

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

До захисту допущено:

Завідувач кафедри

_____ Сергій Стіренко

« ____ » _____ 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Комп'ютерні системи та мережі»

спеціальності 123 «Комп'ютерна інженерія»

на тему: Система планування маршрутів подорожі громадським транспортом (веб-застосунок)

Виконав: студент IV курсу, групи ІО-91

Тарасенко Андрій Володимирович

Керівник

асистент кафедри ОТ, Трочун Є. В.

Консультант із

ст. викладач Виноградов Ю. М.

нормоконтролю

Рецензент

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів
без відповідних посилань.

Студент _____

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Рівень вищої освіти — перший (бакалавр)

Освітньо-професійна програма «Комп'ютерні системи та мережі»

Спеціальність 123 «Комп'ютерна інженерія»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Сергій Стіренко

«_____» _____ 2023 р.

ЗАВДАННЯ

на бакалаврський дипломний проєкт студенту

Тарасенко Андрію Володимировичу

1. Тема проєкту: «Система планування маршрутів подорожі громадським транспортом (веб-застосунок)»,

керівник проєкту асистент кафедри ОТ, Трочун Є. В.,

затверджені наказом по університету від «» травня 2023 р. № 000.

2. Термін здачі студентом закінченого проєкту: «» червня 2023 р.

3. Вихідні дані до проєкту: технічна документація, теоретичні дані.

4. Зміст розрахунково-пояснювальної записки:

Розділ 1. Аналіз предметної області

Розділ 2. Вибір засобів для реалізації

Розділ 3. Розробка застосунку

Розділ 4. Аналіз результатів та інструкція користувача.

5. Перелік графічного матеріалу: структурна схема додатку, функціональна схема (діаграма класів), алгоритм дій програмного забезпечення для пошуку маршрутів, алгоритм дій програмного забезпечення для керування транспортною мережею.

6. Консультанти розділів роботи:

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Нормоконтроль	ст. викладач Виноградов Ю. М.		

7. Дата видачі завдання: «08» червня 2023 р.

Календарний план

№ з/п	Найменування етапів дипломного проєкту	Термін виконання етапів проєкту	Примітки
1	Затвердження теми проєкту	27.02.2023	
2	Вивчення та аналіз завдання	13.03.2015	
3	Розробка архітектури та загальної структури системи	20.03.2015	
4	Розробка структур окремих підсистем	03.04.2023	
5	Програмна реалізація системи	01.05.2023	
6	Оформлення пояснювальної записки	31.05.2016	
7	Захист програмного продукту	02.06.2023	
8	Передзахист	08.06.2023	
9	Захист	19.06.2023	

Студент-дипломник _____

Тарасенко А. В.

Керівник проєкту _____

Трочун Є. В.

АНОТАЦІЯ

Цей бакалаврський проект спрямований на розробку веб-додатку з використанням Python та Django, який буде знаходити найшвидші та найкоротші маршрути для громадського транспорту на основі даних розкладу його руху. Основна увага зосереджена на розробці самого веб-додатку.

Розроблений проект є актуальним у сучасному суспільстві, оскільки забезпечує вирішення проблеми пошуку ефективних маршрутів громадського транспорту, що може заощадити час та гроші пасажирів. Проект використовує Python та Django, які є широко використовуваними технологіями у розробці веб-додатків.

Основними завданнями проекту є розробка зручного веб-додатку, який використовує дані розкладу для пошуку найкоротших та найшвидших маршрутів громадського транспорту. Проект також буде спрямований на впровадження адаптивного дизайну за допомогою використання Bootstrap.

Таким чином, цей бакалаврський проект буде зосереджений на розробці веб-додатку з використанням Python, Django та Bootstrap, який надаватиме користувачам ефективні маршрути громадського транспорту на основі даних розкладу.

Ключові слова: прокладання маршрутів, громадський транспорт, Python, Django.

ABSTRACT

This bachelor's project is focused on developing a web application using Python and Django that will find the fastest and shortest routes for public transportation based on its schedule data. The main focus is on the development of the web application itself.

The project is relevant in modern society, as it provides a solution to the problem of finding efficient public transportation routes, which can save passengers time and money. The project uses Python and Django, which are widely used technologies in web application development.

The main objectives of the project are to develop a user-friendly web application that uses schedule data to find the shortest and fastest public transportation routes. The project will also focus on implementing responsive design through the use of Bootstrap.

Thus, this bachelor's project will focus on developing a web application using Python, Django, and Bootstrap that will provide users with efficient public transportation routes based on schedule data.

**ПОЯСНЮВАЛЬНА ЗАПИСКА
ДО ДИПЛОМНОГО ПРОЕКТУ**

на тему: «Система планування маршрутів подорожі громадським
транспортом (веб-застосунок)»

ЗМІСТ

Перелік умовних позначень, скорочень і термінів	3
Вступ	4
1 Аналіз предметної області.....	5
1.1 Загальний огляд програм для прокладання маршрутів транспорту	5
1.1.1 Google Maps	6
1.1.2 Moovit	7
1.1.3 Citymapper	8
1.1.4 Transit.....	9
1.1.5 Порівняння існуючих рішень.....	11
1.2 Аналіз потреб користувачів.....	12
1.2.1 Потреби невеликих транспортних компаній.....	12
1.2.2 Потреби людей, що їздять на роботу	13
1.2.3 Потреби туристів.....	14
1.3 Опис продукту, що розробляється	15
1.3.1 Можливості застосунку	15
Висновки до розділу 1	18
2 Вибір засобів для реалізації	19
2.1 Вибір мови програмування	19
2.1.1 Python	20
2.1.2 JavaScript.....	23
2.1.3 Java	25
2.1.4 Go	28
2.1.5 Остаточний вибір мови програмування	30
2.2 Вибір фреймворків	31
2.2.1 Django	32
2.2.2 Flask	34
2.2.3 Остаточний вибір фреймворку	36
Висновки до розділу 2	38

	2
3 Розробка застосунку	39
3.1 Загальний опис роботи системи, її можливості	40
3.1.1 Основний функціонал системи	41
3.1.2 Отримання, обробка та відображення даних про транспорт	42
3.1.3 Пошук маршруту	45
3.1.4 Відображення знайденого маршруту	47
3.2 Дизайн користувацького інтерфейсу	49
3.2.1 Проектування адаптивного інтерфейсу	51
Висновки до розділу 3	54
4 Аналіз результатів та інструкція користувача	56
4.1 Використання розробленого застосунку	57
4.1.1 Огляд користувацького інтерфейсу	58
4.1.2 Огляд функції пошуку маршрутів	59
4.1.3 Контроль доступу до управління даними про транспорт	60
4.1.4 Огляд можливостей управління транспортними даними	60
Висновки до розділу 4	63
Висновки	64
Перелік посилань	66

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

ANFIS — adaptive neuro-fuzzy inference system.

HMM — hidden Markov model.

MANFIS — modified adaptive neuro-fuzzy inference system.

ЕОМ — електронно-обчислювальна машина.

РЕСЛ — розпізнавання емоційного стану людини.

ВСТУП

У сучасному швидкоплинному світі час є цінним ресурсом. Поїздки з одного місця в інше можуть забирати багато часу і виснажувати, особливо якщо ви не знайомі з системою громадського транспорту. Громадський транспорт відіграє важливу роль у сучасному міському суспільстві, забезпечуючи економічно ефективний та екологічний спосіб пересування. Однак пошук найефективнішого маршруту може бути складним завданням, особливо у великих містах зі складною транспортною системою. У цьому контексті розробка веб-додатку, який може знаходити найкоротші та найшвидші маршрути громадського транспорту на основі даних розкладу, може значно покращити життя пасажирів.

Цей бакалаврський проект спрямований на розробку веб-додатку, який буде знаходити найкоротші та найшвидші маршрути громадського транспорту на основі даних розкладу. У проекті будуть використані Python і Django, дві популярні технології розробки веб-додатків, з акцентом на зручність користування та адаптивний дизайн.

Загалом, метою проекту є розробка рішення проблеми пошуку ефективних маршрутів громадського транспорту. Таким чином, проект може допомогти користувачам заощадити час і гроші, а також популяризувати використання громадського транспорту. В даній пояснювальній записці буде представлено методологію проекту, включаючи процес розробки програмного забезпечення та використані технології. Також будуть обговорені проблеми, що виникли під час реалізації проекту, та шляхи їх вирішення.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний огляд програм для прокладання маршрутів транспорту

Програми для пошуку найкоротших маршрутів широко використовуються в різних сферах. Ці програми використовуються для пошуку найкоротшого шляху між двома точками на основі різних критеріїв, таких як відстань, час, вартість і так далі. Деякі з найпопулярніших алгоритмів пошуку найкоротшого маршруту, що використовуються в цих програмах, включають алгоритм Дейкстри, алгоритм A*, алгоритм Беллмана-Форда та алгоритм пошуку в ширину.

Однією з популярних програм для пошуку найкоротших маршрутів громадського транспорту є Google Maps. Google Maps дозволяє користувачам вводити своє початкове місцезнаходження та пункт призначення, а потім пропонує кілька можливих маршрутів на вибір. Програма враховує поточне місцезнаходження користувача, а також дані про трафік і громадський транспорт у реальному часі, щоб запропонувати найефективніший маршрут.

Іншою програмою, яка спеціалізується на пошуку найкоротших маршрутів на громадському транспорті, є Moovit. Moovit пропонує інформацію про громадський транспорт в режимі реального часу, включаючи розклади руху автобусів та потягів, і надає користувачам найшвидші та найефективніші маршрути. Програма також дозволяє користувачам відслідковувати свій автобус чи потяг в режимі реального часу, надаючи точний час прибуття та відправлення.

Також досить популярною є Citymapper - це додаток для громадського транспорту, який пропонує інформацію про транспорт у реальному часі для більш ніж 80 міст по всьому світу. Він пропонує планування маршрутів, реальний час прибуття та відправлення, а також покрокові інструкції для систем громадського транспорту, таких як автобуси, поїзди та метро. Citymapper також надає інформацію про ціни на проїзд і пропонує функцію "дощової" маршрутизації, яка пропонує маршрути, що мінімізують час, проведений на вулиці в погану погоду.

Transit — це додаток для громадського транспорту, який пропонує інформацію

про громадський транспорт у реальному часі для понад 200 міст по всьому світу. Він пропонує планування маршрутів, реальний час прибуття та відправлення, а також покрокові вказівки для систем громадського транспорту, таких як автобуси, поїзди та метро. Transit також надає інформацію про системи спільного користування велосипедами та пропонує інформацію в режимі реального часу про наявність велосипедів та місць для паркування.

Загалом, згадані вище програми пропонують користувачам різні варіанти пошуку найкоротших маршрутів на громадському транспорті. Кожна програма має свої унікальні особливості та переваги, тому вибір, якою з них користуватися, залежатиме від потреб та вподобань користувача.

Нижче кожному з програм буде розглянуто більш детально.

1.1.1 Google Maps

Google Maps – це популярний сервіс, який пропонує широкий спектр функцій, включаючи оновлення дорожнього руху в режимі реального часу, супутникові знімки, а також вказівки щодо проїзду, пішохідних маршрутів і маршрутів громадського транспорту. Зручний інтерфейс та безперешкодна інтеграція з іншими сервісами Google роблять його популярним серед користувачів по всьому світу.

Однією з ключових переваг Google Maps є великий набір даних, який використовується для надання користувачам точної та актуальної інформації про місцезнаходження та маршрути. Сервіс використовує поєднання супутникових знімків, геопросторових даних і створеного користувачем контенту, щоб надати комплексне уявлення про місцезнаходження та навігаційні маршрути.

З точки зору користувацького досвіду, Google Maps пропонують інтуїтивно зрозумілий інтерфейс, який є простим у використанні та навігації. Користувачі можуть ввести бажане місце розташування і отримати докладні вказівки, що включають покрокові інструкції, приблизний час у дорозі та альтернативні

маршрути. Сервіс також дозволяє користувачам налаштовувати вигляд карти та шукати певні об'єкти, такі як ресторани, магазини та визначні пам'ятки.

Google Maps доступний в Україні і надає достовірну інформацію про маршрути та напрямки у великих містах, таких як Київ, Харків та Львів. Однак точність і доступність інформації може відрізнятися в менших містах і сільській місцевості, оскільки Google Maps значною мірою покладається на контент, створений користувачами, який доповнює набір даних.

Загалом, Google Maps – це потужний і зручний сервіс, який пропонує широкий спектр функцій для навігації як у великих, так і в малих містах. Хоча він може не забезпечувати повне покриття всіх областей, він залишається цінним ресурсом для користувачів, які шукають надійні транспортні сполучення та інформацію про місцезнаходження.

1.1.2 Moovit

Moovit – популярний мобільний додаток, який в режимі реального часу інформує про автобусні, залізничні, метрополітен та інші громадські транспортні засоби в більш ніж 3 000 міст по всьому світу, включаючи Україну. Додаток покликаний полегшити користувачам навігацію громадським транспортом, надаючи їм точну та актуальну інформацію про розклади, маршрути та затримки.

Moovit пропонує широкий спектр можливостей, включаючи дані про транзит в режимі реального часу, планування поїздок та навігацію в реальному часі. Користувачі можуть встановлювати свої домашні та робочі місця і отримувати сповіщення про будь-які затримки або зміни в їхніх варіантах пересування.

Користувацький інтерфейс Moovit інтуїтивно зрозумілий і простий у використанні. Користувачі можуть шукати варіанти пересування, вводячи своє поточне місцезнаходження та пункт призначення, і додаток надає список варіантів пересування та приблизний час у дорозі. Користувачі також можуть переглядати

карти та розклади транспорту і зберігати свої улюблені маршрути для подальшого використання.

Moovit отримує дані від різних транспортних агентств та операторів, а також краудсорсингові дані від користувачів. Дані додатку охоплюють широкий спектр варіантів пересування, включаючи автобуси, поїзди, метро та легку залізницю.

Moovit добре працює в Україні, особливо у великих містах, де є більш розвинена інфраструктура громадського транспорту. Однак додаток може бути менш корисним у менших містах та сільській місцевості, де можливості громадського транспорту обмежені. Користувачі також повинні знати, що точність даних додатку в режимі реального часу може змінюватися в залежності від якості даних, наданих місцевими транспортними агентствами.

Загалом, Moovit є корисним інструментом для всіх, хто покладається на громадський транспорт для пересування, надаючи інформацію в режимі реального часу і полегшуючи планування поїздок. Однак, користувачі в невеликих містах або сільській місцевості можуть знайти додаток менш корисним через обмежену доступність громадського транспорту.

1.1.3 Citymapper

Citymapper – популярний транспортний додаток, який надає користувачам інформацію в режимі реального часу та варіанти планування маршрутів для різних видів транспорту. Він доступний у понад 80 містах світу.

