НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Факультет прикладної математики

Кафедра прикладної математики

«До захисту допущено»

Завідувач кафедри

_____ О.Р.Чертов

“___”_____20__ р.

Дипломна робота

на здобуття ступеня бакалавра

зі спеціальності 6.040301 «Прикладна математика»

на тему: Математичне та програмне забезпечення системи пошуку оптимального транспортного маршруту при пересуванні містом

Виконав: студент 4 курсу, групи КМ-12

Алєшко Павло Сергійович _____

Керівник старший викладач Ладогубець Тетяна Сергіївна _____

Консультант з нормоконтролю старший викладач Мальчиков В.В. _____

Рецензент кандидат технічних наук, доцент Петрашенко А.В. _____

Засвідчую, що у цій дипломній роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент _____

Київ – 2015 року

Національний технічний університет України

«Київський політехнічний інститут»

Факультет прикладної математики

Кафедра прикладної математики

Рівень вищої освіти – бакалаврський

Спеціальність 6.040301 «Прикладна математика»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ О.Р.Чертов

«___»_____20__ р.

ЗАВДАННЯ

на дипломну роботу студенту

Алєшку Павлу Сергійовичу

1. Тема роботи «Математичне та програмне забезпечення системи пошуку оптимального транспортного маршруту при пересуванні містом »,

керівник роботи старший викладач Ладогубець Тетяна Сергіївна,

затверджені наказом по університету від «19» травня 2015 р. №1039-С

2. Термін подання студентом роботи «12» червня 2015 р.

3. Вихідні дані до роботи

- Дані про дорожні проблеми на опорних точках
- Карта місцевості

4. Зміст роботи

- провести аналіз існуючих рішень для вирішення поставленої задачі;
- здійснити програмну реалізацію інформаційного забезпечення;

- інфологічне проектування системи;
- даталогічне проектування системи;
- провести тестування розробленого інформаційного забезпечення

5. Перелік ілюстративного матеріалу (із зазначенням плакатів, презентацій тощо)

- діаграма потоків даних нульового рівня;
- діаграма потоків даних першого рівня;
- ERD-модель;
- Результати тестування програмного забезпечення;
- Лістинг програмного забезпечення.

6. Консультанти розділів роботи*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Нормоконтроль	ст. викладач Мальчиков В.В.		

7. Дата видачі завдання 6 жовтня 2014

Календарний план

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за тематикою та збір даних	14.12.2014	
2	Проведення порівняльного аналізу математичних методів пошуку шляху	24.12.2014	
3	Підготовка матеріалів першого розділу дипломної роботи	01.02.2015	
4	Підготовка матеріалів другого розділу дипломної роботи	01.03.2015	
5	Підготовка матеріалів третього розділу дипломної роботи	15.03.2015	

6	Перевірка розробленого програмного забезпечення	20.05.2015	
7	Підготовка матеріалів четвертого та п’ятого розділів дипломної роботи	27.05.2015	
8	Оформлення дипломної роботи	02.06.2015	

Студент _____ Алєшко П.С.

Керівник роботи _____ Ладогубець Т.С.

АНОТАЦІЯ

Дана дипломна робота присвячена розробці математичної моделі для вибору оптимального шляху між двома точками по критеріям часу, використовуючи дані про дорожні ситуації.

В роботі виконано порівняльний аналіз математичних методів вибору шляху. Для розв'язку задачі було обрано метод динамічного програмування, а для графічного представлення сервіс Google Maps.

Розроблено програмне забезпечення, яке використовує вищеописані методи для складання оптимального маршруту. Інформаційне забезпечення реалізовується шляхом використання СКБД. Алгоритм реалізовано засобами мови програмування C# з використанням фреймворку Visual Studio. Для відображення карти було використано бібліотеку GMap.

Робота виконана на 72 аркушах, містить список посилань з 7 найменувань. У роботі наведено 14 рисунків, 1 таблиця, 2 додатки.

ABSTRACT

This degree work is devoted to the development of a mathematical model to select the optimal path between two points following the criteria of the route time, by using data about traffic situation.

In the work the comparative analysis of the mathematical methods of route selection is done. The method of dynamic programming was chosen to solve the task and the service Google Maps was chosen for the graphical representation.

The software which uses the mentioned methods for working out an optimal route was developed. The information management is implemented through using DBMS. The algorithm is implemented by means of the programming language C # using the framework Visual Studio. For the map display the library GMap was used.

