

Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра системного аналізу та інформаційних технологій

ІНФОРМАЦІЙНА СИСТЕМА МОНІТОРИНГУ ПАСАЖИРСЬКИХ АВТОПЕРЕВЕЗЕНЬ

Пояснювальна записка до магістерської кваліфікаційної роботи

Виконав: студент 2 курсу, групи 2ІСТ-19м
спеціальності 126 – «Інформаційні системи
та технології»
Машницький П.П.

Керівник: к.т.н., доц. Крижановський
Є.М. _____

Рецензент: к.т.н., доц. Паламарчук Є. А.

Вінниця ВНТУ – 2020 року

Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики
Кафедра системного аналізу та інформаційних технологій

Освітньо-кваліфікаційний рівень магістр

Спеціальність 126 - Інформаційні системи та технології

ЗАТВЕРДЖУЮ

Завідувач кафедри САІТ

_____ д.т.н., проф. В. Б. Мокін

“__” _____ 2020 р.

ЗАВДАННЯ

на магістерську кваліфікаційну роботу студенту
Машницькому Павлу Петровичу

1. Тема роботи: «Інформаційна система моніторингу пасажирських автоперевезень»,
керівник роботи: Крижановський Є. М., к.т.н., доц. каф. САІТ,
затверджені наказом закладу вищої освіти від “__” _____ 2020 року №__
2. Строк подання студентом роботи _____
3. Вихідні дані до роботи:
 - перелік атрибутивних даних про транспортні засоби;
 - перелік атрибутивних даних про розклад руху автобусів Вінницької області;
 - узагальнена електронна карта України.
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
 - обґрунтування проблеми створення інформаційної системи моніторингу пасажирських автоперевезень;
 - формалізація моделі інформаційної системи моніторингу пасажирських автоперевезень;
 - розробка інформаційної системи пасажирських автоперевезень.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень):
 - загальна структура інформаційної системи пасажирських автоперевезень;
 - діаграма варіантів використання;
 - діаграма діяльності для користувацької частини;
 - діаграма діяльності для адміністративної частини;
 - діаграма послідовності;
 - реляційна структура бази даних;
 - загальний вигляд інтерфейсу веб-ресурсу;
 - вигляд спливаючого вікна.

6. Консультанти розділів МКР

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
4	Руда Л.П., к.е.н., доц. каф. ЕПВМ		

7. Дата видачі завдання _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів МКР	Строк виконання етапів роботи	Примітка
1	Аналіз предметної області		
2	Огляд проблем створення інформаційної системи моніторингу пасажирських автоперевезень		
3	Розробка інформаційної системи		
4	Економічна частина		
5	Реалізація електронної карти		
6	Розробка інструкції користувача		
7	Оформлення матеріалів до захисту МКР		

Студент _____

Машницький П. П.

Керівник роботи _____

Крижановський Є. М.

Рецензент _____

Паламарчук Є. А.

РЕФЕРАТ

Магістерська кваліфікаційна робота: 93 стор., 5 табл., 54 рис., 19 джерел.

Об'єкт досліджень – процес створення інформаційної системи моніторингу пасажирських автоперевезень.

Мета роботи – розробка інформаційної системи моніторингу пасажирських автоперевезень.

В роботі звернено увагу на проблему моніторингу автоперевезення в регіоні. Запропоновані технології, які допоможуть оптимально реалізувати дану систему з точки зору можливості її використання на персональних комп'ютерах із вільним для розповсюдження програмним забезпеченням. Також запропоновано структуру веб-системи моніторингу пасажирських автоперевезень та описано функціональне призначення кожної складової. Здійснено реалізацію веб системи моніторингу пасажирських автоперевезень, а також розроблено рекомендації по її впровадженню та експлуатації.

ІНФОРМАЦІЙНА СИСТЕМА, МОНІТОРИНГ, РОЗРАХУНОК РЕЙТИНГУ, АВТОПЕРЕВЕЗЕННЯ.

ABSTRACT

Master's qualification work: 93 pages, 5 tables, 54 pictures, 19 sources.

The object of research is the process of creating an information system for monitoring passenger traffic.

The purpose of the work is to develop an information system for monitoring passenger traffic.

The paper draws attention to the problem of monitoring road transport in the region. Technologies are proposed that will help to optimally implement this system in terms of the possibility of its use on personal computers with free software. The structure of the web system for monitoring passenger traffic is also proposed and the functional purpose of each component is described. The implementation of a web system for monitoring passenger road transport has been implemented, as well as recommendations for its implementation and operation have been developed.

**INFORMATION SYSTEM, MONITORING, RATING CALCULATION,
ROAD TRANSPORT.**

ЗМІСТ

ВСТУП	8
1 ОБҐРУНТУВАННЯ ПРОБЛЕМИ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ПАСАЖИРСЬКИХ АВТОПЕРЕВЕЗЕНЬ.....	10
1.1 Аналіз предметної області.....	10
1.2 Огляд існуючих систем моніторингу автоперевезень.....	11
1.3 Висновки	17
2 ФОРМАЛІЗАЦІЯ МОДЕЛІ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ПАСАЖИРСЬКИХ АВТОПЕРЕВЕЗЕНЬ	18
2.1 Ідея інформаційної системи моніторингу пасажирських автоперевезень.....	18
2.2 Формалізація моделі рейтингу водіїв.....	19
2.3 Огляд та вибір технологій для розробки інформаційної системи моніторингу пасажирських автоперевезень	20
2.4 Висновки	24
3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПАСАЖИРСЬКИХ АВТОПЕРЕВЕЗЕНЬ	25
3.1 Моделі інформаційної системи та їх опис.....	25
3.2 Розробка front-end та back-end частини інформаційної системи моніторингу пасажирських автоперевезень	31
3.3 Проектування бази даних системи моніторингу пасажирських автоперевезень.....	44
3.4 Розробка SQL-запиту для вибірки даних.....	48
3.5 Розробка back-end частини для інформаційної системи пасажирських автоперевезень.....	51
3.6 Висновки	52
4 ЕКОНОМІЧНА ЧАСТИНА	54
4.1 Оцінювання комерційного потенціалу розробки «Інформаційна система моніторингу пасажирських автоперевезень»	54
4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи	57

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки	61
4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності	63
4.5 Висновки	66
ВИСНОВКИ	67
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	69
Додаток А – Технічне завдання	71
Додаток Б – Інструкція користувача	73
Додаток В – Лістинг програми	76
Додаток Г – Графічна частина	78

ВСТУП

Актуальність магістерської кваліфікаційної роботи полягає в тому, що дана система допоможе користувачам одержати актуальну інформацію про розташування транспортного засобу, наприклад, при його затримці, зміні фактичного розкладу, тощо для перевізника дана система допоможе відстежувати чи водій не відхиляється від маршруту і чи рухається згідно розкладу руху. Також в системі буде закладена можливість оцінювання перевізників і розрахунок їх рейтингів на основі вказаних оцінок.

Метою даної роботи є розробка інформаційної системи моніторингу пасажирських автоперевезень. Для цього необхідно запропонувати оптимальні технології та формати для реалізації даної системи з точки зору можливості її використання на персональних комп'ютерах із платним ліцензійним забезпеченням, так і з вільним для розповсюдження програмним забезпеченням.

Об'єктом дослідження магістерської кваліфікаційної роботи є процес моніторингу пасажирських автоперевезень у Вінницькій області.

Предметом магістерської кваліфікаційної роботи є технології та засоби розробки інформаційної системи моніторингу пасажирських автоперевезень.

Розробка інформаційної системи моніторингу пасажирських автоперевезень передбачає виконання задач таких етапів:

- розробка загальної архітектури інформаційної системи моніторингу пасажирських автоперевезень;
- вибір оптимальних технології та форматів для реалізації інформаційної системи моніторингу пасажирських автоперевезень;
- розробка бази даних та інтерфейсу користувача для інформаційної системи моніторингу пасажирських автоперевезень;
- розробка моделі рейтингування водіїв;
- створення веб-ресурсу інформаційної системи моніторингу пасажирських автоперевезень.

Наукова новизна одержаних результатів. Дістала подальший розвиток спеціалізована модель розрахунку рейтингу водіїв транспортних засобів, за рахунок введення гнучкого підходу до задання ваг формування рейтингу, використання якого дозволить покращити сервіс перевізникам, а користувачам вибрати перевізника із найкращим рейтингом.

Практичне значення роботи полягає у можливості використання розробленої інформаційної системи, для отримання певної інформації про транспортний засіб в певний момент часу.

Публікації. В ході виконання магістерської кваліфікаційної роботи опубліковано 2 тез та було взято участь у XLVIII науково-технічній конференції [1].

1 ОБҐРУНТУВАННЯ ПРОБЛЕМИ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ПАСАЖИРСЬКИХ АВТОПЕРЕВЕЗЕНЬ

1.1 Аналіз предметної області

Одним з найкращих способів моніторингу пасажирських автоперевезень є GPS-моніторинг.

GPS-моніторинг — це спеціальна система, розроблена, щоб контролю та спостереження стану рухомого об'єкту в будь-яких час. Системи моніторингу можливо встановити на будь-який автотранспорт, спеціалізовану техніку, водний транспорт та мототехніку [1].

Система повинна містити декілька компонентів GPS-трекера та програмного забезпечення, для взаємодії компонентів. Трекер монтується у певний транспортний засіб, щоб надсилати координати визначені за допомогою супутника та передавати інформацію на спеціалізовані та захищені сервери. Програмне забезпечення призначене для отримання даних і виведення координат на карту, щоб показана розміщення транспорту, та намалювати маршрут, здійснений транспортним засобом за деякий проміжок часу. Крім відображення маршруту та місця розташування, можливо одержувати інформацію де були здійснені зупинки, який був пробіг та витрати палива, а також відхилився від призначеного маршруту [1].

У наш час системи моніторингу транспорту необхідні для кожному транспортному підприємству. Вони допомагають вирішити бізнес завдання, які пов'язані з оптимізацією процесу в організаціях перевізників. Та за допомогою систем моніторингу покращується безпека, водіїв, та вантажів під час руху [2].

Іноді причиною нанесення збитковості компанії, яка надає послуги з перевезення товару є неправильно побудовані трудові процеси. Навіть, коли точно прораховуються, або плануються можливі витрати (кілометраж, паливо,

заробіток водія тощо) для отримання достатнього прибутку, але результат деяких підприємств бажає кращого, вони терплять крах, працюють без прибутку. В результаті відбувається підвищення ціни за послуги, та приносять втрату конкурентоспроможності.

Система для здійснення моніторингу розроблена на основі GPS/GSM легко дає можливість виявити недоліки, що можуть привести до незапланованих витрат:

- Крадіжка палива;
- Невірно побудований маршрут;
- Водії, які отримують заробітну плату почасово стараються робити побільше простоїв.

Виправивши недоліки транспортне підприємство збільшує шанси уберегтися лишніх витрат, та значно покращити прибуток із збереженням цінової політики [3].

Супутникові системи навігації дають можливість:

- Вести контроль за витратами палива;
 - Для різних маршрутів розраховувати оптимальний прибуток;
 - Контролювати водіїв, не даючи можливості водіям здійснювати перевезення, для себе за рахунок підприємства.
- покращення безпеки співробітників та підняти ефективність управління підприємством.

1.2 Огляд існуючих систем моніторингу автоперевезень.

Однією із систем моніторингу є GPSM система, на рисунку 1.1, приведена схема роботи даної системи.



Рисунок 1.1 – Схема роботи GPSM системи

Система працює досить точно та акуратно маючи високу ефективність. На всі транспортні засоби встановлюються спеціальні прилади, так звані GPS - маячки. Дані пристрої записують усі основні параметри роботи автомобілів, а в придачу переміщення та точне розташування. Використовуючи мобільні засоби для зв'язку із системою та передачі, отримання інформації та відображення диспетчеру на його екрані в реальному часі. Що допомагає цілодобово відстежувати де знаходиться, кожен автомобіль [4].

За допомогою отриманих даних про рейс фахівці мають можливість порівняти актуальні дані з певним розрахунком та визначити причини відхилень. Такий аналіз допомагає спланувати більш ефективно робочий процес підприємства та визначити менш затратні маршрути маршруту.

Цілодобовий нагляд за транспортним засобом надає контролювати переміщення, технічний стан автомобіля та надає змогу швидко вирішувати поставлені завдання, та оперативно допомогти під-час непередбачених ситуацій, які можуть виникнути в дорозі. У разі непередбачуваної поломки транспорту, людина, яка контролює володіє актуальною інформацією про автомобіль може відправити послати вчасно допомогу, або події допоміжну машину. Швидке втручання надає можливість швидко доставити вантаж не порушуючи термін поставки. При викраденні автомобіля, можливо швидко

подати інформацію про місце розташування органам, які будуть здійснювати його пошук. Також, система GPSM має функцію віддаленого блокування автомобіля, та деяких його агрегатів для уникнення викраденню [5].

Система моніторингу є недорогою, що забезпечує швидку окупність.

Система GPSM має містить функцій призначених спеціально для підприємств, які займаються перевезенням вантажів. Даний функціонал надає змогу здійснювати контрольні функції під-час знаходження автомобіля в дорозі:

- в реальному часі відстежувати місцезнаходження автомобіля;
- протягом року мати історію переміщення автомобіля;
- кількість установлених міток необмежена;
- різноманітні формати звіту;
- отримання параметрів за допомогою повідомлення відісланого на електронну пошту або на мобільний телефон за допомогою СМС;
- для мобільного телефону або планшета, існує мобільний додаток, що допомагає зручному контролювати усі транспортні засоби;
- можливість контролювати рівень палива;
- прокладання оптимального маршруту;
- для кожного водія ведеться облік даних.

Система freetrack дозволяю повністю отримувати достовірну інформацію для аналізу та ведення статистики, яка є актуальною та точною. Аналіз розроблений на основі даних, які надходять від різних датчиків та контролерів, які розміщені на автомобілях, що дозволяє оператор відслідковувати всі важливі параметри, котрі мають впливають на економічну ефективність [6].

Супутниковий моніторинг – це механізм для покращення роботи підприємств, які займаються перевезеннями, та підвищення прибутку підприємств [7].

Система freetrack може працювати завдяки сучасним технологіям: GPS, GSM, GPRS и Інтернету. В транспортний засіб вмонтовується пристрій з

активованою sim-картою будь-якого українського GSM-оператора. З GPS та GSM-антенами до пристрою можуть бути підключені (за бажанням Клієнта) паливний датчик, тривожна кнопка, сигналізація тощо. Вмонтований пристрій програмується під систему моніторингу freetrack для подальшої обробки інформації. На рисунку 1.2 представлено схему роботи системи freetrack [8].



Рисунок 1.2 – Схема роботи freetrack системи

Принцип дії freetrack:

GPS-приймач вираховує власне місце розташування, вимірюючи час за допомогою отримання сигналу із GPS-супутників. Супутники постійно надсилають повідомлення, де знаходиться інформація про час, коли було відправлено повідомлення, положення супутника, від якого отримано повідомлення (ефемерис), та стан всієї системи і приблизні дані про положення всіх інших супутників, які входять в угруповання системи GPS (альманах). Дані сигнали мають швидкість світла тому швидко розповсюджуються у всесвіті, трішки повільніше в атмосфері [9].

GSM-пристрій зв'язується з національним оператором (чи будь-яким іншим оператором, якщо транспортний засіб за кордоном – то в роумінгу) і по системі GPRS відсилає координати (відомості про паливо, тривожну кнопку,

температурні режими і т.д.) на сервер freetrack, де ці дані обробляються (наносяться на карти, формуються екстренні повідомлення, звіти тощо).

Одними із представників закордонних систем моніторингу, які надають послуги моніторингу за вашим автотранспортом є такі компанії Venish GPS. Розглянемо представлення технології компанії Venish GPS.

Динамічна еволюція технологій створює нові умови для існування та розвитку компаній. Сучасні рішення дозволяють вирішувати глобальні завдання економіки та оптимізувати комерційні активи при мінімальних витратах за короткий час. Компанія Venish GPS пропонує унікальні рішення для управління комерційним флотом на основі інновацій GPS [10].

Високотехнологічний моніторинг GPS гарантує вам і вашій діловій інформації про напрямки автомобіля, його поточне розташування та стан.

GPS моніторинг - це комплексне рішення, спрямоване на підвищення прибутковості бізнесу при мінімізації його витрат. Обладнання GPS-системи не вимагає додаткових людських ресурсів або контролю з боку замовника [10].

Безпека. Після впровадження системи GPS-моніторингу автомобіль знаходиться під постійним супутниковим спостереженням. При цьому власник завжди може визначити фактичне розташування транспортного засобу у випадку викрадення.

Організація робочого часу. Впровадження GPS моніторингу комерційного автопарку дозволяє більш ефективно управляти часом і потенціалом працівників.

Оптимізація трафіку маршруту, запобігання несанкціонованих витрат на паливо.

Цілодобова підтримка споживачів GPS-моніторингу.

Інтуїтивний, зручний інтерфейс програми моніторингу GPS.

GPS-контроль відповідності маршруту зазначеним точкам і негайне повідомлення, якщо вони ігноруються.

Можливість адаптації системи до індивідуальних потреб клієнта.

Можливість тестування системи управління та моніторингу транспортного засобу.

GPS моніторинг і управління автопарком є інструментом для управління компанією, орієнтованою на розвиток і прибуток. Європейська практика використання системи моніторингу та контролю автомобілів GPS свідчить про ефективність цього методу оптимізації бізнес-процесів. На рисунку 1.3 представлена схема роботи системи [11].



Рисунок 1.3-Схема роботи Benish GPS системи

Зосередьтеся на партнерських і взаємовигідних відносинах з клієнтом.

Фінансова стійкість організації дозволяє реалізовувати складні проекти, які потребують значних інвестицій сильні відносини співробітництва з Державним департаментом контролю за рухом. Benish GPS розробляє та продає унікальні та інноваційні GPS-рішення, враховуючи специфіку клієнтів обладнання виробляється в країнах Європейського Союзу та Ізраїлю [11].

Обладнання сертифіковане відповідно до міжнародного права і відповідає найвищим вимогам до якості.

1.3 Висновки

В даному розділі було проаналізовано проблеми, пов'язані із розробкою інформаційних систем моніторингу пасажирських автоперевезень. Розглянуто переваги систем контролю для перевізників в цілому та необхідність оснащення ними транспортних засобів.

Проведено огляд існуючих систем моніторингу за транспортними засобами, та зроблено відповідні висновки. Системи мають свої плюси та мінуси, в основному дані системи орієнтовані на великі підприємства та на транспортні перевезення, тому знаючи найкращі їхні сторони добавимо свої функції та зробимо її більш кращою для перевізників та пасажирів, які будуть користуватися системою.

2 ФОРМАЛІЗАЦІЯ МОДЕЛІ ІНФОРМАЦІЙНОЇ СИСТЕМИ МОНІТОРИНГУ ПАСАЖИРСЬКИХ АВТОПЕРЕВЕЗЕНЬ

2.1 Ідея інформаційної системи моніторингу пасажирських автоперевезень

На даний момент існує чимала кількість інформаційних систем стеження за автотранспортом, але мало таких, які б надавали користувачам інформацію про місце знаходження автобусу в певний момент часу. Така можливість дозволяє користувачеві розуміти, чи виконують його співробітники ту роботу, для якої їх наймали, чи використовують службову техніку не за призначенням. Також відстеження машини допоможе у випадку, якщо автобус або маршрутка раптово зламалися, і зв'язатися по мобільному зв'язку у водія немає можливості. Хтось із співробітників може виїхати у вказане системою місце та надати водієві необхідну технічну допомогу [11].

По – перше, для користувача необхідно формувати цікаву, для нього інформацію.

По-друге система має містити інформацію про маршрути, та інформацію про місце знаходження транспортного засобу.

По-третьому система має містити дані про перевізників та водіїв, які працюють в перевізників. Водії мають певний рейтинг, який формується із його оцінок, які оставляють користувачі системи.

Система повинна мати пошуковий модуль, де можливо вибрати маршрут, враховуючи рейтинг водія.

Також система повинна мати адміністративну частину, де можна буде додавати інформацію до системи.

2.2 Формалізація моделі рейтингу водіїв

Побудуємо модель, яка буде обраховувати рейтинг водіїв. Оскільки для водія може бути велика кількість записів із різними характеристиками, тому необхідно для кожної характеристики обрахувати середнє значення. Отже, обрахунок середнього значення, для однієї характеристики водія:

$$\sum \frac{x_i}{i}, \quad (2.1)$$

де x – оцінка водія за певну характеристику, а i – кількість оцінок за одну характеристику.

Після того, як ми знаємо середню оцінку по характеристиці нам необхідно взнати, який відсоток вона займає від загальної кількості, для цього отримане значення за формулою (2.1) поділимо на 10, та поділимо на кількість характеристик, в результаті ми отримали формулу

$$\left(\frac{\sum \frac{x_{1i}}{i}}{10} \right) / n * 100, \quad (2.2)$$

де n – загальна кількість характеристик, а i – кількість оцінок за одну характеристику.