Однією з головних переваг Citymapper є його зручний інтерфейс, який дозволяє користувачам легко вводити пункт призначення та переглядати різні варіанти маршрутів, орієнтовний час у дорозі, а також оновлення в режимі реального часу про затримки та перебої в роботі транспорту. Додаток також пропонує персоналізовані функції, такі як можливість зберігати часто використовувані місця та маршрути, а також кнопку "Go" яка швидко розраховує найшвидший маршрут на основі поточного

місцезнаходження користувача.

Citymapper має доступ до широкого спектру транспортних даних, включаючи дані про транзит у реальному часі від місцевих транспортних служб, дані про поїздки на попутках від таких провайдерів, як Uber і Lyft, та дані про велопрокат від таких провайдерів, як Lime і JUMP. Це дозволяє додатку надавати точну та актуальну інформацію про час у дорозі, розклад руху транспорту та іншу важливу транспортну інформацію.

З точки зору покриття в Україні, Citymapper наразі доступний лише в Києві. Однак широке покриття додатку в інших великих містах світу свідчить про те, що він має потенціал для розширення в інших містах України та надання цінної транспортної інформації користувачам у менших містах.

Загалом, Citymapper – це потужний та зручний транспортний додаток, який пропонує широкий спектр функцій та доступ до різноманітних наборів транспортних даних. Його доступність у Києві та інших великих містах світу робить його корисним інструментом як для мандрівників, так і для тих, хто користується громадським транспортом.

1.1.4 Transit

Transit – це популярний мобільний додаток, який надає користувачам інформацію про громадський транспорт у режимі реального часу. Додаток доступний на платформах iOS та Android і пропонує безліч функцій, які допомагають користувачам орієнтуватися в системах громадського транспорту в різних містах світу.

Однією з головних переваг Transit є його зручний інтерфейс. Додаток розроблений так, щоб бути інтуїтивно зрозумілим і простим у використанні, з чіткими і стислими інструкціями для доступу до різних функцій. Користувачі можуть легко шукати транзитні маршрути та отримувати оновлення в режимі реального часу

про стан обраних ними маршрутів, включаючи затримки та збої в роботі.

Для надання користувачам точної та актуальної інформації Transit покладається на низку різних джерел даних. Окрім офіційних даних транспортних агентств, додаток також інтегрує дані зі сторонніх джерел, таких як краудсорсингова інформація та дані зі служб спільного користування велосипедами.

Що стосується доступності та покриття в Україні, то наразі Transit доступний лише в обмеженій кількості міст. Однак, додаток надає інформацію для великих українських міст - Києва, Львова та Одеси, а також кількох інших менших міст.

Однією з унікальних особливостей Transit є його інтеграція з іншими транспортними опціями, такими як сервіси спільного користування автомобілями та системами прокату велосипедів. Це дозволяє користувачам легко планувати і здійснювати мультимодальні поїздки з використанням різних видів транспорту.

Загалом, Transit є корисним і зручним інструментом для користувачів, які хочуть орієнтуватися в системах громадського транспорту в різних містах. Хоча його покриття в Україні наразі обмежене, додаток пропонує точну та достовірну інформацію для найбільших міст країни.

1.1.5 Порівняння існуючих рішень

Застосунок	Можливості	Користувацький досвід	Набори даних	Доступність в Україні
Google Maps	Планування маршрутів, оновлення маршрутів у реальному часі	Простий у використанні, інтуїтивно зрозумілий інтерфейс	Глобальне охоплення, включає дані від місцевих транспортних компаній	Доступний лише у великих містах України
Moovit	Планування маршрутів, оновлення маршрутів у реальному часі	Чіткий та зрозумілий інтерфейс	Глобальне покриття, включає дані від місцевих транспортних компаній	Доступний лише у великих містах України
Citymapper	Планування маршрутів, оновлення маршрутів у реальному часі	Цікавий, містить інформацію про погоду та розваги	Обмежене покриття за межами великих міст	Не доступний в Україні
Transit	Планування маршрутів, оновлення маршрутів у реальному часі	Простий і зрозумілий	Обмежене покриття за межами великих міст	Доступний лише у великих містах України

1.2 Аналіз потреб користувачів

Для формування вимог до створюваного продукту слід визначити потреби користувачів, а для цього потрібно визначити хто буде користуватись продуктом. Для даного сервісу можна виділити кілька груп потенційних користувачів:

- Невеликі транспортні компанії
- Люди, які щодня їздять на роботу
- Туристи

Нижче буде розглянуто кожну групу користувачів окремо.

1.2.1 Потреби невеликих транспортних компаній

Малі компанії мають різні потреби, коли йдеться про планування маршрутів власним транспортом. Ось деякі з основних потреб:

Донесення власних послуг до користувачів: для транспортних компаній, особливо невеликих, важливим є інформуванні клієнтів про свої послуги. Так як транспортних компаній існує дуже багато, ненав'язливим способом показати клієнтам можливості своєї компанії є сервіс, з яким користувач сам знайде необхідні для нього транспортні з'єднання та потрібну інформацію про них. Це є чудовим способом отримати нових клієнтів, не набридаючи рекламою.

Налаштовування та менеджмент маршрутів: невеликим компаніям потрібен сервіс, який дозволяє слідкувати за даними транспортних маршрутів, вносити зміни, бачити недоліки в створених маршрутах. Також завдяки цьому сервісу, можна побачити маршрути з боку користувача, прокладаючи маршрути між різними зупинками. Також компаніям може знадобитись можливість вносити тимчасові зміни в маршрути, наприклад закривати зупинку через проведення ремонтних робіт.

Зручний інтерфейс: малим компаніям потрібен сервіс, який простий у

використанні і не вимагає великих технічних знань. Тому сервіс має бути інтуїтивно зрозумілим і зручним для користувача.

Ефективне прокладання маршрутів: малим компаніям потрібен сервіс, здатний ефективно прокладати маршрути. Це потрібно для того щоб користувач бачив всі можливості транспортної компанії.

Оновлення в режимі реального часу: малим компаніям потрібен сервіс, який надає оновлення в реальному часі про трафік, погоду та інші фактори, які можуть вплинути на час в дорозі.

Надійна користувацька підтримка: невеликим компаніям потрібна послуга, яка забезпечує надійну підтримку клієнтів у разі виникнення питань або проблем. У них може не вистачати ресурсів для самостійного усунення технічних проблем, тому їм потрібен сервіс, який пропонує оперативну та кваліфіковану підтримку.

1.2.2 Потреби людей, що їздять на роботу

Люди, які щодня їздять на роботу, мають кілька потреб, коли мова йде про сервіс для планування маршруту:

Найшвидші та найефективніші маршрути: людям, які їздять на роботу щодня, потрібно шукати найшвидші та найефективніші маршрути, щоб дістатися до місця призначення. Вони можуть хотіти уникнути заторів на дорогах і знайти найшвидший спосіб дістатися до роботи.

Кілька варіантів транспорту: Людям, які їздять на роботу, може знадобитися послуга, що надає їм кілька варіантів транспорту.

Зручний інтерфейс: Зручний інтерфейс важливий для пасажирів, які хочуть швидко і легко планувати свої щоденні поїздки. Їм може знадобитися простий у використанні сервіс, який надає чітку та стислу інформацію про варіанти маршрутів.

Недорогі маршрути: Користувачі також можуть шукати економічно ефективні рішення для своїх щоденних поїздок на роботу. Це означає, що користувачі нададуть

перевагу робити якомога менше пересадок. Користувачі також можуть обирати транспорт за власним вподобанням.

Оновлення в реальному часі: Пасажирам може знадобитися інформація в режимі реального часу про стан дорожнього руху та затримки на їхньому маршруті. Ця інформація може допомогти їм відповідно спланувати свою подорож і уникнути заторів.

Загалом, сервіс для планування маршрутів для людей, які щодня їздять на роботу, повинен бути ефективним, мати оновлення в режимі реального часу та кілька видів транспорту, а також бути зручним у користуванні.

1.2.3 Потреби туристів

Туристи мають різні потреби, в тому числі і в послугі планування маршруту:

Управління часом: туристи часто мають щільний графік і хочуть максимально ефективно використати свій час. Хороший сервіс планування маршруту повинен враховувати час, доступний туристу, і пропонувати найефективніші маршрути.

Зручність: туристи хочуть досліджувати місто, не гублячись. Послуга планування маршрутів повинна надавати прості вказівки та пропозиції щодо найкращих видів транспорту.

Знання місцевості: туристи можуть бути не знайомі з містом та його системою громадського транспорту. Хороший сервіс планування маршрутів повинен надавати детальну інформацію про місцеву транспортну систему, включаючи карти і розклади.

Персоналізація: туристи можуть мати особливі вподобання чи інтереси. Хороший сервіс планування маршруту повинен враховувати ці вподобання і надавати персоналізовані рекомендації щодо маршруту туриста.

Загалом, хороший сервіс планування маршрутів для туристів має бути зручним у користуванні, надійним і надавати актуальну інформацію про визначні пам'ятки міста та систему громадського транспорту.

1.3 Опис продукту, що розробляється

Сервіс, що розроблюється - це комплексний і складний веб-застосунок для планування та оптимізації маршрутів для малого бізнесу та приватних осіб. Цей застосунок покликаний допомогти транспортним компаніям та користувачам громадського транспорту знайти один одного, шляхом полегшення планування щоденних поїздок на роботу, в навчальні заклади, та інших поїздок, забезпечуючи їм високий рівень ефективності та зручності.

Цей застосунок пропонує ряд функцій, які гарантують, що кожен користувач може знайти потрібний маршрут. Він надає детальну інформацію про маршрути та розклади руху громадського транспорту, а також оновлення інформації про затори в режимі реального часу та альтернативні варіанти маршрутів.

1.3.1 Можливості застосунку

В ході дослідження існуючих рішень було виділено декі ряд ключових функцій, які дозволяють користувачам ефективно та раціонально керувати своїми маршрутами. Список основних функцій, які повинні бути присутні в застосунку:

- Додавання маршрутів: Функція додавання маршрутів дозволяє адміністраторам додавати нові маршрути до системи. Вони можуть вводити такі дані, як пункт відправлення, пункт призначення, проміжні зупинки та будь-яку іншу додаткову інформацію, що стосується маршруту. Вони також можуть надати інформацію про розклад, включаючи час відправлення та прибуття, частоту та будь-які зміни на основі робочих та вихідних днів. Інтерфейс адміністратора надає зручну форму або інтерфейс для введення всіх необхідних даних, забезпечуючи точність і повноту.

- Редагування маршрутів: Функція редагування маршруту дозволяє адміністраторам змінювати існуючі маршрути. Вони можуть оновлювати таку інформацію, як розклад, зупинки або будь-які інші важливі деталі. Адміністратори можуть вносити зміни в режимі реального часу, гарантуючи, що додаток відображає точну та актуальну інформацію про маршрут. Процес редагування може включати коригування карти маршруту, додавання або видалення зупинок, зміну розкладу або оновлення будь-якої іншої інформації, пов'язаної з маршрутом.

- Видалення маршрутів: Функція видалення маршрутів дозволяє адміністраторам видаляти маршрути, які більше не використовуються або стали застарілими. Коли маршрут видалено, він більше не буде видимим для користувачів, що запобігає доступу до застарілої або невірної інформації. Видалення маршрутів гарантує, що база даних додатку залишається впорядкованою та актуальною, надаючи користувачам надійні та актуальні варіанти маршрутів.

- Приховування маршрутів: Функція приховування маршрутів дозволяє адміністраторам не відображати маршрути, які більше не використовуються або стали застарілими, при цьому не видаляючи їх.

- Пошук маршрутів: Функція пошуку маршрутів дозволяє користувачам шукати маршрути на основі бажаних початкових і кінцевих точок. Вони можуть вводити адреси, орієнтири або цікаві місця, щоб знайти найзручніші маршрути. Додаток використовує алгоритми для розрахунку оптимальних варіантів маршруту, враховуючи такі фактори, як відстань, приблизний час у дорозі та доступні види транспорту. Користувачі можуть обирати між різними видами транспорту, такими як автобуси, трамваї, поїзди та іншими видами транспорту.

- Перегляд деталей маршруту: Функція деталізації маршруту дозволяє користувачам переглядати інформацію про прокладений маршрут. Детальна інформація про кожну зупинку, включаючи час прибуття і відправлення, може бути

надана, що дозволяє користувачам ефективно планувати свою подорож. Користувачі також можуть отримати доступ до додаткових даних, таких як приблизний час у дорозі та відстань.

Також серед існуючих додатків більшість існують перш за все як мобільний додаток, ще не підходить для всіх пристроїв. В свою чергу веб-додаток може відображатися як на телефоні так і на комп'ютері чи інших присторх, які мають браузер.

ВИСНОВКИ ДО РОЗДІЛУ 1

За останні роки сфера планування маршрутів і навігації зазнала значного прогресу, зумовленого зростаючим попитом на ефективні рішення з пошуку маршрутів. Існуючі рішення, такі як Google Maps, Moovit, Citymapper та Transit, стали популярним вибором, пропонуючи широкий спектр можливостей та функцій для задоволення різноманітних потреб користувачів. Ці платформи надають вичерпну інформацію про маршрути, оновлення в режимі реального часу та зручні інтерфейси, що покращує загальний досвід навігації.

Однак існують певні обмеження та проблеми, пов'язані з існуючими рішеннями, особливо в контексті невеликих міст або регіонів з менш розвиненою транспортною мережею. У таких районах доступність і точність маршрутної інформації може бути обмеженою, що заважає користувачам ефективно планувати свої подорожі.

Надаючи адміністраторам можливість керувати та оновлювати маршрути, додаток гарантує доступність актуальної та точної інформації. Тим часом користувачі можуть використовувати інтуїтивно зрозумілий інтерфейс для пошуку маршрутів, перегляду детальної інформації та ефективного планування своїх поїздок. Додаток має на меті спростити процес навігації, покращити доступ до транспортних можливостей та покращити загальний досвід подорожей для користувачів.

Завдяки орієнтації на потреби користувачів, точній інформації про маршрути та зручному інтерфейсу, додаток прагне стати цінним інструментом для приватних осіб, малих компаній, туристів та мешканців невеликих міст. Вирішуючи конкретні проблеми, з якими стикаються в цих сферах, і забезпечуючи надійну функціональність планування маршрутів, додаток має потенціал для того, щоб мати значний вплив і зарекомендувати себе як успішне рішення на ринку.

Насамкінець, завдяки своїм спеціальним функціям, таким як можливість додавання маршрутів власноруч, додаток надає можливість представникам невеликих транспортних компаній знаходити клієнтів для своїх послуг.

2 ВИБІР ЗАСОБІВ ДЛЯ РЕАЛІЗАЦІЇ

Вибір мов програмування та фреймворків відіграє ключову роль в успіху будь-якого проекту з розробки програмного забезпечення. Кожна технологія має свій власний набір сильних і слабких сторін, а також придатність для конкретних випадків використання. Вибір відповідних технологій може суттєво вплинути на ефективність, масштабованість та зручність обслуговування програми. У цьому розділі розглядається важливий процес вибору правильних мов програмування та фреймворків для вашого проекту.