This work contains 72 pages, 14 elements illustrated material, 1 table, 2 bibliographical applications and 7 bibliographic items.

ЗМІСТ

СПИСОК ТЕРМІНІВ, УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

DFD (Data Flow Diagram) – модель проектування, графічне представлення «потоків» даних в інформаційній системі.

IDEF (Integrated DEFinition) – методологія моделювання і стандарт документування процесів, що відбуваються в системі.

NP (англ. Complexity class NP) — клас складності, до якого належать задачі, що можна розв'язати недетермінованими алгоритмами за поліноміальний час.

СКБД (Система Керування Базою Даних) – комплекс програм, що забезпечує користувачам можливість створення, збереження, оновлення, пошук інформації та контролю доступу в базах даних.

ВСТУП

В умовах сьогодення проблема розв'язання задачі пошуку оптимальних шляхів є надзвичайно актуальною, особливо враховуючи постійну зайнятість населення та бажання ефективно використовувати свій час. Але це завдання є значно складнішим, ніж задача пошуку найкоротшого шляху на транспортній мережі, оскільки в деяких випадках виникає необхідність зміни маршруту та (або) виду транспорту (тобто, здійснення пересадки). Слід також врахувати, що не завжди безпересадочний шлях є найкоротшим за критерієм тривалості чи відстані, оскільки тривалість пересадки найчастіше складається з тривалості пішого переходу пасажира між зупинками (залежить від просторового розміщення зупинок) та тривалості очікування транспорту, що залежить від інтервалу руху транспорту.

1 ПОСТАНОВКА ЗАДАЧІ

Метою дипломної роботи є розробка программного забезпечення для вибору маршруту за мінімальним часом його подолання використовуючи дані про ситуації на дорогах.

Предметною областю – пошук маршруту перевезення.

Об'єкт дослідження – підприємства, які спеціалізовані на доставці товарів.

Предмет дослідження – пошук маршруту для відділу доставки підприємства, яке є об'єктом дослідження.

Програмне забезпечення повинно реалізувати наступні функції:

а) пошук оптимального шляху за критерієм часу;

б) формування маршруту зображеного на інтерактивній карті.

Дане програмне забезпечення повинно використовувати пошук маршрутів тільки в межах міста.

1.1 Вимоги до Програмного забезпечення

Програмне забезпечення повинно стабільно працювати на персональному комп'ютері з наступними технічними характеристиками:

а) процесор Intel Core 2 Duo 2.1 GHz або вище;

б) операційна система Windows XP/Vista/7/8;

в) 1 Гб оперативної пам'яті;

г) 100 Мб вільного місця на диску;

д) активне підключення до інтернету;

е) клавіатура та миша.

Для роботи з Google Api необхідне постійне підключення до інтернету для виконання запитів, щодо існуючих маршрутів та завантаження карти місцевості.

1.2 Вимоги до Інформаційного забезпечення

Інформаційне забезпечення повинно відповідати наступним вимогам:

а) забезпечення цілісності даних;

б) захист від несанкціонованого доступу;

в) мінімізація введення та виведення інформації.

1.3 Висновки

В даному розділі було визначено проблемну і предметну область роботи, об'єкт та предмет дослідження, сформульовано мету роботи. Висунуто вимоги для програмного та інформаційного забезпечення.

2 ОГЛЯД ІСНУЮЧИХ СИСТЕМ

2.1 Google Maps

Google Maps — набір додатків, побудованих на основі безкоштовного картографічного сервісу і технологій, які надає компанія Google.

Станом на серпень 2013 на Картах Google представлені розклади руху транспорту для більш ніж мільйона зупинок, розташованих у понад 800 найбільших містах світу, в т.ч. ця інформація доступна для 23 міст України, дані надані партнером Google в Україні компанією EasyWay.

Функції:

а) Інформація про розклади та маршрути руху громадського транспорту

б) Пошук маршруту за точками

в) Рівень завантаженості доріг

г) Показується швидкість, очікуваний час прибуття і відстань до місця прибуття.

д) Програма показує схеми будинків та автоматично перемикається із тривимірного режиму у двовимірний. Доступний вид із супутника.

е) Підтримує векторні карти України.

ж) Є пошук потрібних об'єктів поблизу (кінотеатри, ресторани, заправки тощо).

з) Маршрут прокладається через проміжні точки, є голосовий пошук.

и) Має функцію голосових підказок.