Оскільки в нас декілька характеристик, тому для знаходження рейтингу нам необхідно знайти суму всіх відсотків, що допоможе знайти рейтинг водія, результат показано на формулі

$$R = \left(\sum \left(\left(\frac{\sum \frac{x_{1i}}{i}}{10} \right) \left(\frac{\sum \frac{x_{2i}}{i}}{10} \right) + \left(\frac{\sum \frac{x_{3i}}{i}}{10} \right) + \dots + \left(\frac{\sum \frac{x_{ni}}{i}}{10} \right) \right) \right) / n * 100. \quad (2.3)$$

2.3 Огляд та вибір технологій для розробки інформаційної системи моніторингу пасажирських автоперевезень

Оскільки це буде інформаційна система, яка використовує не одну технологію, для серверної частини та користувацької.

Для розробки серверної частини необхідно обрати мову програмування, для аналізу взято три мови:

- Ruby;
- PHP 7;
- Node.js.

Переваги PHP 7:

- Практичність – дана мова призначена, для допомоги програмісту при виконанні поставленої задачі надаючи можливість. За допомогою своєї великої функціональності PHP добре підходить, щоб вирішити великі спектри завдань;
- Простота – не містить сувору типізацію, в порівнянні із Java або C++, що полегшує вивчення даної мови. Наприклад, коли необхідно створенні змінної нам не обов'язково вказувати для неї тип;
- Традиційність – PHP розроблялася як настройка для Perl, через що мова містить в собі кращі сторони Perl і C. Програмний код PHP нагадує C, що полегшує вивчення цієї мови;
- Ефективність – один з вагомих факторів під-час вибору мови програмування. Через своє "ядро", сценарії в PHP виконуються з досить швидко, що дає можливість створювати серйозні на WEB-додатки PHP [12];
- Гнучкість - оскільки PHP є вмонтованою мовою, що надає можливість гнучкості в процесах розробки. Найчастіше PHP використовуються в HTML сторінках, але при необхідності можуть використовуватись JavaScript, XML, WML та інші мови [13];
- Бази даних – одним з найвагоміших переваг PHP оскільки мають підтримку більше ніж 20 видів баз даних. Також, PHP підтримує DBX, щоб

працювати на абстрактному рівні та ODBC, дозволяє працювати із різними базами даних, які враховують ці стандарти.

Переваги Ruby:

– Стабільність – розробники називають Ruby набагато потужнішим, ніж інші мови програмування Perl і набагато більш об'єктно-орієнтованим, ніж відома мова програмування Python.

– Простота – що стосується початківців, Ruby є кращим серед них оскільки легко вивчити. Він доступний на ринку безкоштовно, а також має широкі можливості копіювання, використання, модифікації або розповсюдження є досить доступними для програмістів, щоб просто продовжувати кодування, не відчуваючи обмежень.

– Швидкість – Ruby швидко пише. Окрім того, що Ruby є найшвидшою мовою з точки зору обробки та запуску запитів, Ruby також може похвалитися тим, що розробники витрачають приблизно на 40 відсотків менше часу на написання коду та розробку програмних продуктів.

– Ruby є об'єктно-орієнтованою мовою. Коли ми створюємо об'єкт з класу, він може отримати власний клас. З цієї причини кожен об'єкт у Ruby дійсно може бути унікальним та мати свої власні методи та властивості. Інші мови програмування часто мають непотрібні обмеження в цьому відношенні.

Переваги Node.js:

– Node.js пропонує легку масштабованість. Однією з ключових переваг Node.js є те, що розробникам легко масштабувати програми в горизонтальному та вертикальному напрямках. Програми можна масштабувати горизонтально, додаючи додаткові вузли до існуючої системи.

– Node.js пропонує розробникам розкіш писати серверні програми в JavaScript. Це дозволяє розробникам Node.js писати як інтерфейсну, так і внутрішню веб-програму в JavaScript, використовуючи середовище виконання. І їм не потрібно використовувати будь-яку іншу мову програмування на стороні сервера. Це також спрощує розгортання веб-додатків, оскільки майже всі веб-браузери підтримують JavaScript;

- Перевага Fullstack полягає в повнотекстовому JavaScript для обслуговування як клієнтських, так і серверних додатків;
- Продуктивність: Node.js інтерпретує код JavaScript за допомогою механізму JavaScript V8 від Google. Цей механізм відповідає коду JavaScript і безпосередньо машинному коду. Це спрощує та пришвидшує ефективне впровадження коду;
- Перевага кешування. Середовище виконання з відкритим кодом Node.js також забезпечує можливість кешування окремих модулів. Щоразу, коли є запит на перший модуль, він потрапляє в кеш пам'яті програми. Розробникам не потрібно повторно виконувати коди, оскільки кешування дозволяє програмам швидше завантажувати веб-сторінки та швидше реагує на користувача [14];
- Пропонує свободу розробки додатків. Ще однією перевагою, яку Node.js пропонує розробникам, є свобода розробки програм та програмного забезпечення. Це одна суттєва особливість, яка залишається відсутнім у Ruby on Rails, що накладає певні обмеження. Ви можете починати все з нуля під час розробки додатків;
- Отримання підтримки для загальноновживаних інструментів. За допомогою Node.js розробники можуть отримати розширену підтримку різних часто використовуваних інструментів. Візьмемо приклад. Припустимо, ви хочете протестувати вихідний код програми Node.js; Ви можете зробити це, використовуючи Jasmine та інші подібні засоби модульного тестування [14];
- Одночасно обробляє запити. Оскільки Node.js надає можливість неблокуючих систем вводу-виводу, це відносно допомагає вам обробляти кілька запитів одночасно. Система може обробляти паралельні обробки запитів ефективніше, ніж інші, включаючи Ruby або Python. Вхідні запити вишикуються і виконуються швидко та систематично.

Отже, перевага полягає в тому, що вам не потрібно наймати окремих розробників для бекенда, а також для інтерфейсної розробки. Це економить і ваші цінні гроші, і час.

Для розробки серверної частини інформаційної системи було вибрано мову Node.js, через її переваги над Ruby і PHP 7 [15].

Для користувацької частини буде використовуватися HTML, CSS і JavaScript. Також будуть використовуватися фреймворки та бібліотеки, для полегшення розробки додатку та пришвидшення роботи, для аналізу виберемо ReactJS і AngularJS, які є гігантами веб-індустрії.

Angular – це фронтенд-фреймворк для розробки веб-додатків, заснований на TypeScript, створений людьми в Google і значним внеском у який є величезна спільнота розробників з усього світу. Це фреймворк односторонньої програми (SPA), що означає, що Angular здатний і в основному використовується для створення односторінкових веб-додатків або SPA.

Angular розробляється тією ж командою людей, яка розробила популярний фреймворк AngularJS. Сучасний Angular вимагає від вас знання мови Typescript, яка є суворим синтаксичним набором JavaScript і додає необов'язкові статичні типи та деякі інші функції до мови, якої в іншому випадку бракує JavaScript [16].

Структура AngularJS була переписана і названа "Angular 2", але це призвело до великої плутанини та паніки у спільноті розробників. Для уточнення команда Angular почала використовувати окремі терміни для кожного фреймворку. На даний момент "AngularJS" тепер посилався на версії 1.X, а "Angular" без "JS" посилався на версії 2 та новіші. Відтоді всі версії фреймворку, крім 1.X, називаються просто Angular. Однак ви можете вказати номер версії, щоб бути точнішим [17].

Переваги ReactJS:

– Простота – використання компонентів React.js забезпечує велику перевагу в розробці. Починаючи з дрібних речей, розробники

використовують їх для створення більших речей, а потім для створення веб-програм. Кожен компонент має власну логіку, контролює власну візуалізацію і може бути використаний повторно, де це потрібно. Можливість розширювати невеликі компоненти, пропонує більшу гнучкість, необхідну для проекту;

– Продуктивність – додатки з великою кількістю взаємодії з користувачами та оновлення даних зазвичай вимагають ретельного розгляду того, як структура програми вплине на продуктивність. Об'єктна модель документа дуже повільно змінюється та оновлюється. Сервер постійно перевіряє різницю, спричинену змінами, щоб надати необхідну відповідь. Щоб правильно відповісти, йому потрібно оновити дерева DOM у цілому документі, що не є ергономічно дійсним: сьогодні дерева DOM містять тисячі елементів. Ця проблема вирішується за допомогою віртуального DOM [18].

Отже, для забезпечення швидкості, простоти та масштабованості, використовуємо React.js, який допомагає створювати та підтримувати складні додатки та допомагає розробникам та покращую взаємодію користувачів з системами. Багаторазові компоненти React.js збільшує швидкість та якість розробки. Поєднуючи швидкість JavaScript з простотою та покращеною продуктивністю, бібліотека може суттєво оптимізувати зусилля веб-розробки.

2.4 Висновки

В даному розділі було сформовано ідею, розробки інформаційних систем моніторингу пасажирських автоперевезень, яка спрямована на допомогу підприємцям та простим користувачам, для вибору водія, із яким можливо добратися до пункту призначення, або проаналізувати його роботу.

Розроблено модель розрахунку рейтингу водіїв транспортного засобу.

Проведено огляд технологій за допомогою яких буде розроблятися система, наведено їх переваги та недоліки та обрано технології за допомогою яких буде реалізована система.

3 РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ПАСАЖИРСЬКИХ АВТОПЕРЕВЕЗЕНЬ

3.1 Моделі інформаційної системи та їх опис

Під час детального аналізу вхідних даних, необхідного функціоналу майбутньої веб-системи, форматів збереження даних та просторових даних розроблено наступну структурну схему веб-системи моніторингу пасажирських автоперевезень (рис. 3.1).

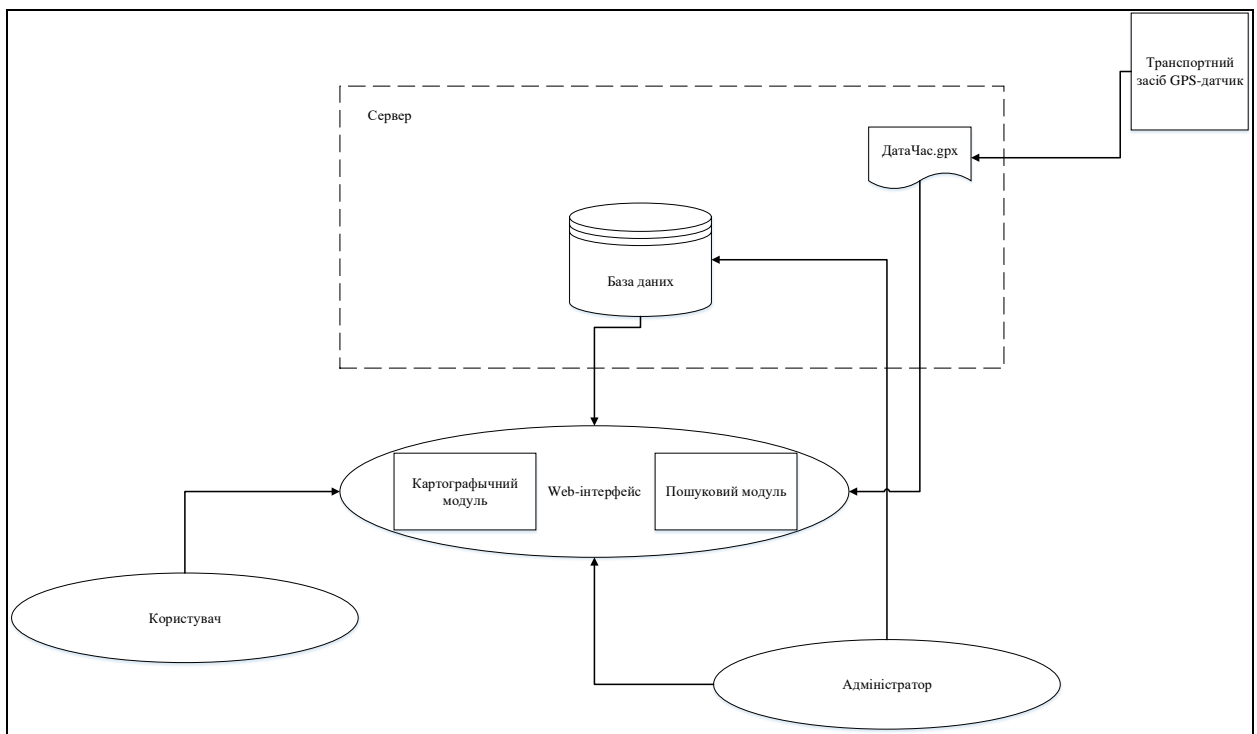


Рисунок 3.1– Структурну схему веб-системи моніторингу пасажирських автоперевезень

Структурна схема системи складається із таких складових:

- GPS-датчик, який розміщений на транспортному засобі, та відсилає дані про місце розташування та час.
- Файли ДатаЧас.gpx, це файли, які зберігають інформацію, яка надходить від GPS-датчика.

- База даних – місце збереження даних системи.
- Web-інтерфейс відповідає за представлення системи користувачу та містить засоби для роботи з: розкладом руху транспортного засобу, місцями розташування транспортного засобу, картографічним модулем, інформацією про перевізника та вокзал, пошуковим модулем.

- Користувач – людина яка користується системою.

- Адміністратор – людина яка відповідає за актуальність даних.

Для того, щоб показати роботу інформаційної системи моніторингу пасажирських автоперевезень, обрано мову моделювання UML.

Для більш кращого представлення переваг та можливостей системи розробимо UML-діаграми:

- діаграму варіантів використання;

- діаграму діяльності;

- діаграму послідовностей.

Діаграма варіантів використання під час розробки програмного забезпечення або в системному проектуванні описує поведінку системи, її відповіді при зовнішніх запитах. Якщо сказати по іншому, використовує опис, «хто» і «що» може робити із системою. Метод застосовується для виявлення вимог, для поведінки системи [19].

Опис діаграми використання:

- Сфера (Scope): інформаційна система пасажирських автоперевезень (System);

- Рівень (Level): User-goal;

- Ключовий актор (Primary Actor): користувач;

- Зацікавлені сторони і інтереси (Stakeholders and interests);

- Користувач: необхідно, як найшвидше знайти необхідну інформацію;

- Адміністратор: необхідно оновлювати дані в системі.

Головний успішний сценарій (Main Success Scenario):

- Користувач запускає систему перейшовши на сайт, для отримання певної інформації.

- Користувач із панелі навігацій обирає необхідний розділ із інформацію.

- Натискає на кнопку «Пошук».

- Користувач вводить параметри пошук.

- Отримує потрібну інформацію

- Задоволений користувач покидає систему, щоб вернутись ще раз.

Розширення сценарію (Extensions):

- Альтернативний сценарій №1 (успіх);

- Користувач натискає на кнопку «Розклад»

- Отримує інформацію про розклад руху автобусів.

- Закриває систему.

- Альтернативний сценарій №2 (успіх);

- Користувач натискає на кнопку «Перевізники»

- Отримує інформацію про перевізників.

- Вибирає інформацію про водія.

- Обирає оцінити водія, оцінює водія.

- Закриває систему.

- Альтернативний сценарій №3 (провал);

- Не знаходить необхідну інформацію

- Засмучений користувач покидає сайт.

Описавши сценарії, побудуємо систему

Після опису сценаріїв, побудуємо діаграму варіантів використання, та зобразимо на рисунку 3.2.

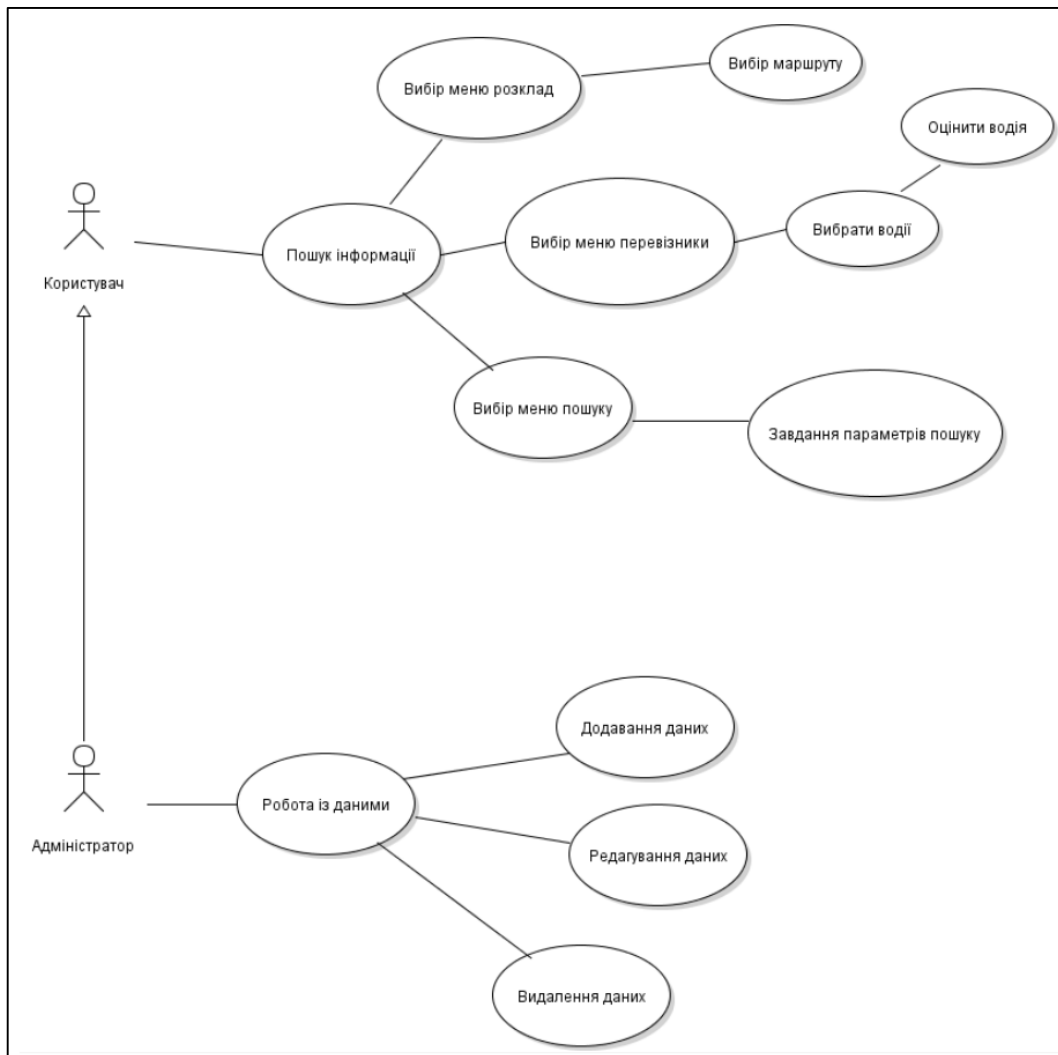


Рисунок 3.2 – Діаграма варіантів використання

Побудуємо діаграму діяльності для користувацької частини, яка зображена на рисунку 3.3.

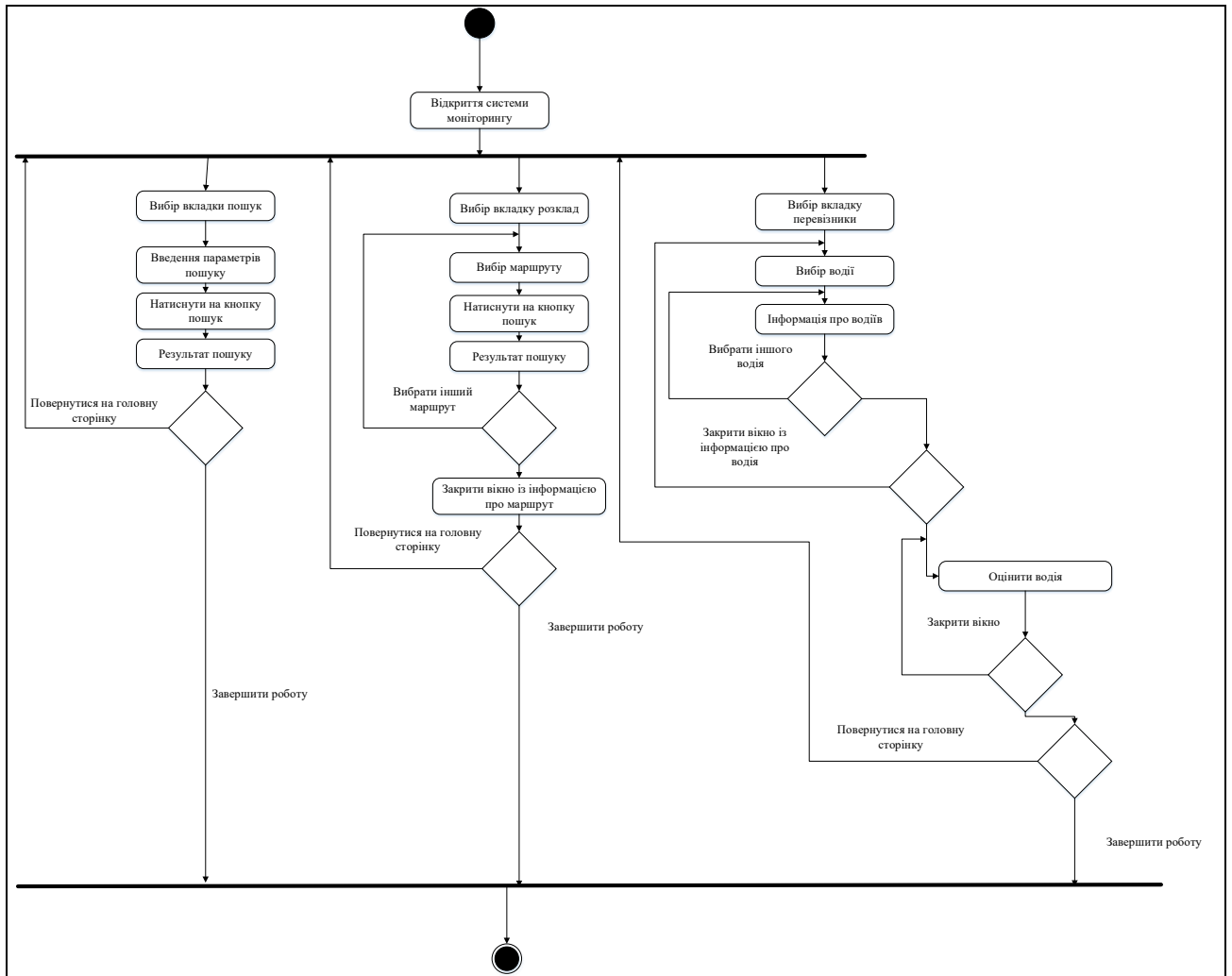


Рисунок 3.3 – Діаграма діяльності для користувацької частини

Побудуємо діаграму діяльності для адміністративної частини, яка зображена на рисунку 3.4.