Вибір правильної мови програмування та фреймворку вимагає ретельного аналізу вимог проекту, цільової аудиторії та цілей розробки. Мова програмування слугує основою для розробки додатків, визначаючи синтаксис, структуру та можливості, доступні для розробників. Фреймворки, з іншого боку, надають заздалегідь визначені структури, бібліотеки та інструменти, які спрощують розробку та підвищують продуктивність.

Отже, вибір мови програмування та фреймворків є критично важливим аспектом будь-якого проекту з розробки програмного забезпечення. Ця глава має на меті надати цінну інформацію про міркування та фактори, які впливають на процес прийняття рішень. Наприкінці цього розділу ви матимете міцний фундамент, який допоможе вам вибрати правильні мови програмування та фреймворки для вашого проекту, що створить передумови для успішної розробки та впровадження.

2.1 Вибір мови програмування

Для розробки веб-застосунку, який би вирішував поставлену задачу існує цілий ряд мов програмування, які пропонують відповідні можливості. Мова програмування Python, відома своєю простотою і зрозумілістю, надає надійні бібліотеки і

фреймворки, такі як Flask і Django, які полегшують веб-розробку і безперешкодну інтеграцію алгоритмів пошуку маршрутів. Мова програмування JavaScript, будучи мовою де-факто для веб-рішень, пропонує фреймворки, такі як Node.js, і бібліотеки, такі як Leaflet.js і Mapbox.js, що робить її широко прийнятим вибором для створення інтерактивних застосунків. Мова програмування Java, з її акцентом на масштабованість і розгалужену екосистему, надає такі фреймворки, як Spring і JavaServer Faces (JSF), які дозволяють розробляти ефективні веб-додатки з інтегрованою функцією пошуку маршрутів. Крім того, такі мови програмування, як Ruby, Go і TypeScript, також пропонують достойні варіанти, кожен з яких має свої сильні сторони і фреймворки, що відповідають вимогам веб-розробки та пошуку маршрутів. Вибір мови програмування повинен визначатися такими факторами, як специфічні вимоги проекту, кваліфікація команди розробників та наявність відповідних бібліотек і фреймворків.

2.1.1 Python

Python - це універсальна і широко поширена мова програмування, відома своєю простотою, читабельністю та розгалуженою екосистемою. Це інтерпретована мова високого рівня, яка робить акцент на читабельності та виразності коду, що робить її чудовим вибором для розробки веб-додатків з можливостями пошуку маршрутів.

Мова програмування Python пропонує багату колекцію бібліотек і фреймворків, які спрощують процеси веб-розробки. Flask і Django - два популярні фреймворки, які забезпечують надійну основу для створення веб-додатків на Python. Flask - це легкий і гнучкий фреймворк, який фокусується на простоті і зручності використання, що робить його ідеальним для малих і середніх проектів. Django, з іншого боку, є надійним і всеосяжним фреймворком, який слідує архітектурному шаблону Model-View-Controller (MVC), пропонуючи такі функції, як маршрутизація URL-адрес, ORM (об'єктно-реляційне відображення) баз даних і автентифікація

користувачів "з коробки".

Окрім фреймворків, мова програмування Python може похвалитися розгалуженою екосистемою бібліотек, які є безцінними для додатків для пошуку маршрутів. Наприклад, бібліотека NetworkX надає структури графових даних та алгоритми, які необхідні для реалізації алгоритмів маршрутизації. Бібліотека GeoPandas дозволяє маніпулювати просторовими даними та аналізувати їх, забезпечуючи інтеграцію з джерелами географічних даних для точного планування маршрутів. Крім того, такі популярні бібліотеки, як NumPy та SciPy, надають ефективні можливості чисельних обчислень, які можна використовувати в алгоритмах оптимізації маршрутів.

Універсальність мови програмування Python виходить за рамки веб-розробки, оскільки її також можна використовувати для написання сценаріїв, аналізу даних, машинного навчання тощо. Простота мови та широка підтримка спільноти роблять її привабливим вибором для розробників усіх рівнів кваліфікації. Крім того, мова програмування Python має широку екосистему сторонніх пакетів, доступних через Python Package Index (PyPI), що забезпечує доступ до широкого спектру інструментів і ресурсів для вдосконалення додатків для пошуку маршрутів.

Отже, Python - це потужна та універсальна мова програмування, яка пропонує широкий спектр інструментів, фреймворків та бібліотек для розробки веб-додатків з можливостями пошуку маршрутів. Її простота, зрозумілість та розгалужена екосистема роблять її популярною серед розробників, дозволяючи ефективно та результативно розробляти рішення для планування та оптимізації маршрутів.

Переваги Python:

- **Читабельність:** мова програмування Python робить акцент на читабельності коду завдяки чистому і простому синтаксису, що робить його легким для розуміння і підтримки.
- **Універсальність:** Python - це універсальна мова, яку можна використовувати для веб-розробки, аналізу даних, наукових обчислень, штучного інтелекту тощо.
- **Велика спільнота та екосистема:** мова програмування Python має велику спільноту розробників, які роблять свій внесок у розгалужену екосистему бібліотек, фреймворків та інструментів, надаючи рішення для різних сфер та додатків.

- Крос-платформна сумісність: код на мові програмування Python може працювати на різних операційних системах, включаючи Windows, macOS і різні Unix-подібні системи, що забезпечує портативність і гнучкість.

- Швидкий розвиток: лаконічний синтаксис мови програмування Python та широка підтримка бібліотек сприяють швидкій розробці, дозволяючи розробникам створювати додатки швидко та ефективно.

- Можливості інтеграції: Python легко інтегрується з іншими мовами, такими як C/C++, Java та .NET, дозволяючи розробникам використовувати існуючий код та бібліотеки.

Недоліки Python:

- Обмеження продуктивності: порівняно з мовами нижчого рівня, такими як C або C++, Python може мати обмеження продуктивності через свою інтерпретовану природу, що може вплинути на швидкість виконання обчислювально інтенсивних завдань.

- Споживання пам'яті: споживання пам'яті мова програмування Python може бути вищим у порівнянні з деякими іншими мовами, що може вплинути на масштабованість та ресурсоемність додатків.

- Не підходить для всіх випадків використання: хоча мова програмування Python є універсальною мовою, вона може бути не найкращим вибором для певних випадків використання, які вимагають низькорівневого контролю або суворої оптимізації продуктивності.

- Сумісність версій: мова програмування Python має декілька версій, що може призвести до проблем сумісності при інтеграції зі сторонніми бібліотеками або роботі з кодовими базами, розробленими у різних версіях Python.

2.1.2 JavaScript

JavaScript - це універсальна і широко використовувана мова програмування, яка в першу чергу використовується для розробки інтерактивних і динамічних веб-додатків. Вона підтримується всіма основними веб-браузерами, що робить її фундаментальною технологією для інтерфейсної веб-розробки. JavaScript - це високорівнева, інтерпретована мова з простим і гнучким синтаксисом, яка дозволяє розробникам створювати багатий і цікавий користувацький досвід в Інтернеті.

Однією з ключових переваг мови програмування JavaScript є її здатність працювати безпосередньо в браузері, що дозволяє створювати сценарії на стороні клієнта. Це означає, що код JavaScript може виконуватися на пристрої користувача, зменшуючи потребу в обробці на стороні сервера і підвищуючи швидкість відгуку веб-додатків. Мова програмування JavaScript може маніпулювати елементами HTML, обробляти користувацькі події, виконувати перевірку вводу та динамічно оновлювати вміст і зовнішній вигляд веб-сторінок.

Окрім розробки інтерфейсів, мова програмування JavaScript також використовується для розробки бекенд-версій з появою серверних фреймворків JavaScript, таких як Node.js. Node.js дозволяє розробникам створювати масштабовані та ефективні веб-сервери та додатки за допомогою мови програмування JavaScript. За допомогою Node.js JavaScript можна використовувати для обробки серверної логіки, виконання операцій з базами даних та створення RESTful API.

Мова програмування JavaScript має велику екосистему бібліотек та фреймворків, які розширюють його можливості та спрощують завдання веб-розробки. Такі популярні бібліотеки, як React, Angular та Vue.js, надають потужні інструменти для створення складних користувацьких інтерфейсів та управління станом у великомасштабних додатках. Ці фреймворки підвищують продуктивність завдяки багаторазовому використанню компонентів, ефективному зв'язуванню даних та широкій підтримці спільноти.

Більше того, мова програмування JavaScript вийшов за межі веб-розробки і тепер

використовується для розробки мобільних додатків (React Native, Ionic), десктопних додатків (Electron) і навіть машинного навчання (TensorFlow.js). Така універсальність дозволяє розробникам використовувати свої навички JavaScript на різних платформах і в різних сферах.

Незважаючи на свої сильні сторони, мова програмування JavaScript має деякі обмеження. Він може бути схильний до проблем сумісності з браузерами, оскільки різні браузери можуть дещо по-різному інтерпретувати JavaScript-код. Крім того, гнучкість мови програмування JavaScript може призвести до потенційних пасток, якщо використовувати його необережно, наприклад, проблеми зі змінними в області видимості та примусове використання типів. Однак, правильне кодування та використання сучасних функцій та інструментів мови програмування JavaScript може допомогти пом'якшити ці проблеми.

Загалом, JavaScript - це потужна і широко розповсюджена мова програмування, яка дозволяє розробникам створювати динамічні та інтерактивні веб-додатки. Її універсальність, велика екосистема та широка підтримка браузерами роблять її популярним вибором для фронтенд- та бекенд-веб-розробки, а також для інших сфер застосування.

Переваги:

- Широка сумісність з браузерами: JavaScript підтримується всіма основними веб-браузерами, що забезпечує широку сумісність веб-додатків.
- Інтерактивна взаємодія з користувачем: JavaScript дозволяє створювати динамічні та інтерактивні користувацькі інтерфейси, покращуючи загальний користувацький досвід.
- Широка екосистема: JavaScript має широку екосистему бібліотек, фреймворків та інструментів, які спрощують завдання розробки та прискорюють реалізацію проектів.
- Багаторазове використання: бібліотеки та фреймворки JavaScript надають багаторазові компоненти та модулі, що полегшує повторне використання коду та пришвидшує розробку.

Недоліки:

- Сумісність з браузерами: різні браузери можуть по-різному інтерпретувати

код JavaScript, що вимагає додаткових зусиль для тестування та налагодження кросбраузерної сумісності.

- Обмежена продуктивність: інтерпретована природа JavaScript може призвести до зниження продуктивності порівняно з мовами нижчого рівня, особливо для завдань з інтенсивними обчисленнями.

- Проблеми безпеки: як сценарії на стороні клієнта, код JavaScript доступний користувачам, що робить його вразливим до таких атак, як впровадження коду та міжсайтовий скриптинг, якщо він не захищений належним чином.

- Відсутність строгої типізації: JavaScript має динамічну типізацію, що може призвести до потенційних помилок і ускладнити підтримку коду.

- Крива навчання: JavaScript має криву навчання, особливо для розробників, які переходять з інших мов, через його унікальні особливості та асинхронну модель програмування.

2.1.3 Java

Java - це широко використовувана мова програмування, відома своєю універсальністю, продуктивністю та надійністю. Вона особливо популярна у сфері корпоративної веб-розробки, де її об'єктно-орієнтована природа та розгалужена екосистема роблять її чудовим вибором для створення масштабованих та безпечних веб-додатків.

Основна перевага мови програмування Java полягає в її незалежності від платформи. Код Java компілюється у байт-код, який може працювати на будь-якій платформі за допомогою віртуальної машини Java (JVM). Це дозволяє розробникам писати код один раз і розгортати його на різних операційних системах, що робить Java ідеальним вибором для веб-додатків, які повинні бути сумісними з різними середовищами.

У контексті веб-розробки мова програмування Java зазвичай використовується

для програмування на стороні сервера. Фреймворки на основі Java, такі як JavaServer Pages (JSP), JavaServer Faces (JSF) та Java Servlets, забезпечують основу для створення динамічних та інтерактивних веб-додатків. Ці фреймворки дозволяють розробникам обробляти HTTP-запити, генерувати динамічний веб-контент, керувати сесансами користувачів і взаємодіяти з базами даних.

Веб-додатки на Java часто використовують архітектуру Model-View-Controller (MVC), яка сприяє розділенню завдань і модульній розробці. Такі фреймворки, як Spring MVC та JavaServer Faces (JSF), забезпечують надійну реалізацію MVC, дозволяючи розробникам структурувати свої додатки та ефективно керувати складною бізнес-логікою.

Розгалужена екосистема Java також включає широкий спектр бібліотек та фреймворків, які покращують веб-розробку. Spring Framework, наприклад, надає комплексну підтримку для створення додатків корпоративного рівня, включаючи такі функції, як встановлення залежностей, доступ до даних та безпека. Інші фреймворки, такі як Hibernate, спрощують роботу з базами даних, а Apache Struts пропонує потужний фреймворк MVC для веб-додатків.

Крім того, мова програмування Java підтримує використання стандартної бібліотеки тегів JavaServer Pages (JSTL), яка надає набір тегів для поширених завдань веб-розробки, таких як ітерації, умови та форматування. Це полегшує розробникам створення динамічного контенту на веб-сторінках.

Крім того, мова програмування Java пропонує потужні функції безпеки та розвинений набір інструментів і бібліотек для захисту веб-додатків. Служба автентифікації та авторизації Java (JAAS) та фреймворки, такі як Spring Security, дозволяють розробникам впроваджувати автентифікацію, контроль доступу та інші заходи безпеки для захисту своїх веб-додатків.

Однак важливо зазначити, що веб-розробка на Java, як правило, передбачає більше налаштувань та конфігурацій порівняно з іншими мовами. Крива навчання може бути крутішою, особливо для початківців, через складність мови та різноманітність доступних фреймворків. Крім того, веб-додатки на Java можуть вимагати більше пам'яті та обчислювальної потужності порівняно з легшими альтернативами.

Загалом, надійність, незалежність від платформи та розгалужена екосистема мови програмування Java роблять її чудовим вибором для веб-розробки, особливо в корпоративному середовищі. Її продуктивність, функції безпеки та масштабованість роблять її добре придатною для створення великомасштабних, критично важливих веб-додатків.

Переваги Java:

- Незалежність від платформи: код на Java працює на будь-якій платформі з віртуальною машиною Java (JVM), що забезпечує чудову портативність.
- Об'єктно-орієнтоване програмування: об'єктно-орієнтована природа Java дозволяє створювати модульний і багаторазовий код, що сприяє легкості супроводу і масштабованості.
- Надійність: суворі перевірки під час компіляції та обробка винятків у Java сприяють створенню стабільних та надійних додатків.
- Велика екосистема: Java має велику екосистему бібліотек, фреймворків та інструментів, які підтримують різні аспекти розробки, прискорюючи процес розробки.
- Безпека: Java надає вбудовані засоби безпеки, такі як пісочниця, перевірка байт-коду та криптографічні бібліотеки для розробки безпечних додатків.