к) Інформація про маршрути доступна в усіх браузерах, включно з мобільними.

2.2 Google Street View

Google Street View надає користувачам можливість «поблукати» в тривимірній проекції вулиць через інтернет. Така функціональність стала можлива за допомоги кругового фотографування місцевості спеціальним обладнанням в режимі реального часу. В підсумку створюється багато сферичних панорам з прив'язкою до географічних координат та надається можливість перемикатись між ними, використовуючи для цього інтуїтивно зрозумілий інтерфейс, створюючи відчуття пересування в просторі. Є можливість розпізнавати пласкі поверхні на фотографії, такі як фасади будівель, також присутня можливість отримання найвдалішого ракурсу для перегляду вибраного виду.

Існує можливість використовувати сервіс для створення своїх продуктів сторонніми компаніями. На сьогоднішній день це безкоштовна служба, але можливість додати рекламу залишена на майбутнє.

Для розробників сайтів зручно буде використати JavaScript для керування функціональністю карт, правда кількість запитів з одного сервера обмежена. Google Static Maps API дозволяє будувати статичні мапи за допомогою спеціальних url'ів . Також існують версії додатків під різні види мобільних пристроїв.

Функції:

а) Фотографування місцевості в режимі реального часу

б) Розпізнавання пласких поверхонь будівель

в) Програма показує схеми будинків та автоматично перемикається із тривимірного режиму у двовимірний.

г) Доступний вид із супутника.

2.3 Яндекс Карти

В Україні запущений у 2006 році, містить карти 204 українських міст.Користувачі можуть переглядати карту в будь-якому з трьох відображень: схему, супутниковий знімок і гібрид (поєднану).

За допомогою сервісу можна шукати за адресами, вулицями міст, регіонами, країнами та організаціями, вимірювати відстані між географічним об'єктами та прокладати автомобільні маршрути. Для певних міст доступний індикатор ситуації на дорогах — сервіс Яндекс.Затори. Рівень завантаженості доріг подано як чотириколірну графічну та десятибальну цифрову шкалу. Дані Яндекс заторів можуть враховуватися під час автоматичного прокладання маршрутів. Інформація про дорожні події, що отримується з мобільних Яндекс карт, також впливає на рекомендований маршрут. На картах є інтерактивні схеми транспортних розв'язок з рекомендаціями, як оптимально ними проїхати. На сервісі можна вреальному часі переглядати зображення з веб-камер.

Функції:

а) Показується швидкість, очікуваний час прибуття і відстань до місця прибуття.

б) Програма показує схеми будинків та автоматично перемикається із тривимірного режиму у двовимірний. Доступний вид із супутника.

в) Підтримує векторні карти України.

г) Є пошук потрібних об'єктів поблизу (кінотеатри, ресторани, заправки тощо).

д) Доступна навігація усіма містами України і трасами між Україною, Росією та Білоруссю.

е) Маршрут прокладається через проміжні точки, є голосовий пошук.

ж) Має функцію голосових підказок.

з) У програмі також є розпізнавання голосових команд.

и) Україномовний інтерфейс.

к) Доступна для смартфонів та планшетів на платформах iOS, Android, Windows Phone.

2.4 Висновки

На ринку є програмні продукти , які працюють з картами та обирають кращі маршрути. Проте ці севріси не моделюють ситуацію, а тільки показують найшкоротші або найдешевші маршрути, і тому вони не можуть змоделювати ситуацію в якій ризик потрапити в затор якомога менший.

3 ОПИС ПРЕДМЕТНОЇ ОБЛАСТІ ТА АНАЛІЗ МЕТОДІВ РІШЕННЯ ЗАДАЧ

Завдання даної роботи це створення програмного забезпечення для використання підприємствами, які займаються доставкою товарів, що швидко псуються. Завдання даного програмного забезпечення – вибір найшвидшого маршруту використовуючи дані про ситуації на дорогах в момент запиту маршруту.Побудова маршруту відбувається наступним чином:

а) Запит до Google Api щодо опорних точок, між точками початку та кінця маршруту;

б) Запит до Google Api щодо дорожних ситуацій між опорними точками та формування матриці часу(ваги);

в) Побудова маршруту використовуючи дані опорних точок, опорних проміжків та матриці часу(ваги).