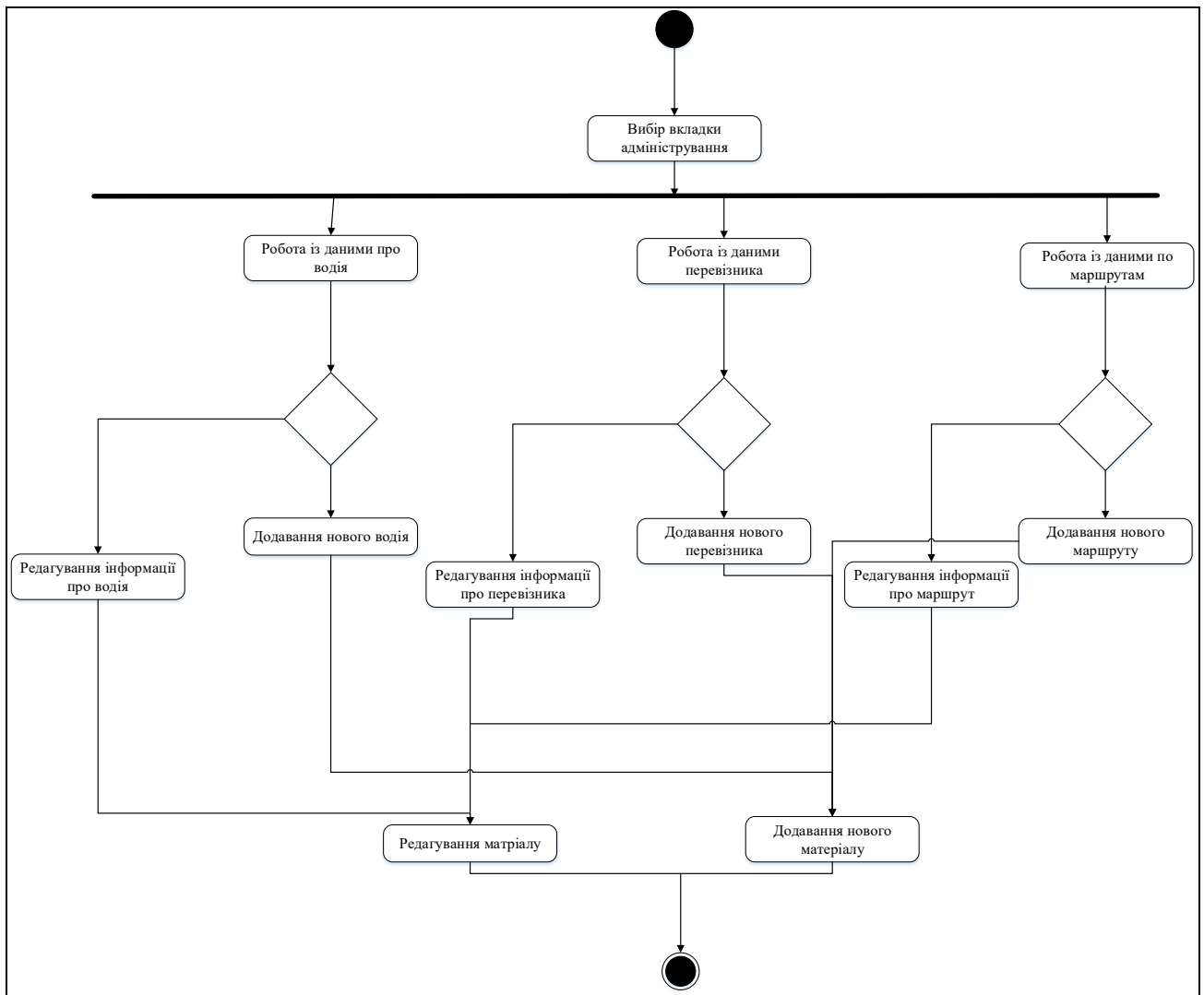


Рисунок 3.4 – Діаграма діяльності для адміністративної частини

Діаграма послідовності — діаграма, яка показує взаємодії об'єктів пов'язаних за часом. Така діаграма зображує задіяні об'єкти та відправлених повідомлень, враховуючи час.

На діаграмі відображено у вигляді прямих ліній процеси або різні об'єкти, які одночасно виконуються одночасно. Надіслані повідомлення показуються у вигляді горизонтальних ліній, та враховуючи час відправлення.

Діаграма послідовності, де зображені процеси взаємодії об'єктів, у нашій інформаційній системі. Діаграма послідовності зображена на рисунку 3.5.

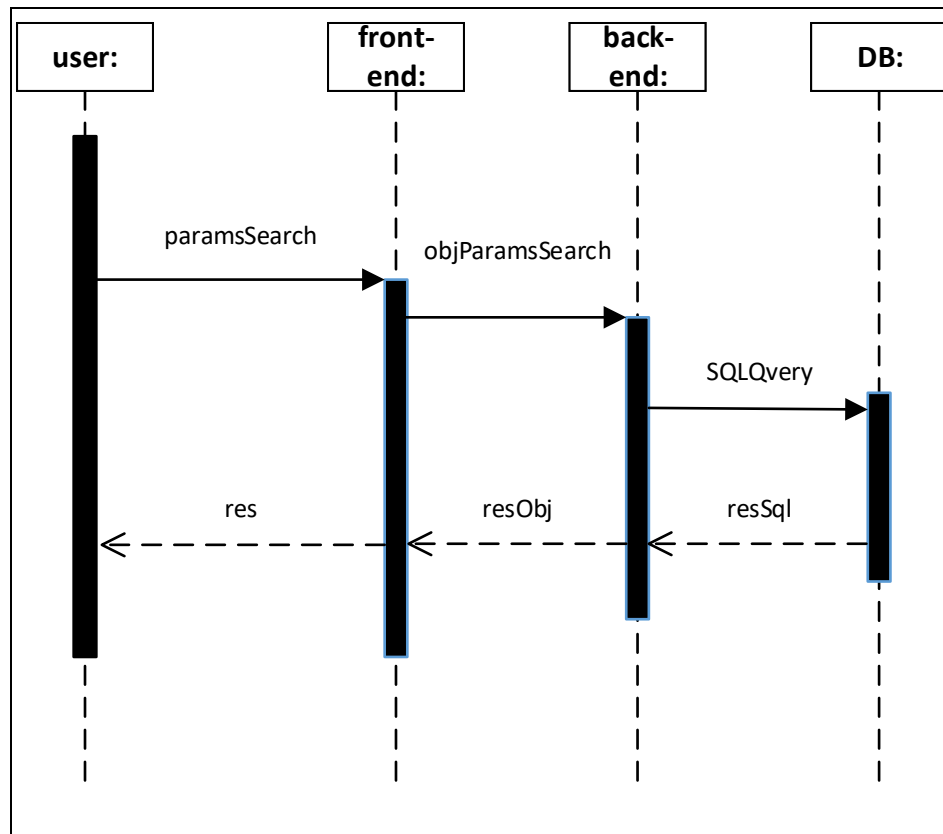


Рисунок 3.5 – Діаграма послідовності

3.2 Розробка front-end та back-end частини інформаційної системи моніторингу пасажирських автопревезень

Першим кроком Front-end частини необхідно зробити макет системи, який зроблено за допомогою html та css.

Html не є мовою програмування, а лише відноситься до мови розмітки, яка надає інформацію браузеру, як зображувати вміст сторінки, для нашого перегляду. До HTML входить серія (елементів), які застосовуються, щоб вміщувати, або згортати різні частини контенту, для відображення в будь-який спосіб. Елементи використовуючи початкові та кінцеві (теги) можуть відобразити слова або зображення посиланням для переходу на іншу сторінку також можуть показувати текст курсивом, змінювати шрифт тощо.

Приклад використання мови html представлений на рисунку 3.6.

```

<div >

<label className={c.myLeble} >Початкова станція:</label><br/>
<input type="text" className={c.myInput} value={this.state.Value_Start}
  onChange={this.onInputStart}
  /></div>
<div>
  <label className={c.myLeble}>Кінцева станція:</label><br/>
<input type="text" className={c.myInput} value={this.state.Value_Finish}
  onChange={this.onInputFinish}
  />
</div>
<div>
<button className={c.myButton} onClick={this.FileterinMapInform}>Пошук</button>
</div>
</div>
</div>

```

Рисунок 3.6 – Використання мови html

CSS – це спеціальна мова (мова стилів), за допомогою якої описують вигляд документів (як і де відобразити елементи веб-сторінки), написаних мовами розмітки даних.

Приклад використання мови CSS представлений на рисунку 3.7.

```

color:black;
grid-area: f;
})
}.text {
  font-style: italic;
  text-align: center;
})
}.myButton {
  -moz-box-shadow: 3px 4px 0px 0px #899599;
  -webkit-box-shadow: 3px 4px 0px 0px #899599;
  box-shadow: 3px 4px 0px 0px #899599;
  background:-webkit-gradient(linear, left top, left bottom, color-stop(0.05,
  background:-moz-linear-gradient(top, #ededed 5%, #bab1ba 100%);
  background:-webkit-linear-gradient(top, #ededed 5%, #bab1ba 100%);
  background:-o-linear-gradient(top, #ededed 5%, #bab1ba 100%);
  background:-ms-linear-gradient(top, #ededed 5%, #bab1ba 100%);
  background:linear-gradient(to bottom, #ededed 5%, #bab1ba 100%);
  filter:progid:DXImageTransform.Microsoft.gradient(startColorstr='#ededed',
  background-color:#ededed;
  -moz-border-radius:15px;
  -webkit-border-radius:15px;
  border-radius:15px;
  border:1px solid #d6bcd6;
  display:inline-block;

```

Рисунок 3.7 – Використання мови CSS

Для гарної взаємодії між сервером використовувалась бібліотека react.

Веб-сторінка нашої веб-системи моніторингу пасажирських автоперевезень має мати свою структуру, до якої має входити шапка, меню навігації, та основна частина.

Створення макету шапки веб-сторінки показано на рисунку 3.8.

```
import React from 'react';
import c from './header.module.css';

const Header = () => {
  return <header className= {c.header}>
    <img className={c.img} src='http://arenda-avtobusa.by/images/%D0%BB%D0%BE%D0%B3%D0%BE.svg?crc=136901531' />
    <h3 className={c.tekst}>Система моніторингу пасажирських автоперевезень</h3>
  </header>
};

export default Header;
```

Рисунок 3.8 – Створення шапки веб-сторінки

Результат створення макету шапки веб-сторінки показано на рисунку 3.9.



Рисунок 3.9 – Результат створення шапки веб-сторінки

Створення макету меню навігації веб-сторінки показано на рисунку 3.10.

```
import React from 'react';
import c from './navbar.module.css';
import {NavLink} from "react-router-dom";

let Navbar = () => {
  return ( <nav className = {c.nav}>

    <NavLink to='/' className={c.myButton}> Головна</NavLink>

    <NavLink to='/schedule' className={c.myButton}> Розклад</NavLink>

    <NavLink to='/search' className={c.myButton}> Пошук</NavLink>

  </nav> );
};

export default Navbar;
```

Рисунок 3.10 – Створення меню навігації веб-сторінки

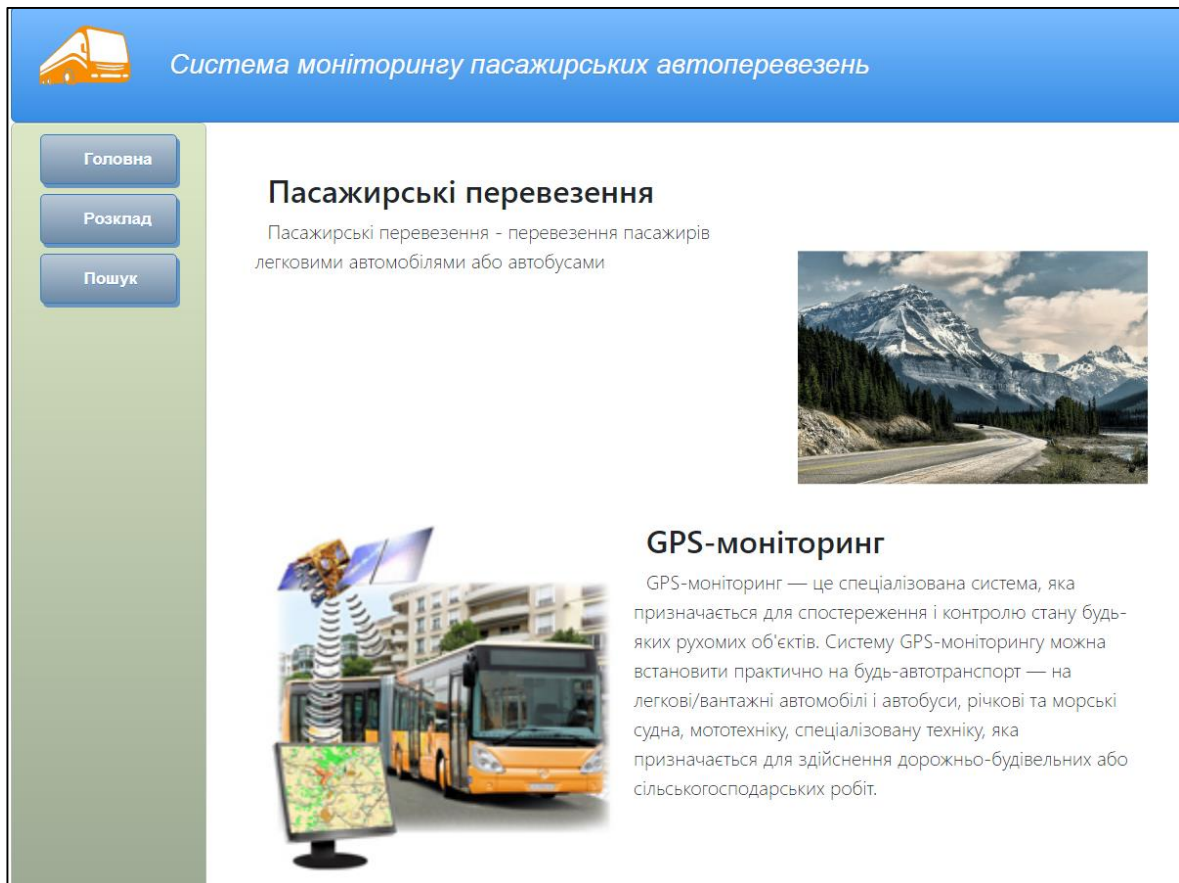


Рисунок 3.13 – Вигляд головної сторінки, при розширенні 1366x768 пікселів

Для адаптивності, тобто мобільності сайту, Bootstrap має систему сіток. За допомогою її ми зможемо легко адаптувати сторінку для мобільних пристроїв, це можна побачити на рисунку 3.14.

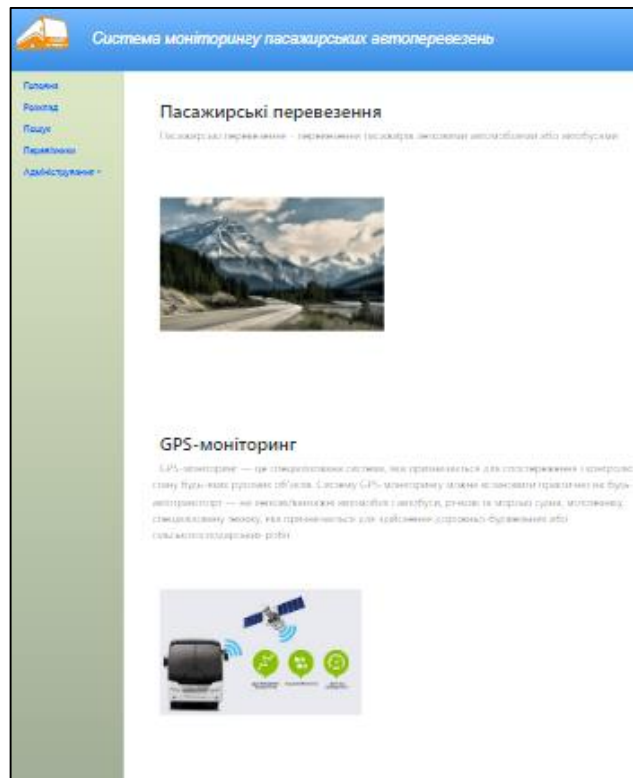


Рисунок 3.14 – Вигляд головної сторінки, при розширенні 375x667 пікселів

Щоб дати можливість розглянути детальніше інформацію про маршрути, зробимо кнопки із назвами маршруту. Вигляд кнопок зображено на рисунку (рис. 3.15):

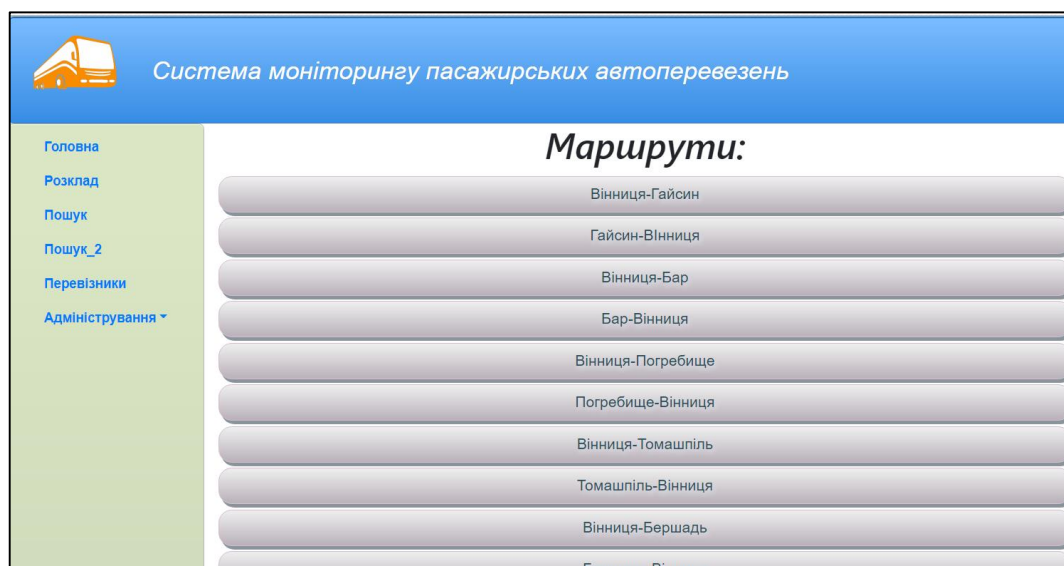


Рисунок 3.15 – Вигляд кнопок із назвами маршрутів

Наша інформаційна система містить дані про маршрут, місце знаходження автобуса в певний момент часу, для представлення на карті ми будемо використовувати картографічний модуль OpenStreetMap, що буде відображатися на спливаючому вікні. Відобразимо макет спливаючого вікна на рисунку 3.16.

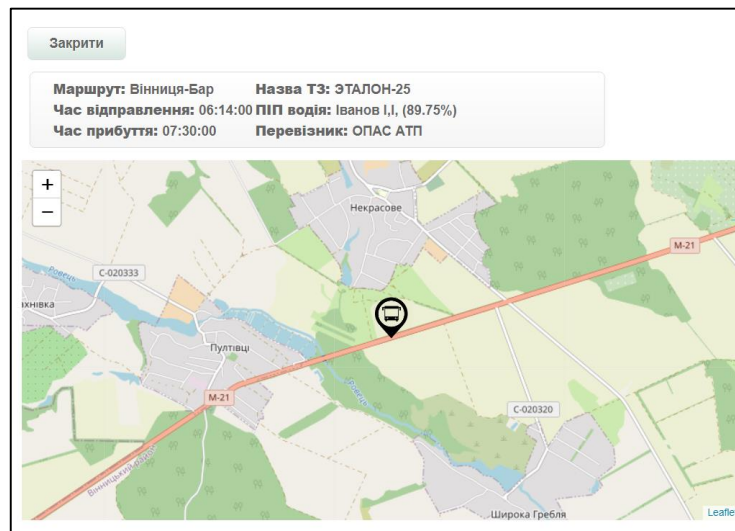


Рисунок 3.16 – Вигляд спливаючого вікна із картою

Макет сторінки пошуку, містить параметри пошуку, та результат. Зображено на рисунку 3.17 макет сторінки пошуку.