Недоліки Java:

- Крива навчання: Java має круту криву навчання, особливо для початківців, через складність синтаксису та необхідність розуміння об'єктно-орієнтованих принципів.
- Багатослівність: синтаксис Java може бути багатослівним, що вимагає більшої кількості рядків коду порівняно з іншими мовами для досягнення подібної функціональності.
- Споживання пам'яті: програми на Java зазвичай потребують більше пам'яті порівняно з легшими мовами, що може бути важливим фактором для середовищ з обмеженими ресурсами.
- Час запуску: програми на Java, як правило, мають довший час запуску порівняно з інтерпретованими мовами, оскільки вони потребують ініціалізації JVM та завантаження залежностей.

- Обмежений паралелізм: хоча Java забезпечує підтримку паралелізму за допомогою потоків і бібліотек, таких як `java.util.concurrent`, керування паралельним програмуванням може бути складним і схильним до помилок.

2.1.4 Go

Go - це статично типізована, скомпільована мова програмування, розроблена компанією Google. Вона створена для того, щоб бути ефективною, лаконічною та добре масштабованою. Хоча Go є мовою загального призначення, вона набула популярності у спільноті веб-розробників завдяки своїй чудовій підтримці для створення високопродуктивних веб-додатків.

Простота і мінімалізм мови програмування Go роблять її привабливим вибором для веб-розробки. Її синтаксис чистий і легкий для читання, що дозволяє розробникам писати лаконічний і виразний код. Мова програмування Go робить акцент на читабельності коду, що полегшує його розуміння та підтримку великих кодових баз з часом.

Продуктивність мови програмування Go є визначною особливістю. Це скомпільована мова, яка створює власний машинний код, що дозволяє створювати високоефективні та швидкодіючі додатки. Вбудована в Go модель паралелізму, Goroutines, дозволяє легко та ефективно програмувати паралельно, що робить її добре придатною для обробки високого трафіку та паралельних запитів у веб-додатках.

Що стосується фреймворків для веб-розробки, мова програмування Go пропонує кілька популярних варіантів. Одним з найпоширеніших фреймворків є стандартний пакет бібліотек Go `"net/http"`. Він надає надійний набір інструментів для створення веб-серверів, управління маршрутизацією, обслуговування статичних файлів та управління HTTP-запитами і відповідями. Простота і потужність стандартної бібліотеки дозволяють розробникам створювати веб-додатки без

необхідності важких зовнішніх залежностей.

Ще одним помітним фреймворком в екосистемі мови програмування Go є Gin. Gin - це легкий і високопродуктивний веб-фреймворк, який надає додаткові функції та абстракції на додаток до стандартної бібліотеки. Він пропонує швидкий маршрутизатор, підтримку проміжного програмного забезпечення та різноманітні утиліти для спрощення типових завдань веб-розробки.

Сильний акцент мови програмування Go на простоті та ефективності поширюється і на його розгортання. Додатки на Go компілюються в автономні двійкові файли, що спрощує розгортання та усуває необхідність у додаткових залежностях під час виконання. Ця характеристика робить Go добре придатною для створення мікросервісів або розгортання додатків на хмарних платформах.

Хоча мова програмування Go пропонує численні переваги для веб-розробки, важливо враховувати її відносну незрілість у порівнянні з більш усталеними мовами. Екосистема Go все ще розвивається, і доступність сторонніх бібліотек та інструментів може бути більш обмеженою порівняно з такими мовами, як Python або JavaScript. Проте, основна мова та стандартна бібліотека є стабільними та добре підтримуються спільнотою Go.

Отже, Go - це потужна мова програмування для веб-розробки, яка пропонує простоту, ефективність та високу продуктивність. Її чистий синтаксис, вбудована модель паралелізму та всеосяжна стандартна бібліотека роблять її чудовим вибором для створення масштабованих та високопродуктивних веб-додатків. Завдяки зростаючій екосистемі та підтримці спільноти, Go продовжує набирати популярність як найкраща мова для проектів веб-розробки.

Переваги Go (Golang):

- Продуктивність: компільована природа Go та ефективна модель паралелізму сприяють створенню високопродуктивних веб-додатків.
- Спрощеність: чистий синтаксис та мінімалістичний дизайн Go сприяють читабельності коду та легкості його супроводу.
- Конкурентність: підпрограми та вбудовані примітиви паралельності спрощують паралельне програмування, дозволяючи ефективно обробляти декілька запитів.

- Стандартна бібліотека: стандартна бібліотека Go надає основні інструменти для веб-розробки, що полегшує початок роботи без важких зовнішніх залежностей.

- Розгортання: програми на Go компілюються у автономні двійкові файли, що спрощує розгортання і усуває необхідність у додаткових залежностях під час виконання.

Недоліки Go (Golang):

- Зрілість екосистеми: екосистема Go є відносно молодю, і доступність сторонніх бібліотек та інструментів може бути більш обмеженою у порівнянні з більш усталеними мовами.

- Крива навчання: Go має власний набір парадигм і концепцій програмування, які можуть потребувати певного вивчення і пристосування для розробників, які не знайомі з мовою.

- Невелика кількість бібліотек: хоча екосистема Go розвивається, кількість доступних бібліотек і фреймворків може бути меншою у порівнянні з більш популярними мовами, такими як Python або JavaScript.

- Відсутність узагальнень: наразі Go не підтримує узагальнення, що іноді може призвести до дублювання коду або більш багатослівних рішень.

- розмір спільноти: хоча спільнота Go є активною і доброзичливою, вона може бути меншою порівняно зі спільнотами більш поширених мов.

2.1.5 Остаточний вибір мови програмування

Python є ідеальним вибором для розробки даного застосунку, завдяки своїй універсальності, великим бібліотекам та простоті використання. Завдяки своїй багатій екосистемі пакетів, Python надає численні бібліотеки та фреймворки, які можуть значно полегшити пошук маршрутів та веб-розробку. Наявність таких фреймворків як Django або Flask для веб-розробки є надійними та ефективними інструментами. Важливим є те, що мова програмування Python є популярним вибором

для роботи з даними та графами, що значно полегшує роботу над частиною проекту з пошуком маршруту. Крім того, простий і зрозумілий синтаксис Python дозволяє пришвидшити розробку та полегшити підтримку коду. Велика та активна спільнота цієї мови програмування означає, що існує безліч ресурсів, навчальних посібників та підтримки для подолання будь-яких проблем, з якими можна зіткнутися в процесі розробки. Крім того, сумісність Python з іншими технологіями та платформами робить цю мову програмування підходящим вибором для створення масштабованих та крос-платформних додатків. Загалом, поєднання потужності, гнучкості та підтримки спільноти робить Python найкращим вибором для розробки додатку для пошуку маршрутів.

2.2 Вибір фреймворків

Python пропонує широкий вибір фреймворків, які добре підходять для розробки веб-додатків. Django, з його широким набором функцій, надає комплексне рішення для створення складних веб-додатків. Він включає в себе ORM, систему автентифікації та потужний шаблонізатор. Flask, з іншого боку, пропонує легкий і гнучкий підхід, що робить його ідеальним для малих і середніх додатків. Він дозволяє розробникам обирати конкретні компоненти, які їм потрібні, що призводить до створення більш адаптованого рішення. Pyramid дотримується мінімалістичної філософії, зосереджуючись на простоті та масштабованості. Він підтримує різні движки шаблонів і системи баз даних, що робить його адаптивним до різних вимог проекту. Інші фреймворки, такі як Bottle, CherryPy, Tornado і web2py, мають свої унікальні особливості, такі як мінімалізм, високопродуктивні можливості або інтегровані середовища розробки. Завдяки такому різноманітному вибору фреймворків, Python дає можливість розробникам вибрати той, який найкраще відповідає потребам їхнього проекту, забезпечуючи ефективну та результативну розробку веб-додатків.

2.2.1 Django

Django - це високорівневий веб-фреймворк, написаний на мові Python, який відповідає архітектурному шаблону "модель-вигляд-контролер"(MVC). Він покликаний спростити та прискорити процес створення веб-додатків, надаючи надійний набір інструментів та функцій.

Однією з ключових переваг Django є його потужна система об'єктно-реляційного відображення (ORM). Вона дозволяє розробникам взаємодіяти з базою даних за допомогою об'єктів Python, усуваючи необхідність писати складні SQL-запити. ORM підтримує різні бекенди баз даних, включаючи PostgreSQL, MySQL, SQLite і Oracle, забезпечуючи гнучкість і сумісність.

Django також має вбудовану систему автентифікації, що дозволяє легко керувати реєстрацією користувачів, логінами та паролями. Вона забезпечує безпечну і налаштовану систему автентифікації та авторизації користувачів, допомагаючи розробникам впроваджувати контроль доступу і специфічні для користувача функції у своїх додатках.

Ще однією визначною особливістю Django є його механізм шаблонів, який дозволяє відокремлювати HTML-код від коду Python. Це дозволяє розробникам створювати динамічні веб-сторінки, вбудовуючи код Python у шаблони HTML, що спрощує створення динамічного контенту та обробку даних.

Django дотримується принципу "Не повторюй себе"(DRY) і сприяє багаторазовому використанню коду завдяки своєму модульному дизайну. Він пропонує широкий спектр готових компонентів, відомих як "додатки Django які надають такі функціональні можливості, як інтерфейс адміністратора, обробка форм, кешування та інше. Ця розгалужена екосистема багаторазових додатків економить час і зусилля розробників, дозволяючи їм зосередитися на унікальних аспектах своїх додатків.

Крім того, Django підкреслює важливість безпеки. Він включає в себе численні функції безпеки, включаючи захист від поширених веб-уразливостей, таких як

міжсайтовий скриптинг (XSS) і підробка міжсайтових запитів (CSRF). Акцент Django на безпеці допомагає розробникам створювати надійні та безпечні веб-додатки.

Крім того, Django має активну спільноту, яка надає велику документацію, навчальні посібники та пакети сторонніх розробників. Спільнота активно підтримує та оновлює фреймворк, забезпечуючи його стабільність, надійність та сумісність з новими версіями Python та веб-стандартами.

Загалом, Django - це комплексний веб-фреймворк, який спрощує розробку веб-додатків, пропонуючи потужні функції, ефективний ORM, вбудовану автентифікацію, движок шаблонів та безліч компонентів для багаторазового використання. Зосередженість на безпеці, дотримання найкращих практик та процвітаюча спільнота роблять його популярним вибором для створення масштабованих та багатофункціональних веб-додатків.

Переваги Django:

- Комплексний фреймворк з багатим набором функцій та інструментів
- Потужне об'єктно-реляційне відображення (ORM) для взаємодії з базами даних
- Вбудована система автентифікації для управління користувачами
- Движок шаблонів для динамічного створення контенту
- Модульний дизайн, що сприяє повторному використанню коду
- Наголос на безпеку з вбудованим захистом від поширених вразливостей
- Широка документація та потужна підтримка спільноти

Недоліки Django:

- Обмежена гнучкість у виборі компонентів
- Накладні витрати на продуктивність у певних сценаріях через надійність та абстракції фреймворку
- Вимагає дотримання угод Django, які можуть обмежувати певні налаштування або відхилятися від особистих уподобань

2.2.2 Flask

Flask

Flask - це легкий та легко адаптований веб-фреймворк для Python, що працює за принципом мікрофреймворку. Він розроблений, щоб запропонувати простоту, гнучкість і можливості кастомізації, що робить його ідеальним вибором для розробки малих і середніх веб-додатків і API.

Flask дотримується мінімалістичної та неупередженої філософії дизайну, надаючи розробникам широкий контроль над структурою додатку та дизайнерськими рішеннями. Він має потужну систему маршрутизації, що дозволяє точно зіставляти URL-адреси з конкретними функціями або поданнями, тим самим полегшуючи визначення кінцевих точок у додатку. Крім того, Flask інтегрує шаблонізатор Jinja2, що полегшує створення динамічних HTML-сторінок, які поєднують статичний вміст з даними з програми.

Однією з найсильніших сторін Flask є його розширюваність. Фреймворк може похвалитися потужною екосистемою розширень, які надають додаткові функціональні можливості, такі як безшовна інтеграція з базами даних, механізми валідації форм та підтримка автентифікації. Ці розширення можуть бути легко інтегровані в додатки Flask, що дозволяє розробникам адаптувати фреймворк до своїх конкретних вимог.

Flask вирізняється своєю легкістю та масштабованістю. Його мінімалістичний дизайн забезпечує ефективність та адаптивність, не накладаючи жорстких залежностей чи вимог. Це дозволяє розробникам вибірково обирати компоненти, які відповідають потребам їхнього проекту. Крім того, Flask легко інтегрується з іншими бібліотеками та інструментами Python, полегшуючи використання існуючої функціональності для розширення можливостей програми.

Хоча Flask пропонує численні переваги, важливо враховувати його обмеження. На відміну від деяких комплексних фреймворків, Flask не постачається з попередньо налаштованими функціями та компонентами. Розробники повинні вибрати та

інтегрувати необхідні розширення, щоб додати розширену функціональність. Крім того, акцент Flask на гнучкості може розглядатися як недолік розробниками, які віддають перевагу фреймворкам, що керують структурою та дизайном додатків. Нарешті, Flask зосереджується на наданні основних функціональних можливостей, таких як маршрутизація та шаблони, а більш просунуті функції вимагають використання додаткових розширень або ручної реалізації.

Загалом, Flask представляє себе як легкий, гнучкий фреймворк для веб-розробки на мові Python. Його простота, розширюваність і можливості безперешкодної інтеграції роблять його чудовим вибором для розробників, які прагнуть отримати детальний контроль над структурою та дизайном додатків. Хоча Flask може вимагати більше ручної конфігурації та інтеграції порівняно з універсальними фреймворками, його гнучкість і різноманітна екосистема розширень надають розробникам необхідні інструменти для створення ефективних і масштабованих веб-додатків.

Переваги Flask:

- Легкий та мінімалістичний фреймворк.
- Пропонує простоту та гнучкість у розробці додатків.
- Забезпечує детальний контроль над структурою програми та дизайнерськими рішеннями.
- Безперешкодна інтеграція з іншими бібліотеками та інструментами Python.
- Розгалужена екосистема розширень для додаткової функціональності.
- Ефективність та масштабованість.

Недоліки Flask:

- Не містить попередньо налаштованих функцій та компонентів.
- Може вимагати ручної інтеграції розширень для розширеної функціональності.
- Сконцентрований на гнучкості, а не на упередженій структурі, що може не відповідати вподобанням усіх розробників.
- Розширені можливості можуть потребувати додаткових розширень або ручної реалізації.

2.2.3 Остаточний вибір фреймворку

На основі детального аналізу різних мов програмування та фреймворків для веб-розробки можна зробити висновок, що Django є найкращим вибором веб-застосунку, що розроблюється. Django - це потужний і надійний веб-фреймворк, який пропонує безліч переваг для розробки веб-додатків.

Однією з ключових переваг Django є його прихильність архітектурному шаблону Model-View-Controller (MVC), відомому в Django як Model-View-Template (MVT). Цей шаблон сприяє чистій організації коду і розділенню завдань, що полегшує підтримку і розширення програми з часом.

