучила 10 мільйонів доларів фінансування серії А від Horizon Capital та Genesis Partners. У 2018 році компанія залучила застосункові 20 мільйонів доларів на фінансування Серії В від Horizon Capital, Genesis Partners та AVentures Capital [13], [14].

Uklon отримав високу оцінку за його зручність і доступність за ціною. Компанію також похвалили за функції безпеки, які включають відстеження в режимі реального часу та цілодобову підтримку клієнтів. Uklon також прагне бути екологічним, і компанія пропонує безліч екологічно чистих варіантів, таких як електромобілі та скутери [13], [14].

Ось головні особливості Uklon.

Зручність: Uklon дозволяє легко пересуватися по місту. Користувач може замовити поїздку лише кількома натисканнями на свій телефон [13], [14].

Доступність: Uklon є більш доступним варіантом, ніж традиційні таксі. Користувач може заощадити гроші на своїх поїздках, використовуючи Uklon.

Безпека: Uklon піклується про вашу безпеку. Всі водії пройшли перевірку і застраховані. Користувач також може відстежувати свою поїздку в режимі реального часу [13], [14].

Екологічність: Uklon пропонує безліч екологічно чистих варіантів, таких як електромобілі та скутери. Користувач може допомогти навколишньому середовищу, вибравши екологічний варіант [13], [14].

Якщо ви шукаєте зручний, доступний і безпечний спосіб пересування по місту, то Uklon - відмінний варіант. З Uklon користувач може швидко і легко дістатися туди, куди вам потрібно [13], [14].

Ось як працює Uklon:

- завантаження застосунків Uklon і створення облікового запису;
- вказання місця самовивозу і висадки;
- вибір типу транспортного засобу, який хоче користувач;
- оплата поїздки за допомогою кредитної картки або мобільного платіжного застосунка;
- відстеження своєї поїздки в режимі реального часу.
- зустріч зі своїм водієм у місці зустрічі.
- виконання поїздки.

Uklon доступний більш ніж в 28 містах України, а також в декількох містах Польщі, Румунії та Грузії. Компанія постійно розширює своє охоплення, і очікується, що найближчим часом вона буде доступна в більшій кількості міст [13], [14].

Uklon – відмінний варіант для тих, хто шукає зручний, доступний і безпечний спосіб пересування по місту. Компанія пропонує безліч варіантів транспортування і піклується про вашу безпеку і екологічності. З Uklon ви можете швидко і легко дістатися туди, куди вам потрібно [13], [14].

Приклад інтерфейсу GitHub наведено на рис. 1.3.

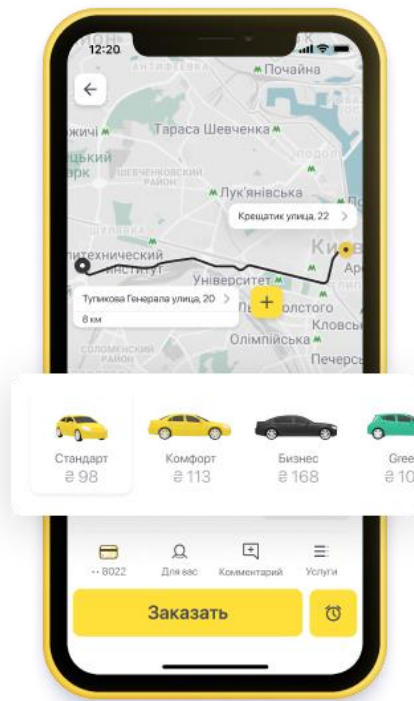


Рисунок 1.3 – Приклад інтерфейсу Uklon [13]

1.3.4 Порівняння існуючих аналогів

Виконаємо порівняння існуючих аналогів за наступними критеріями:

- модерація постів – чи наявні механізми модерації контенту, що розміщується на платформі, та на якому рівні (перевірка достовірності, спам-фільтри і т.п.)
- робота в Україні – чи працює сервіс наразі в Україні;
- організація колективних поїздок – чи дозволяє застосунок організовувати колективні (корпоративні чи родинні) пасажирські поїздки;
- способи оплати поїздок – способи оплати, що надає застосунок;
- механізми захисту користувачів – рівень захисту особистих даних, які надає користувач при реєстрації для користування застосунком;
- програми підтримки та лояльності для користувачів – чи наявні в застосунку механізми для ведення статистики по користувачу для оформлення програми лояльності.

Результати порівняння існуючих аналогів наведено у табл. 1.1.

Таблиця 1.1 – Програмне забезпечення для підтримки освітньої діяльності

Критерії порівняння	Uber	Bolt	Uklon	Your Taxi
Робота в Україні	Частково призупинена	Так	Так	Так
Організація колективних поїздок	Ні	Ні	Ні	Так
Способи оплати поїздок	Картка, готівка	Картка, готівка	Картка, готівка	Картка, готівка
Механізми захисту користувачів	Високий ступінь (спілкування виключно через систему)	Середній ступінь (дані доступні через застосунок)	Середній ступінь (дані доступні через застосунок)	Високий ступінь (спілкування виключно через систему)
Програми підтримки лояльності та для користувачів	Відсутня	Відсутня	Відсутня	На основі даних про поїздки користувача

При ознайомленні з результатами порівняння можна зробити висновки, що в більшості випадків застосунки розраховані на особисте використання користувача та не надають функцій організації групових поїздок (як для корпоративного користування, так і для родинного). Більш того більшість існуючих рішень не в достатній мірі забезпечують збереження даних користувачів, адже надають особисту інформацію про них водіям (номери телефону, ім'я та інше).

Саме тому, розробка нового застосунку у вигляді інформаційного ресурсу служби таксі є актуальним прикладним завданням.

1.4 Постановка завдання

Головним завданням роботи є створення системи, яка дозволяє користувачу замовити таксі через вебсайт.

Система повинна сповіщати користувача про можливість виконання замовлення, яке надходить від нього через сайт.

Система буде складатися з декількох модулів. Модулі електронної платіжної системи, база даних MySQL, збір статистики Google будуть додані до системи у готовому вигляді, інші модулі, наприклад модуль сортування буде створений в ході розробки вебзастосунку.

1.4.1 Функціональність системи

Функціональні вимоги:

- звукове сповіщення про статус замовлення;
- розрахунок вартості поїздки;
- підбір вільних авто;
- розрахунок часу прибуття;
- індексування користувачів.

1.4.2 Вимоги до інтерфейсу

Інтерфейс повинен бути в фірмових кольорах, інтуїтивно зрозумілий, та в приємних кольорах. Всі послуги та акції компанії повинні бути перераховані на вебсайті.

1.4.3 Вимоги до продуктивності

Система повинна забезпечувати швидку передачу даних про замовлення;

Система повинна стабільно працювати при великій кількості замовлень.

1.4.4 Вимоги до безпеки

Дані платіжної системи, локальне сховище бази даних користувачів, інформація про користувачів будуть використовувати потоковий шифр, для забезпечення безпеки користувачів системи.

1.4.5 Експлуатаційні вимоги

Для коректної роботи системи на комп'ютерах користувачів повинно бути:

- будь-яка операційна система (ОС);
- процесор 1.8 GHz +;
- підключення до мережі Інтернет.

Також для стабільної роботи системи повинен бути як мінімум один комп'ютер-сервер та людина, що буде контролювати його коректну роботу.

1.4.6 Не функціональні вимоги до продукту

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини мобільного застосунку:

- продуктивний хостинг;
- вебсервер Apache;
- підтримка PHP (починаючи з версії 5.4) та WordPress.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- персональний комп'ютер (ПК) з графічною ОС;
- процесор із тактовою частотою від 1,8 GHz;
- обсяг оперативної пам'яті від 3GB;
- підключення до мережі Internet.

1.5 Висновки за розділом 1

В першому розділі основну увагу було приділено дослідженню проблемної області транспортних пасажирських перевезень. Спочатку було розглянуто сучасний стан проблемної області. Після цього було досліджено можливості часткової автоматизації процесу виконання замовлення пасажирського транспортного перевезення, через використання програмних засобів. На основі попереднього дослідження було виділено основних лідерів галузі та детально проаналізовано їх переваги та недоліки.

З рештою, це все дозволило виконати формування переліку функціональних та нефункціональних вимог до майбутнього нового рішення, що і було головною метою цього розділу.

2 МАТЕРІАЛИ І МЕТОДИ

В першому розділі основну увагу було приділено дослідженню проблемної області транспортних пасажирських перевезень. Спочатку було розглянуто сучасний стан проблемної області. Після цього було досліджено можливості часткової автоматизації процесу виконання замовлення пасажирського транспортного перевезення, через використання програмних засобів. На основі попереднього дослідження було виділено основних лідерів галузі та детально проаналізовано їх переваги та недоліки. З рештою, це все дозволило виконати формування переліку функціональних та нефункціональних вимог до майбутнього нового рішення, що і було головною метою цього розділу.

В другому розділі буде виконуватися робота з вибору та затвердження технічних засобів розробки: технологія реалізації застосунку, мова програмування, надбудови для організації розробки, патерн розробки для уніфікації. Порівняння та аналіз переваг та недоліків дозволить зробити ґрунтовний вибір.

2.1 Вибір мови програмування

PHP – широко використовувана і популярна серверна скриптова мова для веброзробки. Незважаючи на появу численних мов програмування і фреймворків, PHP продовжує зберігати свою актуальність і популярність. У цій відповіді ми розглянемо, чому PHP залишається популярним, і дамо огляд того, як він працює [15]–[18].

Простота і зручний у використанні. Однією з головних причин популярності PHP є його простота і зручність у використанні. PHP був розроблений таким чином, щоб бути зручним для користувачів, дозволяючи розробникам швидко писати та розгортати вебпрограми. Синтаксис PHP схожий на C та інші мови програмування, що полегшує його вивчення та

використання розробниками. Крім того, PHP легко інтегрується з HTML, що робить його природним вибором для веброзробки [15]–[18].

Велика документація та підтримка спільноти. PHP виграє від широкої та активної спільноти розробників. Ця мова існує вже більше двох десятиліть, що призвело до безлічі документації, навчальних посібників та інтернет-ресурсів. Розробники можуть легко знаходити відповіді на свої запитання, усувати неполадки та ділитися своїми знаннями у спільноті PHP. Наявність такої великої підтримки робить PHP привабливим вибором як для початківців, так і для досвідчених розробників [15]–[18].

Велика кодова база та сторонні бібліотеки. PHP володіє величезною кодовою базою і великою колекцією сторонніх бібліотек і фреймворків. Ці бібліотеки пропонують готові модулі та функції, які спрощують складні завдання, такі як взаємодія з базою даних, обробка файлів та зображень. Популярні фреймворки PHP, такі як Laravel, Symfony та CodeIgniter, забезпечують надійні структури та Компоненти для створення масштабованих та підтримуваних вебзастосунків. Доступність цих ресурсів прискорює час розробки та дозволяє розробникам ефективно створювати багатофункціональні програми [15]–[18].

Кросплатформна Сумісність. PHP призначений для роботи в різних операційних системах, включаючи Windows, macOS, Linux та Unix. Така кросплатформна сумісність гарантує, що PHP-застосунки можуть бути розгорнуті на різних вебсерверах без істотних змін. Будь то Apache, Nginx, Microsoft IIS або будь-який інший вебсервер, PHP може легко інтегруватися з ними, пропонуючи гнучкість у варіантах розгортання [15]–[18].

Можливості інтеграції. PHP легко інтегрується з іншими технологіями та платформами. Він має широку підтримку взаємодії з базами даних, включаючи MySQL, PostgreSQL, Oracle та багато інших. PHP також може взаємодіяти із зовнішніми службами та API за допомогою різних протоколів, таких як HTTP, SOAP, REST та XML-RPC. Ці можливості інтеграції роблять PHP чудовим вибором для створення динамічних та інтерактивних

вебзастосунків, які взаємодіють з кількома системами та джерелами даних [15]–[18].

Продуктивність і швидкість. PHP пройшов довгий шлях з точки зору оптимізації продуктивності. З кожним випуском команда розробників PHP вносила численні вдосконалення для покращення швидкості виконання та зменшення споживання пам'яті. Останні версії PHP, такі як PHP 7 та PHP 8, продемонстрували значне підвищення продуктивності порівняно з попередніми версіями. Така покращена продуктивність гарантує, що PHP зможе справлятися з високими навантаженнями на трафік і ефективно обробляти складні операції [15]–[18].

Масштабованість та гнучкість. Масштабованість та гнучкість PHP роблять його придатним для проектів різних розмірів та складності. Незалежно від того, чи створюєте ви невеликий персональний вебсайт або масштабну корпоративну програму, PHP може задовольнити ваші потреби. Крім того, гнучкість PHP дозволяє розробникам вибирати з різних парадигм програмування, таких як процедурна або об'єктно-орієнтована, залежно від вимог їх проекту. Така адаптивність сприяє широкому використанню PHP у різних доменах [15]–[18].

Як працює PHP. PHP в основному використовується як серверна мова сценаріїв, що означає, що він працює на вебсервері, а не в браузері користувача. Ось огляд того, як працює PHP [15]–[18].

Запит клієнта. Коли користувач запитує вебсторінку, що містить PHP-код, вебсервер ідентифікує PHP-код за розширенням файлу, як правило .php. Сервер визнає, що йому потрібно обробити PHP-код, перш ніж надсилати вихідні дані назад клієнту [15]–[18].