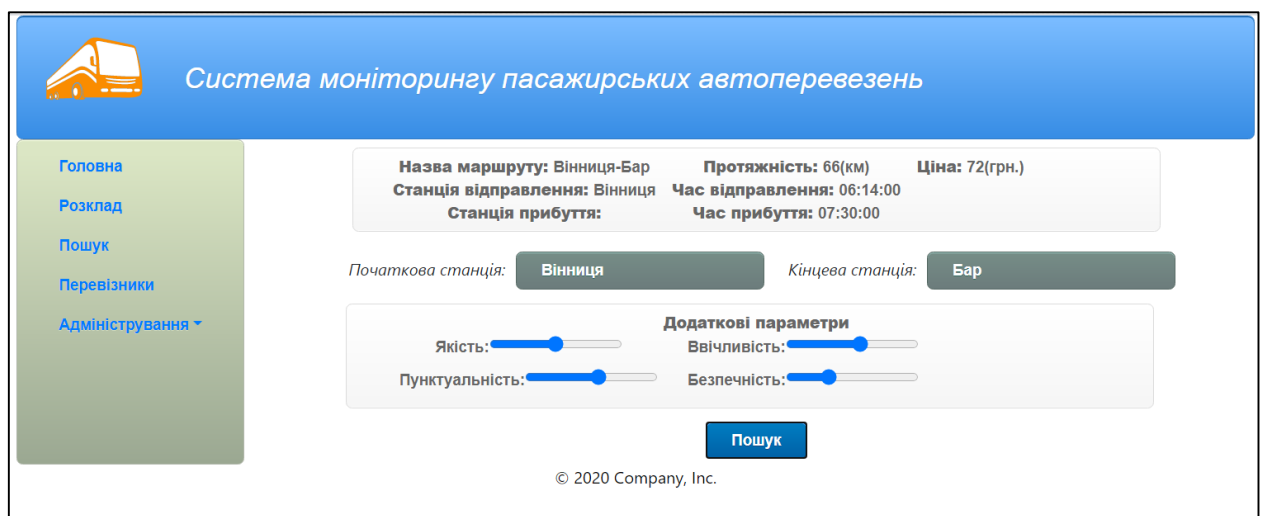


Рисунок 3.17 – Макет сторінки пошуку

Для сторінки перевізників зроблено макет у вигляді блоків із інформацією про перевізників, результат зображено рисунку 3.18.

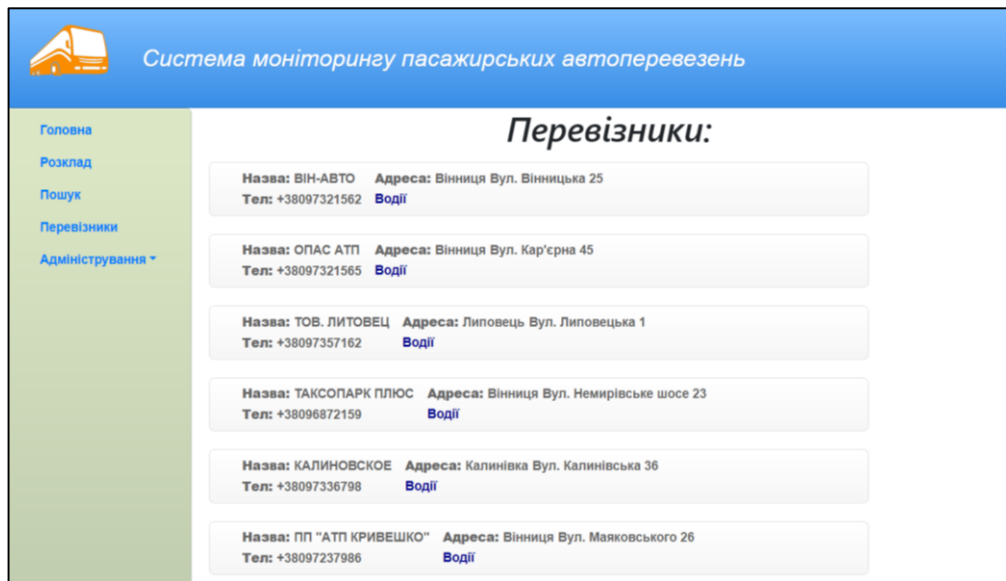


Рисунок 3.18 – Сторінка перевізники

Для представлення інформації про водіїв, які працюють у перевізника було зроблено макет спливаючого вікна, який зображено на рисунку 3.19.

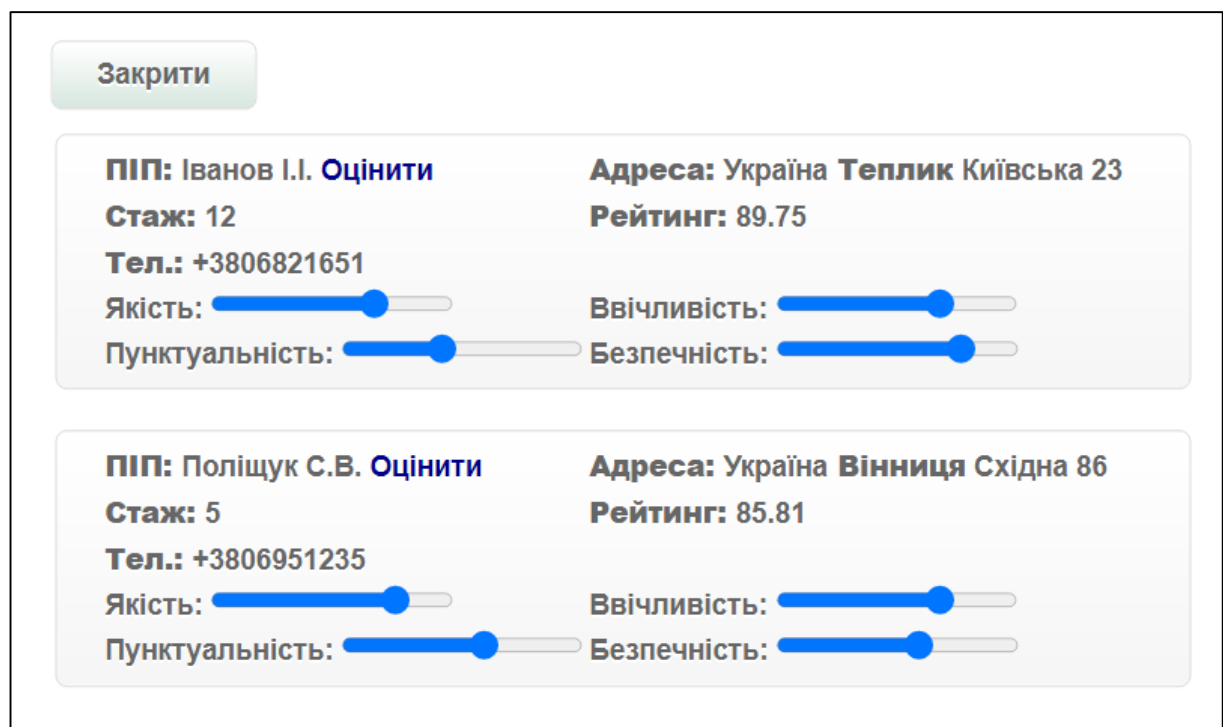


Рисунок 3.19 – Спливаюче вікно

Для представлення оцінювання водіїв, які працюють у перевізника було зроблено макет спливаючого вікна, який зображено на рисунку 3.20.

Рисунок 3.20 – Спливаюче вікно оцінювання водія

Для редагування та додавання інформації, розроблено макет сторінки адміністрування, де розміщена таблиця із даними приклад зображено на рисунку 3.21.

Система моніторингу пасажирських автоперевезень

Водії:

#	ПІБ	Телефо	Стаж	Країна	Місто	Вулиця	Будинок	Квартира		
1	Іванов І.І.	+3806821651	12	Україна	Теплик	Київська	23			
2	Петров П.П.	+3806556232	16	Україна	Немирів	Сонячна	15			
3	Поліщук С.В.	+3806951235	5	Україна	Вінниця	Східна	86			
4	Савіцький А.А.	+3806955654	9	Україна	Київ	Квітуча	132	6		
5	Мудрий М.І.	+3806841235	7	Україна	Вінниця	Нагорна	89			
6	Яремчук Я.І.	+3806925164	7	Україна	Жмеренка	Вінницька	98			

Рисунок 3.21 – Приклад сторінки редагування та додавання

Для редагування розроблено вікно редагування та додавання. Вікно зображено на рисунку 3.22.

Водій

ПІБ: Савіцький А.А. Телефон: +3806955654 Стаж: 9

Країна: Україна Місто: Київ

Вулиця: Квітуча Будинок: 132 квартира: 6

Зберегти

Рисунок 3.22 – Вікно редагування та додавання

Після розробки макета сторінок та спливаючих вікон, необхідно розробити функціонал, що буде вирішуватися за допомогою JavaScript, React.

Для того, щоб реалізувати сторінку маршрути необхідно, звернутися до бази даних, яка поверне інформацію про маршрути, та описати зміни, які відбуваються при натисканні на кнопки, реалізація зображена на рисунку на рисунку 3.23.

```

import React, {Component} from 'react';
import c from './querystopbus.module.css';
import {Link, Route} from 'react-router-dom';

class Querystopbus extends Component {
  state = {
    datestop: [],
  };
  componentDidMount(){
    this.getDatestop();
  }
  getDatestop = () =>{
    fetch('http://localhost:4900/stopbus')
      .then(response => response.json())
      .then(response => this.setState({datestop:response.data}))
      .catch(err=>console.error(err));
  };

  render(){
    const {datestop}= this.state;
    console.log(datestop);
    return (
      <div className="App">
        <h1 className={c.text}>Mapuyт</h1>
        <datestop.map((id_stop,name_stop,id_vehicle,id_route,time_arriveos,time_departure,id_route,name_route,legnly,id_station_u,id_station_b,id_vehicle,name
          <div className={c.item }>
            <Link to="/map/${id_vehicle}" key={id_vehicle}>button className={c.myButton }>{name_route}</button></Link></div>
          </div>
        </div>
      );
    };
  };
  export default Querystopbus;

```

Рисунок 3.23 – Реалізація сторінки маршрути

Метод `render()` відповідає за виведення інформації користувачу, а метод `getDateastop()` відповідає за надсилання запиту до бази даних, та обробку відповіді.

Для отримання детальної інформації про маршрут, та відобразити місце знаходження транспортного засобу на карті, необхідно звернутися, до бази даних, обробити отриману інформацію та відобразити її користувачу фрагмент реалізації, спливаючого вікна показано на рисунку 3.24.

```

afterOpenModal() {
}
getDriveone=>{
  fetch('http://localhost:4000/drive/${parseInt(this.props.match.params.id)}')
    .then(response => response.json())
    .then(response => { console.log(response.drive.root.children[1].attributes); this.setState({cordn:response.drive.root.children[1].attributes})})
    .catch(err=>console.error(err))
};
componentDidMount(){
  this.getDriveone();
}
closeModal() {
  this.setState({modalIsOpen: false});
}
componentWillUnmount(){
}
render() {
  return (
    <div>
      <Modal
        isOpen={this.state.modalIsOpen}
        onAfterOpen={this.afterOpenModal}
        onRequestClose={this.closeModal}
        style={customStyles}
        contentLabel="Example Modal"
      />
      <Link to="/schedule"> <button className={c.myButton} onClick={this.closeModal}>Закрити</button><br/></Link>
      <table className={c.myTable}> <td><div><b>Маршрут:</b></div><div><b>Час відправлення:</b></div><div><b>Час прибуття:</b></div></td><td><div><b>Машина Т3:</b></div><div><b>Піп водія:</b></div></td></tr>
      </table>
      <MapTwo lat={this.state.cordn.lat} lon={this.state.cordn.lon} />
    </div>
  );
}

```

Рисунок 3.24 – Реалізація спливаючого вікна

За відображення карти відповідає компонент `MapTwo` реалізація якого зображена на рисунку 3.25, за отримання інформації з сервера відповідає метод `getDriveone()`, за закриття вікна відповідає метод `closeModal()`.

```

let MapTwo =(props)=> {
  let cordn={};
  let Filter=(a,b)=>{
    if(a===undefined || b===undefined){
      return {lon:49.235428,
              lat:28.484953}
    }
    else{return {lon:(parseFloat(a)),
                 lat:(parseFloat(b))}
    }
  };
  cordn=Filter(props.lon,props.lat);
  return (
    <LeafletMap
      center=[[cordn.lat, cordn.lon]]
      zoom={13}
      maxZoom={18}
      attributionControl={true}
      zoomControl={true}
      doubleClickZoom={true}
      scrollWheelZoom={true}
      dragging={true}
      animate={true}
      easeLinearity={0.35}
    >
      <TileLayer
        url='http://[s].tile.osm.org/{z}/{x}/{y}.png'
      />
      <Marker position=[[cordn.lat,cordn.lon]] zoom={13} icon={myIcon}>
        <Popup>
          {console.log(cordn)}
        </Popup>
      </Marker>
    </LeafletMap>
  );
};

```

Рисунок 3.25 – Компонент MapTwo

Для сторінки пошуку необхідно, обробити введені параметри пошуку та надіслати запит, який верне дані враховуючи введені значення. Реалізацію зображено на рисунку 3.26

```

}
getDateSearch = _ => {
  let data = {
    Value_Start: this.state.Value_Start,
    Value_Finish: this.state.Value_Finish,
    quality_rt_v1: this.state.quality_rt_v1,
    civility_rt_v1: this.state.civility_rt_v1,
    nicety_rt_v1: this.state.nicety_rt_v1,
    safety_rt_v1: this.state.safety_rt_v1
  };
  fetch('http://localhost:4000/search_info/add', {
    method: 'POST',
    body: JSON.stringify(data),
    headers: {
      'Content-Type': 'application/json'
    }
  })
  .then(response => response.json())
  .then(response => { console.log('response'); console.log(response); this.setState({ dateSearch: response }) })
  .catch(err => console.error(err))
};
onInputStart = (e) => {
  this.setState({
    Value_Start: e.target.value
  })
};
onInputFinish = (e) => {
  this.setState({
    Value_Finish: e.target.value
  })
};
handleChangeQualityRt(event) {
  this.setState({ quality_rt_v1: event.target.value });
}
handleChangeCivilityRt(event) {
  this.setState({ civility_rt_v1: event.target.value });
}
handleChangeNicetyRt(event) {
  this.setState({ nicety_rt_v1: event.target.value });
}
handleChangeSafetyRt(event) {
  this.setState({ safety_rt_v1: event.target.value });
}
renderDateSearch = ({ id_stop, name_stop, id_vehicle, id_roud, time_arriveos, time_departure, id_rout, name_rout, lengly,
  <div className={c.item}>
    <table className={c.myTable}>
      <tr>

```

Рисунок 3.26 – Реалізація сторінки пошук

Метод `GetDatesearch()` відповідає за надсилання на сервер вхідних параметрів та отримання відповіді та обробку значень визвавши `renderDatesearch()` який відповідає за представлення інформації.

Для сторінки адміністратора, необхідно заповнити даними таблицю, та забезпечити зв'язок із сервером та взаємодію із спливаючими вікнами. За отримання даних із сервера відповідає метод `getDriverInfo()`, за надсилання даних, які необхідно зберігати відповідає, метод `handleSubmit()`. Реалізація методу `getDriverInfo()` на рисунку 3.27, а `handleSubmit()` на рисунку 3.28.

```

getDriverInfo() {
  let data = {
    id_driver: this.props.id_driver
  };

  fetch('http://localhost:4000/driver_table/driver', {
    method: 'POST',
    body: JSON.stringify(data),
    headers: {
      'Content-Type': 'application/json'
    }
  })
  .then(response => response.json())
  .then(response => {
    console.log('response'); console.log(response[0]);
    this.setState({
      id_driver: this.props.id_driver,
      driver_pip: response[0].driver_pip,
      phone: response[0].phone,
      expireance: response[0].expireance,
      country: response[0].country,
      city: response[0].city,
      street: response[0].street,
      num_house: response[0].num_house,
      flat: response[0].flat
    })
  })
  .catch(err => console.error(err));
}

```

Рисунок 3.27 – Реалізація методу `getDriverInfo()`

```

if (this.state.driver_pip != ''
&& this.state.phone != ''
&& this.state.expireance != ''
&& this.state.city != ''
&& this.state.street != ''
&& this.state.country != '') {
  if (this.state.id_driver){
    fetch('http://localhost:4000/driver_table/driver_upd',{
      method: 'POST',
      body: JSON.stringify(data),
      headers: {
        'Content-Type': 'application/json'
      }
    })
    .then(response => response.json())
    .then(response => { console.log('response'); console.log(response);
      this.props.closeIndw();
      //event.preventDefault();/**this.setState({ driver: response }) */)
      .catch(err => console.error(err));
  }else{
    fetch('http://localhost:4000/driver_table/driver_insrt',{
      method: 'POST',
      body: JSON.stringify(data),
      headers: {
        'Content-Type': 'application/json'
      }
    })
    .then(response => response.json())
    .then(response => { console.log('response'); console.log(response);
      this.props.closeIndw();
      //event.preventDefault();/**this.setState({ driver: response }) */)
      .catch(err => console.error(err));
  }
} else {
  alert('Занесіть всі дані');
  event.preventDefault();
}

```

Рисунок 3.28 – Реалізація методу `handleSubmit()`

3.3 Проектування бази даних системи моніторингу пасажирських автоперевезень

База даних інформаційної системи моніторингу пасажирських автоперевезень повинна містити такі таблиці:

- Вокзал;
- Зупинка;
- Маршрути;
- Перевізник;
- Транспортний засіб;
- Водій;
- Журнал поїздок.

Таблиці бази даних інформаційної системи моніторингу показані на (рис. 3.29).

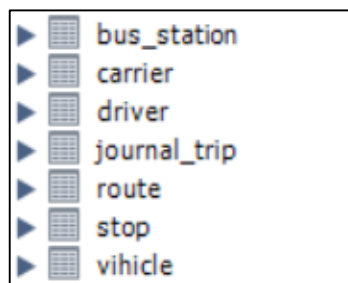


Рисунок 3.29 – Назви таблиць бази даних інформаційної системи моніторингу пасажирських автоперевезень

У таблиці `bus_station` представлено назви автобусних станцій Вінницької області:

Назви автобусних станцій показані на рисунку 3.30.

+ Параметри						
	id_station	name_station	city	address	phone_number	
Изменить Копировать Удалить	1	Вінниця	Вінниця	вул. Київська, 8	0432-671345	
Изменить Копировать Удалить	2	Бар	Бар	вул. Св. Миколая, 7	01.05.2019	
Изменить Копировать Удалить	3	Гайсин	Гайсин	вул. 1-го Травня, 129	0433421176	
Изменить Копировать Удалить	4	Бершадь	Бершадь	вул. Чкалова, 3	0435221083	
Изменить Копировать Удалить	5	Калинівка	Калинівка	вул. Леніна, 49	0433321385	
Изменить Копировать Удалить	6	Могиліє-Подільський	Могиліє-Подільський	вул. Пушкінська, 41	0433767071	
Изменить Копировать Удалить	7	Хмільник	Хмільник	вул. В.Порика, 7	0433823029	
Изменить Копировать Удалить	8	Іллінці	Іллінці	вул. К.Маркса, 78	0434523483	
Изменить Копировать Удалить	9	Тульчин	Тульчин	вул. Леніна, 15	0433522930	
Изменить Копировать Удалить	10	Козятин	Козятин	вул. Миколана(Довженка), 43/12	0434223925	
Изменить Копировать Удалить	11	Тростянець	Тростянець	вул. Леніна, 26	0434322532	
Изменить Копировать Удалить	12	Немирів	Немирів	вул. Горького, 10	0433122989	
Изменить Копировать Удалить	13	Лтин	Лтин	вул. Радянська, 7	0434721700	
Изменить Копировать Удалить	14	Ямпіль	Ямпіль	вул. Жовтнева, 52	0434621348	
Изменить Копировать Удалить	15	Крижопіль	Крижопіль	вул. К.Маркса, 116/ 8-го березня, 1	0434021537	
Изменить Копировать Удалить	16	Жмеринка	Жмеринка	вул. Заводська, 16А	056522-08-00	
Изменить Копировать Удалить	17	Липовець	Липовець	вул. Леніна, 121	0435821039	
Изменить Копировать Удалить	18	Шаргород	Шаргород	вул. Леніна, 260	043442234	
Изменить Копировать Удалить	19	Теплик	Теплик	вул. І.Франка, 29	0435321530	
Изменить Копировать Удалить	20	Погребище	Погребище	вул. Козацька, 1	0434621540	
Изменить Копировать Удалить	21	Томашпіль	Томашпіль	вул. Радянська, 1-а	0434821557	
Изменить Копировать Удалить	22	Муровані Курилівці	Муровані Курилівці	вул. Комарова, 4	0435621059	
Изменить Копировать Удалить	23	Тиверів	Тиверів	вул. Жовтнева, 5	0435521440	
Изменить Копировать Удалить	24	Оратів	Оратів	вул. Невського, 20	0433021436	
Изменить Копировать Удалить	25	Пщанка	Пщанка	вул. Кошового, 10	0434921109	
Изменить Копировать Удалить	26	Чечельник	Чечельник	вул. Радянська, 78	0435121000	
Изменить Копировать Удалить	27	Чернівці	Чернівці	вул. Головна, 19	0372-508849	

Рисунок 3.30 – Назви автобусних станцій

У таблиці carrier наведена інформація про перевізників, які виконують перевезення пасажирів.