Django також вирізняється вбудованим адміністративним інтерфейсом, який дозволяє легко керувати та налаштовувати моделі даних додатку. Ця функція економить час і зусилля розробників, надаючи готове рішення для обробки CRUD-операцій (створення, читання, оновлення, видалення) над записами бази даних.

Крім того, шар ORM (об'єктно-реляційне відображення) Django абстрагує базову базу даних, дозволяючи розробникам працювати з записами бази даних за допомогою коду Python замість написання SQL запитів. Це спрощує роботу з базою даних і сприяє повторному використанню коду.

Ще однією перевагою Django є його сильний акцент на безпеці. Він включає в себе численні заходи безпеки, такі як захист від поширених веб-уразливостей, таких як міжсайтовий скриптинг (XSS) і підробка міжсайтових запитів (CSRF). Система автентифікації та авторизації Django також надає надійні можливості управління користувачами, гарантуючи безпеку та захист вашого додатку.

Крім того, Django має велику та активну спільноту, яка сприяє постійному вдосконаленню фреймворку. Це означає доступ до вичерпної документації, навчальних посібників та широкого спектру сторонніх пакетів і бібліотек, які можуть ще більше розширити функціональність вашого додатку.

Таким чином, Django пропонує комплексний і багатофункціональний

фреймворк, який вирізняється масштабованістю, організацією коду, безпекою та підтримкою спільноти. Враховуючи потужність і хорошу зарекомендованість, Django є чудовим вибором для створення веб-додатку.

ВИСНОВКИ ДО РОЗДІЛУ 2

Отже, процес вибору відповідної мови програмування та фреймворку для веб-додатку є важливим рішенням, яке суттєво впливає на успіх проекту. Після ретельної оцінки та розгляду різних варіантів, Python був обраний як найкраща мова програмування для цього проекту. Python пропонує безліч переваг, серед яких простота, читабельність, велика бібліотечна екосистема та придатність для веб-розробки.

З точки зору веб-фреймворків, Django виявився найбільш сприятливим вибором. Завдяки широкому набору функцій, дотриманню найкращих практик та потужній підтримці спільноти, Django забезпечує міцну основу для розробки надійних та масштабованих веб-додатків. Його архітектура Model-View-Template, вбудований адміністративний інтерфейс, рівень ORM та акцент на безпеці роблять його переконливим вибором для потреб нашого проекту.

Використовуючи потужність Python та можливості фреймворку Django, можна спростити розробку, забезпечити якість коду, підвищити безпеку та отримати доступ до величезної екосистеми ресурсів. Поєднання Python та Django дозволяє нам створювати високопродуктивні, зручні в обслуговуванні та багатофункціональні веб-додатки, які відповідають вимогам та очікуванням наших користувачів.

Хоча інші мови програмування та фреймворки мають свої переваги, рішення використовувати Python та Django відображає ретельне врахування таких факторів, як ефективність розробки, підтримка коду, підтримка спільноти та відповідність нашим конкретним вимогам проекту. Маючи в своєму розпорядженні Python і Django, ми маємо все необхідне для того, щоб розпочати шлях розробки і створити винятковий веб-додаток.

3 РОЗРОБКА ЗАСТОСУНКУ

Розробка веб-сервісу, який надає можливості пошуку маршрутів, є серйозним завданням, що вимагає ретельного планування, ефективної реалізації та безперешкодної інтеграції різних технологій. У сучасному швидкоплинному світі, де подорожі та транспорт відіграють вирішальну роль у нашому повсякденному житті, потреба в ефективному та надійному рішенні для пошуку маршрутів є першочерговою. Незалежно від того, чи це стосується пасажирів, які шукають найкоротший шлях до місця призначення, чи туристів, які досліджують незнайомі міста, добре розроблений веб-сервіс може спростити процес пошуку оптимальних маршрутів і покращити загальний користувацький досвід.

Цей розділ заглиблюється у сферу розробки веб-сервісу, досліджуючи тонкощі, пов'язані зі створенням надійних, масштабованих і зручних для користувачів онлайн-сервісів. Ми заглибимося у фундаментальні концепції, методології та технології, які лежать в основі розробки сучасних веб-сервісів. Ця глава має на меті забезпечити комплексне розуміння процесу розробки веб-сервісів - від дизайну та архітектури до стратегій впровадження та розгортання.

У ході вивчення цього розділу ми розглянемо різні аспекти розробки веб-сервісів, включаючи вибір відповідних мов програмування, фреймворків та інструментів, принципи проектування для створення інтуїтивно зрозумілих користувацьких інтерфейсів, реалізацію безпечних механізмів автентифікації та авторизації, а також стратегії розгортання для забезпечення оптимальної продуктивності та масштабованості. Крім того, ми торкнемося таких ключових аспектів, як управління даними, інтеграція API та методології тестування, які є важливими для створення надійного та ефективного веб-сервісу.

Отримавши уявлення про тонкощі розробки веб-сервісів, ми зможемо ефективно планувати, розробляти та впроваджувати надійні та орієнтовані на користувача веб-сервіси, які задовольнятимуть потреби нашої цільової аудиторії. Знання та розуміння, отримані з цього розділу, забезпечать нас необхідними інструментами та методами, щоб орієнтуватися в постійно мінливому ландшафті розробки веб-сервісів і надавати

інноваційні та ефективні послуги в Інтернеті.

3.1 Загальний опис роботи системи, її можливості

Система планування маршрутів, що розроблюється, є складною та інтуїтивно зрозумілою, полегшує планування маршрутів та навігацію для користувачів. Вона пропонує широкий спектр функцій та інструментів, забезпечуючи ефективний та безперебійний пошук оптимальних маршрутів різними видами транспорту.

В основі системи лежить потужна функція пошуку маршрутів. Користувачі можуть ввести початкову точку і пункт призначення, а система планування маршрутів швидко розрахує і представить найбільш підходящий маршрут на основі різних факторів, таких як час у дорозі, час у транспорті та час прибуття.

Підтримка різних видів транспорту та різноманітних маршрутів є ключовою перевагою системи. Адміністратор транспортної компанії може додавати будь-які транспортні маршрути. Ця різноманітність дозволяє пропонувати індивідуальні маршрути, які відповідають індивідуальним уподобанням та конкретним вимогам до подорожей.

Дані, надані адміністратором чи представником транспортної компанії, використовуються для того щоб ефективно та зручно для користувача надавати актуальну інформацію про умови дорожнього руху, розклад громадського транспорту та додаткову інформацію про маршрути.

Персоналізація - ще один важливий аспект системи. Адміністратор чи представник транспортної компанії може додати будь-який вид маршруту, такий як прямий маршрут, маршрут з проміжними зупинками, кільцевий маршрут, чи маршрут який складається з багатьох відрізків. Також маршрутам можна задавати будь-який розклад, як регулярні та часті відправлення, так і поїздки які відбуваються раз на місяць.

Завдяки всеосяжній базі даних про визначні місця, система планування

маршрутів дозволяє користувачам досліджувати пам'ятки, визначні місця, ресторани та інші об'єкти, розташовані вздовж їхніх маршрутів. Ця багата інтеграція покращує процес прийняття рішень і полегшує навігацію, надаючи контекстну інформацію та відповідні рекомендації.

Візуальне представлення є важливим компонентом інтерфейсу системи. Користувачі можуть побачити на карті місцезнаходження будь-якої обраної зупинки, серед іншої інформації про зупинку. Користувачі можуть легко збільшувати та зменшувати масштаб, панорамувати карту та отримувати доступ до додаткової інформації за потреби.

Визнаючи важливість мобільного доступу, система планування маршрутів має адаптивний дизайн для сумісності з різними пристроями, включаючи настільні комп'ютери, смартфони та планшети. Така адаптивність гарантує, що користувачі можуть отримати доступ до її потужних функцій і використовувати їх на ходу, зміцнюючи її статус як зручного і незамінного навігаційного інструменту.

Загалом, система планування маршрутів втілює прагнення спростити процес планування маршрутів, надаючи точні та ефективні навігаційні рішення. Поєднуючи передові алгоритми, інтеграцію даних у режимі реального часу та зручні інтерфейси, вона створює передумови для безперебійного та ефективного планування маршрутів, що задовольняє різноманітні потреби користувачів.

3.1.1 Основний функціонал системи

Сервіс пошуку маршрутів, який розроблюється, пропонує широкий спектр функцій, покликаних спростити та покращити досвід подорожей для наших користувачів. Незалежно від того, чи пересуваються вони жвакими міськими вулицями, чи планують щоденні поїздки на роботу, чи вирушають у нову подорож, цей сервіс надає інструменти та функціональність, які допоможуть їм знайти найкращі маршрути відповідно до їхніх уподобань. Від оновлень трафіку в

реальному часі до персоналізованого планування маршрутів - нижче наведені деякі з ключових функцій, які пропонує даний сервіс пошуку маршрутів транспорту.

Перелік основних функцій доступних в системі:

- Пошук маршруту: Можливість знайти найкоротший або оптимальний маршрут між двома або більше місцями на основі різних критеріїв, таких як відстань, час у дорозі, вид транспорту або вподобання користувача.

- Різні види транспорту: Підтримка різних видів транспорту, включаючи громадський транспорт (автобус, потяг, метро), автомобіль, пішохідний, велосипедний та інші, що дозволяє користувачам обирати найбільш підходящий спосіб пересування для своєї подорожі.

- Маршрути з кількома зупинками та пунктами призначення: Підтримка пошуку та відображення маршрутів, які складаються з кількох видів транспорту.

- Покрокові вказівки: Покрокові вказівки та інструкції, на протязі всього прокладеного маршруту, включаючи інформацію про час відправлення, проміжні зупинки, станції для пересадки, карта з розміщенням зупинки.

- Сумісність: Сумісність з різними браузерами та пристроями, включаючи настільні комп'ютери, смартфони, планшети та що дозволяє користувачам отримувати доступ до сервісу з їхніх улюблених пристроїв.

Ці та багато інших функцій роблять сервіс для пошуку маршрутів, що розроблюється, цінним інструментом для людей, які шукають ефективні та оптимізовані маршрути для щоденних поїздок на роботу, подорожей та інших цілей. Завдяки зручному інтерфейсу та розширеній функціональності можна забезпечити безперебійну та приємну роботу для всіх користувачів.

3.1.2 Отримання, обробка та відображення даних про транспорт

В основі сервісу пошуку маршрутів лежить ефективна обробка та використання транспортних даних. У цьому розділі буде заглиблено у складний

процес отримання, обробки та відображення транспортних даних, який є основою функціональності сервісу для пошуку маршрутів.

Для надання точної та актуальної транспортної інформації в сервісі використовується інтерактивний користувацький інтерфейс, за допомогою якого адміністратор чи представник транспортної компанії може вводити дані про транспортні маршрути, щоб користувачі могли ними користуватись. Також інтерфейс дозволяє користувачам шукати маршрути, що підходять їх потребам, вказавши бажаний час відправлення, початкову зупинку та пункт призначення.

Після отримання даних з інтерфейсу система для пошуку маршрутів використовує складні алгоритми і методи обробки даних для перетворення введених користувачем даних у придатний для використання формат. Це включає перевірку введених даних, зіставлення їх з наявними транспортними даними та визначення найкращих можливих маршрутів на основі вподобань користувача.

Оброблені дані потім організовуються в комплексну і взаємопов'язану транспортну мережу, де встановлюються зв'язки між різними маршрутами, зупинками і пунктами призначення. Це дозволяє сервісу генерувати точні та ефективні маршрути на основі даних та вподобань користувачів. Транспортні дані доповнюються додатковою контекстною інформацією, такою як дані про об'єкти, орієнтири та відповідні геопросторові дані, що збагачує загальний користувацький досвід.

Відображення оброблених транспортних даних у зручний та візуально привабливий спосіб має вирішальне значення для забезпечення безперешкодної роботи користувачів. В даному сервісі використовуються сучасні методи візуалізації даних та інтерактивні карти для представлення зупинок. Інтерфейс розроблений таким чином, щоб бути інтуїтивно зрозумілим і зручним для навігації, що дозволяє користувачам легко досліджувати та взаємодіяти з транспортними даними.

Зображення сторінки для додавання даних

Зображення сторінки для відображення доданих маршрутів

Щоб зробити можливим додавання транспортних маршрутів до системи пошуку маршрутів, потрібно реалізували функціонал, який дозволяє авторизованим адміністраторам чи відповідальним особам додавати нові маршрути.

Використовуючи архітектуру Django Model-View-Controller (MVC), потрібно створити спеціальну функцію для відображення транспортних маршрутів та форму для створення транспортного маршруту. Форма збирає необхідну інформацію для створення маршруту. Після відправки даних, вони перевіряються спеціальною функцією і створюється новий об'єкт маршруту в базі даних, використовуючи ORM Django. Це гарантує, що доданий маршрут зберігається постійно і до нього можна отримати доступ для подальшої обробки та відображення.

Обробка транспортних маршрутів включає в себе маніпуляції та аналіз даних маршруту для отримання змістовної інформації. Для цього використовуються потужні можливості запитів Django для отримання відповідних маршрутів з бази даних на основі певних критеріїв. Наприклад, можна отримати маршрути, які задовольняють певним параметрам, таким як відстань або час. Маючи отримані дані про маршрут, можна виконувати розрахунки і застосовувати алгоритми для визначення таких факторів, як найкоротший маршрут, оптимальний розклад або альтернативні маршрути. Ці оброблені результати можна зберігати або використовувати для подальших операцій, таких як рекомендації щодо маршруту або оптимізація розкладу.

Щоб надати користувачам інтуїтивно зрозуміле та інформативне відображення транспортних маршрутів, використовується система шаблонів Django та HTML/CSS для створення динамічних і візуально привабливих веб-сторінок. За допомогою цього розробляються шаблони, які ефективно представляють деталі маршруту, включаючи пункт відправлення, пункт призначення, проміжні зупинки, пересадки, час у дорозі та види транспорту.

Реалізавши ці функції в рамках проекту на Python та Django, можна легко обробляти додавання, обробку та відображення транспортних маршрутів. Це забезпечує ефективне управління даними, точні розрахунки маршрутів і візуально привабливий користувацький інтерфейс, що дозволяє користувачам приймати обґрунтовані рішення щодо подорожей на основі доступних варіантів транспорту.

3.1.3 Пошук маршруту

Сторінка пошуку маршрутів є основною функцією сервісу для пошуку маршрутів, надаючи користувачам зручний та інтуїтивно зрозумілий інтерфейс для планування своїх подорожей. У цьому розділі розглядаються ключові функції та можливості сторінки для пошуку маршрутів.

По суті, сторінка для пошуку маршрутів пропонує користувачам можливість легко планувати свої маршрути з одного місця в інше. Користувачі можуть ввести початкову точку і пункт призначення, а сервіс для пошуку маршрутів, використовуючи складні алгоритми і методи обробки даних, генерує оптимальні маршрути відповідно до запитів користувачів. Цей спрощений процес позбавляє користувачів необхідності орієнтуватися в складних транспортних мережах або вручну шукати маршрути найкоротші маршрути серед купи транспорту.