Інтерпретатор PHP. Вебсервер передає PHP-код інтерпретатору PHP, який є програмним модулем, відповідальним за виконання PHP-скриптів. Інтерпретатор PHP зчитує та обробляє PHP-код рядок за рядком, виконуючи різні дії на основі наданих інструкцій [15]–[18].

Змішування PHP з HTML. Однією з сильних сторін PHP є його здатність легко змішувати PHP-код з HTML-розміткою. PHP-код укладений у спеціальні роздільники, як правило `<?php i?>`. Коли інтерпретатор PHP стикається з цими роздільниками, він переходить з режиму синтаксичного аналізу HTML в режим обробки PHP. У режимі PHP він виконує PHP-код і генерує динамічний вміст або виконує певні завдання [15]–[18].

Доступ до баз даних та зовнішніх ресурсів. PHP має вбудовану підтримку взаємодії з базами даних. Розробники можуть використовувати функції та розширення, надані PHP, для підключення до баз даних, Виконання запитів, отримання результатів та маніпулювання даними. PHP також підтримує різні протоколи та бібліотеки для взаємодії із зовнішніми ресурсами, такими як вебсервіси та API [15]–[18].

Створення динамічного контенту. Однією з основних цілей PHP є генерація динамічного контенту. Оскільки інтерпретатор PHP обробляє PHP-код, він може виконувати обчислення, отримувати доступ до баз даних, маніпулювати файлами та взаємодіяти із зовнішніми службами. Вихідні дані, що генеруються PHP-кодом, можуть включати HTML, CSS, JavaScript або будь-який інший тип вмісту, що підтримується вебсервером [15]–[18].

Відправка відповіді клієнту. Як тільки інтерпретатор PHP закінчує обробку PHP-коду, він надсилає згенерований вихід назад на вебсервер. Потім вебсервер пересилає відповідь браузеру користувача, який відображає HTML, CSS та JavaScript для відображення кінцевої вебсторінки [15]–[18].

Таким чином, PHP залишається популярним завдяки своїй простоті, великій документації, великій підтримці спільноти та Великій кодовій базі. Він забезпечує сумісність між платформами, можливості інтеграції та підвищення продуктивності. Здатність PHP генерувати динамічний контент, взаємодіяти з базами даних і легко змішуватися з HTML робить його відмінним вибором для проектів веброзробки різного розміру і складності [15]–[18].

Проведемо порівняння мов програмування для реалізації вебзастосунків, результати порівняння представимо в табличній формі (табл. 2.1). У якості критеріїв будемо використовувати наступні:

- надбудови для структуризації розробки – чи наявні зараз якісь рішення, на кшталт, фреймворків, надбудов або ж систем керування вмістом (CMS) для структуризації та уніфікації розробки;
- серверна підтримка – наскільки розповсюджена підтримка технології серед серверів для розгортання застосунку;
- масштабованість розробок – наскільки можливе та просте подальше масштабування розроблених застосунків;
- складність підтримки – оцінка складності подальшої підтримки розробок.

Таблиця 2.1 – Порівняння мов програмування

Критерій	PHP	Python	ASP.NET
Надбудови для структуризації розробки	Багато	Мало	Мало
Серверна підтримка	Розповсюджена	Недостатньо розповсюджена	Недостатньо розповсюджена
Масштабованість розробок	Проста	Проста	Проста
Складність підтримки	Низька	Середня	Середня

Надзвичайне поширення та розповсюдженість підтримки технології PHP робить його оптимальним рішенням для веброзробки в порівнянні з іншими аналогами популярними на сьогоднішній день.

2.2 Вибір та обґрунтування CMS

WordPress – це широко використовувана та популярна CMS, яка підтримує значну частину вебсайтів в Інтернеті. Незважаючи на доступність

численних варіантів CMS, WordPress продовжує зберігати своє домінуюче становище [19]–[21].

Зручний інтерфейс. Однією з основних причин популярності WordPress є його Зручний інтерфейс. Він був розроблений з урахуванням простоти, дозволяючи користувачам, які майже не мають технічних знань, легко створювати та керувати своїми вебсайтами. Інтуїтивно зрозуміла інформаційна панель, функція перетягування та візуальні редактори роблять створення та обслуговування своїх вебсайтів доступними для початківців та нетехнічних користувачів, не покладаючись на розробників [19]–[21].

Широкі теми та плагіни. WordPress пропонує велику колекцію тем і плагінів, які розширюють його функціональність. Темі дозволяють користувачам змінювати дизайн та макет своїх вебсайтів лише кількома клацаннями миші, дозволяючи їм створювати візуально привабливі та унікальні вебсайти. Плагіни, з іншого боку, надають застосункові функції та можливості, такі як Електронна комерція, оптимізація пошукових систем (SEO), контактні форми, Інтеграція соціальних мереж тощо. Маючи широкий спектр тем і плагінів, WordPress легко налаштовується та адаптується до різних вимог вебсайту [19]–[21].

Активна спільнота та підтримка. WordPress виграє від активної та динамічної спільноти розробників, дизайнерів та користувачів. Ця спільнота надає широку підтримку, документацію, навчальні посібники, форуми та інтернет-ресурси, які допомагають користувачам вирішувати проблеми, знаходити рішення та розширювати свої знання. Характер WordPress, орієнтований на спільноту, гарантує, що користувачі мають доступ до величезної кількості інформації та допомоги, що полегшує подолання проблем та вдосконалення своїх вебсайтів WordPress [19]–[21].

Зручність SEO. WordPress спочатку розроблений таким чином, щоб бути зручним для пошукових систем, що має вирішальне значення для видимості вебсайту та органічного трафіку. Він генерує чистий і структурований код, пропонує зручні для SEO URL-адреси, дозволяє

налаштовувати мета-теги, надає плагіни для оптимізації контенту і підтримує адаптивний дизайн. Ці функції в поєднанні з найкращими практиками SEO роблять WordPress кращим вибором для користувачів, які надають пріоритет оптимізації пошукових систем [19]–[21].

Масштабованість та гнучкість. WordPress має високу масштабованість та гнучкість, що дозволяє вебсайтам рости в міру розвитку їхніх потреб. Незалежно від того, ведете ви невеликий блог або великий корпоративний вебсайт, WordPress може обробляти різні рівні трафіку та вмісту. Він підтримує створення декількох типів вмісту, таких як записи в блогах, сторінки, портфоліо, продукти електронної комерції тощо. WordPress також дозволяє користувачам легко оновлювати та керувати своїм вмістом, що робить його ідеальною платформою для вебсайтів, які потребують частих оновлень [19]–[21].

Екосистема та інтеграція. WordPress має велику екосистему, яка розширює її функціональність та можливості інтеграції. Він легко інтегрується з різними сторонніми послугами, такими як платіжні шлюзи, маркетингові платформи електронної пошти, аналітичні інструменти та соціальні мережі. Ця можливість інтеграції дозволяє користувачам використовувати можливості зовнішніх сервісів і оптимізувати роботу свого вебсайту [19]–[21].

Як працює WordPress. WordPress дотримується архітектури клієнт-сервер і використовує для роботи комбінацію PHP, бази даних (зазвичай MySQL) та вебсервера. Ось огляд того, як працює WordPress [19]–[21].

Обробка на стороні сервера. Коли користувач запитує вебсторінку, створену за допомогою WordPress, вебсервер отримує запит і передає його інтерпретатору PHP. Інтерпретатор PHP обробляє PHP-код у файлах WordPress, взаємодіє з базою даних та виконує різні дії на основі запиту користувача [19]–[21].

Взаємодія з базою даних. WordPress зберігає свій вміст, налаштування та конфігурації в базі даних. PHP-код у WordPress отримує та оновлює інформацію з бази даних за потреби. Це включає в себе вибірку повідомлень,

сторінок, коментарів, відомостей про користувачів та інших елементів контенту. База даних забезпечує структуровану систему зберігання даних для управління та упорядкування даних вебсайту [19]–[21].

Рендеринг теми. WordPress використовує теми для управління зовнішнім виглядом та макетом вебсайту. Коли користувач отримує доступ до сторінки WordPress, PHP-код витягує відповідні файли теми та динамічно генерує HTML, CSS та JavaScript, необхідні для візуалізації сторінки. Темми визначають загальний вигляд, дизайн та функціональність вебсайту, включаючи такі елементи, як колонтитули, бічні панелі, меню тощо [19]–[21].

Функціональність плагіна. Плагіни в WordPress розширюють його функціональність, додаючи певні функції та можливості. Коли користувач встановлює та активує плагін, PHP-код у плагіні інтегрується з кодовою базою WordPress. Плагіни можуть підключатися до різних подій і дій в WordPress, дозволяючи їм розширювати або змінювати основні функціональні можливості. Це може включати додавання нових типів вмісту, реалізацію користувацьких функцій, інтеграцію із зовнішніми службами тощо [19]–[21].

Взаємодія з користувачем і управління контентом. WordPress надає зручний інтерфейс інформаційної панелі, де власники вебсайтів та адміністратори можуть керувати своїм вмістом, налаштовувати Налаштування, встановлювати теми та плагіни та виконувати інші адміністративні завдання. Інформаційна Панель дозволяє користувачам створювати та редагувати записи, сторінки, медіа та інші елементи вмісту за допомогою візуального редактора або безпосередньо редагуючи базовий код [19]–[21].

Формування та доставка вихідних даних. Як тільки PHP-код у WordPress обробляє запит Користувача, він генерує кінцевий результат, як правило, у вигляді HTML, CSS та JavaScript. Веб-сервер надсилає ці результати назад у браузер користувача, який малює та відображає вебсторінку на основі отриманого вмісту. На цьому цикл обробки запиту

користувача і доставки відповідної вебсторінки, згенерованої на WordPress, завершується [19]–[21].

Таким чином, WordPress залишається популярним завдяки своєму зручному інтерфейсу, великій екосистемі теми та плагінів, активній підтримці спільноти, зручності SEO, масштабованості та можливостям інтеграції. Він використовує архітектуру клієнт-сервер, використовує PHP та базу даних для обробки на стороні сервера та управління вмістом, а також динамічно генерує вебсторінки на основі запитів користувачів [19]–[21].

Порівняльна характеристика WordPress з іншими найпоширенішими CMS для розробки вебзастосунку у таблиці 2.2.

Серед критеріїв порівняння будемо розглядати наступні:

- вимоги до хостингу – вимоги зі сторони застосунку та CMS до хостингу, де розгортається застосунок;
- впорядкованість застосунку – наявність чіткої архітектури для спрощення підтримки;
- можливість масштабованості – можливість та простота масштабованості попередньо розробленого застосунку, на основі використання сторонніх модулів;
- робота з СКБД – простота та захищеність роботи із СКБД;
- підтримка та сторонні розширення – підтримка зі сторони спільноти та оцінка кількості якісних сторонніх розширень для спрощення розробки застосункового функціоналу.

Таблиця 2.2 – Порівняння фреймворків

	WordPress	October
Швидкість та простота розробки	Проста та швидка розробка	Ускладнена розробка
Модульна структура	Так	Так
Використання сторонніх модулів	Так (є офіційний каталог протестованих розширень)	Є внутрішній каталог (обмежений)
Захищеність даних	Висока	Середня
Серверна підтримка	Активна	Активна

З порівняння можна зробити висновок, що WordPress є однією з найпопулярніших альтернатив, команда якого турбується про захист даних і загальну безпеку застосунків, тому постійно й активно оновлює CMS.

2.3 Структурна схема розробки

Система складається з двох частин:

- клієнтська;
- серверна.

Загальна схема системи за обраною темою на рис. 2.1.

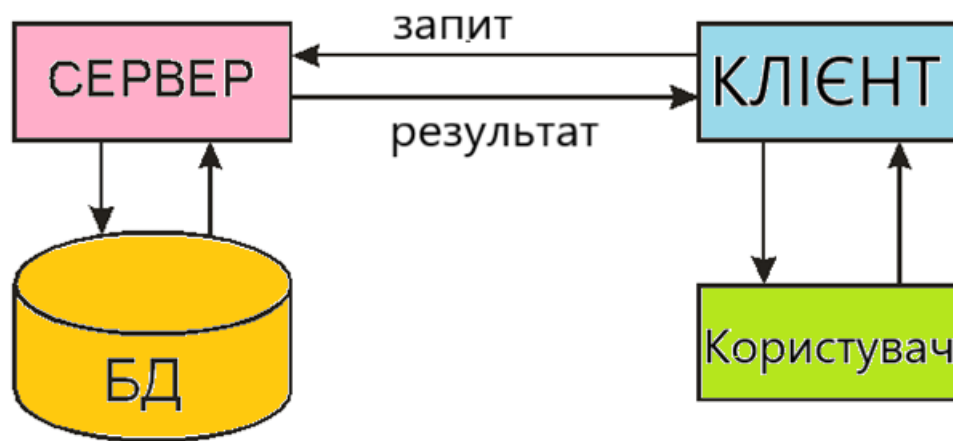


Рисунок 2.1 – Загальна схема системи за обраною темою

Архітектура вебзастосунку базується на моделі Модель-вид-контролер (MVC).

Модель здійснює роботу з базою даних та саме у моделях повинні бути зосереджені методи роботи з даними.

Контролер оброблює дані з моделі, якщо вони потрібні, і генерує Вид з цими даними, де інформація доноситься до користувача з відповідними графічними елементами. Взаємодія користувача з застосунком відбувається через Контролер.

Архітектура вебзастосунку наведена на рис. 2.2.