Інформація про перевізників зображено на рисунку 3.31.

+ Параметри						
	id_carrier	name_carrier	number_phone	city	address	
Изменить Копировать Удалить	1	ВІН-АВТО	+38097321562	Вінниця	Вул. Вінницька 25	
Изменить Копировать Удалить	2	ОПАС АТП	+38097321565	Вінниця	Вул. Кар'єрна 45	
Изменить Копировать Удалить	3	ТОВ. ЛИТОВЕЦ	+38097357162	Липовець	Вул. Липовецька 1	
Изменить Копировать Удалить	4	ТАКСОПАРК ПЛЮС	+38096872159	Вінниця	Вул. Немирівське шосе 23	
Изменить Копировать Удалить	5	КАЛИНОВСКОЕ	+38097336798	Калинівка	Вул. Калинівська 36	
Изменить Копировать Удалить	6	ПП "АТП КРИВЕШКО"	+38097237986	Вінниця	Вул. Маяковського 26	
Изменить Копировать Удалить	7	ГАЙСИНСКОЕ	+38091236578	Гайсин	Вул. Маяковського 100	
Изменить Копировать Удалить	8	ТУЛЬЧИНСКОЕ	+38062265489	Тульчин	Вул. Соборна 24	
Изменить Копировать Удалить	9	БЕРШАДСЬКЕ АТП 10527	+38068352489	Бершадь	Вул. Хмельницького 37	
Изменить Копировать Удалить	10	ТЕПЛИКСКОЕ	+38097658978	Теплик	Вул. Київська 159	
Изменить Копировать Удалить	11	НЕМИРОВСКОЕ	+38097567845	Немирів	Вул. Вінницька 69	
Изменить Копировать Удалить	12	ЯМПОЛЬСКОЕ АТП 10551	+38097485612	Ямпіль	Вул. Київська 61	
Изменить Копировать Удалить	13	КРИЖОПОЛЬСКОЕ	+38068256478	Крижопіль	Вул. Вінницька 13	
Изменить Копировать Удалить	14	ВИНПАСТРАНС ТОВ	+38097781526	Вінниця	Вул. Липовецька 98	
Изменить Копировать Удалить	15	РАЙ АТП 1	+38068758642	Вінниця	Вул. Калинівська 89	

Рисунок 3.31 – Інформація про перевізників

У таблиці route приведена інформація про маршрути.

Таблиця інформації про маршрути зображена на рисунку 3.32.

+ Параметри						
	id_rout	name_rout	lengly	id_station_a	id_station_b	price
<input type="checkbox"/>	1	Вінниця-Бар	66	1	2	72
<input type="checkbox"/>	2	Вінниця-Гайсин	96	1	3	108
<input type="checkbox"/>	3	Вінниця-Бершадь	157	1	4	172
<input type="checkbox"/>	4	Вінниця-Калинівка	27	1	5	28
<input type="checkbox"/>	5	Вінниця-Мог.Подільський	114	1	6	128
<input type="checkbox"/>	6	Вінниця-Хмільник	65	1	7	72
<input type="checkbox"/>	7	Вінниця-Іплінці	72	1	8	73
<input type="checkbox"/>	8	Вінниця-Тулечин	83	1	9	90
<input type="checkbox"/>	9	Вінниця-Козятин	72	1	10	68
<input type="checkbox"/>	10	Вінниця-Тростянець	120	1	11	128
<input type="checkbox"/>	11	Вінниця-Немирів	46	1	12	46
<input type="checkbox"/>	12	Вінниця-Літин	29	1	13	26
<input type="checkbox"/>	13	Вінниця-Ямпіль	138	1	14	150
<input type="checkbox"/>	14	Вінниця-Крижопіль	133	1	15	150
<input type="checkbox"/>	15	Вінниця-Жмеринка	40	1	16	37
<input type="checkbox"/>	16	Вінниця-Липовець	44	1	17	43
<input type="checkbox"/>	17	Вінниця-Шаргород	78	1	18	86
<input type="checkbox"/>	18	Вінниця-Теплик	135	1	19	150
<input type="checkbox"/>	19	Вінниця-Погребище	76	1	20	69
<input type="checkbox"/>	20	Вінниця-Томашпіль	123	1	21	139
<input type="checkbox"/>	21	Вінниця-Муровані Куриловці	99	1	22	108
<input type="checkbox"/>	22	Вінниця-Тиврів	28	1	23	31
<input type="checkbox"/>	23	Вінниця-Оратів	94	1	24	93
<input type="checkbox"/>	24	Вінниця-Піщанка	155	1	25	172
<input type="checkbox"/>	25	Вінниця-Чечельник	160	1	26	172
<input type="checkbox"/>	26	Вінниця-Чернівці	111	1	27	128

Рисунок 3.32 – Інформації про маршрути

У таблиці vehicle наведена інформація про транспортні засоби.

Наведена інформація про транспортні засоби зображено на рисунку 3.33.

+ Параметры							
	id_vehicle	name	year_manufacture	number_seats	pip_driver	technical_condition	id_carrier
<input type="checkbox"/>	1	ЭТАЛОН-25	2001	29	Іванов І.І.	Пройдено	2
<input type="checkbox"/>	2	МЕРСЕДЕС-17	1999	18	Петров П.П.	Пройдено	1
<input type="checkbox"/>	3	БОГДАН-20	2009	20	Поліщук П.І.	Пройдено	3
<input type="checkbox"/>	4	ЭТАЛОН-26	2007	26	Савіцький А.А.	Не пройдено	4
<input type="checkbox"/>	5	ПАЗ-3205-35	2009	35	Василик В.А.	Пройдено	5
<input type="checkbox"/>	6	МЕРСЕДЕС-18	2003	18	Жук А.В.	Не пройдено	6
<input type="checkbox"/>	7	ИВЕКО-18	2014	18	Гречанок А.С.	Не пройдено	8
<input type="checkbox"/>	8	МЕРСЕДЕС Б.50	2006	25	Слободянюк А.І.	Пройдено	7
<input type="checkbox"/>	9	БОГДАН-21	2001	28	Коваль М.В.	Пройдено	9
<input type="checkbox"/>	10	ЭТАЛОН 23	1998	29	Бойко С.В.	Пройдено	10
<input type="checkbox"/>	11	БОГДАН-20	2005	28	Фаталюк В.І.	Пройдено	11
<input type="checkbox"/>	12	БОГДАН-29	2006	23	Цибилько С.Б.	Не пройдено	12
<input type="checkbox"/>	13	МЕРСЕДЕС-16	2015	18	Волощак В.С.	Пройдено	14
<input type="checkbox"/>	14	БОГДАН-25	2016	28	Шкурак С.К.	Пройдено	13
<input type="checkbox"/>	15	НЕОПЛАН-50	2017	48	Яремчук Я.І.	Пройдено	15
<input type="checkbox"/>	16	БОГДАН-25	2015	24	Подольч П.П.	Пройдено	7
<input type="checkbox"/>	17	МЕРСЕДЕС-47	2006	46	Цибюля С.Б.	Пройдено	5
<input type="checkbox"/>	18	МЕРСЕДЕС-18	2011	18	Мудрий М.І.	Не пройдено	10
<input type="checkbox"/>	19	НЕОПЛАН	2002	40	Семко С.В.	Пройдено	2
<input type="checkbox"/>	20	ПАЗ-3205-18/1	2007	18	Поліщук С.В.	Пройдено	2
<input type="checkbox"/>	21	ПАЗ-4234	1999	21	Коваль С.Б.	Пройдено	7
<input type="checkbox"/>	22	СЕТРА-45	2005	40	Василик Я.С.	Пройдено	11
<input type="checkbox"/>	23	ТУР 28	2003	29	Мудрий В.І.	Пройдено	12

Рисунок 3.33 – Інформація про транспортні засоби

У таблиці stop наведена інформація про зупинки транспортні засоби.

Наведена інформація про зупинки транспортні засоби зображено на рисунку 3.34.

+ Параметры							
	id_stop	name_stop	id_vehicle	id_roud	time_arriveos	time_departun	
<input type="checkbox"/>	1	Вінниця	1	1	06:14:00	07:30:00	
<input type="checkbox"/>	2	Бар	1	27	13:50:00	15:00:00	
<input type="checkbox"/>	3	Вінниця	2	2	07:50:00	09:45:00	
<input type="checkbox"/>	4	Гайсин	2	28	15:15:00	17:10:00	
<input type="checkbox"/>	5	Вінниця	3	3	10:20:00	13:40:00	
<input type="checkbox"/>	6	Бершадь	3	29	14:00:00	17:20:00	
<input type="checkbox"/>	7	Вінниця	4	4	08:10:00	08:50:00	
<input type="checkbox"/>	8	Калинівка	4	30	10:40:00	11:20:00	
<input type="checkbox"/>	9	Вінниця	5	5	11:00:00	13:00:00	
<input type="checkbox"/>	10	Могилів Подільський	5	31	13:30:00	15:30:00	
<input type="checkbox"/>	11	Вінниця	6	6	07:00:00	08:15:00	
<input type="checkbox"/>	12	Хмільник	6	32	08:45:00	10:00:00	
<input type="checkbox"/>	13	Вінниця	7	7	14:00:00	15:25:00	
<input type="checkbox"/>	14	Іллінци	7	33	16:00:00	17:25:00	
<input type="checkbox"/>	15	Вінниця	8	8	06:30:00	08:10:00	
<input type="checkbox"/>	16	Тулчин	8	34	08:40:00	10:20:00	
<input type="checkbox"/>	17	Вінниця	9	9	11:30:00	12:50:00	
<input type="checkbox"/>	18	Козятин	9	35	13:20:00	14:40:00	
<input type="checkbox"/>	19	Вінниця	10	10	07:45:00	10:06:00	
<input type="checkbox"/>	20	Тростянець	10	36	15:00:00	17:21:00	
<input type="checkbox"/>	21	Вінниця	11	11	06:30:00	07:21:00	
<input type="checkbox"/>	22	Немирів	11	37	07:50:00	08:45:00	
<input type="checkbox"/>	23	Вінниця	12	12	06:20:00	06:48:00	
<input type="checkbox"/>	24	Літин	12	38	07:28:00	08:00:00	
<input type="checkbox"/>	25	Вінниця	13	13	15:00:00	17:55:00	

Рисунок 3.34 – Інформація про зупинки транспортні засоби

Зв'язки між таблицями показані на рисунку 3.35.

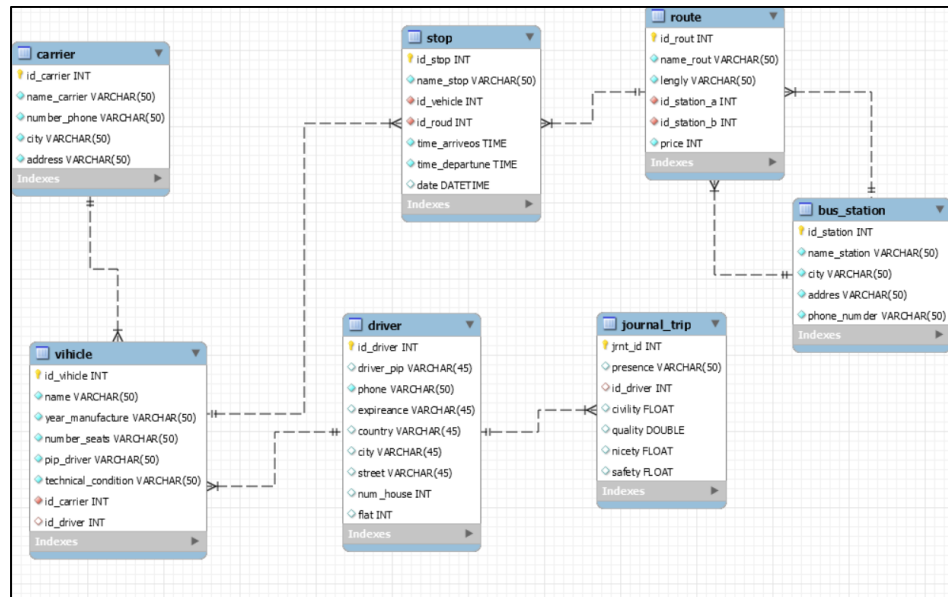


Рисунок 3.35 – Зв'язки між таблицями

3.4 Розробка SQL-запиту для вибірки даних

Запит — це формулювання своєї інформаційної потреби користувачем деякої бази даних або інформаційної системи, наприклад, пошукової системи.

Для роботи системи було складено 3 основних запити, які дозволяють сформулювати потрібну інформація, для користувача.

1 запит – Надає інформацію про зупинки транспортних засобів, інформацію про самий транспортний засіб, також дані про перевізника.

На рисунку 3.36 представлено сформований перший запит, за допомогою мови SQL.

Результат виконання запиту 1 зображено на рисунку 3.37.

```

1 SELECT * FROM stop
2 INNER JOIN route ON stop.id_roud = route.id_rout
3 INNER JOIN vehicle ON stop.id_vehicle = vehicle.id_vihicle
4 INNER JOIN carrier ON vehicle.id_carrier = carrier.id_carrier
    
```

Рисунок 3.36 – Формування SQL запиту

id_stop	name_stop	id_vehicle	id_roud	time_arriveos	time_departune	id_rout	name_rout	lengly	id_station_a	id_station_b	price	id_vihicle	name
3	Вінниця	2	2	07:50:00	09:45:00	2	Вінниця-Гайсин	96	1	3	108	2	МЕРСЕДЕС-17
4	Гайсин	2	28	15:15:00	17:10:00	28	Гайсин-Вінниця	96	3	1	108	2	МЕРСЕДЕС-17
1	Вінниця	1	1	06:14:00	07:30:00	1	Вінниця-Бар	66	1	2	72	1	ЭТАЛОН-25
2	Бар	1	27	13:50:00	15:00:00	27	Бар-Вінниця	66	2	1	72	1	ЭТАЛОН-25
37	Вінниця	19	19	09:05:00	10:30:00	19	Вінниця-Погребище	76	1	20	69	19	НЕОПЛАН
38	Погребище	19	45	15:15:00	17:00:00	45	Погребище-Вінниця	76	20	1	69	19	НЕОПЛАН
39	Вінниця	20	20	08:45:00	10:18:00	20	Вінниця-Томашпіль	123	1	21	139	20	ПАЗ-3205-18/1
40	Томашпіль	20	46	11:30:00	13:05:00	46	Томашпіль-Вінниця	123	21	1	139	20	ПАЗ-3205-18/1
5	Вінниця	3	3	10:20:00	13:40:00	3	Вінниця-Бершадь	157	1	4	172	3	БОГДАН-20
6	Бершадь	3	29	14:00:00	17:20:00	29	Бершадь-Вінниця	157	4	1	172	3	БОГДАН-20
7	Вінниця	4	4	08:10:00	08:50:00	4	Вінниця-Калинівка	27	1	5	28	4	ЭТАЛОН-26
8	Калинівка	4	30	10:40:00	11:20:00	30	Калинівка-Вінниця	27	5	1	28	4	ЭТАЛОН-26
9	Вінниця	5	5	11:00:00	13:00:00	5	Вінниця-Мог.Подільський	114	1	6	128	5	ПАЗ-3205-35
10	Могилів Подільський	5	31	13:30:00	15:30:00	31	Мог.Подільський-Вінниця	114	6	1	128	5	ПАЗ-3205-35
33	Вінниця	17	17	07:10:00	08:45:00	17	Вінниця-Шаргород	78	1	18	86	17	МЕРСЕДЕС-47
34	Шаргород	17	43	13:35:00	15:10:00	43	Шаргород-Вінниця	78	18	1	86	17	МЕРСЕДЕС-47
11	Вінниця	6	6	07:00:00	08:15:00	6	Вінниця-Хмільник	65	1	7	72	6	МЕРСЕДЕС-18
12	Хмільник	6	32	08:45:00	10:00:00	32	Хмільник-Вінниця	65	7	1	72	6	МЕРСЕДЕС-18
51	Вінниця	6	26	11:45:00	14:50:00	26	Вінниця-Чернівці	111	1	27	128	6	МЕРСЕДЕС-18
52	Чернівці	6	52	18:10:00	20:18:00	52	Чернівці-Вінниця	111	27	1	128	6	МЕРСЕДЕС-18

Рисунок 3.37 – Результат виконання запиту 1

2 запит – Надає інформацію про автостанції, зупинки транспортних засобів та інформацію про маршрути.

На рисунку 3.38 представлено сформований перший запит, за допомогою мови SQL.

Результат виконання запиту 1 зображено на рисунку 3.39.

```

SELECT rt.*, stp.*, vhcl.*, drv.*,
(select count(jt.presence) from bus_station.journal_trip jt where jt.id_driver = drv.id_driver
and jt.presence = 'true') as cnt_trip,
(select avg(jt.nicety) from bus_station.journal_trip jt where jt.id_driver = drv.id_driver) as avg_nicety,
(select avg(jt.civility) from bus_station.journal_trip jt where jt.id_driver = drv.id_driver) as avg_civility,
(select avg(jt.quality) from bus_station.journal_trip jt where jt.id_driver = drv.id_driver) as avg_quality
FROM bus_station.route as rt
inner join bus_station.stop as stp on stp.id_roud = rt.id_rout
inner join bus_station.vihicle as vhcl on vhcl.id_vihicle = stp.id_vehicle
inner join bus_station.driver drv on drv.id_driver = vhcl.id_driver
inner join bus_station.journal_trip jrnt on jrnt.id_driver = drv.id_driver;

```

Рисунок 3.38 – Формування SQL запиту

id_stop	name_stop	id_vehicle	id_rout	time_arriveos	time_departune	id_rout	name_rout	lengly	id_station_a
2	Бар	1	27	13:50:00	15:00:00	27	Бар-Вінниця	66	2
4	Гайсин	2	28	15:15:00	17:10:00	28	Гайсин-Вінниця	96	3
6	Бершадь	3	29	14:00:00	17:20:00	29	Бершадь-Вінниця	157	4
8	Калинівка	4	30	10:40:00	11:20:00	30	Калинівка-Вінниця	27	5
10	Могилів Подільський	5	31	13:30:00	15:30:00	31	Мог.Подільський- Вінниця	114	6
12	Хмільник	6	32	08:45:00	10:00:00	32	Хмільник-Вінниця	65	7
14	Іллінці	7	33	16:00:00	17:25:00	33	Іллінці-Вінниця	72	8
16	Тулчин	8	34	08:40:00	10:20:00	34	Тулчин-Вінниця	83	9
18	Козятин	9	35	13:20:00	14:40:00	35	Козятин-Вінниця	72	10
20	Тростянець	10	36	15:00:00	17:21:00	36	Тростянець-Вінниця	120	11
22	Немирів	11	37	07:50:00	08:45:00	37	Немирів-Вінниця	46	12
24	Літин	12	38	07:28:00	08:00:00	38	Літин-Вінниця	29	13
26	Ямпіль	13	39	18:25:00	22:40:00	39	Ямпіль-Вінниця	138	14
28	Крижопіль	14	40	16:40:00	19:06:00	40	Крижопіль-Вінниця	133	15
30	Жмеринка	15	41	11:25:00	12:10:00	41	Вінниця-Жмеринка	40	16
32	Липовець	16	42	11:40:00	12:35:00	42	Вінниця-Липовець	44	17
34	Шаргород	17	43	13:35:00	15:10:00	43	Шаргород-Вінниця	78	18
36	Теплик	18	44	14:00:00	16:45:00	44	Теплик-Вінниця	135	19
38	Погребище	19	45	15:15:00	17:00:00	45	Погребище-Вінниця	76	20
40	Томашпіль	20	46	11:30:00	13:05:00	46	Томашпіль-Вінниця	123	21
42	Муровані Куриловці	21	47	12:30:00	14:30:00	47	Муровані Куриловці- Вінниця	99	22
44	Тиврів	22	48	12:10:00	12:37:00	48	Тиврів-Вінниця	28	23
46	Оратів	23	49	11:25:00	13:13:00	49	Оратів-Вінниця	94	24
48	Піщанка	8	50	18:00:00	21:05:00	50	Піщанка-Вінниця	155	25
50	Чечельник	11	51	16:50:00	20:05:00	51	Чечельник-Вінниця	160	26
52	Чернівці	6	52	18:10:00	20:18:00	52	Чернівці-Вінниця	111	27

Рисунок 3.39 – Результат виконання запиту 2

3 запит – Надає інформацію про перевізника та водія відображаючи його рейтинг.

На рисунку 3.40 представлено сформований перший запит, за допомогою мови SQL.