Однією з основних причин, чому користувачі покладаються на сторінку пошуку маршрутів, є її здатність економити час і зусилля. Завдяки комплексному огляду доступних маршрутів, включаючи різні види транспорту, альтернативні шляхи та орієнтовний час у дорозі, користувачі можуть приймати обґрунтовані рішення без необхідності проведення тривалих досліджень або спроб і помилок. Інтуїтивно зрозумілий дизайн сторінки та зручний інтерфейс роблять її доступною для користувачів з будь-якою технічною підготовкою, що ще більше підвищує важливість сторінки для пошуку маршрутів, її привабливість та зручність використання.

Окрім економії часу та зусиль, сторінка для пошуку маршрутів пропонує користувачам можливість відкрити для себе нові та ефективні способи навігації в транспортній мережі. Вона представляє користувачам низку транспортних варіантів. Це дозволяє користувачам обирати вид транспорту, який найкраще відповідає їхнім потребам, беручи до уваги такі фактори, як зручність, вартість, вплив на навколишнє середовище та особисті уподобання. Динамічний та адаптивний характер сторінки гарантує, що користувачі можуть знайти маршрути, які відповідають їхнім

конкретним вимогам.

Крім того, сторінка пошуку маршрутів слугує надійним помічником для користувачів, які покладаються на мережі громадського транспорту. Вона надає детальну інформацію про розклад руху, зупинки та пересадки, що дозволяє користувачам точно і впевнено планувати свої подорожі.

Таким чином, сторінка для пошуку маршрутів пропонує користувачам комплексний та ефективний інструмент для планування своїх поїздок. Спрощуючи процес пошуку маршруту, вона дозволяє користувачам приймати обґрунтовані рішення, економити час і знаходити оптимальні маршрути, які відповідають їхнім уподобанням. Зручний інтерфейс, гнучкість та інтеграція з даними в режимі реального часу роблять сервіс для пошуку маршрутів незамінним ресурсом для людей, які шукають швидку, зручну та надійну навігацію у своїх повсякденних подорожах.

Зображення сторінки для пошуку маршрутів

Для того щоб розробити вище описану сторінку, для розробки фронтенду потрібно зробити інтуїтивно зрозумілий користувацький інтерфейс, який дозволяє користувачам вводити бажані станції відправлення та призначення. Для цього використовуються сучасні веб-технології, такі як HTML, CSS та JavaScript, щоб створити візуально привабливий та адаптивний інтерфейс.

Внутрішня частина сторінки пошуку маршрутів відіграє життєво важливу роль в обробці даних, введених користувачем, і отриманні відповідних даних. Використовуючи фреймворк Django, буде розроблено необхідні функції для представлення та обробки для обробки вхідних запитів та ініціювання процесу пошуку маршруту. Ці функції для представлення витягують дані, введені користувачем, і викликають логіку серверної частини, щоб знайти найбільш підходящий маршрут на основі попередньо визначених алгоритмів і моделей даних.

Бекенд використовує надійні можливості Django для отримання даних з основної бази даних. Завдяки чітко визначеним моделям, що представляють станції, маршрути, сполучення та точки маршруту, сервіс може ефективно отримувати необхідну інформацію для розрахунку маршруту. Використовуючи ORM (об'єктно-реляційне відображення) Django, легко долається розрив між операціями з

базою даних та написаним кодом на Python.

Для визначення оптимального маршруту застосовується алгоритм, який ретельно враховує такі фактори, як відстань, час та бажані види транспорту. Бекенд обробляє дані, виконує необхідні обчислення і повертає результати на фронтенд для відображення.

Завдяки такій злагодженій взаємодії фронтенду та бекенду користувачі можуть легко шукати маршрути, переглядати відповідні деталі та приймати обґрунтовані рішення щодо подорожі. Сторінка для пошуку маршрутів є найголовнішою сторінкою застосунку, поєднуючи елегантний користувацький інтерфейс з надійною функціональністю бекенду для надання точних та ефективних пропозицій щодо маршрутів.

3.1.4 Відображення знайденого маршруту

Сторінка для відображення знайденого маршруту є важливим компонентом сервісу для пошуку маршрутів, надаючи користувачам комплексне уявлення про підібрані для них маршрути. У цьому розділі розглядаються функції та можливості сторінки для відображення знайденого маршруту.

Основна функція сторінки для відображення знайденого маршруту - надати користувачам візуально привабливе та інформативне представлення підібраних для них маршрутів. Після того, як користувачі здійснили пошук маршруту, на цій сторінці відображається знайдений маршрут разом з різними деталями та інтерактивними елементами. Користувачі можуть переглядати покрокові вказівки, карти з зупинками, пересадки, приблизний час у дорозі та іншу інформацію, яка допомагає їм ефективно орієнтуватися під час подорожі.

Однією з ключових причин, чому користувачі покладаються на сторінку відображення маршруту, є її здатність забезпечити чітке розуміння всього маршруту від початку до кінця. Вона представляє користувачам огляд прокладеного маршруту,

виділяючи послідовність маршрутних точок, об'єктів і будь-яких необхідних змін режиму руху. Таке комплексне відображення дає змогу користувачам підготуватися до подорожі, забезпечуючи цілісне розуміння майбутнього маршруту та передбачаючи будь-які потенційні виклики або орієнтири на шляху.

Крім того, сторінка відображення маршруту пропонує користувачам низку інтерактивних функцій для покращення вивчення маршруту. Користувачі можуть збільшувати і зменшувати масштаб мапи і досліджувати конкретні об'єкти або орієнтири вздовж маршруту. Ці інтерактивні елементи надають користувачам динамічний і цікавий досвід, дозволяючи їм ознайомитися з навколишнім середовищем, визначити орієнтири і приймати обґрунтовані рішення під час подорожі.

Крім того, на сторінці відображення маршруту пріоритетом є доступність та зручність для користувачів. Вона розроблена таким чином, щоб бути адаптивною та зрозумілою, забезпечуючи безперебійний перегляд на різних пристроях та екранах різного розміру. Це дозволяє користувачам отримувати доступ до обраних маршрутів в такому вигляді, в якому їм зручно, і взаємодіяти з ними, незалежно від того, чи використовують вони стаціонарний комп'ютер, планшет, мобільний пристрій чи будь-який інший пристрій, де є браузер.

Таким чином, сторінка для відображення знайденого маршруту відіграє вирішальну роль у наданні користувачам комплексного та інтерактивного відображення обраних ними маршрутів. Пропонуючи детальні вказівки, інтерактивні карти, вона дозволяє користувачам впевнено та ефективно орієнтуватися у транспортних мережах під час подорожі, чи на етапі її планування. Зручний інтерфейс, цікаві інтерактивні елементи та доступність роблять сторінку для відображення знайденого маршруту цінним ресурсом для тих, хто шукає зручне та інформативне відображення маршрутів.

Зображення сторінки для пошуку маршрутів

Щоб реалізувати функцію відображення маршруту у нашому веб-додатку, ми використовуємо потужні можливості та гнучкість фреймворку Django. В основі реалізації лежить використання представлень і шаблонів Django. Ми створюємо спеціальне представлення, яке отримує обрану інформацію про маршрут з бекенду і

відображає її за допомогою відповідного шаблону. Представлення отримує необхідні дані, такі як покрокові інструкції, точки маршруту, орієнтовний час у дорозі та інші відповідні деталі, з нашої внутрішньої бази даних або зовнішніх API.

Система шаблонів Django дозволяє нам структурувати і формувати інформацію про маршрут у візуально привабливий спосіб. За допомогою використання HTML, CSS та мови шаблонів Django можна створити зручний та функціональний інтерфейс, який представляє деталі маршруту в чіткій, зручній, зрозумілій та організованій формі. Шаблон може включити інтерактивні елементи, такі як карти, функцію масштабування та перемикач між різними видами карт, щоб покращити користувацький досвід.

Крім того, система автентифікації та авторизації користувачів Django дозволяє представникам транспортної компанії керувати доступом до управління транспортними даними і надавати доступ до цієї функції лише відповідальним особам. Це дозволяє відділити управління транспортом від функцій доступних звичайним користувачам, таким як перегляд маршрутів, пошук маршрутів і інших.

Крім того, підтримка Django адаптивного веб-дизайну та мобільної оптимізації гарантує, що сторінка з маршрутом буде доступною і функціональною на різних пристроях і з різними розмірами екрану. Використовуючи адаптивні CSS фреймворки, такі як Bootstrap, ми можемо досягти адаптивного і візуально привабливого інтерфейсу користувача, який адаптується до пристрою користувача, будь то настільний комп'ютер, планшет або мобільний телефон.

3.2 Дизайн користувацького інтерфейсу

Дизайн користувацького інтерфейсу відіграє вирішальну роль у створенні привабливого та інтуїтивно зрозумілого інтерфейсу додатку для користувачів. Він зосереджується на розробці візуальних та інтерактивних елементів додатку, з якими користувачі взаємодіють для виконання своїх завдань і досягнення цілей. Добре

продуманий інтерфейс не лише покращує зручність та функціональність додатку, але й сприяє загальному задоволенню та залученню користувачів.

Процес розробки користувацького інтерфейсу починається з розуміння цільової аудиторії, її потреб, вподобань та поведінки. Проведення користувацьких досліджень, інтерв'ю та юзабіліті-тестування допомагає отримати уявлення про очікування та вимоги користувачів. Такий підхід, орієнтований на користувача, гарантує, що дизайн інтерфейсу відповідає ментальним моделям користувачів і забезпечує безперебійну та інтуїтивно зрозумілу роботу з ним.

Ключові принципи дизайну інтерфейсу включають простоту, послідовність, ясність та інтуїтивність. Простота передбачає усунення непотрібної складності та захащеності інтерфейсу, зосередження уваги на основних елементах і функціях. Чистий і не захащений дизайн дозволяє користувачам швидко зрозуміти призначення додатку і легко орієнтуватися в його функціях.

Послідовність має вирішальне значення для створення цілісного та передбачуваного користувацького досвіду. Підтримка однакових візуальних елементів, таких як типографіка, кольори та іконки, на різних екранах і в різних взаємодіях допомагає користувачам розвивати впізнаваність і зменшує когнітивне навантаження. Послідовність також поширюється на розміщення інтерактивних елементів і розташування контенту, гарантуючи, що користувачі можуть легко знаходити і взаємодіяти з потрібними функціями.

Ясність у дизайні інтерфейсу гарантує, що інформація та дії представлені у чіткій та зрозумілій формі. Використання описових міток, надання корисних відгуків і вказівок, а також відповідні візуальні підказки дозволяють користувачам зрозуміти інтерфейс і функціональність програми без двозначностей. Добре продумані повідомлення про помилки та інформативні підказки сприяють запобіганню та вирішенню помилок.

Інтуїтивність - ключовий аспект ефективного дизайну користувацького інтерфейсу. Інтерфейс повинен відповідати ментальним моделям користувачів і поширеним моделям взаємодії, щоб полегшити їм вивчення і навігацію в додатку. Звичні шаблони та конвенції дизайну, такі як використання стандартних іконок, жестів та навігаційних структур, сприяють загальній інтуїтивності додатку.

Візуальна естетика також відіграє важливу роль у дизайні користувацького інтерфейсу. Вибір кольорів, типографіки та візуальних елементів повинен узгоджуватися з брендингом та призначенням додатку, створюючи візуально привабливий та цілісний досвід. Увага до деталей, таких як інтервали, вирівнювання та візуальна ієрархія, покращує читабельність та візуальну ясність.

Під час проектування користувацького інтерфейсу дизайнери використовують різні інструменти та методи для створення каркасів, макетів та прототипів. Ці візуальні представлення допомагають зацікавленим сторонам і командам розробників візуалізувати макет, взаємодію та потік додатку. Ітеративні процеси проектування, що включають зворотний зв'язок і доопрацювання, гарантують, що дизайн користувацького інтерфейсу відповідає вимогам додатку і відповідає очікуванням користувачів.

Отже, дизайн користувацького інтерфейсу є критично важливим аспектом розробки додатків, який фокусується на створенні цікавого, інтуїтивно зрозумілого та візуально привабливого користувацького інтерфейсу для користувачів. Дотримуючись принципів простоти, послідовності, ясності та інтуїтивності, дизайнери можуть створювати інтерфейси, які підвищують зручність використання, полегшують виконання завдань і забезпечують чудовий користувацький досвід. Ефективний дизайн інтерфейсу в поєднанні з бездоганною функціональністю та надійним бекендом сприяє загальному успіху додатку в задоволенні потреб користувачів і досягненні поставлених цілей.

3.2.1 Проектування адаптивного інтерфейсу

Адаптивний дизайн інтерфейсу - це підхід до створення користувацьких інтерфейсів, які можуть динамічно адаптуватися та підлаштовуватися під різні пристрої, розміри екранів та орієнтацію. Зі збільшенням різноманітності пристроїв і платформ дуже важливо забезпечити доступність інтерфейсу додатку та його

оптимізацію для безперешкодної роботи користувача на різних пристроях, включаючи мобільні телефони, планшети та комп'ютери.

Одним з популярних фреймворків, який дозволяє створювати адаптивний дизайн інтерфейсу, є Bootstrap. Bootstrap - це фронтенд-фреймворк, який надає набір компонентів CSS і JavaScript, а також адаптивну систему сітки, щоб спростити процес створення адаптивних веб-інтерфейсів. Він пропонує ряд заздалегідь розроблених елементів інтерфейсу, адаптивних класів і варіантів макетів, які допомагають розробникам створювати інтерфейси, що автоматично підлаштовуються під різні розміри екрану.

Використовуючи Bootstrap, розробники можуть використовувати його адаптивну систему сітки для створення гнучкого макета, який адаптується до різних розмірів екрану. Система сітки дозволяє організувати вміст у адаптивні стовпці та рядки, забезпечуючи розумне переливання та перегруповання елементів залежно від доступного простору екрану. Це забезпечує послідовний та оптимізований досвід перегляду для користувачів на різних пристроях.

Bootstrap також пропонує набір адаптивних класів CSS, які розробники можуть застосовувати до елементів, щоб керувати їх видимістю або поведінкою на різних розмірах екрану. Наприклад, ви можете використовувати клас "hidden-xs щоб приховати елемент на дуже маленьких екранах, або клас "col-md-offset-3 щоб змістити позиціонування елемента на екранах середнього розміру.

Крім того, Bootstrap надає широкий спектр попередньо стилізованих компонентів інтерфейсу, таких як навігаційні панелі, кнопки, форми і модальні елементи, які розроблені для мобільних пристроїв і адаптуються до різних розмірів екранів. Ці компоненти мають вбудовану адаптивність, що гарантує, що вони масштабуються та адаптуються належним чином на різних пристроях.

Використовуючи функції та компоненти Bootstrap, розробники можуть створювати адаптивний інтерфейс, який автоматично підлаштовується під різні розміри екранів і забезпечує однаковий користувацький досвід на різних пристроях. Це допомагає скоротити час і зусилля на розробку, забезпечуючи при цьому доступність додатку для ширшого кола користувачів.