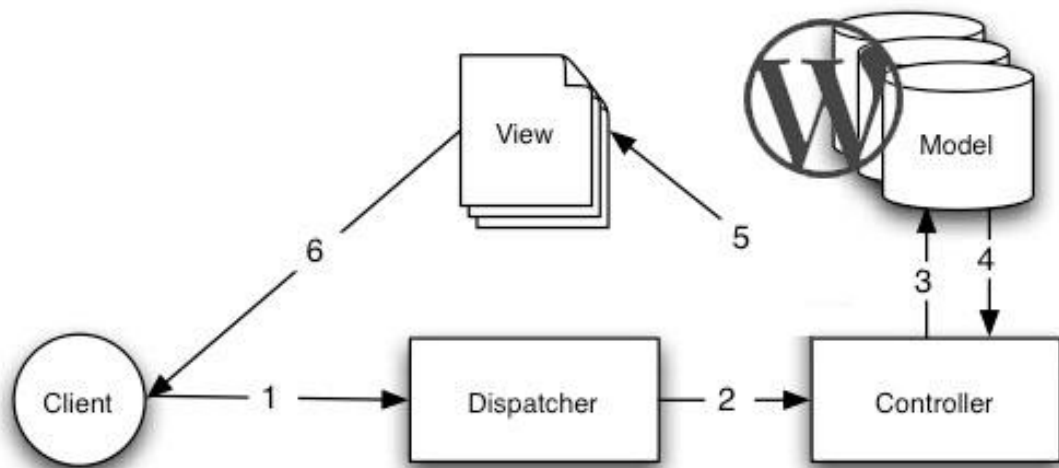


Рисунок 2.2 – Архітектура вебзастосунку

2.4 Висновки за розділом 2

В другому розділі було виконано роботу з вибору та затвердження технічних засобів розробки: технологія реалізації застосунку, мова програмування, надбудови для організації розробки, патерн розробки для уніфікації. Порівняння та аналіз переваг та недоліків дозволили зробити ґрунтовний вибір: мова PHP, CMS WordPress, патерн MVC.

3 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

3.1 Проектування моделей системи

Моделювання на уніфікованій мові моделювання (UML) важливо при веброзробці з кількох причин. UML надає стандартизовану візуальну мову для моделювання та документування програмних систем, включаючи вебпрограми. Ось ключові причини, чому моделювання UML є важливим у веброзробці [22], [23].

Візуалізація та передача інформації про структуру системи. Моделі UML допомагають розробникам візуалізувати та передавати структуру вебзастосунків. Використовуючи такі діаграми, як діаграми класів, діаграми компонентів та діаграми пакетів, розробники можуть представляти різні компоненти, модулі та їх взаємозв'язки всередині системи. Це допомагає зацікавленим сторонам, включаючи розробників, дизайнерів та клієнтів, отримати чітке уявлення про архітектуру та організацію системи [22], [23].

Проектування та планування поведінки системи. UML дозволяє розробникам проектувати та планувати поведінку вебзастосунків. Діаграми випадків використання можуть бути використані для фіксації та представлення функціональних вимог та взаємодій між учасниками (користувачами чи системами) та самою системою. Діаграми послідовностей та дій можуть застосунково проілюструвати потік подій та покрокову поведінку системи. Ці діаграми допомагають визначити та вдосконалити функціональність системи, гарантуючи, що вони відповідають бажаним взаємодіям користувачів та бізнес-процесам [22], [23].

Моделювання користувальницьких інтерфейсів. UML може бути використаний для моделювання та проектування користувальницьких інтерфейсів (UI) вебзастосунків. Прототипи інтерфейсу користувача, каркаси та макети можуть бути представлені за допомогою UML-діаграм, таких як діаграми класів та діаграми дій. Це дозволяє розробникам та дизайнерам

ефективно співпрацювати, забезпечуючи чітке розуміння елементів інтерфейсу користувача, потоків навігації та взаємодії з користувачами. UML-моделі користувацьких інтерфейсів служать основою для реалізації візуально привабливих і зручних для користувача вебінтерфейсів [22], [23].

Управління та документування складності системи. Вебзастосунки можуть бути складними, включаючи багато компонентів, модулів та взаємодій. UML забезпечує системний підхід до управління та документування цієї складності. За допомогою діаграм UML розробники можуть представляти структуру, поведінку та взаємодію системи структурованим та організованим способом. Ця документація служить цінним довідником для розуміння та обслуговування системи з часом, допомагаючи в усуненні неполадок, налагодженні та майбутніх удосконаленнях [22], [23].

Підтримка співпраці та командної роботи. UML виступає спільною мовою для розробників, дизайнерів та інших зацікавлених сторін, які беруть участь у проектах веброзробки. Це полегшує ефективне спілкування та співпрацю між членами команди, дозволяючи їм обмінюватися ідеями, обговорювати вимоги та узгоджувати своє розуміння системи. Діаграми UML служать візуальним засобом передачі концепцій, зменшуючи непорозуміння та сприяючи кращій командній роботі та спільній співпраці [22], [23].

Полегшення системного аналізу та тестування. Моделі UML допомагають у системному аналізі та тестуванні під час веброзробки. Наприклад, діаграми класів можна використовувати для ідентифікації та уточнення класів та їх взаємозв'язків, допомагаючи у створенні тестових наборів. Діаграми кінцевих автоматів можуть бути використані для моделювання поведінки різних станів системи, полегшуючи ідентифікацію тестового сценарію. Ці UML-діаграми підтримують валідацію і верифікацію системи, гарантуючи, що вона функціонує належним чином [22], [23].

Забезпечення розвитку та технічного обслуговування системи. Вебзастосунки часто зазнають змін та оновлень з часом. Моделі UML забезпечують основу для управління еволюцією системи та її

обслуговуванням [22], [23]. Вони служать джерелом документації для розуміння існуючої архітектури системи, полегшуючи визначення областей, які потребують вдосконалення, планування оновлень та оцінку впливу змін. Моделі UML можуть оновлюватися з урахуванням мінливої системи, забезпечуючи точну та актуальну документацію для майбутніх зусиль з розробки [22], [23].

Таким чином, моделювання на UML важливо при веброзробці, оскільки воно допомагає візуалізувати структуру системи, поведінку при проектуванні, моделювати користувацькі інтерфейси, управляти складністю, підтримувати спільну роботу, полегшувати Системний аналіз і тестування, а також забезпечувати еволюцію і обслуговування системи. Використовуючи UML, розробники можуть покращити комунікацію, покращити дизайн системи та спростити процес розробки, що в кінцевому підсумку призведе до більш успішних вебзастосунків [22], [23].

Різні типи моделей UML служать різним цілям протягом життєвого циклу розробки програмного забезпечення. Кожен тип UML-моделі фокусується на певному аспекті розроблюваної системи. Ось огляд різних типів моделей UML та їх використання [22], [23].

Діаграми класів (рис. 3.1). Діаграми класів представляють статичну структуру системи, зосереджуючись на класах, їх атрибутах, методах та взаємозв'язках. Вони зображують об'єкти та їх взаємодії всередині системи. Діаграми класів використовуються для моделювання сутностей, структур даних та бізнес-логіки системи. Вони допомагають ідентифікувати різні класи, їх властивості та поведінку, а також те, як вони співвідносяться між собою. Діаграми класів також корисні для проектування схем баз даних та представлення ієрархій успадкування [22], [23].

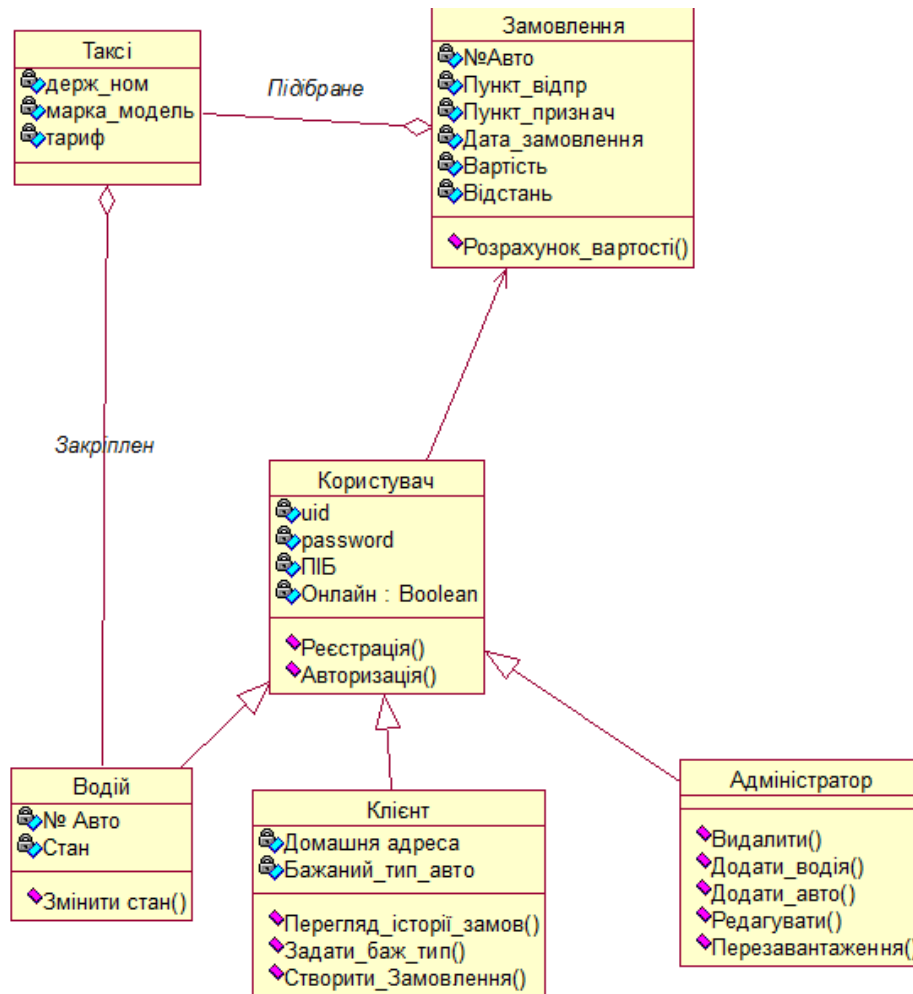


Рисунок 3.1 – Діаграма класів

Діаграма пакетів (рис. 3.2) ілюструє фізичні компоненти системи та їх взаємозв'язки.

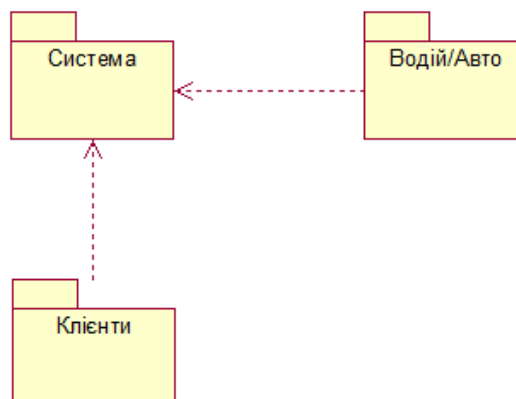


Рисунок 3.2 – Діаграма пакетів

Діаграма компонентів (рис. 3.3) ілюструє логічні компоненти системи та їх взаємозв'язки. Вони забезпечують високорівневе уявлення про архітектуру системи, приділяючи особливу увагу будівельним блокам і залежностям. Діаграми компонентів корисні для моделювання модульної структури системи, ідентифікації повторно використовуваних компонентів та визначення інтерфейсів між компонентами. Вони допомагають зрозуміти розгортання та організацію системних компонентів та допомагають у системній інтеграції [22], [23].

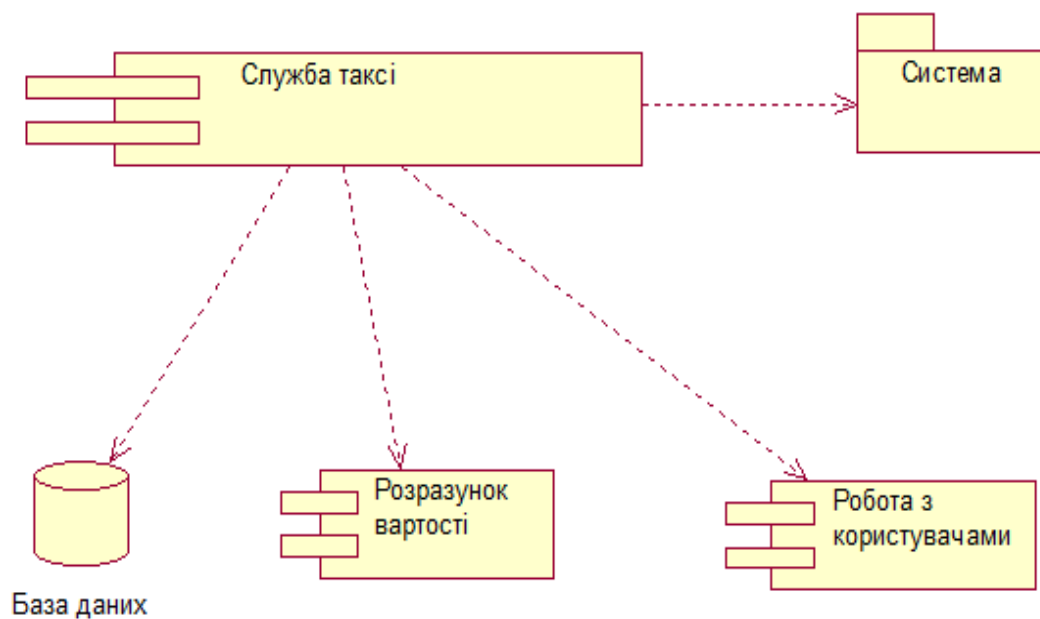


Рисунок 3.3 – Діаграма компонентів

Діаграми прецедентів (рис. 3.4). Діаграми прецедентів (або ж випадків використання) зображують взаємодію між учасниками (Користувачами або системами) та системою, що розробляється. Вони ілюструють функціональні вимоги системи з точки зору користувача. Діаграми випадків використання корисні для визначення цілей користувачів, визначення функціональних можливостей системи та визначення сфери застосування системи. Вони надають огляд високого рівня того, як користувачі взаємодіють із системою, та різні випадки використання (дії користувачів та системні реакції) [22], [23].