Результат виконання запиту 1 зображено на рисунку 3.41.

```

SELECT stp.*, vhcl.*, rt.*, crr.*,
(select count(jt.presence) from bus_station.journal_trip jt where jt.id_driver = vhcl.id_driver) as cnt_trip,
(select count(jt.presence) from bus_station.journal_trip jt where jt.id_driver = vhcl.id_driver
and jt.presence = 'true') as cnt_trip_tr,
(select avg(jt.nicety) from bus_station.journal_trip jt where jt.id_driver = vhcl.id_driver) as avg_nicety,
(select avg(jt.civility) from bus_station.journal_trip jt where jt.id_driver = vhcl.id_driver) as avg_civility,
(select avg(jt.quality) from bus_station.journal_trip jt where jt.id_driver = vhcl.id_driver) as avg_quality
FROM bus_station.stop stp
INNER JOIN bus_station.route rt ON stp.id_rout = rt.id_rout
INNER JOIN bus_station.vehicle vhcl ON stp.id_vehicle = vhcl.id_vehicle
INNER JOIN bus_station.carrier crr ON vhcl.id_carrier = crr.id_carrier

```

Рисунок 3.40 – Формування SQL запиту

3	Вінниця	2	2	07:50:00	09:45:00	2020-10-14 20:02:28	2	МЕРСЕДЕС-17	1999	18
4	Гайсин	2	28	15:15:00	17:10:00	2020-10-14 20:02:28	2	МЕРСЕДЕС-17	1999	18
1	Вінниця	1	1	06:14:00	07:30:00	2020-10-14 20:02:28	1	ЭТАЛОН-25	2001	29
2	Бар	1	27	13:50:00	15:00:00	2020-10-14 20:02:28	1	ЭТАЛОН-25	2001	29
37	Вінниця	19	19	09:05:00	10:30:00	2020-10-14 20:02:28	19	НЕОПЛАН	2002	40
38	Погребище	19	45	15:15:00	17:00:00	2020-10-14 20:02:28	19	НЕОПЛАН	2002	40
39	Вінниця	20	20	08:45:00	10:18:00	2020-10-14 20:02:28	20	ПАЗ-3205-18/1	2007	18
40	Томашпіль	20	46	11:30:00	13:05:00	2020-10-14 20:02:28	20	ПАЗ-3205-18/1	2007	18
5	Вінниця	3	3	10:20:00	13:40:00	2020-10-14 20:02:28	3	БОГДАН-20	2009	20
6	Бершадь	3	29	14:00:00	17:20:00	2020-10-14 20:02:28	3	БОГДАН-20	2009	20
7	Вінниця	4	4	08:10:00	08:50:00	2020-10-14 20:02:28	4	ЭТАЛОН-26	2007	26
8	Калинівка	4	30	10:40:00	11:20:00	2020-10-14 20:02:28	4	ЭТАЛОН-26	2007	26
9	Вінниця	5	5	11:00:00	13:00:00	2020-10-14 20:02:28	5	ПАЗ-3205-35	2009	35
10	Могилів П...	5	31	13:30:00	15:30:00	2020-10-14 20:02:28	5	ПАЗ-3205-35	2009	35
33	Вінниця	17	17	07:10:00	08:45:00	2020-10-14 20:02:28	17	МЕРСЕДЕС-47	2006	46
34	Шаргород	17	43	13:35:00	15:10:00	2020-10-14 20:02:28	17	МЕРСЕДЕС-47	2006	46
11	Вінниця	6	6	07:00:00	08:15:00	2020-10-14 20:02:28	6	МЕРСЕДЕС-18	2003	18
12	Хмільник	6	32	08:45:00	10:00:00	2020-10-14 20:02:28	6	МЕРСЕДЕС-18	2003	18
51	Вінниця	6	26	11:45:00	14:50:00	2020-10-14 20:02:28	6	МЕРСЕДЕС-18	2003	18

Рисунок 3.41 – Результат виконання запити 3

3.5 Розробка back-end частини для інформаційної системи пасажирських автоперевезень.

Основна задача back-end частини, це взаємодія із базою даних та відтворення інформації із грх файлів.

Для підключення до бази даних було прописано connection, який працює з бібліотекою mysql2. Об'явлення connection показано на рисунку 3.42.

```

const UPDATE_TO_DRIVER_TB = `UPDATE bus_station.driver drv
                             set driver_pip = ?,
                                 phone = ?,
                                 expireance = ?,
                                 country = ?,
                                 city = ?,
                                 street = ?,
                                 num_house = ?,
                                 flat = ?
                             WHERE drv.id_driver = ?
`;

const connection = mysql.createConnection({
  host: '127.0.0.1',
  user: 'root',
  password: 'pavlo',
  port: 3306,
  database: 'bus_station'
});

const connection2 = mysql2.createConnection({
  host: '127.0.0.1',
  user: 'root',
  password: 'pavlo',
  port: 3306,
  database: 'bus_station'
}).promise();

connection.connect(err => {
  console.log(err);
  if (err) {
    return err;
  }
});

```

Рисунок 3.42 – Підключення до бази даних

Приклад створення API, до якого будемо звертатися із front-end частини, приклад зображено на рисунку 3.43.

```

app.post("/driver_table/driver_insnt", jsonParser, function (req, res) {
  let filter = [];
  if (!req.body) return res.sendStatus(400);
  filter.push(String(req.body.driver_pip));
  filter.push(String(req.body.phone));
  filter.push(Number(req.body.expireance));
  filter.push(String(req.body.country));
  filter.push(String(req.body.city));
  filter.push(String(req.body.street));
  filter.push(Number(req.body.num_house));
  filter.push(Number(req.body.flat));
  //console.log(String(filter));
  connection2.query(INSERT_TO_DRIVER_TB, [filter])
    .then((rows, fields) => {
      let i = 1;
      //console.log(rows);
      return res.send(rows)
    })
    .catch(err => {
      console.log(err);
    });
});

```

Рисунок 3.43 – Приклад створення API

Для створення API, при зверненні по якому буде вертатися інформація про місцезнаходження транспортного засобу, для цього використаємо бібліотек parser, яка буде розшифровувати інформацію із gpx файлу. Реалізація зображена на рисунку 3.44.

```

app.get('/drevel', (req, res) => {
  let fs = require('fs');
  let parse = require('xml-parser');
  let gpx = fs.readFileSync('./gpxfile/drevel/avto.gpx', 'utf8');
  let obj = parse(gpx);
  res.json({
    drive: obj
  });
});

```

Рисунок 3.44 – Приклад створення API з інформацією про місце знаходження транспортного засобу

3.6 Висновки

Отже, в даному розділі було розкрито основну ідею створення інформаційної системи моніторингу пасажирських автоперевезень. За

допомогою схеми та побудови UML діаграм показано принципи роботи нашої системи.

Після розробки макетів сторінок розроблено front-end для сторінок, описано зміни при натисканні на різні об'єкти, також для зв'язку із базою даних та grx файлами було розроблено back-end частину.

Спроектвано базу даних відповідно до розробленої моделі, та розроблено запити, для отримання інформації, або редагування.

4 ЕКОНОМІЧНА ЧАСТИНА

4.1 Оцінювання комерційного потенціалу розробки «Інформаційна система моніторингу пасажирських автоперевезень»

Метою проведення технологічного аудиту є оцінювання комерційного потенціалу розробки. Для проведення технологічного аудиту було залучено 2-х незалежних експертів.

Здійснюємо оцінювання комерційного потенціалу розробки за 12-ма критеріями, наведеними у таблиці 4.1, за 5-ти бальною шкалою.

Таблиця 4.1 – Критерії оцінювання комерційного потенціалу розробки та їх бальна оцінка

Бали (за 5-ти бальною шкалою)					
Кри-те-Рій	0	1	2	3	4
Технічна здійсненність концепції:					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздатність продукту в реальних умовах
Ринкові переваги (недоліки):					
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо вища за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижче за ціни аналогів	Ціна продукту значно нижче за ціни аналогів

Продовження таблиці 4.1

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

Продовження таблиці 4.1

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

Результати оцінювання комерційного потенціалу розробки наведено в таблиці 4.2.

Таблиця 4.2 – Результати оцінювання комерційного потенціалу розробки

Критерії	Прізвище, ініціали, посада та науковий ступінь експерта	
	1. Жуков С.О., к.т.н., доц. кафедри САІТ	2. Козачко О.М., к.т.н., доц. кафедри САІТ
	Бали, виставлені експертами:	
1	4	4
2	0	1
3	4	4
4	2	3
5	4	3
6	2	3
7	2	3
8	3	3
9	4	4
10	4	4
11	4	3
12	4	4
Сума балів	СБ ₁ = 37	СБ ₂ = 39
Середньоарифметична сума балів СБ	$\overline{СБ} = \frac{\sum_{i=1}^3 СБ_i}{2} = 38$	

Отже, з отриманих даних таблиці 4.1 видно, що рівень потенціалу нової розробки – вище середнього. Дана розробка є конкурентоспроможною з аналогами, так як для її розробки було проаналізовано недоліки та переваги аналогових продуктів, та на основі цього впроваджену у розробку. Вона має соціологічний вплив, так як покращує ефективність контролю бюджету користувачів, що є важливим у житті людей з обмеженими фінансами.

4.2 Прогнозування витрат на виконання науково-дослідної роботи та конструкторсько–технологічної роботи

Для розробки нового програмного продукту необхідні такі витрати.

Основна заробітна плата для розробників визначається за формулою:

$$Z_o = \frac{M}{T_p} \cdot t, \quad (4.1)$$

де M – місячний посадовий оклад конкретного розробника;

T_p – кількість робочих днів у місяці, $T_p = 20$ днів;

t – число днів роботи розробника, $t = 37$ днів.

Розрахунки заробітних плат для керівника і програміста наведені в таблиці 4.3.

Таблиця 4.3 – Розрахунки основної заробітної плати

Працівник	Оклад M , грн.	Оплата за робочий день, грн.	Число днів роботи, t	Витрати на оплату праці, грн.
Науковий керівник	7500	375	4	1500
Інженер	9600	480	33	15840
Всього:				17340

Розрахуємо додаткову заробітну плату, вона розраховується як 10-12% від суми основної заробітної плати всіх розробників:

$$Z_{\text{дод}} = 0,1 \cdot 17340 = 1734 \text{ (грн)}. \quad (4.2)$$

Нарахування на заробітну плату $H_{\text{зп}}$ для працівників бюджетної сфери становить 22% від суми основної та додаткової заробітної плати:

$$H_{\text{зп}} = (Z_o + Z_p) \cdot \frac{\beta}{100}, \quad (4.3)$$

$$H_{\text{зп}} = (17340 + 1734) \cdot \frac{22}{100} = 4196,28 \text{ (грн)}.$$

Розрахунок амортизаційних витрат для програмного забезпечення виконується за такою формулою:

$$A = \frac{C \cdot H_a}{100} \cdot \frac{T}{12}. \quad (4.4)$$

де C – балансова вартість обладнання, грн;

H_a – річна норма амортизаційних відрахувань % (для програмного забезпечення 25%);

T – Термін використання ($T=2$ міс.).

Розрахунки амортизаційних витрат для програмного забезпечення наведені в таблиці 4.4.

Таблиця 4.4 – Розрахунок амортизаційних відрахувань

Найменування програмного забезпечення	Балансова вартість, грн.	Норма амортизації, %	Термін використання, міс.	Величина амортизаційних відрахувань, грн
Персональний комп'ютер	12800	25	2	533,3
GPS-трекер	2250	25	2,5	117,5
Всього:				640,8

Розрахуємо витрати на комплектуючі. Витрати на комплектуючі розрахуємо за формулою:

$$K = \sum_1^n N_i \cdot C_i \cdot K_i, \quad (4.5)$$

де n – кількість комплектуючих;

N_i – кількість комплектуючих i -го виду;

C_i – покупна ціна комплектуючих i -го виду, грн;

K_i – коефіцієнт транспортних витрат (прийmemo $K_i = 1,1$).

Таблиця 4.5 - Витрати на комплектуючі, що були використані для розробки ПЗ.

Найменування матеріалу	Одиниці виміру	Ціна, грн.	Витрачено	Вартість витрачених матеріалів, грн.
Зошит	шт.	45	1	45
Пачка паперу	уп.	100	1	100
Ручка	шт.	4	1	4
Всього з урахуванням транспортних витрат				166,9

Витрати на силову електроенергію розраховуються за формулою:

$$V_e = V \cdot P \cdot \Phi \cdot K_{\Pi}, \quad (4.6)$$

де V – вартість 1кВт-години електроенергії ($V=2.64$ грн/кВт);

P – установлена потужність комп'ютера ($P=0,6$ кВт);

Φ – фактична кількість годин роботи комп'ютера ($\Phi=93$ год.);

K_{Π} – коефіцієнт використання потужності ($K_{\Pi} < 1$, $K_{\Pi} = 0,7$).

$$V_e = 2,64 \cdot 0,6 \cdot 93 \cdot 0,7 = 79.56 \text{ (грн).}$$

Розрахуємо інші витрати $V_{ін}$. Інші витрати I_B можна прийняти як (100...300)% від суми основної заробітної плати розробників та робітників, які були виконували дану роботу, тобто:

$$V_{ін} = (1..3) \cdot (Z_o + Z_p), \quad (4.7)$$

Отже, розрахуємо інші витрати:

$$V_{ін} = 1 * (17340 + 1734) = 19074 \text{ (грн)}.$$

Сума всіх попередніх статей витрат дає витрати на виконання даної частини роботи:

$$B = Z_o + Z_d + H_{зп} + A + K + B_e + I_B, \quad (4.8)$$

$$B = 17340 + 1734 + 4196,28 + 640,8 + 166,9 + 79,56 + 19074 = 43257,24 \text{ (грн)}.$$

Розрахуємо загальну вартість наукової роботи $B_{заг}$ за формулою:

$$B_{заг} = \frac{B_{ін}}{\alpha}, \quad (4.9)$$

де α – частка витрат, які безпосередньо здійснює виконавець даного етапу роботи, у відн. одиницях = 1.

$$B_{заг} = \frac{43257}{1} = 43257.$$

Прогнозування загальних витрат $ЗВ$ на виконання та впровадження результатів виконаної наукової роботи здійснюється за формулою:

$$ЗВ = \frac{B_{заг}}{\beta}, \quad (4.10)$$

де β – коефіцієнт, який характеризує етап (стадію) виконання даної роботи.

Отже, розрахуємо загальні витрати:

$$ЗВ = \frac{43257}{0,9} = 48\,078 \text{ (грн).}$$

4.3 Прогнозування комерційних ефектів від реалізації результатів розробки

Спрогнозуємо кількісно, яку вигоду можна отримати у майбутньому від впровадження результатів виконаної наукової роботи.

Виконання наукової роботи та впровадження її результатів буде здійснюватися протягом одного року. Основні позитивні результати від впровадження розробки очікуються протягом 3-х років після її впровадження. Одним із основних позитивних результатів є зростання величини прибутку.

При реалізації результатів наукової розробки покращується якість програмного продукту, що дозволяє підвищити ціну його реалізації на 150 грн. Кількість одиниць реалізації програмного засобу також збільшиться: протягом першого року — на 700 шт., протягом другого року — ще на 450 шт., протягом третього року — ще на 300 шт.

Реалізація продукції до впровадження результатів наукової розробки складала 50 шт., а ціна — 250 грн.

Спрогнозуємо збільшення чистого прибутку підприємства від впровадження результатів наукової розробки у кожному році відносно базового за такою формулою:

$$\Delta\Pi_i = \sum_1^n (\Delta\Pi_{\text{я}} \cdot N + \Pi_{\text{я}}\Delta N)_i, \quad (4.11)$$

де $\Delta\Pi_0$ – покращення основного оціночного показника від впровадження результатів розробки у даному році. Зазвичай таким показником може бути ціна одиниці нової розробки;

N – основний кількісний показник, який визначає діяльність підприємства у даному році до впровадження результатів наукової розробки;

ΔN – покращення основного кількісного показника діяльності підприємства від впровадження результатів розробки;

Π_0 – основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки;

n – кількість років, протягом яких очікується отримання позитивних результатів від впровадження розробки.

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт 0,8333.

ρ – коефіцієнт, який враховує рентабельність продукту. Рекомендується приймати $= 0,2 \dots 0,3$;

v – ставка податку на прибуток. $v = 18\%$.

Тоді, збільшення чистого прибутку підприємства протягом першого року складе:

$$\Delta\Pi_1 = [150 \cdot 50 + (250 + 150) \cdot 700] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 39290,09 \text{ (грн)},$$

Протягом другого року:

$$\Delta\Pi_2 = [150 \cdot 50 + (250 + 150) \cdot (700 + 450)] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 61\,710 \text{ (грн)},$$

Протягом третього року:

$$\Delta\Pi_3 = [150 \cdot 50 + (250 + 150) \cdot (700 + 450 + 300)] \cdot 0,8333 \cdot 0,2 \cdot \left(1 - \frac{18}{100}\right) = 79900 \text{ (грн)}.$$

Отже, протягом трьох років підприємство може розраховувати на збільшення чистого прибутку від реалізації наукової розробки.

4.4 Розрахунок ефективності вкладених інвестицій та період їх окупності

Визначимо абсолютну і відносну ефективність вкладених інвестором інвестицій та розрахуємо термін окупності.

Абсолютна ефективність $E_{абс}$ вкладених інвестицій розраховується за формулою:

$$E_{абс} = (ПП - PV), \quad (4.12)$$

де ПП – приведена вартість всіх чистих прибутків, які отримає підприємство (організація) від реалізації результатів наукової розробки, грн.;

PV – теперішня вартість інвестицій $PV = ЗВ = 48\,078$ (грн).

Рисунок, що характеризує рух платежів (інвестицій та додаткових прибутків) буде мати вигляд, як на рисунку 4.1.

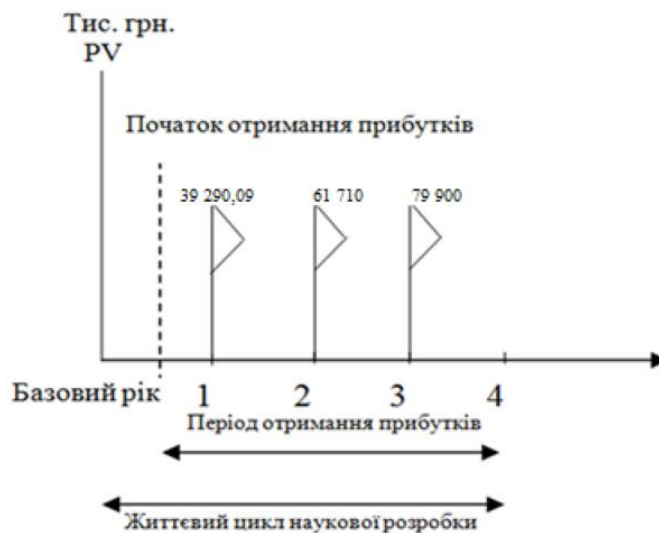


Рисунок 4.1 – Вісь часу з фіксацією платежів, що мають місце під час розробки та впровадження результатів наукової роботи

У свою чергу, приведена вартість всіх чистих прибутків ПП розраховується за формулою:

$$ПП = \sum_1^T \frac{\Delta\Pi_i}{(1+\tau)^t}, \quad (4.13)$$

де $\Delta\Pi_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої НДДКР, грн;

t – період часу, протягом якого виявляються результати впровадженої НДДКР, 3 роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні; для України цей показник знаходиться на рівні 0,1;

t – період часу (в роках) від моменту отримання чистого прибутку до точки “0”.

Отже, розрахуємо вартість чистого прибутку:

$$ПП = \frac{48\,063,6}{(1+0,1)^0} + \frac{39\,290,09}{(1+0,1)^2} + \frac{61\,710}{(1+0,1)^3} + \frac{79\,900}{(1+0,1)^4} = 181\,474,87 \text{ (грн)}.$$

Тоді розрахуємо $E_{абс}$:

$$E_{абс} = 181\,474,87 - 48\,078 = 133\,411,27 \text{ (грн)}.$$

Оскільки $E_{абс} > 0$, то вкладання коштів на виконання та впровадження результатів НДДКР буде доцільним.

Розрахуємо відносну (щорічну) ефективність вкладених в наукову розробку інвестицій E_v за формулою:

$$E_v = \sqrt[T]{1 + \frac{E_{абс}}{PV}} - 1, \quad (4.14)$$

де $E_{абс}$ – абсолютна ефективність вкладених інвестицій, грн;

PV – теперішня вартість інвестицій $PV = 3B$, грн;

T – життєвий цикл наукової розробки, роки.

Тоді будемо мати:

$$E_B = \sqrt[3]{1 + \frac{133\,411,27}{48\,078}} - 1 = 0,56 \text{ або } 56(\%).$$

Далі, розраховану величина E_B порівнюємо з мінімальною (бар'єрною) ставкою дисконтування $\tau_{\text{мін}}$, яка визначає ту мінімальну дохідність, нижче за яку інвестиції вкладатися не будуть. У загальному вигляді мінімальна (бар'єрна) ставка дисконтування $\tau_{\text{мін}}$ визначається за формулою:

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

де d – середньозважена ставка за депозитними операціями в комерційних банках; в 2020 році в Україні $d = 0,2$;

f – показник, що характеризує ризикованість вкладень, величина $f = 0,1$.

$$\tau = 0,2 + 0,1 = 0,3.$$

Оскільки $E_B = 55(\%) > \tau_{\text{мін}} = 0,3 = 30(\%)$, то інвестор буде зацікавлений вкладати гроші в дану наукову розробку.