Зверніть увагу, що хоча Bootstrap є популярним вибором для створення

адаптивного дизайну інтерфейсу, існують також інші фреймворки та підходи, які можуть допомогти досягти подібних результатів. Вибір фреймворку може залежати від таких факторів, як вимоги проекту, досвід команди розробників і конкретні цілі дизайну.

Основні особливості Bootstrap:

- Система адаптивної сітки
- Попередньо стилізовані компоненти інтерфейсу
- Адаптивні утиліти
- Налаштовувані теми
- Сумісність з браузерами
- Документація та активна спільнота користувачів

ВИСНОВКИ ДО РОЗДІЛУ 3

У цьому розділі було розглянуто різні аспекти розробки сервісу планування маршрутів. Спочатку було досліджено ключові функціональні можливості сервісу для пошуку маршрутів транспорту, включаючи пошук маршрутів, відображення маршрутів, додавання, обробку та відображення транспортних маршрутів. Функціонал пошуку маршрутів дозволяє користувачам знаходити найоптимальніші маршрути відповідно до їхніх уподобань та вимог. Функціонал відображення маршрутів представляє знайдені маршрути у зрозумілій та інформативній формі, надаючи користувачам покрокові інструкції, приблизний час у дорозі та будь-які важливі деталі.

Крім того, було обговорено, як Python та Django використовуються для реалізації цих функцій. Надійний фреймворк Django та розгалужена екосистема забезпечують ефективну та безпечну обробку даних, плавну інтеграцію з базами даних та безперешкодну взаємодію з інтерфейсними компонентами. Поєднання Python та Django забезпечує міцну основу для розробки надійного та масштабованого сервісу планування маршрутів.

Інтегруючи всі ці компоненти та функціональні можливості, розроблений сервіс планування маршрутів дозволяє користувачам легко шукати маршрути, переглядати детальну інформацію та приймати обґрунтовані рішення щодо своїх транспортних потреб. Продуманий дизайн, зручний інтерфейс та ефективна внутрішня реалізація працюють разом, щоб забезпечити комплексний та цінний досвід для користувачів.

У розділі також підкреслюється важливість користувацького досвіду та дизайну інтерфейсу. Завдяки ефективному дизайну інтерфейсу, зокрема адаптивному макету, інтуїтивно зрозумілій навігації та візуально привабливим елементам, користувачі можуть легко взаємодіяти з сервісом і отримувати доступ до потрібних функцій. Реалізація таких функцій, як пошук маршрутів, відображення маршрутів та взаємодія з транспортними даними, забезпечує безперебійну роботу користувачів.

Підсумовуючи, у розділі було розглянуто принципи проектування,

функціональні можливості та методи реалізації, необхідні для розробки надійного та орієнтованого на користувача сервісу планування маршрутів. Застосовуючи ці принципи та використовуючи можливості Python і Django, сервіс для пошуку маршрутів має на меті покращити досвід користувачів у плануванні поїздок та оптимізувати їхні транспортні рішення.

4 АНАЛІЗ РЕЗУЛЬТАТІВ ТА ІНСТРУКЦІЯ КОРИСТУВАЧА

Аналіз результатів та створення інструкції користувача є невід’ємними етапами розробки та впровадження будь-якої програмної системи, в тому числі і сервісу планування маршрутів. Аналіз результатів дозволяє оцінити продуктивність та ефективність сервісу на основі різних метрик та показників. Уважно вивчаючи зібрані дані та проводячи ретельний аналіз, можна виявити закономірності, тенденції та потенційні сфери для покращення. Цей аналіз дозволяє приймати обґрунтовані рішення, оптимізувати алгоритми роботи сервісу для пошуку маршрутів та підвищити його загальну функціональність і ефективність.

Крім того, проведення ретельного аналізу системи є вкрай важливим для забезпечення надійної та точної роботи сервісу планування маршрутів. Завдяки комплексному аналізу системи можна переконатися, що сервіс функціонує належним чином, прокладаючи точні та оптимальні маршрути на основі даних користувача. Ретельний аналіз роботи системи допомагає виявити будь-які потенційні проблеми.

Після того, як система пройшла ретельний аналіз і перевірку, інструкція користувача стає важливим компонентом. Інструкція користувача слугує всеосяжним керівництвом, що містить детальні інструкції про те, як користуватися сервісом, орієнтуватися в його можливостях та максимально ефективно використовувати його функціональні можливості. Ця інструкція пропонує пояснення різних опцій та налаштувань, а також поради щодо усунення несправностей, щоб допомогти користувачам у взаємодії з сервісом. Інструкція користувача має на меті забезпечити безперебійну роботу користувачів, що дозволить їм легко орієнтуватися в сервісі планування маршрутів і досягати бажаних результатів.

Отже, аналіз результатів дозволяє отримувати інформацію та приймати рішення на основі даних, а ретельне тестування забезпечує надійність і точність сервісу. Інструкція користувача є цінним ресурсом, який надає користувачам знання та рекомендації, необхідні для ефективного використання послуги. Разом ці компоненти сприяють успішному розгортанню та впровадженню сервісу планування маршрутів, підвищуючи рівень задоволеності користувачів та сприяючи

безперешкодному та ефективному плануванню маршрутів.

4.1 Використання розроблено застосунку

Застосунок покликаний допомогти користувачам у пошуку оптимальних маршрутів між потрібними їм пунктами, беручи до уваги різні фактори, такі як відстань, час у дорозі, види транспорту та вподобання щодо розкладу. Завдяки інтуїтивно зрозумілому користувацькому інтерфейсу та потужним алгоритмам пошуку, застосунок має на меті спростити процес планування маршруту та надати користувачам точну та надійну інформацію про маршрут.

Основна мета програми - запропонувати зручний та ефективний спосіб пошуку маршрутів на основі індивідуальних уподобань та вимог. Незалежно від того, чи потрібно вам зорієнтуватися в галасливому місті, дістатися до роботи або спланувати поїздки, додаток оснащений необхідними інструментами, які допоможуть знайти найкращі маршрути, що відповідають потребам користувачів.

Щоб розпочати пошук маршруту, користувачі можуть просто ввести початкову та кінцеву зупинки. Користувачі можуть уточнити пошук, вказавши час відправлення, що дозволяє їм планувати свої подорожі з урахуванням запланованих заходів або очікуваних дорожніх умов.

Результати пошуку представлені в чіткій та інтуїтивно зрозумілій формі, відображаючи рекомендовані маршрути разом з детальною інформацією, такою як орієнтовний час у дорозі, відстань, та будь-яка відповідна інформація про маршрут.

У цьому розділі буде розглянуто різні можливості та функції додатку, надаючи детальні пояснення та покрокові інструкції щодо максимального використання його потенціалу. Наприкінці цього розділу з'явиться повне уявлення про те, як використовувати додаток на повну потужність і планувати свої маршрути з легкістю та впевненістю.

4.1.1 Огляд користувацького інтерфейсу

Розуміння інтерфейсу має вирішальне значення, оскільки воно закладає основу для безперешкодної навігації та ефективного використання функцій програми. Цей підрозділ має на меті ознайомитися з елементами інтерфейсу, їхнім призначенням і тим, як вони впливають на загальний користувацький досвід.

Завдяки детальному вивченню інтерфейсу користувачі отримають повне уявлення про візуальну ієрархію програми, навігаційні меню, функціональність пошукового рядка, відображення карти та інші інтерактивні елементи. Буде висвітлено інтуїтивно зрозумілі дизайнерські рішення, які були реалізовані для забезпечення простоти використання та швидкого доступу до основних функцій.

У підрозділі аналізу інтерфейсу також обговорюватимуться найкращі практики та поради щодо ефективної навігації та використання елементів інтерфейсу. Користувачі дізнаються, як взаємодіяти з різними компонентами, налаштовувати свої уподобання, а також користуватися комбінаціями клавіш і жестами, щоб покращити загальний досвід роботи.

Заглибившись в аналіз інтерфейсу, можна отримати можливість легко орієнтуватися в додатку і використовувати всі його можливості. Незалежно від того, чи ви новачок, чи досвідчений користувач, цей підрозділ надасть цінну інформацію для оптимізації взаємодії та спрощення процесу планування маршрутів.

Коли користувач вперше заходить на сайт, він бачить початкову сторінку. На ній відображені ...

Зображення початкової сторінки

Інші сторінки буде розглянуто в наступних розділах.

Завдяки використанню Bootstrap інтерфейс підходить для екранів будь-якого розміру

Зображення інтерфейсу на екранах різних розмірів.

4.1.2 Огляд функції пошуку маршрутів

За допомогою функції пошуку маршруту користувачі можуть ввести початкову та кінцеву зупинки, щоб розпочати процес пошуку. Додаток використовує передові алгоритми для розрахунку найефективніших і найшвидших маршрутів на основі різних факторів, таких як відстань, час у дорозі та час очікування.

Результат пошуку відображається в чіткій та організованій. Маршрут супроводжується детальною інформацією, включаючи приблизний час у дорозі, відстань та покрокові вказівки.

Надаючи комплексну та зручну функцію пошуку маршрутів, застосунок має на меті спростити процес пошуку та навігації оптимальних маршрутів. Незалежно від того, чи користувачі їдуть на роботу, чи досліджують нове місто, чи планують подорож, вони можуть покластися на функцію пошуку маршрутів, яка надасть їм точні, ефективні та надійні варіанти маршрутів.

Пошук маршруту для користувача починається зі сторінки для вводу даних пошуку. На ній користувач може ввести початкову зупинку та кінцеву зупинку, а також бажаний час відправлення.

Зображення сторінки для вводу даних пошуку

Після вводу даних для пошуку, користувач натискає кнопку "Знайти маршрут". Застосунок оброблює введені дані, знаходить маршрут та відображає його.

Зображення сторінки зі знайденим маршрутом

Користувач також може переглядати деталі про маршрут, такі як ...

Зображення деталей зупинки

Зображення деталей маршруту

4.1.3 Контроль доступу до управління даними про транспорт

Застосунок надає надійну систему облікових записів користувачів, що дозволяє адміністраторам мати спеціальні облікові записи з підвищеними привілеями. Ці облікові записи слугують шлюзами для доступу до адміністративних функцій і можливостей, які недоступні звичайним користувачам. Завдяки окремим обліковим записам адміністраторів програма підтримує чітке розмежування між адміністративними завданнями і взаємодією з користувачами, забезпечуючи ефективне і контрольоване управління системою.

За допомогою облікових записів адміністраторів уповноважений персонал мають можливість створювати, редагувати та видаляти дані про транспортні маршрути.

Функція управління доступом з обліковими записами для адміністраторів дає змогу уповноваженому персоналу ефективно керувати та адмініструвати маршрути транспорту, забезпечуючи безперебійну, безпечну роботу системи відповідно до вимог організації.

Зображення сторінки входу

Можливості адміністратора буде розглянуто в наступному розділі

4.1.4 Огляд можливостей управління транспортними даними

Застосунок пропонує комплексний набір функцій управління транспортними даними, які дозволяють адміністраторам підтримувати та оновлювати базу даних транспортних маршрутів, розкладів та іншої пов'язаної з ними інформації. Ці функції дозволяють легко інтегрувати нові маршрути, вносити зміни до існуючих маршрутів та ефективно обробляти зміни в розкладі руху транспорту.

Однією з ключових особливостей функцій управління транспортними даними є

можливість додавання нових транспортних маршрутів до системи. Адміністратори можуть вводити необхідні дані, такі як пункт відправлення, пункт призначення, проміжні зупинки та пов'язану з ними інформацію про розклад для кожного маршруту. Це дозволяє додатку надавати користувачам точні та вичерпні варіанти маршрутів під час пошуку.

Функції управління транспортними даними також включають можливість видалення застарілих або зайвих маршрутів із системи. Це гарантує, що додаток підтримує чисту та актуальну базу даних, підвищуючи ефективність процесу пошуку та покращуючи загальний користувацький досвід.

Для спрощення управління транспортними даними додаток надає інтуїтивно зрозумілі інтерфейси та інструменти, які полегшують навігацію та редагування інформації про маршрути. Адміністратори можуть використовувати ці інтерфейси для візуалізації маршрутів, перегляду розкладів і внесення необхідних змін у зручний для користувача спосіб. Система також може включати механізми перевірки даних для забезпечення точності та узгодженості введеної інформації, мінімізації помилок та покращення цілісності даних.

Використовуючи функції управління транспортними даними, адміністратори можуть ефективно підтримувати та оновлювати транспортні дані в додатку, забезпечуючи користувачам доступ до надійної та актуальної інформації про маршрути. Це дозволяє користувачам приймати обґрунтовані рішення, ефективно планувати свої поїздки та впевнено орієнтуватися в транспортній мережі.

Адміністратор може створювати нові маршрути для того щоб вони потім відображались при пошуку маршрутів

Зображення сторінки для додавання нового маршруту

Зображення сторінки з заповненими даними про маршрут

Після створення маршрутів, їх можна переглядати в списку зі всіма маршрутами.

Зображення списку існуючих маршрутів

Кожен маршрут можна відкрити на окремій сторінці для перегляду деталей про нього.

Зображення сторінки з переглядом деталей про маршрут

Якщо в маршруті є помилки, дані про маршрут можна редагувати, чи навіть

видалити маршрут.

Зображення редагування маршруту

Зображення видалення маршруту

ВИСНОВКИ ДО РОЗДІЛУ 4

У цьому розділі було розглянуто широкі функціональні можливості розробленого застосунку для пошуку маршрутів. Застосунок пропонує комплексний набір функцій, покликаних допомогти користувачам у пошуку оптимальних маршрутів, управлінні транспортними даними та ефективній навігації. Підсумуємо ключові аспекти функціональності додатку.

Функція пошуку маршрутів надає користувачам зручний інтерфейс для введення бажаних пунктів відправлення та призначення, а також використовує передові алгоритми для розрахунку найшвидших та найефективніших маршрутів. Завдяки можливості враховувати різні фактори, такі як відстань, час у дорозі та час очікування, користувачі можуть впевнено приймати обґрунтовані рішення щодо своїх поїздок.

Функція управління доступом забезпечує безпечний доступ до програми через облікові записи для адміністраторів. Це дозволяє ефективно управляти транспортними даними та забезпечує цілісність функціональності додатку.

Функції управління транспортними даними дозволяють адміністраторам додавати, редагувати та видаляти транспортні маршрути, забезпечуючи актуальність і точність бази даних. Додаток дозволяє легко інтегрувати нові маршрути, вносити зміни до існуючих маршрутів та обробляти зміни в розкладі, надаючи користувачам надійну та актуальну інформацію про маршрути.

Дизайн інтерфейсу користувача програми зосереджений на забезпеченні інтуїтивно зрозумілого та зручного досвіду, що дозволяє користувачам легко орієнтуватися в різних функціях. Адаптивний дизайн забезпечує оптимальний перегляд і взаємодію на різних пристроях, задовольняючи різноманітні потреби користувачів.