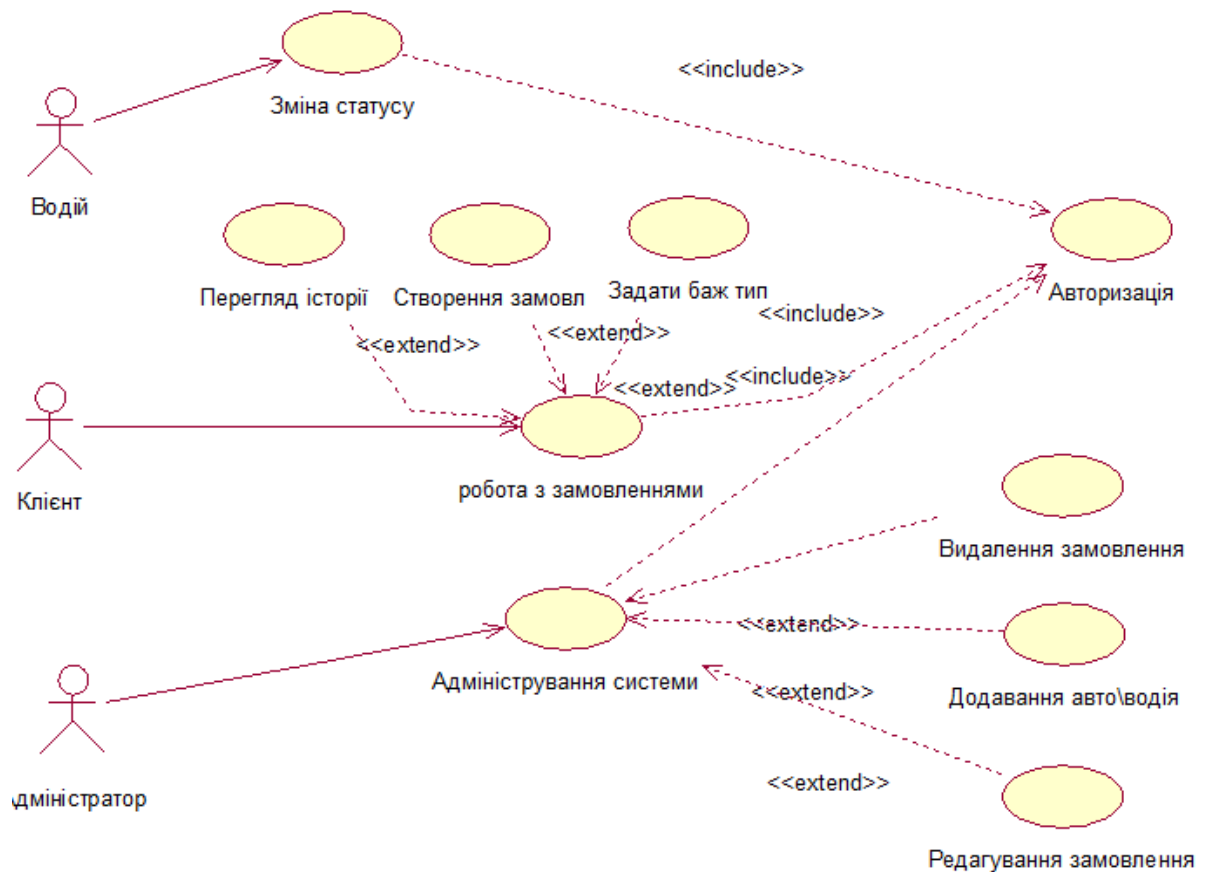


Рисунок 3.4 – Діаграма прецедентів

Діаграма станів (рис. 3.5). Діаграми станів представляють різні стани та переходи об'єкта чи системи з часом. Вони моделюють поведінку об'єктів у відповідь на події та стимули. Діаграми кінцевих автоматів особливо корисні для моделювання реактивних систем або систем зі складною поведінкою, що залежить від стану. Вони допомагають ідентифікувати різні стани, події та переходи, яким об'єкт або система можуть піддаватися протягом свого життєвого циклу. Діаграми кінцевих автоматів корисні для розуміння поведінки системи, визначення обмежень та розробки логіки управління [22], [23].

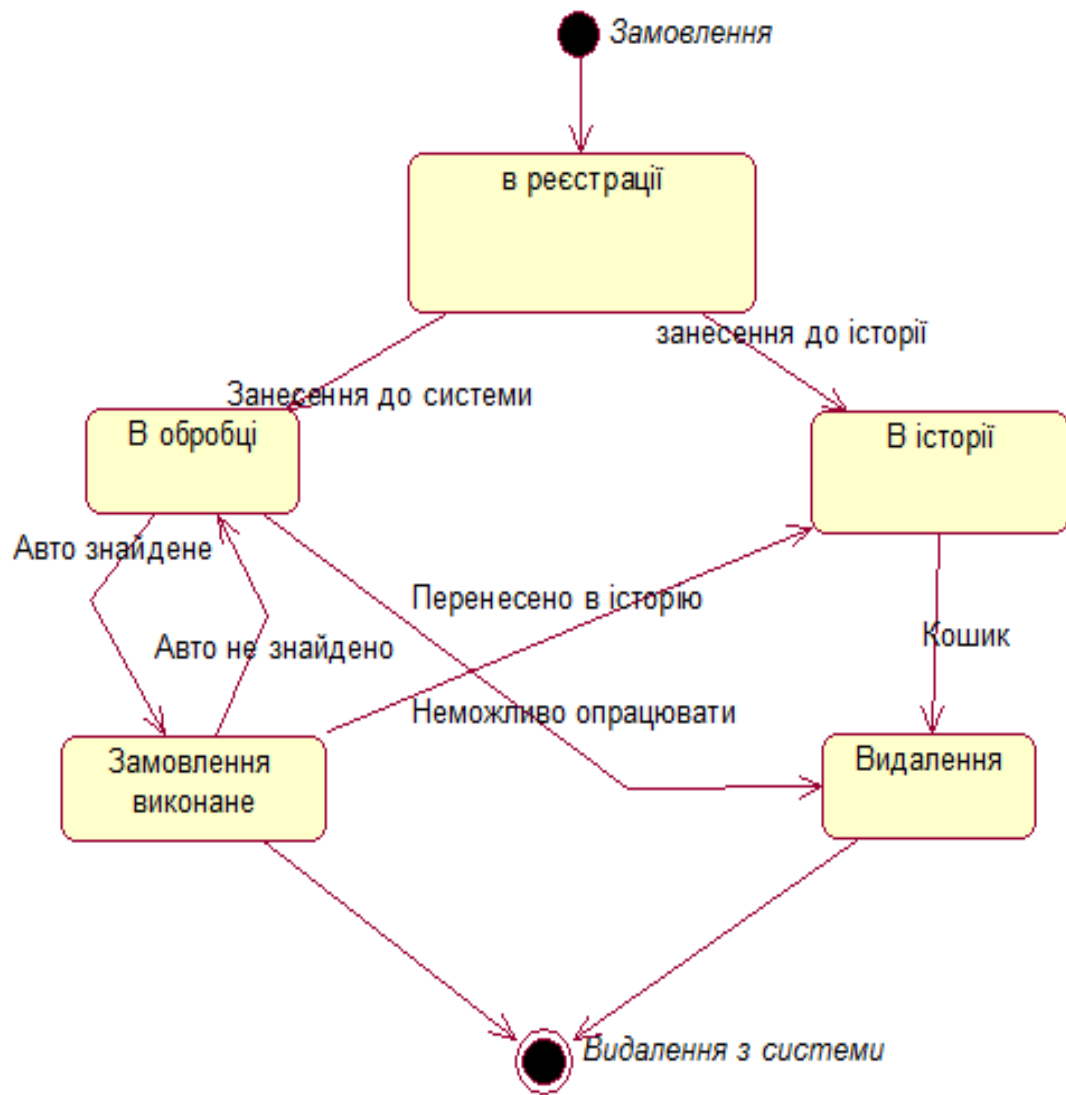


Рисунок 3.5 – Діаграма станів

Діаграми послідовностей ілюструють динамічну поведінку системи, фіксуючи взаємодію між об'єктами з часом. Вони показують послідовність повідомлень, якими обмінюються об'єкти, підкреслюючи порядок виконання. Діаграми послідовностей особливо корисні для моделювання потоку подій та взаємодії між об'єктами або компонентами в конкретному сценарії. Вони допомагають визначити порядок викликів методів, передачі повідомлень і відповідей системи під час виконання випадку використання [22], [23].

Загальна діаграма послідовностей наведена на рис. 3.6.

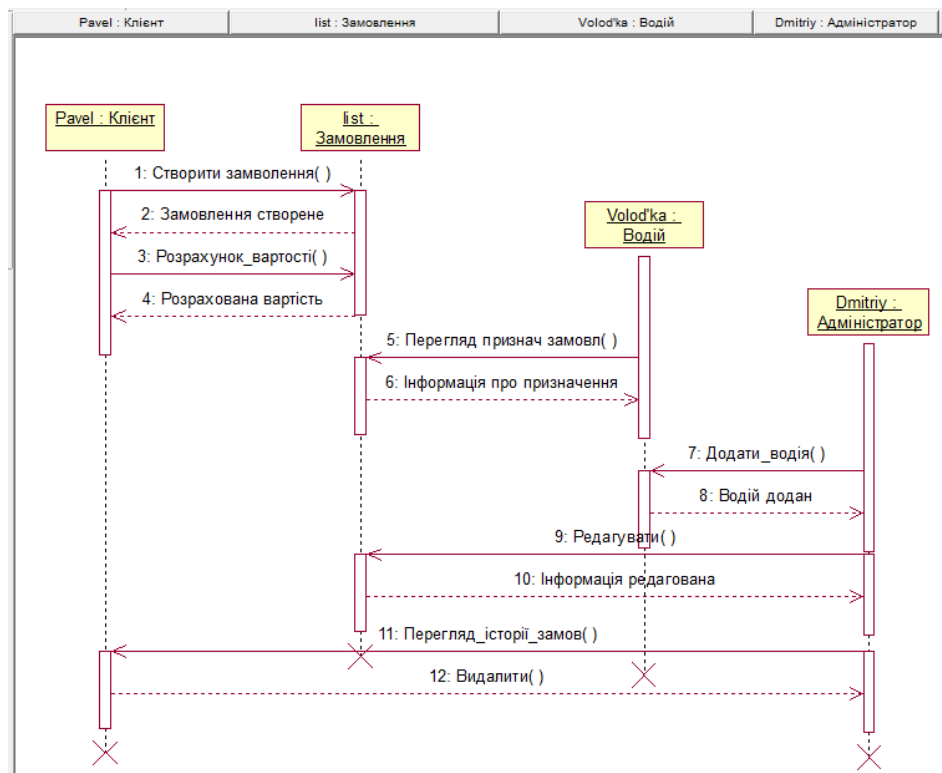


Рисунок 3.6 – Діаграма послідовності

Діаграма послідовностей для редагування інформації в системі наведена на рис. 3.7.

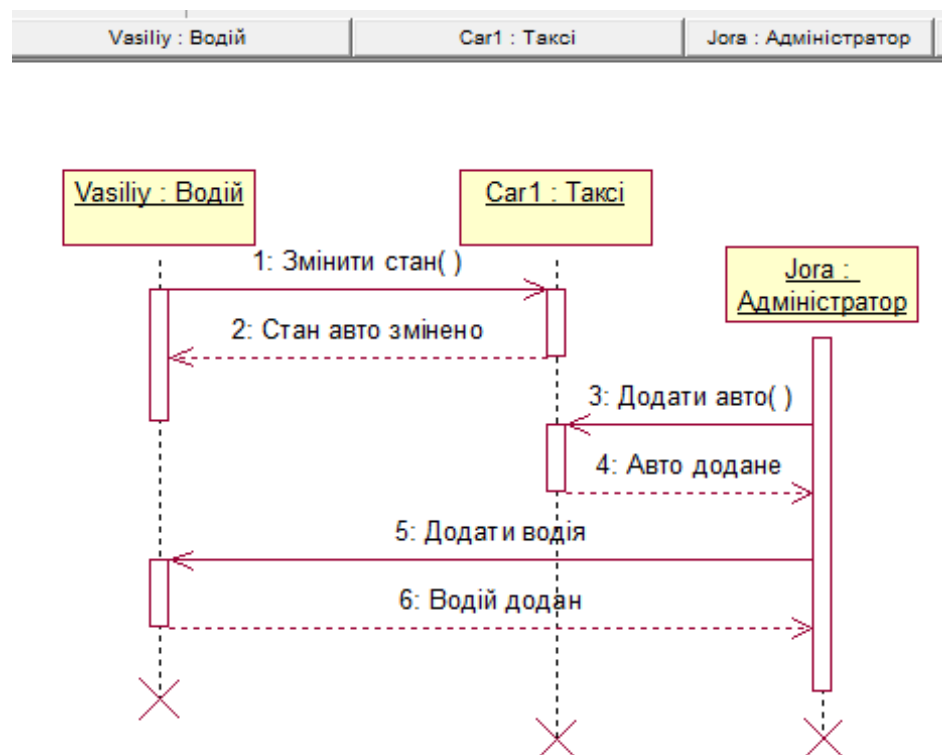


Рисунок 3.7 – Діаграма послідовності (редагування інформації в системі)

Діаграми кооперацій відображають фізичне розгортання системних компонентів на апаратних вузлах або інфраструктурі. Вони показують, як програмні компоненти розподіляються по апаратному або програмному середовищу системи. Схеми розгортання використовуються для моделювання архітектури розгортання, топології мережі, серверів, баз даних та інших компонентів інфраструктури. Вони допомагають у плануванні розгортання системи, визначенні вимог до обладнання та розумінні середовища виконання системи [22], [23]. Діаграма кооперації системи наведена на рис. 3.8.