Термін окупності вкладених у реалізацію наукового проекту інвестицій $T_{\text{ок}}$ розраховується за формулою:

$$T_{\text{ок}} = \frac{1}{E_B}, \quad (4.16)$$

Якщо $T_{\text{ок}} < 3\dots 5$ -ти років, то фінансування наукової розробки є доцільним.

$$T_{\text{ок}} = \frac{1}{0,56} = 1,82 \text{ (років)}.$$

Обрахувавши термін окупності даної наукової розробки, можна зробити висновок, що фінансування даної наукової розробки буде доцільним, оскільки $T_{ок} < 3$ років.

4.5 Висновки

Результати проведених розрахунків дають можливість зробити висновок про доцільність розробки інформаційної системи моніторингу пасажирських автоперевезень та впровадження нашої наукової роботи. Це підтверджують такі показники як:

- абсолютна ефективність вкладених інвестицій, яка дорівнює 133 411,27 (грн)., що є більшим 0 і вказує на те, що інвестор може бути зацікавленим у нашій розробці;

- відносна ефективність наукової розробки становить 56%, що є вищим за мінімальну ставку дисконтування (30%), тому вкласти кошти у нашу розробку є вигідніше, ніж покласти кошти на депозит;

- термін окупності вкладених у реалізацію наукового проекту інвестицій складе 1,82 років, що є менше 3-ох і вказує швидку окупність вкладених інвестицій.

Крім того, розраховано, що наукова розробка принесе підприємству додатковий прибуток протягом 3-х років за рахунок покращення її якості порівняно з існуючими аналогами.

Усе це, узятє разом, забезпечує прийняття рішення про доцільність виготовлення нового продукту.

ВИСНОВКИ

В магістерській кваліфікаційній роботі розроблено інформаційну систему моніторингу пасажирських автоперевезень. В роботі приведено огляд проблем розробки веб-системи моніторингу пасажирських автоперевезень. Запропоновані оптимальні технології та формати для реалізації даної системи з точки зору можливості її використання на персональних комп'ютерах із платним ліцензійним забезпеченням, так і з вільним для розповсюдження програмним забезпеченням. Здійснено програмну реалізацію системи на прикладі Вінницької області. Розроблено рекомендації по впровадженню та експлуатації розробленої інформаційної системи моніторингу пасажирських автоперевезень.

В першому розділі під час проведення огляду проблем створення інформаційної системи моніторингу пасажирських автоперевезень було доведено актуальність розробки даної системи, а також розглянуто український та зарубіжний досвід в даному напрямку.

В другому розділі у ході проведення огляду програмних засобів було вибрані оптимальні засоби для створення інформаційної системи моніторингу пасажирських автоперевезень. Також було приведено загальну ідею та формалізацію моделі рейтингу водія.

В третьому розділі розроблено загальну архітектуру системи, розроблено базу даних та інтерфейс користувача. Також детально описано підходи до розробки серверної та клієнтської частини системи.

В четвертому розділі було виконано економічну частину кваліфікаційної роботи під час розрахунків було доведено:

– абсолютна ефективність вкладених інвестицій, яка дорівнює 133 411,27 грн., що є більшим 0 і вказує на те, що інвестор може бути зацікавленим у нашій розробці;

– відносна ефективність наукової розробки становить 56%, що є вищим за мінімальну ставку дисконтування (30%), тому вкласти кошти у нашу розробку є вигідніше, ніж покласти кошти на депозит.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Розробка веб-системи моніторингу пасажирських автоперевезень / П. П. Машницький, Є. М. Крижановський// XLVIII Науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (2019), електронне наукове видання матеріалів конференції, м. Вінниця. – Режим доступу: <https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2019/rt/printerFriendly/7215/0>
2. Сапронов О. М. Автомобільний транспорт як складова міського пасажирського транспорту у контексті енергетичної, соціальної, економічної та екологічної безпеки / О. М. Сапронов // Освіта і управління. – 2007. – № 2. – С. 152–158.
3. Прейгер Д. К. Стратегічні напрями розвитку транспортної галузі України у після кризовий період /Д. К. Прейгер, О. В. Собкевич, О. Ю. Ємельянова. – К. : НІСД, 2011. – 48 с.
4. Статистичний збірник «Транспорт і зв'язок України, 2016». Відповідальний за випуск О. О. Кармазіна.[Електронний ресурс]. – Режим доступу: <http://www.ukrstat.gov.ua>.
5. Стройко Т. В. Інноваційне оновлення регіональної транспортної інфраструктури на принцип інжинірингу / В. В. Бондар, Т. В. Стройко // Вісник Приазовського державного технічного університету. Серія: Економічні науки [Збірник наукових праць]. – Маріуполь: ДВНЗ «Приазовський державний технічний університет», 2017. – Випуск 33 - С. 296-302.
6. Закон України «Про автомобільний транспорт». [Електронний ресурс]. – Режим доступу: zakon.rada.gov.ua/go/2344-14.
7. Транспортна стратегія України на період до 2020 року [Електронний ресурс]. – Режим доступу:<http://www.transport-ukraine.eu/page/transportna-strategiya-ukrayini-na-period-do-2020-roku>.
8. Система моніторингу транспорту - плюси і ... плюси. [Електронний ресурс]. – Режим доступу: <https://intelli.com.ua/ua/statti/systema-monitorynhu-transportu-pliusy-i-pliusy.html>.

9. Принцип роботи системи контролю GPS моніторингу. [Електронний ресурс]. – <https://freetrack.com.ua/technology/>.
10. Системний аналіз та проектування ГІС. – Електронний навчальний посібник / Є. М. Крижановський, В.Б. Мокін, А.Р. Ящолт, Л.М. Скорина. – Вінниця : ВНТУ, 2015. – 127 с.
11. Система GPS моніторингу автомобіля. [Електронний ресурс]. – Режим доступу: https://www.benishgps.com/ge/products/sputnikovaya_sistema_monitoringa_transporta/.
12. Управління флотом. [Електронний ресурс]. – Режим доступу: <https://techsisindia.com/solutions/fleet-management/>.
13. MySQL по максимуму. – Електронний навчальний посібник / Шварц Б., Зайцев П., Ткаченко В.,– Санкт-Петербург, 2018. – 864с.
14. Мартін Груббер. Розуміння SQL. / Мартін Груббер. – Санкт-Петербург: ПИТЕР, 1993. – 289с.
15. OpenLayers. [Електронний ресурс]. – Режим доступу: <https://leafletjs.com/>.
16. Express.js [Електронний ресурс]. – Режим доступу до ресурсу: <https://uk.wikipedia.org/wiki/Express.js>
17. Node.js в дії, посібник / Майк Кантелон, Марк Гартер. – Manning, 2013. – 416 с.
18. React: Up & Running: створення веб-додатків, посібник / Стоян Стефанов. – O'Reilly Media, Inc., 2016. – 222 с.
19. HTML & CSS: розробка та створення веб-сайтів, посібник / Джон Дюкетт. – John Wiley & Sons, 2011. – 512 с.
20. Методичні вказівки до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання / Уклад. В. Б. Мокін, С. О. Жуков, А. Р. Ящолт, О. М. Козачко, Л. М. Скорина. – Електронне видання. – Вінниця : ВНТУ, 2018. – 48 с.

Додаток А

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет комп'ютерних систем і автоматики

ЗАТВЕРДЖУЮ
Завідувач кафедри САІТ
_____ д.т.н., проф. В.Б. Мокін
(підпис)
“ ___ ” _____ 2020 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на магістерську кваліфікаційну роботу
«ІНФОРМАЦІЙНА СИСТЕМИ МОНІТОРИНГУ
ПАСАЖИРСЬКИХ АВТОПЕРЕВЕЗЕНЬ»
08-53.МКР.003.03.000.ТЗ

Керівник магістерської кваліфікацій-
ної роботи
доцент, к.т.н.
_____ Є. М. Крижановський
(підпис)
“ ___ ” _____ 2020 р.

Розробив студент гр. ЗІСТ-19м
_____ П. П. Машницький
(підпис)
“ ___ ” _____ 2020 р.

Вінниця 2020

1. Підстава для проведення робіт

Підставою для виконання роботи є наказ № ___ по ВНТУ від «__» _____ 2020 р., та індивідуальне завдання на МКР, затверджене протоколом № ___ засідання кафедри САІТ від «__» _____ 2020 р.

2. Джерела розробки:

- Геоінформаційні системи в екології. – Електронний навчальний посібник / Є. М. Крижановський, В.Б. Мокін, – Вінниця : ВНТУ, 2014. – 182 с.
- Самойленко В.М. Географічні інформаційні системи та технології. – 2010., К.: Ніка-Центр. – 448 с.3. Мета і призначення роботи

3. Мета і призначення роботи

Розробка інформаційної системи моніторингу пасажирських автоперевезень.

4. Вихідні дані для проведення робіт:

- перелік атрибутивних даних про транспортні засоби;
- перелік атрибутивних даних про розклад руху автобусів Вінницької області;
- узагальнена електронна карта України.

5. Методи дослідження:

- метод формалізації;
- моделювання системи.

6. Етапи роботи і терміни їх виконання

- a) Аналіз предметної області __. – __
- b) Розробка інформаційної технології __ – __
- c) Реалізація інформаційної технології..... __ – __
- d) Розробка інструкції користувача..... __ – __

7. Очікувані результати та порядок реалізації

Отримання системи моніторингу пасажирських автоперевезень.

8. Вимоги до розробленої документації

Пояснювальна записка оформлена у відповідності до вимог «Методичних вказівок до виконання та оформлення магістерських кваліфікаційних робіт для студентів спеціальності 126 – «Інформаційні системи та технології» денної форми навчання».

9. Порядок приймання роботи

Публічний захист..... __. __. 2020 р.

Початок розробки «__» _____ 2020 р.

Граничні терміни виконання МКР «__» _____ 2020 р.

Розробив студент групи 2ІСТ-19м _____ Машницький П.П.

Додаток Б

Інструкція користувача

Для роботи із інформаційною системою необхідно її запусити в браузері. На рисунку Б.1 показано головну сторінку системи.

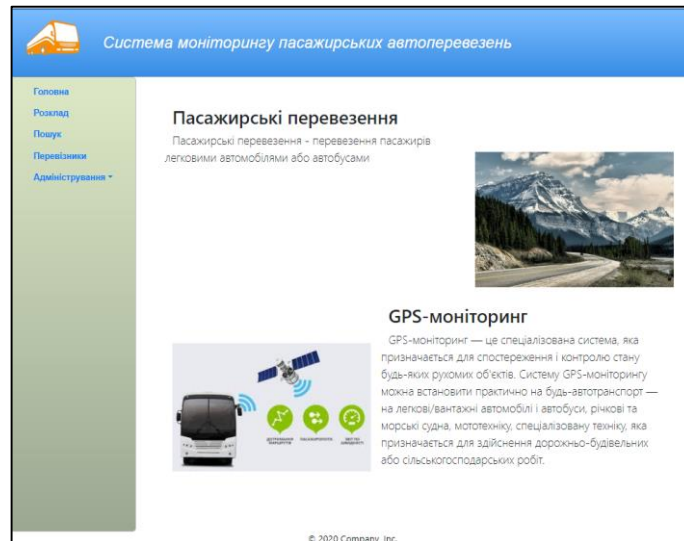


Рисунок Б.1 – Головна сторінка

Щоб подивитись місце знаходження автобусу необхідно в меню навігації вибрати «Розклад», відкриється сторінка «Маршрути», яка показана на рисунку Б.2, де необхідно вибрати маршрут, натиснувши на кнопку із необхідним маршрутом, відкриється спливаюче вікно, яке показано на рисунку Б.3.

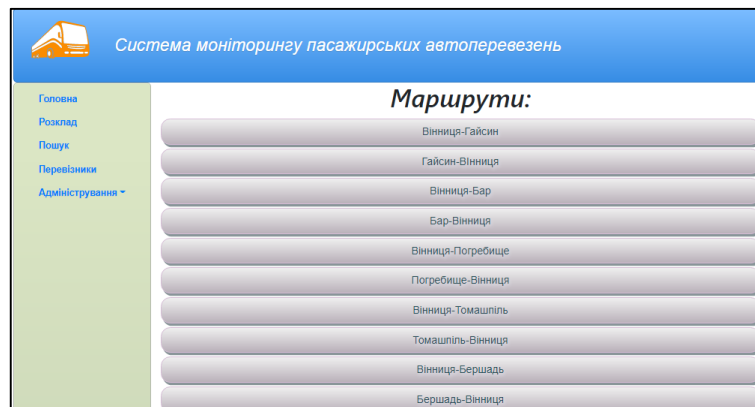


Рисунок Б.2 – Сторінка маршрути

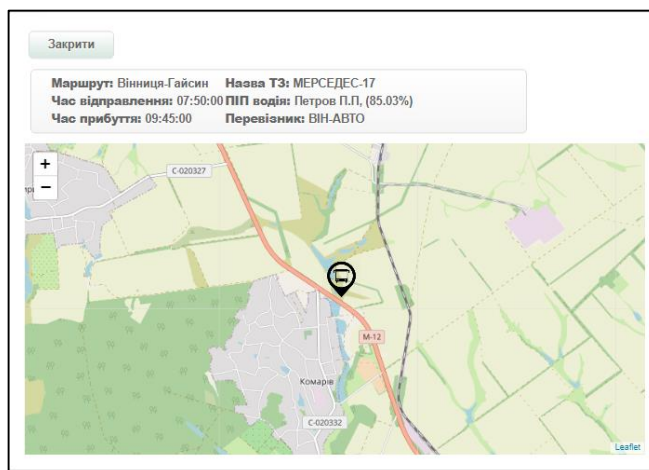


Рисунок Б.3 – Спливаюче вікно

Щоб переглянути інформацію про перевізників необхідно в меню навігації обрати «Перевізники», відкриється сторінка перевізники, яку показано на рисунку Б.4, якщо на сторінці натиснути на кнопку водії, відкриється спливаюче вікно із інформацією про водія. Вигляд спливаючого вікна показано на рисунку Б.5.

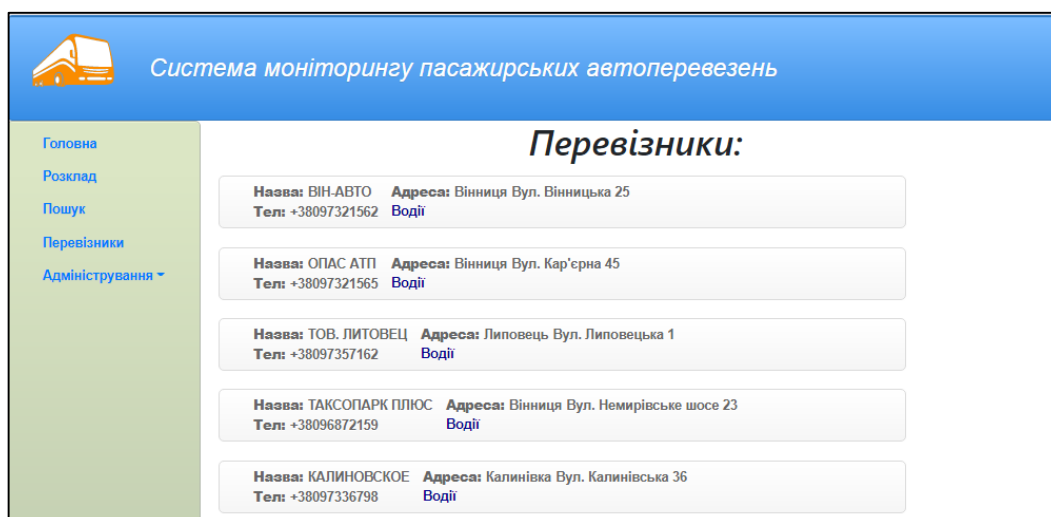


Рисунок Б.4 – Сторінка перевізники

Закрити

ПІП: Іванов І.І. Оцінити	Адреса: Україна Теплик Київська 23
Стаж: 12	Рейтинг: 89.75
Тел.: +3806821651	
Якість: [Slider]	Ввічливість: [Slider]
Пунктуальність: [Slider]	Безпечність: [Slider]

ПІП: Поліщук С.В. Оцінити	Адреса: Україна Вінниця Східна 86
Стаж: 5	Рейтинг: 85.81
Тел.: +3806951235	
Якість: [Slider]	Ввічливість: [Slider]
Пунктуальність: [Slider]	Безпечність: [Slider]

Рисунок Б.5 – Спливаюче вікно

Щоб оцінити водія необхідно на спливаючому вікні натиснути на кнопку «Оцінити». Відкриється вікно на якому можна оцінити водія виставивши оцінку, вікно для оцінювання зображено на рисунку Б.6.

Закрити

Іванов І.І.

Якість: [Slider]

Ввічливість: [Slider]

Пунктуальність: [Slider]

Безпечність: [Slider]

Надіслати

Рисунок Б.6 – Вікно для оцінювання

Додаток В

Лістинг програмного коду.

```
const express = require('express');
const cors = require('cors');
const mysql = require('mysql');

const app = express();

const SELECT_ALL_STOP_BUS_SESTEM_QUERY = 'SELECT *
FROM stop ' +
  'INNER JOIN route ON stop.id_roud = route.id_rout ' +
  'INNER JOIN vehicle ON stop.id_vehicle = vihicle.id_vihicle ' +
  'INNER JOIN carrier ON vehicle.id_carrier = carrier.id_carrier';
const connection = mysql.createConnection({
  host : 'localhost',
  user : 'root',
  password : 'pavlo',
  database : 'bus_system'
});

connection.connect(err => {
  if (err){
    return err;
  }
});

app.use(cors());

app.get('/',(req,res) => {
  res.send('go to to see')
});
app.get ('/stopbus',(req,res) =>{
  connection.query(SELECT_ALL_STOP_BUS_SESTEM_QUERY, (err,
results) =>{

    if (err){
      return res.send(err)
    }
    else {
      return res.json({
        data:results
      })
    }
  })
});
```

```
    })
  }
});
});
```

```
app.get('/stopbus/add', (req,res)=>{
  const SELECT_SEARCH_BUS = `SELECT *FROM ((stop JOIN route
ON stop.id_roud = route.id_rout)JOIN bus_station ON route.id_station_b =
bus_station.id_station)`;
  connection.query(SELECT_SEARCH_BUS,(err,result) =>{
    if (err){
      return res.send(err)
    }
    else {
      return res.json({
        datas:result
      })
    }
  });
});
```

```
app.get('/drevel',(req,res) => {
```

```
  let fs = require('fs');
  let parse = require('xml-parser');
  let gpx = fs.readFileSync('./gpxfile/drive1/avto.gpx', 'utf8');
```

```
  let obj = parse(gpx);
```

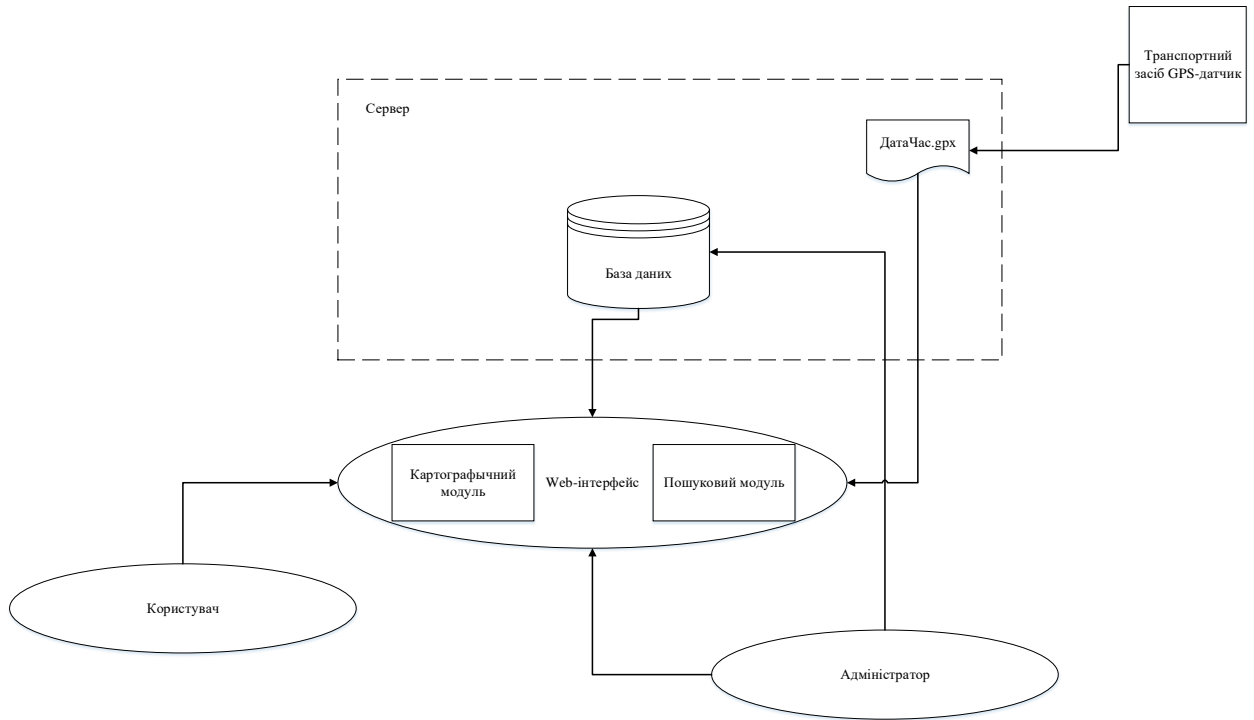
```
  res.json({
    drive:obj
  })
```

```
});
```