Загалом, розроблений застосунок пропонує потужне та ефективне рішення для планування маршрутів та навігації. Поєднуючи передові алгоритми, безпечне управління доступом та комплексне управління транспортними даними, застосунок надає користувачам надійну та зручну платформу для оптимізації їхніх подорожей.

ВИСНОВКИ

Всебічне дослідження процесу розробки веб-застосунку для пошуку маршрутів, висвітлене в чотирьох розділах цього проекту, надало цінну інформацію про розробку надійного і зручного для користувачів додатку для пошуку маршрутів.

У першому розділі було розглянуто існуючі рішення, на прикладі популярних застосунків для пошуку маршрутів, підкресливши важливість ефективної і точної навігації в сучасному швидкоплинному світі. Проаналізувавши існуючі рішення, було отримано глибше розуміння викликів і вимог, а також бажань і потреб користувачів, пов'язаних з розробкою веб-додатку, який надає користувачам оптимальні маршрути.

У другому розділі було заглиблено в різні інструменти та мови програмування, доступні для розробки веб-додатків. Було досліджено сильні та слабкі сторони таких популярних технологій, як Python, JavaScript, Java та Go. Крім того, було проведено порівняння таких відомих фреймворків, як Django та Flask, які пропонують потужні можливості та спрощений досвід розробки для створення веб-застосунків.

Третій розділ присвячений тонкощам створення застосунку для пошуку маршрутів. Вона охоплює такі важливі аспекти, як дизайн адаптивного користувацького інтерфейсу та реалізацію основних функціональних можливостей. Використовуючи відповідні технології та фреймворки, такі як Django ORM, Bootstrap, було забезпечено ефективність, масштабованість та адаптивність застосунку до різних пристроїв.

Останній, п'ятий, розділ дозволив провести комплексну оцінку розробленого застосунку для пошуку маршрутів. За допомогою тестування та перевірки було ретельно проаналізовано функціональність, продуктивність та зручність використання застосунку.

Загалом, шлях від початкового огляду до огляду готового веб-застосунку підкреслив важливість ретельного планування, технологічної експертизи та підходу, орієнтованого на користувача. Завдяки використанню сучасних технологій веб-розробки та дотриманню найкращих практик було створено надійний та

інтуїтивно зрозумілий додаток для пошуку маршрутів.

Таким чином, цей проект забезпечує комплексне дослідження розробки веб-застосунку для пошуку маршрутів. Розглянувши існуючі рішення, обравши відповідні технології, ретельно розробивши застосунок та провівши ретельні перевірки, було успішно створено цінний інструмент для невеликих транспортних компаній та користувачів, які шукають ефективну та надійну навігацію. Шлях від зародження до реалізації підкреслює важливість ретельного планування, порівняння різних веб-технологій та зосередження на створенні виняткового користувацького досвіду.

ПЕРЕЛІК ПОСИЛАНЬ

1. Chen Y., Ma J. Analysis of Route Planning Methods for Tourist Bus Based on GIS Technology / Chen Y., Ma J. — 2020 — [313-328 ст.]
2. Google Maps Wikipedia page [Електронний ресурс]. — Режим доступу до ресурсу: https://en.wikipedia.org/wiki/Google_Maps
3. Moovit [Електронний ресурс]. — Режим доступу до ресурсу: <https://moovit.com/about-us/>
4. Citymapper Wikipedia page [Електронний ресурс]. — Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Citymapper>
5. Transit Wikipedia page [Електронний ресурс]. — Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Transit_\(app\)](https://en.wikipedia.org/wiki/Transit_(app))
6. Klinger T., Dumm R. E. Identifying User Needs and Requirements for Trip Planning Assistance in Multi-Modal Personal Navigation Systems. Proceedings of the 10th International Conference on Location Based Services / Klinger T., Dumm R. E. — 2014 — [313-328 ст.]
7. Python [Електронний ресурс]. — Режим доступу до ресурсу: <https://www.python.org/about/>
8. Java Wikipedia page [Електронний ресурс]. — Режим доступу до ресурсу: [https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))
9. JavaScript Wikipedia page [Електронний ресурс]. — Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/JavaScript>
10. Go [Електронний ресурс]. — Режим доступу до ресурсу: <https://www.python.org/about/>
11. Sebesta R. W. Concepts of Programming Languages / Sebesta R. W. — 11-те вид. — Pearson. — 2015.
12. Django Documentation [Електронний ресурс]. — Режим доступу до ресурсу: <https://docs.djangoproject.com/>
13. Flask Documentation [Електронний ресурс]. — Режим доступу до ресурсу: <https://flask.palletsprojects.com/>

14. Grigore M., Marca A. D., Almasan M. Best Practices in Web Development. Database Systems Journal / M. Grigore, A. D. Marca, M. Almasan — 2016.

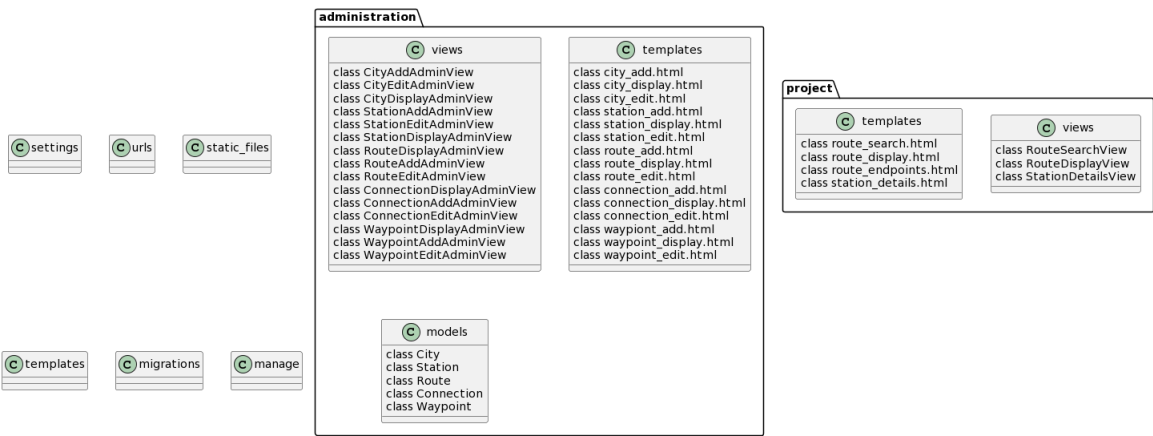
ДОДАТОК 1

Система планування маршрутів подорожі громадським транспортом (веб-застосунок)

Структурна схема системи
ІАЛЦ.123456.789 Д1

Аркушів 1

Київ 2023 р.



ДОДАТОК 2

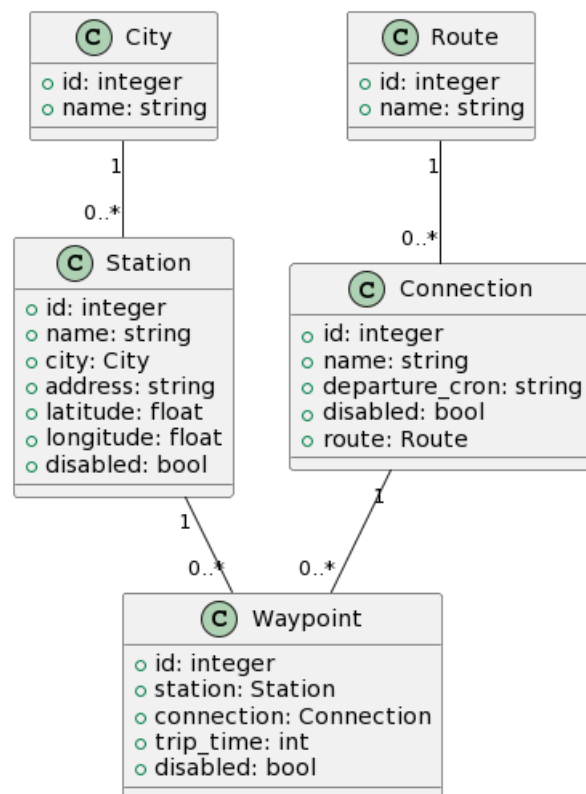
Система планування маршрутів подорожі громадським транспортом (веб-застосунок)

Функціональна схема (діаграма класів)

ІАЛЦ.123456.789 Д2

Аркушів 1

Київ 2023 р.



ДОДАТОК 3

Система планування маршрутів подорожі громадським транспортом (веб-застосунок)

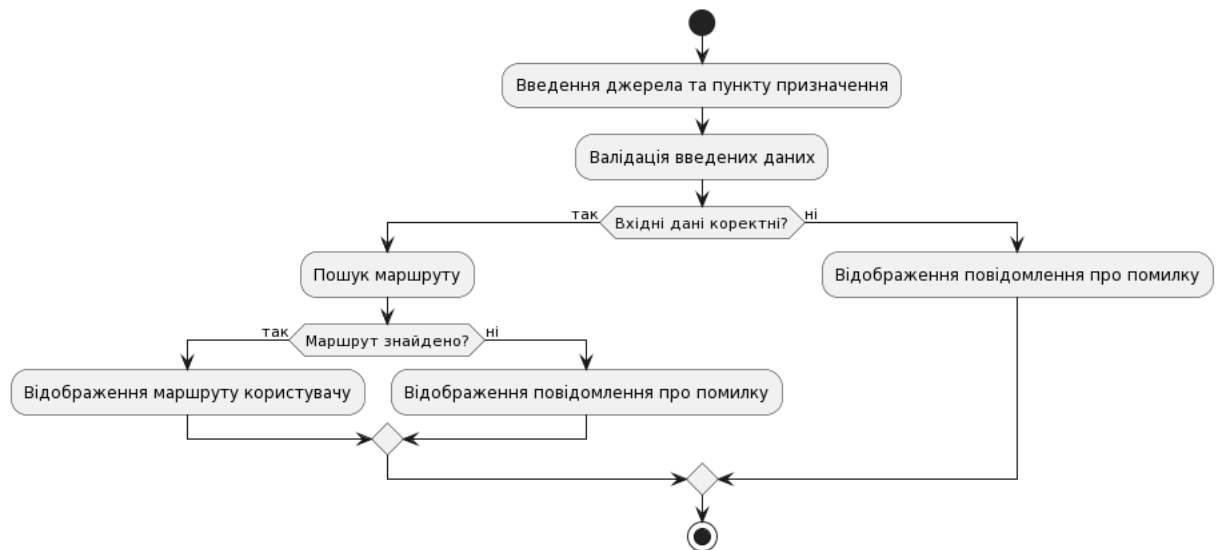
Алгоритм дій програмного забезпечення для пошуку

маршруту

ІАЛЦ.123456.789 ДЗ

Аркушів 1

Київ 2023 р.



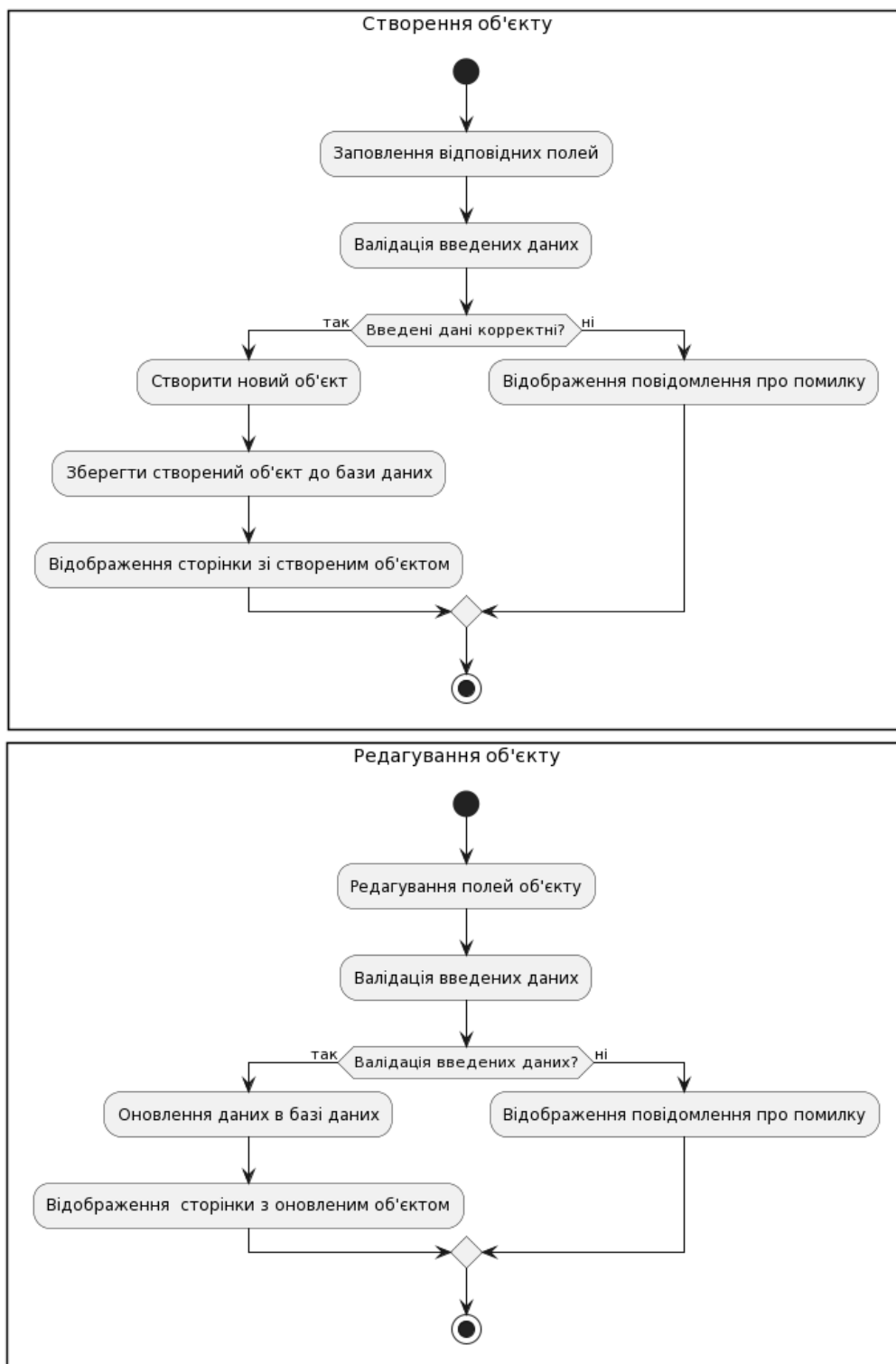
ДОДАТОК 4

Система планування маршрутів подорожі громадським транспортом (веб-застосунок)

**Алгоритм дій програмного забезпечення для керування
транспортною мережею
ІАЛЦ.123456.789 Д4**

Аркушів 1

Київ 2023 р.



ДОДАТОК 5

Система планування маршрутів подорожі громадським транспортом (веб-застосунок)

Текст програмного коду

ІАЛЦ.123456.789 Д5

Аркушів 1

Київ 2023 р.

Лістинг файлу presenter.h — прототип пред'явника-вчителя

```
print("Hello ,□World!")
```