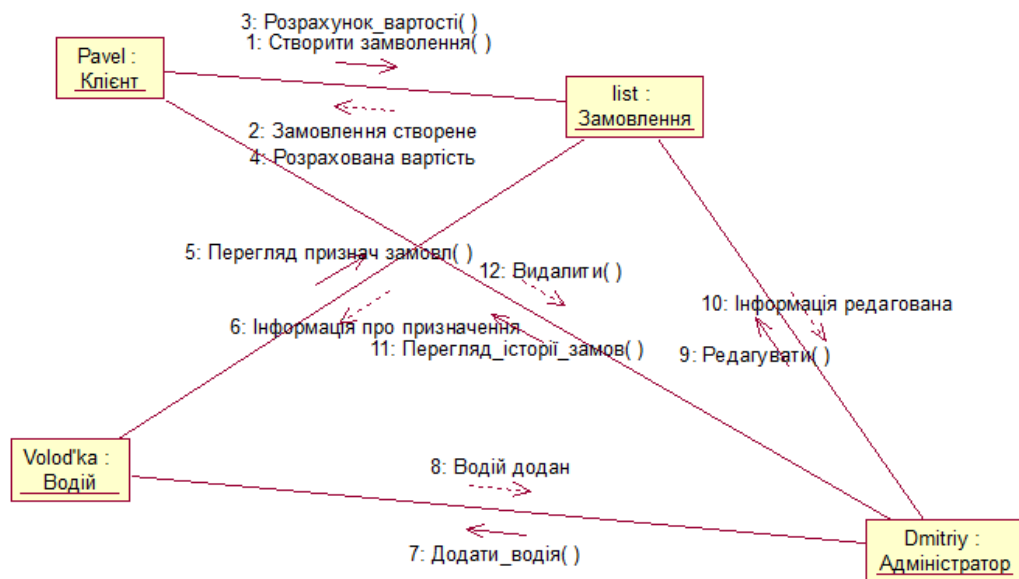


Рисунок 3.8 – Діаграма кооперації системи

Діаграма кооперації для редагування інформації в системі наведена на рис. 3.9.

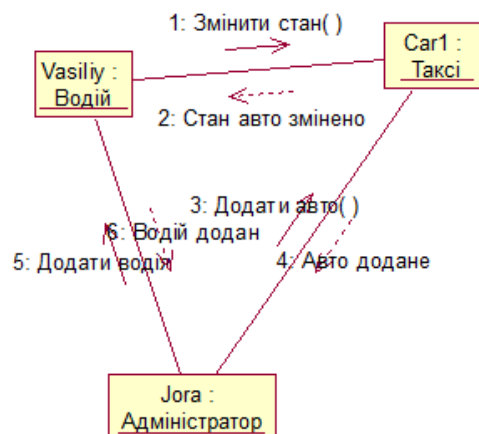


Рисунок 3.9 – Діаграма кооперації (редагування інформації в системі)

3.2 Проектування інтерфейсу

Інтерфейс повинен бути в фірмових кольорах, інтуїтивно зрозумілий, та в приємних кольорах. Всі послуги та акції компанії повинні бути перераховані на вебсайті (рис. 3.10).

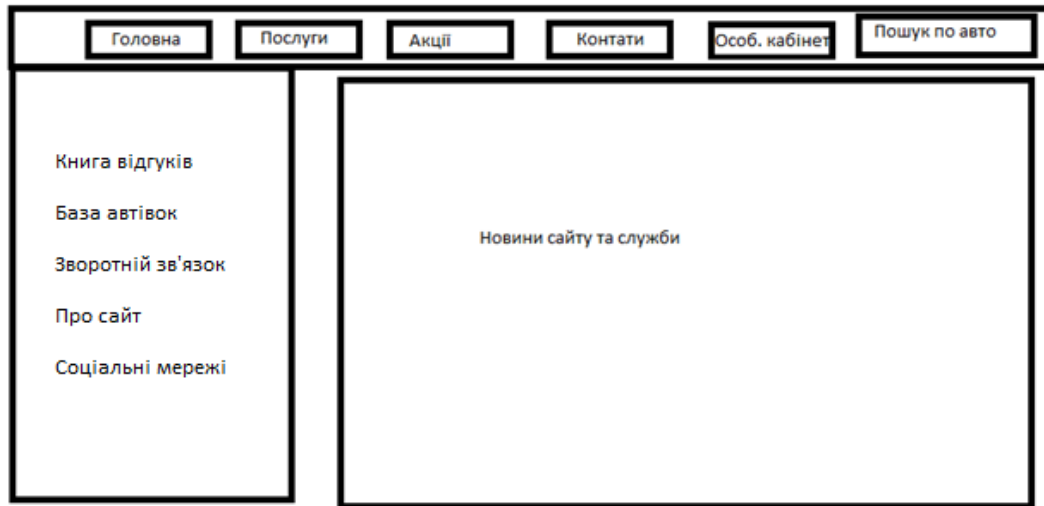


Рисунок 3.10 – Макет інтерфейсу

3.3 Висновки за розділом 3

Використовуючи засоби проектування мови UML, було проведено фізичне та логічне проектування вебзастосунку. Спираючись на визначені принципи розробки графічного інтерфейсу було прийнято рішення, щодо подальшої реалізації ПЗ.

4 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

4.1 Призначення й умови застосування програми

Головним завданням роботи є створення системи, яка дозволяє користувачу замовити таксі через вебсайт.

Система повинна сповіщати користувача про можливість виконання замовлення, яке надходить від нього через сайт.

Система буде складатися з декількох модулів. Модулі електронної платіжної системи, база даних MySQL, збір статистики Google будуть додані до системи у готовому вигляді, інші модулі, наприклад модуль сортування буде створений в ході розробки вебзастосунку.

Функціональні вимоги:

- звукове сповіщення про статус замовлення;
- розрахунок вартості поїздки;
- підбір вільних авто;
- розрахунок часу прибуття;
- індексування користувачів.

4.2 Вимоги до інтерфейсу

Інтерфейс повинен бути в фірмових кольорах, інтуїтивно зрозумілий, та в приємних кольорах. Всі послуги та акції компанії повинні бути перераховані на вебсайті.

4.3 Умови виконання програми

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини мобільного застосунку:

- продуктивний хостинг;

- вебсервер Apache;
- підтримка PHP (починаючи з версії 5.4) та WordPress.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК з графічною ОС;
- процесор із тактовою частотою від 1,8 GHz;
- обсяг оперативної пам'яті від 3GB;
- підключення до мережі Internet.

4.4 Розгортання застосунку

Перше вікно застосунку наведено на рис. 4.1.



Рисунок 4.1 – Вхід до системи

Структура вебзастосунку на основі використання обраної CMS.

Архітектура серверу, на якому розгортається застосунок.

Операційна система: Linux CentOS 6.7 x86_64;

Веб-сервер: Apache+nginx(reverse proxy);

Сервер бази даних: MySQL 5.1.73 Source distribution;

Версія PHP: 5.3.3. + модуль Alt-php.

Порядок встановлення та налаштування CMS наведено на рис.4.2.

 wp-admin	4 KiB	755 [drwxr-xr-x]
 wp-content	4 KiB	755 [drwxr-xr-x]
 wp-includes	4 KiB	755 [drwxr-xr-x]
 1462_70538434e6aef29cf4c9c8e6450311ef.zip	4.11 MiB	644 [-rw-r--r--]
 index.html	2.94 KiB	644 [-rw-r--r--]
 index.php	418	644 [-rw-r--r--]
 license.txt	19.46 KiB	644 [-rw-r--r--]
 readme.html	10.61 KiB	644 [-rw-r--r--]
 wordpress-4.4.2-ru_RU.tar.gz	7.23 MiB	644 [-rw-r--r--]
 wp-activate.php	4.92 KiB	644 [-rw-r--r--]
 wp-blog-header.php	271	644 [-rw-r--r--]
 wp-comments-post.php	1.34 KiB	644 [-rw-r--r--]
 wp-config-sample.php	4.08 KiB	644 [-rw-r--r--]
 wp-cron.php	3.21 KiB	644 [-rw-r--r--]
 wp-links-opml.php	2.32 KiB	644 [-rw-r--r--]
 wp-load.php	3.24 KiB	644 [-rw-r--r--]
 wp-login.php	32.98 KiB	644 [-rw-r--r--]
 wp-mail.php	7.70 KiB	644 [-rw-r--r--]
 wp-settings.php	12.72 KiB	644 [-rw-r--r--]
 wp-signup.php	27.92 KiB	644 [-rw-r--r--]
 wp-trackback.php	3.94 KiB	644 [-rw-r--r--]
 xmlrpc.php	2.99 KiB	644 [-rw-r--r--]

Рисунок 4.2 – Розгорнутий CMS WordPress на сервері

Розширити функціонал вебзастосунку можна за допомогою модулів. Для встановлення модуля скористаємося посиланням <http://t-vlad.com/wp-admin/plugins.php>.

Модулі, що були встановлені:

Akismet – цей модуль захищає наш сайт від спаму в коментарях і зворотні посилання.

Wordfence Security – Антивірус, брандмауер і високошвидкісний кеш.

Advanced Custom Fields – для додавання полів різних типів в контент типи.

Taxi Fare Calculator - TaxiFareFinder for WordPress – розрахунок вартості поїздки.

4.4.1Звернення до програми

Звертання до програми від користувача організовано через відправлення запиту про перехід на сайт, використовуючи адресний рядок у веббраузері. Також звертання до програми відбувається, коли адміністратор змінює налаштування або виконує операції з користувачами.

4.4.2Вхідні й вихідні дані

В якості вхідних даних виступають дані, які надходять від віддаленого клієнта про запитувану сторінку або про виконання деяких дій, а вихідними даними є результати обробки запитів клієнта і формування вебсторінки.

4.5 Методика та результати тестування

Під час тестування було обрано стратегію поетапного виконання алгоритму користування програмою.

На рис. 4.3 наведено зовнішній вигляд вебзастосунку після встановлення шаблону.

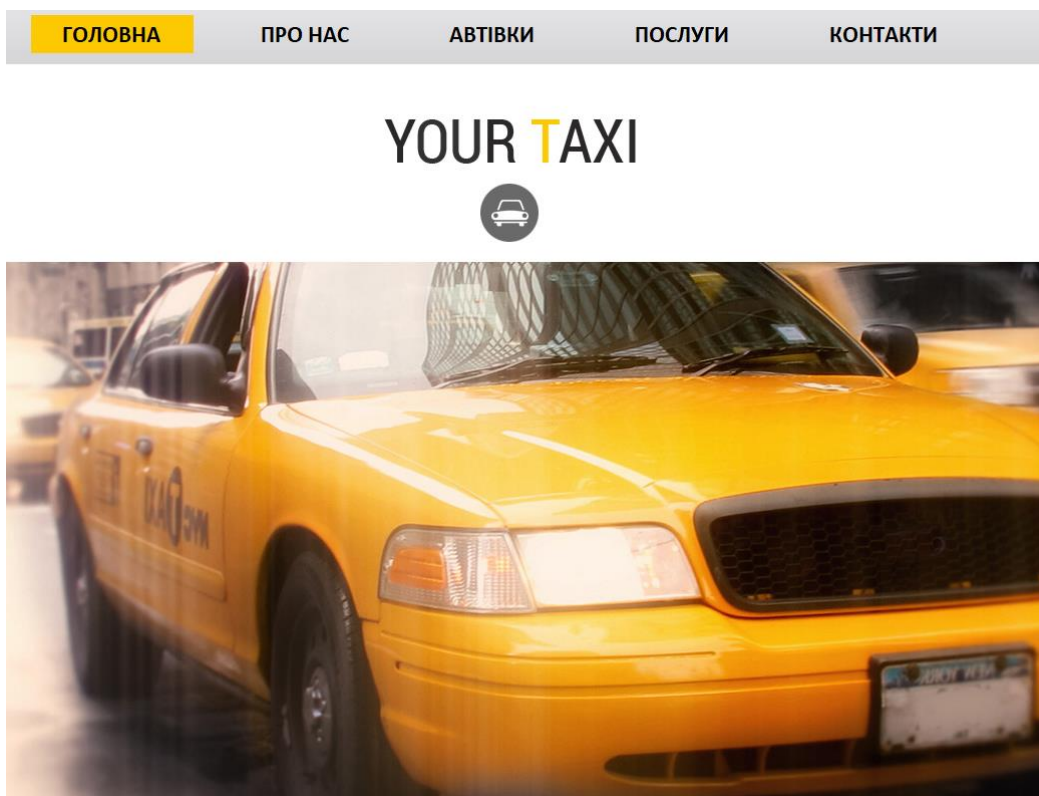


Рисунок 4.3 – Зовнішній вигляд вебзастосунку після встановлення шаблону

На рис. 4.4 наведено сторінку меню розробленої програми.