Додаток Г

Графічна частина

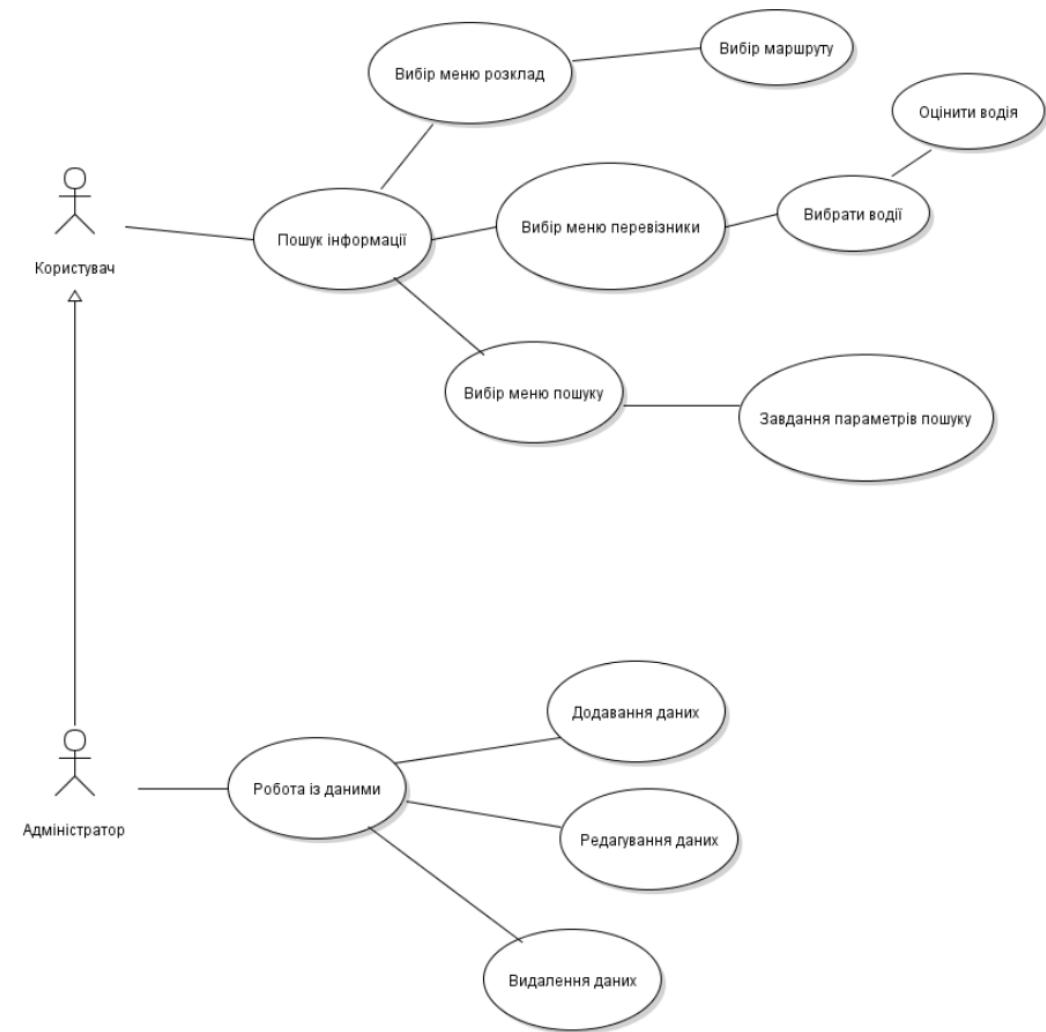
Загальна структура інформаційної системи пасажирських автоперевезень



08–53.МКР.003.02.00.ПЛ

Змн.	Арк.	Лист	Підпис	Дата	Інформаційної системи моніторингу пасажирських автоперевезень	Літ.	Маса	Масштаб
Розробив		Машницький П.П.			Загальна структура інформаційної системи пасажирських автоперевезень	Арк.	1	Аркушів 8
Перевірив		Крижановський С.М.						
Рецензент		Паламарчук С. А.						
Н. Контр.		Жуков С.О.			ВНТУ, 2ІСТ-19м			
Зав. каф.		Мокін В.Б.						

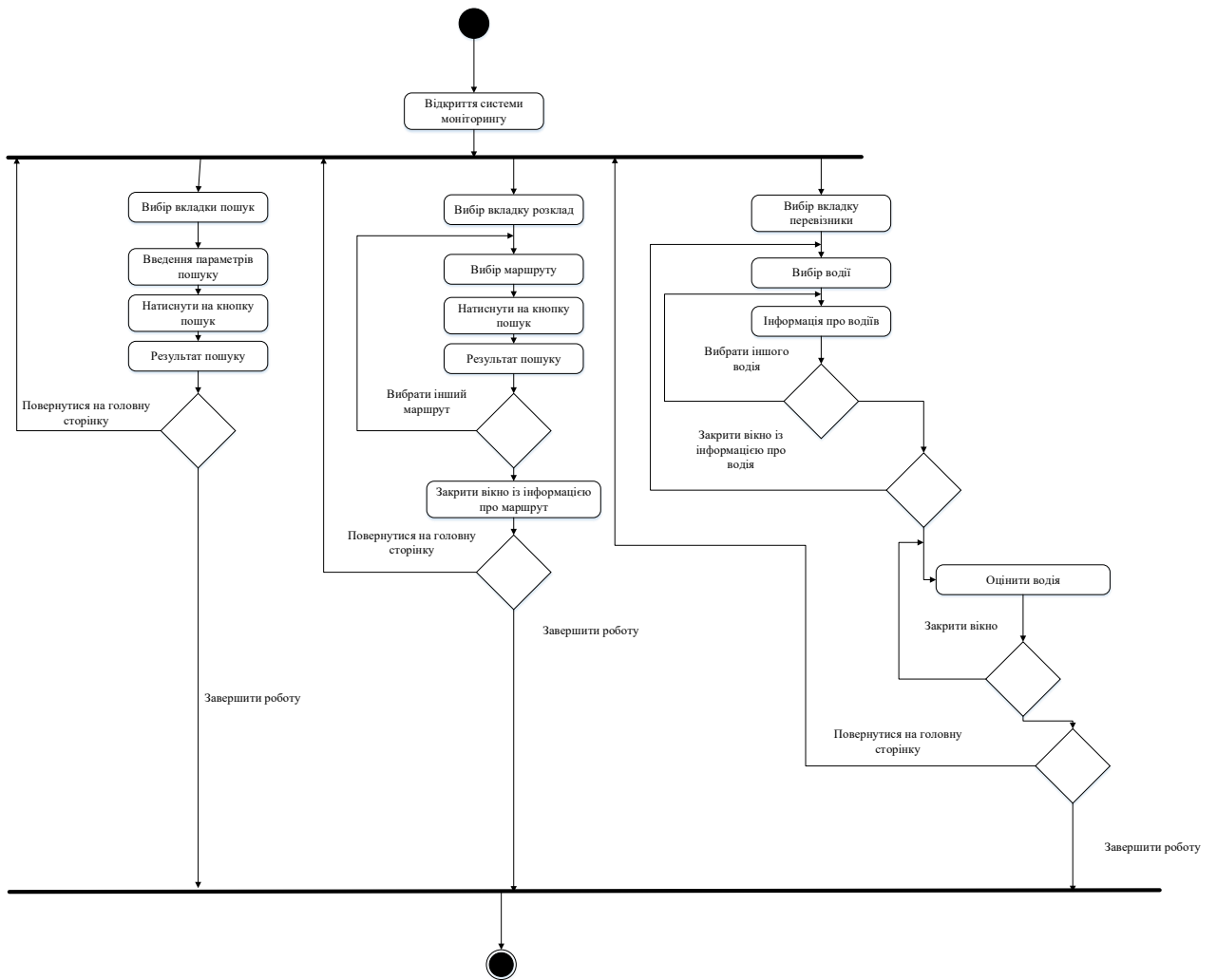
Діаграма варіантів використання



08–53.МКР.003.02.00.ПЛ

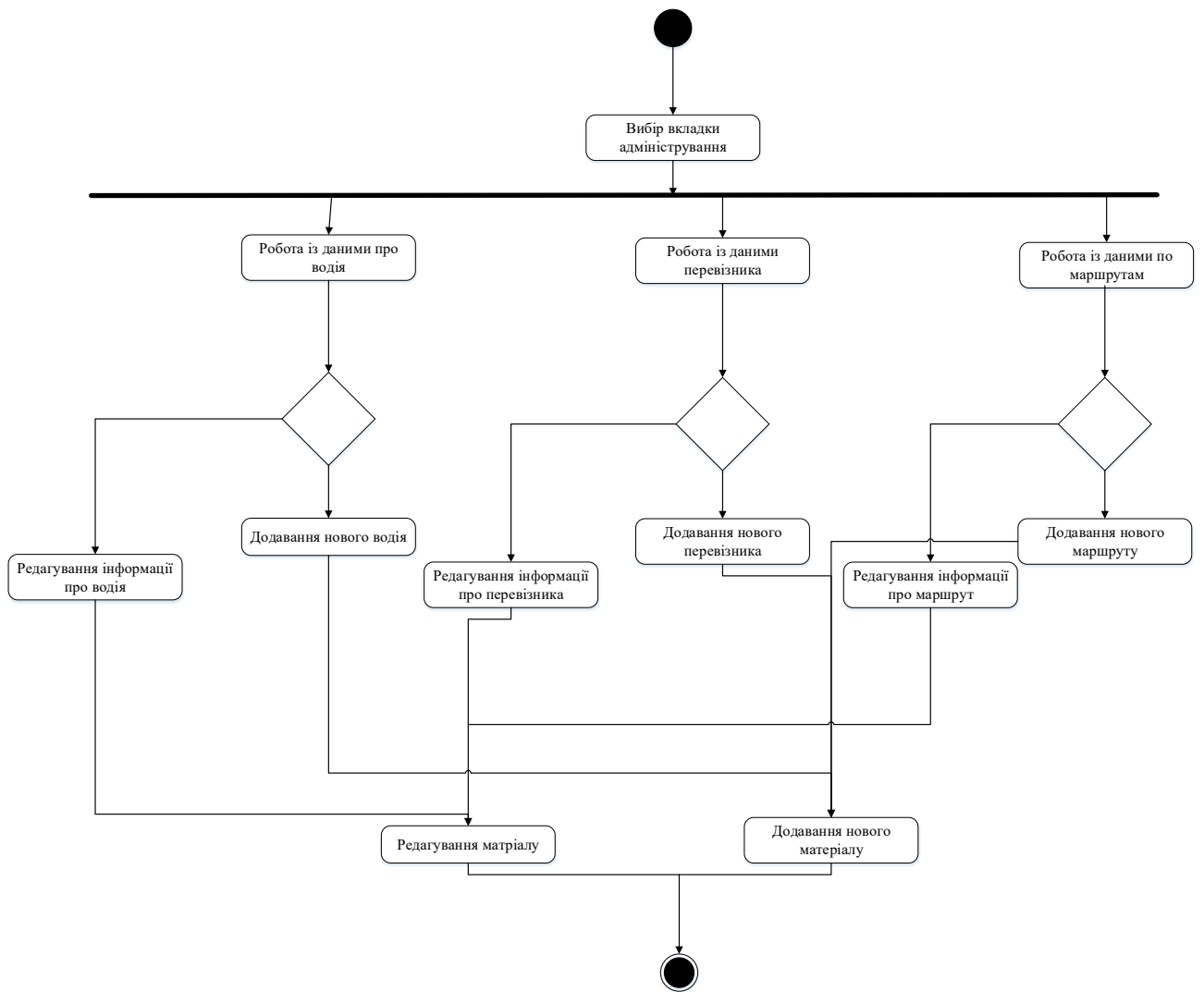
Змн.	Арк.	Лист	Підпис	Дата	Інформаційної системи моніторингу пасажирських автоперевезень	Літ.	Маса	Масштаб
								1 : 1
Розробив		Машницький П.П.						
Перевірив		Крижановський С.М.						
Рецензент		Паламарчук Є. А.			Арк.	2	Аркушів	8
Н. Контр.		Жуков С.О.			Діаграма варіантів використання			
Зав. каф.		Мокін В.Б.						

Діаграма діяльності для користувацької частини



					08–53.МКР.003.02.00.ПЛ				
					Інформаційної системи моніторингу пасажирських автоперевезень	Літ.		Маса	Масштаб
Змн.	Арк.	Лист	Підпис	Дата					1 : 1
Розробив		Машницький П.П.							
Перевірив		Крижановський Є.М							
Рецензент		Паламарчук Є. А.				Арк.	3	Аркушів	8
					Діаграма діяльності для користувацької частини	ВНТУ, 2ІСТ-19м			
Н. Контр.		Жуков С.О.							
Зав. каф.		Мокін В.Б.							

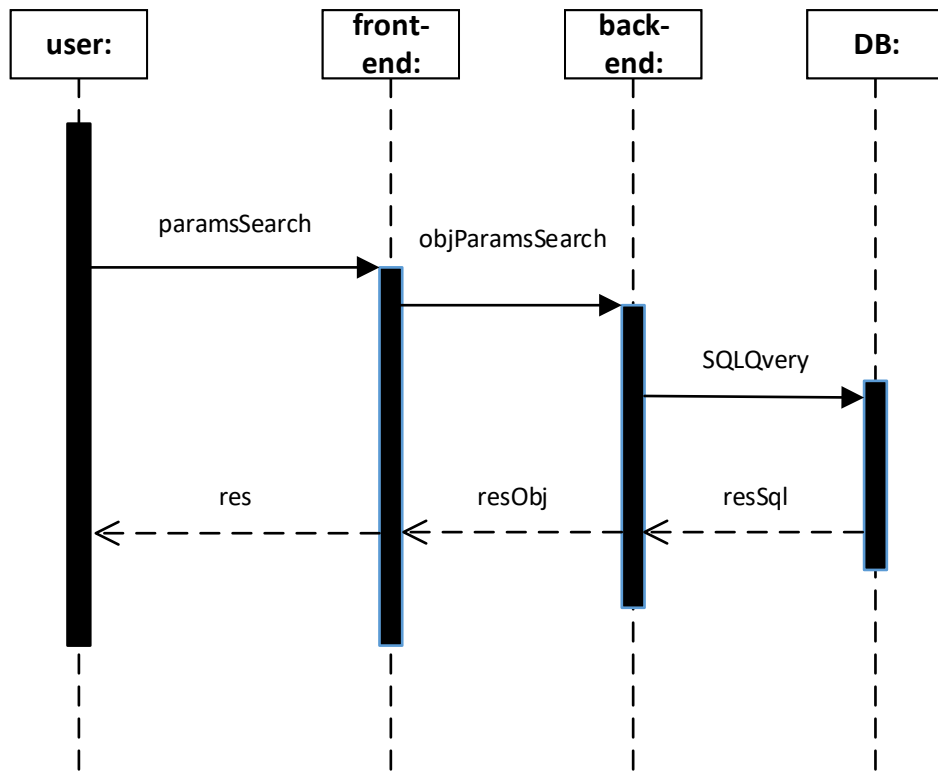
Діаграма діяльності для адміністративної частини



08–53.МКР.003.02.00.ПЛ

Змн.	Арк.	Лист	Підпис	Дата				
					08–53.МКР.003.02.00.ПЛ			
					Інформаційної системи моніторингу пасажирських автоперевезень	Літ.	Маса	Масштаб
								1 : 1
						Арк.	4	Аркушів 8
Н. Контр.		Жуков С.О.			Діаграма діяльності для адміністративної частини	ВНТУ, 2ІСТ-19м		
Зав. каф.		Мокін В.Б.						

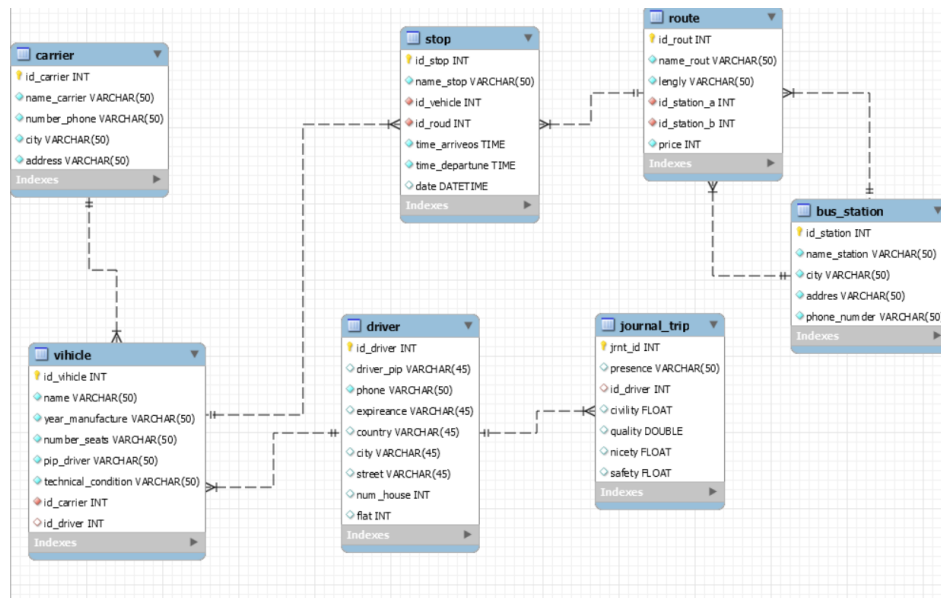
Діаграма послідовності



08–53.МКР.003.02.00.ПЛ

Змн.	Арк.	Лист	Підпис	Дата	Інформаційної системи моніторингу пасажирських автоперевезень	Літ.	Маса	Масштаб
						Арк. 5	Аркушів 8	
Н. Контр.		Жуков С.О.			Діаграма послідовності	ВНТУ, 2ІСТ-19м		
Зав. каф.		Мокін В.Б.						

Реляційна структура бази даних



08–53.МКР.003.02.00.ПЛ

Змн.	Арк.	Лист	Підпис	Дата	Інформаційної системи моніторингу пасажирських автоперевезень	Літ.	Маса	Масштаб
								1 : 1
Розробив		Машницький П.П.						
Перевірів		Крижановський С.М.						
Рецензент		Паламарчук С. А.			Арк.	6	Аркушів	8
Н. Контр.		Жуков С.О.			Реляційна структура бази даних	ВНТУ, 2ІСТ-19м		
Зав. каф.		Мокін В.Б.						


Загальний вигляд інтерфейсу веб-ресурсу

Система моніторингу пасажирських автоперевезень

Головна
Розклад
Пошу
Перевізники
Адміністрування


Пасажирські перевезення

Пасажирські перевезення - перевезення пасажирів автомобілями або автобусами



GPS-моніторинг

GPS-моніторинг - це спеціалізована система, яка призначена для спостереження і контролю стану будь-якої рухомої об'єкта. Систему GPS-моніторингу можна встановити практично на будь-якому транспорті - на легкому/важкому автомобілі/автобусі, тракторі та мототранспорті, мототранспорті, спеціалізованому техніці, яка призначена для обслуговування дорожньої інфраструктури або спеціалізованої роботи.



08–53.МКР.003.02.00.ПЛ

Змн.	Арк.	Лист	Підпис	Дата	Інформаційної системи моніторингу пасажирських автоперевезень	Літ.	Маса	Масштаб
Розробив		Машницький П.П.			Загальний вигляд інтерфейсу веб-ресурсу	Арк.	7	Аркушів 8
Перевірив		Крижановський С.М.						
Рецензент		Паламарчук С. А.						
Н. Контр.		Жуков С.О.						
Зав. каф.		Мокін В.Б.						

Вигляд спливаючого вікна

Закрити

ПІП: Іванов І.І. **Оцінити**

Стаж: 12

Тел.: +3806821651

Якість:

Пунктуальність:

Адреса: Україна **Теплик Київська 23**

Рейтинг: 89.75

Ввічливість:

Безпечність:

ПІП: Поліщук С.В. **Оцінити**

Стаж: 5

Тел.: +3806951235

Якість:

Пунктуальність:

Адреса: Україна **Вінниця Східна 86**

Рейтинг: 85.81

Ввічливість:

Безпечність:

08–53.МКР.003.02.00.ПЛ

Змн.	Арк.	Лист	Підпис	Дата	Інформаційної системи моніторингу пасажирських автоперевезень	Літ.	Маса	Масштаб
								1 : 1
Розробив		Машницький П.П.						
Перевірив		Крижановський С.М.						
Рецензент		Паламарчук С. А.			Арк.	8	Аркушів	8
Н. Контр.		Жуков С.О.			Вигляд спливаючого вікна			
Зав. каф.		Мокін В.Б.			ВНТУ, 2ІСТ-19м			