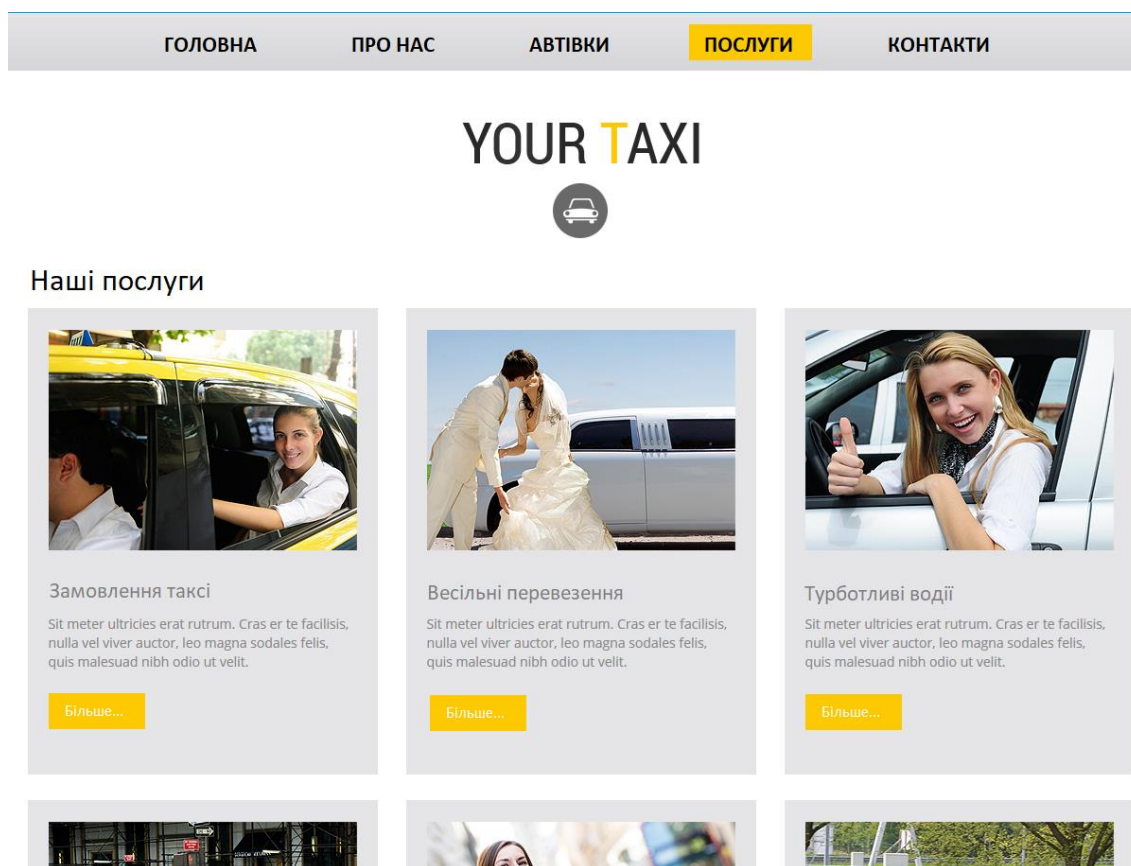


Рисунок 4.4 – Сторінка меню

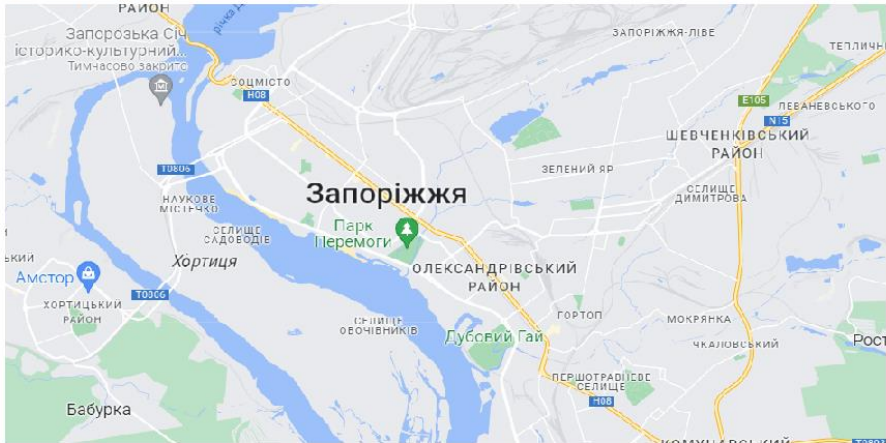
На рис. 4.5 наведено зовнішні сторінки контактів розробленої програми.

ГОЛОВНА
ПРО НАС
АВТІВКИ
ПОСЛУГИ
КОНТАКТИ

YOUR TAXI



Знайти нас



Контактна інформація

Lorem ipsum dolor sit amet, consecteturpiscinger elit um dolor sit amet, consecteturpiscinger.

Форма для зв'язку

Name:	E-mail:	Phone:
Message:		

Не забудьте, что TemplateMonster предоставляет круглосуточную

Рисунок 4.5 – Сторінка Контактів

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи бакалавра було реалізовано головну мету – програмну реалізацію інформаційного ресурсу служби таксі. Для цього було виконано ряд завдань:

- проаналізувати сучасний стан області транспортних застосунків;
- дослідити існуючі реалізації застосунки для замовлення перевезень;
- з дослідження виділити переваги та недоліки таких рішень;
- сформулювати список вимог до нового рішення;
- визначивши засоби реалізації, виконати проєктування нового застосунку;
- виконати розробку та тестування застосунку на основі проєктування;
- скласти необхідну програмну документацію за результатами усіх етапів: аналітика вимог, вибір технічних засобів реалізації, проєктування реалізації, розробка та тестування.

Розроблена структура вебзастосунку була реалізована із використанням PHP CMS – WordPress.

Був розроблений сценарій та проведено тестування розробленого вебзастосунку з використанням сценарію тестування на основі основної логіки використання застосунку. Отриманий результат свідчить про те, що тестування дало позитивний результат.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Вдовиченко, В. О. Пасажирські автомобільні перевезення [Текст] : навч. посіб. / В. О. Вдовиченко, Н. В. Потаман ; Харків. нац. автомоб.-дорож. ун-т. - Харків : ХНАДУ, 2017. - 335 с. : рис., табл. - Бібліогр.: с. 328-335.
2. Пшінько О. М. Сталий розвиток транспорту і пасажирські перевезення у дальньому сполученні [Текст] / О. М. Пшінько, Ю. С. Бараш, Н. І. Верхоглядова, І. П. Корженевич, А. В. Логвиненко, Л. М. Лобойко // Вісник Дніпропетровського національного університету залізничного транспорту імені академіка В. Лазаряна. - 2008. - Вип. 22. - С. 254-257.
3. Бабій О. М. Економічне обґрунтування розміру прибутку в тарифах на пасажирські автотранспортні перевезення [Текст] / О. М. Бабій, К. Х. Сербінова // Вісник соціально-економічних досліджень. - 2013. - Вип. 3(2). - С. 13-18.
4. Барабаш О. О. Механізм визначення оптимального рівня тарифів на пасажирські перевезення [Текст] / О. О. Барабаш, М. В. Богаченко // Науковий вісник Ужгородського університету. Серія : Економіка. - 2015. - Вип. 1(3). - С. 23-28.
5. Крайник Л. В. "Маршрутні таксі" в Україні та структурні зміни типу парку автобусів [Текст] / Л. В. Крайник, А. Редзюк // Вісник машинобудування та транспорту. - 2015. - № 1. - С. 27-32.
6. Згоба М. І. Прогнозування попиту на пасажирські перевезення таксі методами нейронної мережі [Текст] / М. І. Згоба, Ю. І. Грицюк // Науковий вісник НЛТУ України. - 2021. - Т. 31, № 3. - С. 109-119.
7. Ананко І. М. Визначення вимог пасажирів до автотранспортних послуг, що надаються маршрутними таксі м. Харкова [Текст] / І. М. Ананко // Вісник економіки транспорту і промисловості. - 2011. - № 34. - С. 62.
8. Гребняк М. П. Екологічні аспекти праці водіїв маршрутних таксі [Текст] / М. П. Гребняк, Р. А. Федорченко // Український журнал з проблем медицини праці. - 2017. - № 1. - С. 31-38.

9. Uber / [Електронне джерело] – Режим доступу:
<https://www.uber.com/ua/uk/>
10. The History of Uber / [Електронне джерело] – Режим доступу:
<https://www.investopedia.com/articles/personal-finance/111015/story-uber.asp>
11. Bolt / [Електронне джерело] – Режим доступу: <https://bolt.eu/uk-ua/>
12. Bolt history / [Електронне джерело] – Режим доступу:
<https://golden.com/wiki/Bolt-39ANYWK>
13. Uklon / [Електронне джерело] – Режим доступу: <https://uklon.com.ua/>
14. AI gives Ukrainian internet taxi start-up real competitive edge /
[Електронне джерело] – Режим доступу: <https://customers.microsoft.com/en-us/story/uklon-transportation-powerbi-azure-machine-learning-ukraine>
15. PHP / [Електронне джерело] – Режим доступу: <https://www.php.net/>
16. Everything You Need to know about PHP Development / [Електронне джерело] – Режим доступу: <https://www.clariontech.com/php-development>
17. How to Become a PHP Developer? / [Електронне джерело] – Режим доступу: <https://www.simplilearn.com/how-to-become-a-php-developer-article>
18. PHP Development: The Complete Guide to Getting Started /
[Електронне джерело] – Режим доступу: <https://www.trio.dev/blog/php-development-guide>
19. WordPress / [Електронне джерело] – Режим доступу:
<https://wordpress.com/>
20. Український WordPress / [Електронне джерело] – Режим доступу:
<https://wordpress.co.ua/>
21. WordPress що це? / [Електронне джерело] – Режим доступу:
<https://freehost.com.ua/ukr/faq/wiki/wordpress-chto-eto/>
22. Automated Prototyping of User Interfaces based on UML Scenarios /
[Електронне джерело] – Режим доступу:
<http://www.iro.umontreal.ca/~keller/Publications/Papers/2006/ase-2006.pdf>

23. UML Activity Diagram Semantics and Automated GUI Prototyping /

[Електронне джерело] – Режим доступу:

<https://core.ac.uk/download/pdf/55336476.pdf>

ДОДАТОК А
Технічне завдання

Вступ

Область застосування програмного забезпечення – будь-яке підприємство, що потребує систему служби таксі.

Тема для проектування системи – розробка вебзастосунку служби таксі.

Умовне позначення теми розробки – система YourTaxi.

При наявності Інтернету і при наявності будь-якого браузера, користувач може перейти на сайт ознайомитися з потрібною інформацією.

A.1 Підстава для розробки

Підставою для розробки є завдання на дипломну кваліфікаційну роботу на тему «Програмна реалізація інформаційного ресурсу служби таксі», затверджене наказом Національного університету «Запорізька політехніка» № 87 від 04 квітня 2022 р.

A.2 Основні вимоги до програми, що розробляється

Головним завданням роботи є створення системи, яка дозволяє користувачу замовити таксі через вебсайт.

Система повинна сповіщати користувача про можливість виконання замовлення, яке надходить від нього через сайт.

Система буде складатися з декількох модулів. Модулі електронної платіжної системи, база даних MySQL, збір статистики Google будуть додані до системи у готовому вигляді, інші модулі, наприклад модуль сортування буде створений в ході розробки вебзастосунку.

A.2.1 Вимоги до функціональних характеристик

Функціональні вимоги:

- звукове сповіщення про статус замовлення;
- розрахунок вартості поїздки;
- підбір вільних авто;
- розрахунок часу прибуття;
- індексування користувачів.

A.2.2 Вимоги до інтерфейсу програми

Інтерфейс повинен бути в фірмових кольорах, інтуїтивно зрозумілий, та в приємних кольорах. Всі послуги та акції компанії повинні бути перераховані на вебсайті (рис. А.1).

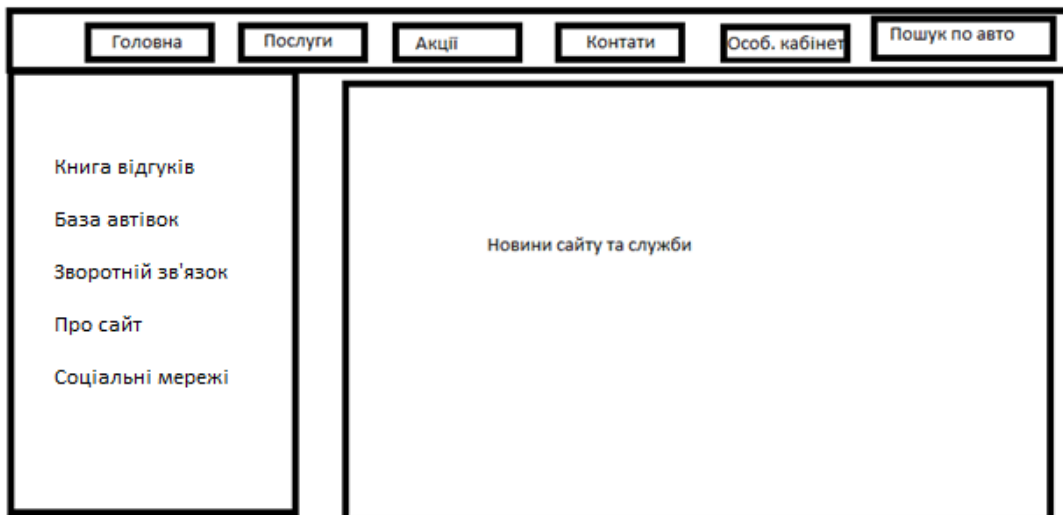


Рисунок А.1 – Макет інтерфейсу

A.2.3 Вимоги до надійності

Дані платіжної системи, локальне сховище бази даних користувачів, інформація про користувачів будуть використовувати потоковий шифр, для забезпечення безпеки користувачів системи.

A.2.4 Умови експлуатації

Для коректної роботи системи на комп'ютерах користувачів повинно бути:

- будь-яка операційна система (ОС);
- процесор 1.8 GHz +;
- підключення до мережі Інтернет.

Також для стабільної роботи системи повинен бути як мінімум один комп'ютер-сервер та людина, що буде контролювати його коректну роботу.

A.2.5 Вимоги до складу та параметрів технічний засобів

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини мобільного застосунку:

- продуктивний хостинг;
- вебсервер Apache;
- підтримка PHP (починаючи з версії 5.4) та WordPress.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК з графічною ОС;
- процесор із тактовою частотою від 1,8 GHz;
- обсяг оперативної пам'яті від 3GB;
- підключення до мережі Internet.

A.3 Вхідні дані до роботи

В якості вхідних даних виступають дані, які надходять від віддаленого клієнта про запитовану сторінку або про виконання деяких дій, а вихідними даними є результати обробки запитів клієнта і формування вебсторінки.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Головним завданням роботи є створення системи, яка дозволяє користувачу замовити таксі через вебсайт.

Система повинна сповіщати користувача про можливість виконання замовлення, яке надходить від нього через сайт.

Система буде складатися з декількох модулів. Модулі електронної платіжної системи, база даних MySQL, збір статистики Google будуть додані до системи у готовому вигляді, інші модулі, наприклад модуль сортування буде створений в ході розробки вебзастосунку.

Б.2 Функціональне призначення

Функціональні вимоги:

- звукове сповіщення про статус замовлення;
- розрахунок вартості поїздки;
- підбір вільних авто;
- розрахунок часу прибуття;
- індексування користувачів.

Б.3 Опис логічної структури

Система складається з двох частин:

- клієнтська;
- серверна.

Загальна схема системи за обраною темою на рис. Б.1.

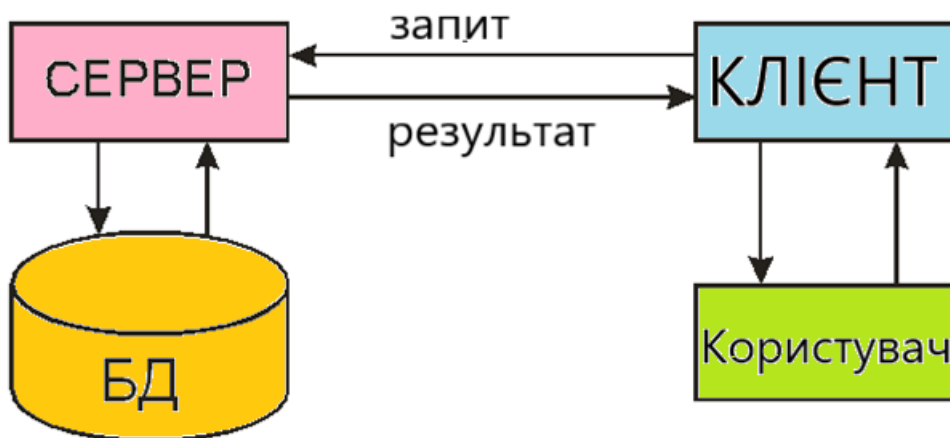


Рисунок Б.1 – Загальна схема системи за обраною темою

Архітектура вебзастосунку базується на моделі Модель-вид-контролер (MVC).

Модель здійснює роботу з базою даних та саме у моделях повинні бути зосереджені методи роботи з даними.

Контролер оброблює дані з моделі, якщо вони потрібні, і генерує Вид з цими даними, де інформація доноситься до користувача з відповідними графічними елементами. Взаємодія користувача з додатком відбувається через Контролер.

На рис.Б.2 наведено архітектуру вебзастосунку.

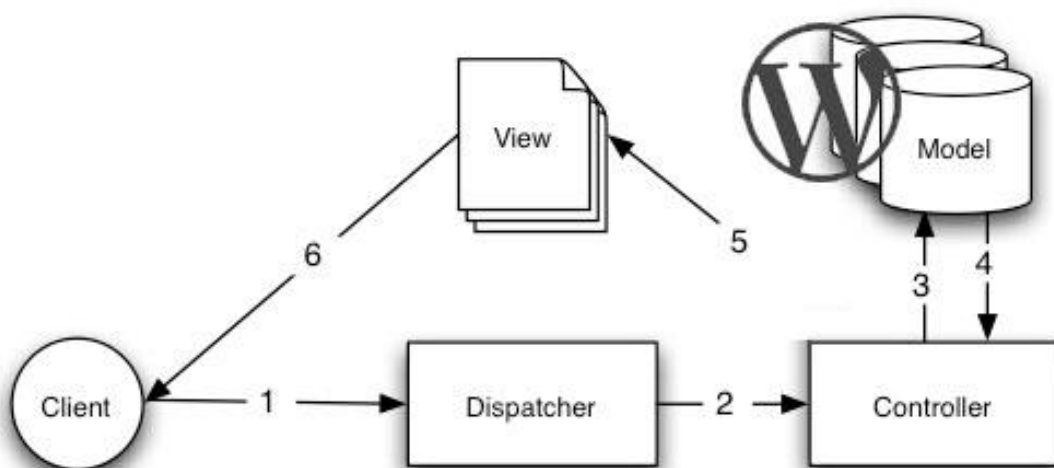


Рисунок Б.2 – Архітектура вебзастосунку

Б.4 Використані технічні засоби

Мінімальні системні вимоги та ПЗ, що необхідні для функціонування серверної частини мобільного застосунку:

- продуктивний хостинг;
- вебсервер Apache;
- підтримка PHP (починаючи з версії 5.4) та WordPress.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- ПК з графічною ОС;
- процесор із тактовою частотою від 1,8 GHz;
- обсяг оперативної пам'яті від 3GB;
- підключення до мережі Internet.

Б.6 Вхідні та вихідні дані

В якості вхідних даних виступають дані, які надходять від віддаленого клієнта про запитувану сторінку або про виконання деяких дій, а вихідними даними є результати обробки запитів клієнта і формування вебсторінки.

ДОДАТОК В
Текст програми

```

<html lang="en">
  <head>
    <title>Home</title>
    <meta charset="utf-8">
    <meta name = "format-detection" content =
"telephone=no" />
    <link rel="icon" href="images/favicon.ico">

    <script>
      $(document).ready(function() {
        jQuery('#camera_wrap').camera({
          loader: false,
          pagination: false ,
          minHeight: '444',
          thumbnails: false,
          height: '28.28125%',
          caption: true,
          navigation: true,
          fx: 'mosaic'
        });
        $().UItoTop({ easingType: 'easeOutQuart'
      });
    });
  </script>
</head>
<body class="page1" id="top">
  <div class="main">

    <!--
=====header=====
=====-->

    <header>
      <div class="menu_block ">
        <div class="container_12">
          <div class="grid_12">
            <nav class="horizontal-nav
full-width horizontalNav-notprocessed">

```

```

        <ul class="sf-menu">
            <li
class="current"><a href="index.html">Головна</a></li>
            <li><a
href="index-1.html">Про нас</a></li>
            <li><a
href="index-2.html">Автівки</a></li>
            <li><a
href="index-3.html">Послуги</a></li>
            <li><a
href="index-4.html">Контакти</a></li>
        </ul>
    </nav>
    <div class="clear"></div>
</div>
    <div class="clear"></div>
</div>
<div class="container_12">
    <div class="grid_12">
        <h1>
            <a href="index.html">
                
            </a>
        </h1>
    </div>
</div>
    <div class="clear"></div>
</header>
<div class="slider_wrapper ">
    <div id="camera_wrap" class="">

        </div>
</div>
<div class="container_12">

```

```

<div class="grid_4">
  <div class="banner">
    <div class="maxheight">
      <div class="banner_title">
        
        <div
class="extra_wrapper">Швидко&amp;
        <div
class="color1">Зручно</div>
      </div>
    </div>
    <a href="#" class="fa fa-
share-square"></a>
  </div>
</div>
<div class="grid_4">
  <div class="banner">
    <div class="maxheight">
      <div class="banner_title">
        
        <div
class="extra_wrapper">Кращі
        <div
class="color1">Ціни</div>
      </div>
    </div>
    <a href="#" class="fa fa-
share-square"></a>
  </div>
</div>
<div class="grid_4">
  <div class="banner">

```

```

        <div class="maxheight">
            <div class="banner_title">
                
                <div
class="extra_wrapper">Пеpeвiз
                    <div
class="color1">Baraxy</div>
                </div>
            </div>
            <a href="#" class="fa fa-
share-square"></a>
        </div>
    </div>
</div>
<div class="clear"></div>
</div>
<div class="c_phone">
    <div class="container_12">
        <div class="grid_12">
            <div class="fa fa-
phone"></div>+ 1800 559 6580
                <span>Замовити зараз</span>
            </div>
            <div class="clear"></div>
        </div>
    </div>
<!--=====Content=====-->
    <div class="content"><div class="ic"></div>
        <div class="container_12">
            <div class="grid_5">
                <h3>Форма замовлення</h3>
                <form id="bookingForm">
                    <div class="fl1">
                        <div class="tmInput">

```

```

<input
name="Name"          placeholder="Имя:"          type="text"          data-
constraints='@NotEmpty @Required @AlphaSpecial'>
</div>
<div class="tmInput">
    <input
name="From"          placeholder="From:"          type="text"          data-
constraints="@NotEmpty @Required ">
</div>
</div>
<div class="fl1">
    <div class="tmInput">
        <input
name="Email"          placeholder="Email:"          type="text"          data-
constraints="@NotEmpty @Required @Email">
</div>
<div class="tmInput
mr0">
    <input name="To"
placeholder="To:"          type="text"          data-constraints="@NotEmpty
@Required">
</div>
</div>
<div class="clear"></div>
<strong>Time</strong>
<div class="tmInput">
    <input name="Time"
placeholder=""          type="text"          data-constraints="@NotEmpty
@Required">
</div>
<div class="clear"></div>
<strong>Date</strong>
<label
class="tmDatepicker">

```

```

<input type="text"
name="Date" placeholder='20/05/2014' data-
constraints="@NotEmpty @Required @Date">
</label>
<div class="clear"></div>
<div class="tmRadio">
<p>Comfort</p>
<input name="Comfort"
type="radio" id="tmRadio0" data-
constraints='@RadioGroupChecked(name="Comfort",
groups=[RadioGroup])' checked/>
<span>Cheap</span>
<input name="Comfort"
type="radio" id="tmRadio1" data-
constraints='@RadioGroupChecked(name="Comfort",
groups=[RadioGroup])' />
<span>Standart</span>
<input name="Comfort"
type="radio" id="tmRadio2" data-
constraints='@RadioGroupChecked(name="Comfort",
groups=[RadioGroup])' />
<span>Lux</span>
</div>
<div class="clear"></div>
<div class="fl1 fl2">
<em>Дорослі</em>
<select name="Adults"
class="tmSelect auto" data-class="tmSelect tmSelect2" data-
constraints="">
<option>1</option>
<option>1</option>
<option>2</option>

```



```

$password = 'passbypass';
$databse = 'taxi_app_db';

// Create a new connection
$conn = new mysqli($host, $username, $password, $database);

// Check the connection
if ($conn->connect_error) {
    die('Connection failed: ' . $conn->connect_error);
}

// Function to insert a new client
function insertClient($name, $phone, $address)
{
    global $conn;

    $sql = "INSERT INTO clients (name, phone, address)
VALUES ('$name', '$phone', '$address')";

    if ($conn->query($sql) === TRUE) {
        echo "Client inserted successfully.";
    } else {
        echo "Error inserting client: " . $conn->error;
    }
}

// Function to retrieve client data
function getClient($clientID)
{
    global $conn;

    $sql = "SELECT * FROM clients WHERE id = $clientID";
    $result = $conn->query($sql);

    if ($result->num_rows > 0) {
        $client = $result->fetch_assoc();
    }
}

```

```

        echo "Client ID: " . $client['id'] . "<br>";
        echo "Name: " . $client['name'] . "<br>";
        echo "Phone: " . $client['phone'] . "<br>";
        echo "Address: " . $client['address'] . "<br>";
    } else {
        echo "Client not found.";
    }
}

// Function to update client details
function updateClient($clientID, $name, $phone, $address)
{
    global $conn;

    $sql = "UPDATE clients SET name = '$name', phone =
'$phone', address = '$address' WHERE id = $clientID";

    if ($conn->query($sql) === TRUE) {
        echo "Client updated successfully.";
    } else {
        echo "Error updating client: " . $conn->error;
    }
}

// Function to delete a client
function deleteClient($clientID)
{
    global $conn;

    $sql = "DELETE FROM clients WHERE id = $clientID";

    if ($conn->query($sql) === TRUE) {
        echo "Client deleted successfully.";
    } else {
        echo "Error deleting client: " . $conn->error;
    }
}

```

```

}

// Example usage

// Insert a new client
insertClient("John Doe", "1234567890", "123 Street, City");

// Retrieve client information
getClient(1);

// Update client details
updateClient(1, "John Smith", "9876543210", "456 Avenue,
Town");

// Delete a client
deleteClient(1);

// Close the database connection
$conn->close();
?>

        <div class="clear"></div>
    </div>
</div>
</div>
<!--
=====footer=====
=====-->

    <footer>
        <div class="container_12">
            <div class="grid_12">

                <div class="copy">
                    <div class="st1">
                        <div      class="brand">Tour<span
class="color1">T</span>axi </div>
                    </div>

```

```

        </div>
        <div class="clear"></div>
    </div>
</footer>
<script>
    $(function () {
        $('#bookingForm').bookingForm({
            ownerEmail: '#'
        });
    })
    $(function() {
        $('#bookingForm input, #bookingForm
textarea').placeholder();
    });
</script>
</body>
</html>

```

ДОДАТОК Д
Слайди презентації



Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

- Об'єкт дослідження – процес програмної реалізації інформаційного ресурсу служби таксі.
- Предмет дослідження – програмні засоби для реалізації інформаційного ресурсу служби таксі.
- Метою роботи є програмна реалізація інформаційного ресурсу служби таксі у вигляді кросплатформного вебзастосунку.

2

Рисунок Д.2 – Слайд 2

Порівняння існуючих аналогів

Критерії порівняння	Uber	Bolt	Uklon	Your Taxi
Робота в Україні	Частково призупинена	Так	Так	Так
Організація колективних поїздок	Ні	Ні	Ні	Так
Способи оплати поїздок	Картка, готівка	Картка, готівка	Картка, готівка	Картка, готівка
Механізми захисту користувачів	Високий ступінь (спілкування виключно через систему)	Середній ступінь (дані доступні через застосунок)	Середній ступінь (дані доступні через застосунок)	Високий ступінь (спілкування виключно через систему)
Програми підтримки та лояльності для користувачів	Відсутня	Відсутня	Відсутня	На основі даних про поїздки користувача

3

Рисунок Д.3 – Слайд 3

Постановка завдання

Головним завданням роботи є створення системи, яка дозволяє користувачу замовити таксі через вебсайт.

Система повинна сповіщати користувача про можливість виконання замовлення, яке надходить від нього через сайт.

Система буде складатися з декількох модулів. Модулі електронної платіжної системи, база даних MySQL, збір статистики Google будуть додані до системи у готовому вигляді, інші модулі, наприклад модуль сортування буде створений в ході розробки вебзастосунку.

4

Рисунок Д.4 – Слайд 4

Порівняння мов програмування

Критерій	PHP	Python	ASP.NET
Надбудови для структуризації розробки	Багато	Мало	Мало
Серверна підтримка	Розповсюджена	Недостатньо розповсюджена	Недостатньо розповсюджена
Масштабованість розробок	Проста	Проста	Проста
Складність підтримки	Низька	Середня	Середня

5

Рисунок Д.5 – Слайд 5

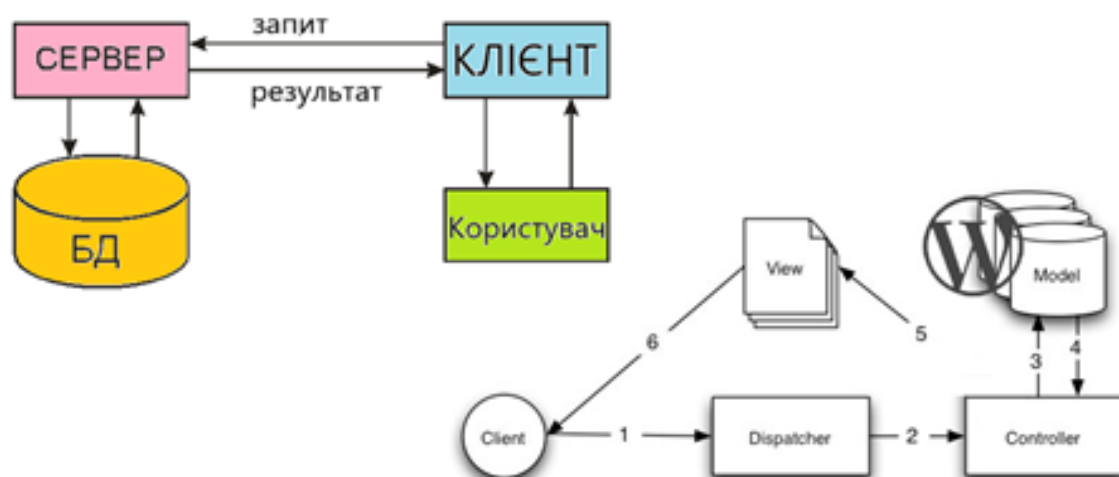
Порівняння CMS

	WordPress	October
Швидкість та простота розробки	Проста та швидка розробка	Ускладнена розробка
Модульна структура	Так	Так
Використання сторонніх модулів	Так (є офіційний каталог протестованих розширень)	Є внутрішній каталог (обмежений)
Захищеність даних	Висока	Середня
Серверна підтримка	Активна	Активна

6

Рисунок Д.6 – Слайд 6

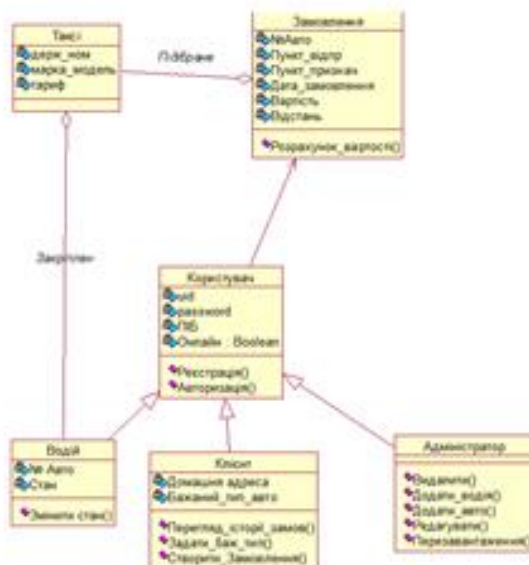
Загальна схема застосунку



7

Рисунок Д.7 – Слайд 7

Діаграма класів



8

Рисунок Д.8 – Слайд 8

Діаграма прецедентів

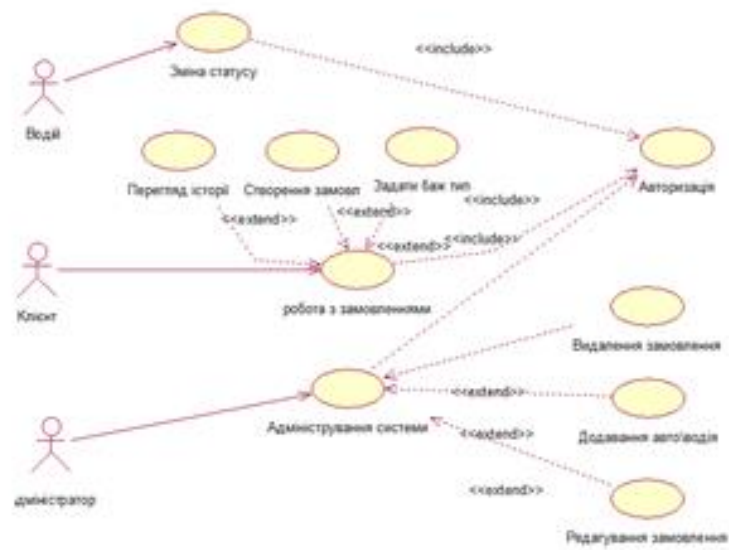


Рисунок Д.9 – Слайд 9

Моделювання взаємодій

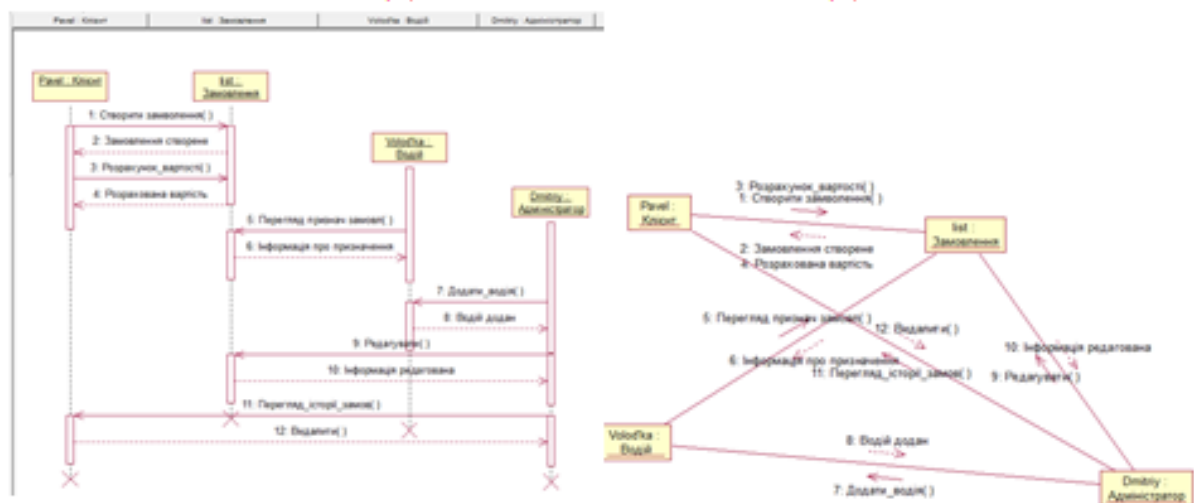


Рисунок Д.10 – Слайд 10

Інтерфейс застосунку

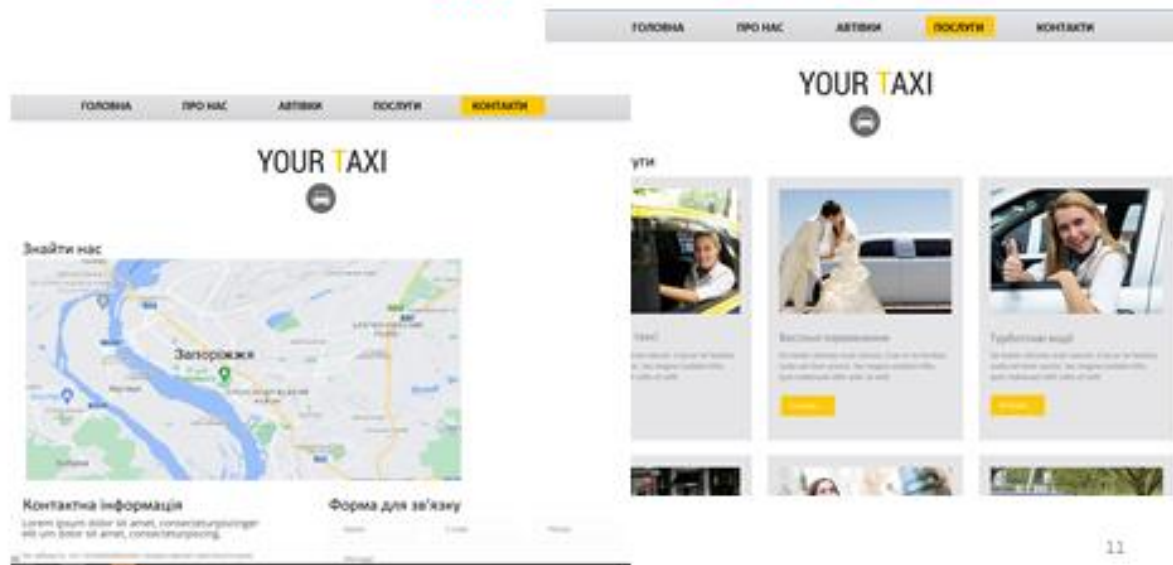


Рисунок Д.11 – Слайд 11

Висновки

В ході виконання дипломної кваліфікаційної роботи бакалавра було реалізовано головну мету – програмну реалізацію інформаційного ресурсу служби таксі. Для цього було виконано ряд завдань:

- проаналізувати сучасний стан області транспортних застосунків;
- дослідити існуючі реалізації застосунків для замовлення перевезень;
- з дослідження виділити переваги та недоліки таких рішень;
- сформулювати список вимог до нового рішення;
- визначивши засоби реалізації, виконати проектування нового застосунку;
- виконати розробку та тестування застосунку на основі проектування;
- скласти необхідну програмну документацію за результатами усіх етапів: аналітика вимог, вибір технічних засобів реалізації, проектування реалізації, розробка та тестування.

В результаті виконання завдання було отримано систему, що відповідає всім вимогам визначеним на початку розробки та закріпленим в технічному завданні. Розроблена система є зручною для використання та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи бакалавра повністю виконано.

Рисунок Д.12 – Слайд 12