В загальному вигляді задача формулюється наступним чином. Існує деяка кількість опорних точок, які з'єднані між собою. Кожний зв'язок має свою вагу – час переходу між точками. Транспортна система в задачі є орієнтовним графом, де N_1 –вхід, а N_n – вихід. Необхідно визначити найшвидший шлях з N_1 в N_n. Вагові коефіцієнти c_{ij} ребер x_{ij} являються часом переходу між опорними точками i та j .Зіставимо кожному ребру графа булеву змінну, тобто $x_{ij} \in \{0;1\}$. Якщо ребро входить у маршрут $x_{ij} = 1$, інакше – $x_{ij} = 0$. Тоді задачу пошуку оптимального шляху перевезення математично можна записати так:

1. Цільова функція:

$$\min Z = \sum_i \sum_j x_{ij} \cdot c_{ij} \tag{3.1}$$

2. Обмеження:

Для перерахунку усіх опорних проміжків k, які переходять у опорну точку i:

$$\sum_k x_{ki} = 1 \ , i = 2..n \tag{3.2}$$

Для перерахунку усіх опорних проміжків j , які переходять з опорної точки i:

$$\sum_j x_{ij} = 1 \ \ , i = 1..n-1 \tag{3.3}$$

$$x_{ij} \in \{0;1\} \tag{3.4}$$

Якщо опорна точка i не входить в визначений маршрут, то для будь-якої опорної точки, крім початкової та кінцевої, має виконуватись умова :

$$\sum_k x_{ki} - \sum_j x_{ij} = 0 \tag{3.5}$$

Отже з (3.1) – (3.5) формул можна зробити висновок, що математичною моделлю задачі пошуку оптимального шляху перевезення є задача цілочисельного лінійного програмування, яку ще називають задачею булевого програмування.

Існує багато способів вирішення даної задачі. Ці методи можна розділити на дві підгрупи:

а) Евристичні методи

б) Точні методи

До Евристичних методів відносять:

а) Жадібний алгоритм

б) Генетичний алгоритм

в) Метод вектора спаду

До точних методів відносять:

а) Метод повного перебору

б) Метод гілок та меж

в) Адитивний алгоритм Балаша

г) Динамічне програмування

3.1 Огляд евристичних методів вирішення задачі

3.1.1 Жадібний алгоритм

Жадібний алгоритм – простий евристичний алгоритм, який приймає найкраще рішення, виходячи з наявних на поточному етапі даних, не турбуючись про можливі наслідки, сподіваючись врешті-решт отримати оптимальне рішення [1]. Складність алгоритму $O(n \log n + n)$

Переваги:

а) простота реалізації;

б) на кожному кроці зведення великої задачі до малої;

в) висока швидкість обчислення.

Недоліки:

а) для отримання оптимального рішення необхідна структура, в якій кожен наступний крок веде до оптимального розв’язку.

3.1.2 Генетичний алгоритм

Генетичний алгоритм — це еволюційний алгоритм пошуку, що використовується для вирішення задач оптимізації і моделювання шляхом послідовного підбору, комбінування і варіації шуканих параметрів з використанням механізмів, що нагадують біологічну еволюцію.

Особливістю генетичного алгоритму є акцент на використання оператора "схрещення", який виконує операцію рекомбінацію рішень-кандидатів, роль якої аналогічна ролі схрещення в живій природі [2].

Переваги:

а) можливість пристосування для кожної функції окремо;

б) вибір довільної кількості хромосом для оптимізації по багатьом критеріям.

Недоліки:

а) складність підбору параметрів;

б) складність перевірки оптимальності результату.

3.1.3 Метод вектора спаду

Метод вектора спаду це евристичний метод пошуку. Обирається початковий допустимий план задачі. З таким можливим варіантом розв’язку пов’язується певна множина варіантів, які утворюють окіл початкового варіанта. Перебираючи елементи околу, здійснюється перехід до кращого плану або фіксується локальна оптимальність певного варіанта. В першому випадку процес локального покращання плану

продовжується, а в другому здійснюється випадковий вибір наступного плану, який беруть за початок пошуку нового оптимального плану[3].

Переваги:

а) Швидкодія методу

Недоліки:

а) Необхідні початкові обмеження типу нерівності

б) Від початкового вибору плану задачі залежить швидкість отримання оптимального результату.

3.2 Огляд точних методів вирішення задачі

3.2.1 Метод повного перебору

Щоб отримати рішення потрібно перебрати усі можливі варіанти. Складність повного перебору залежить від кількості всіх можливих рішень задачі,[3].

Переваги:

а) простота реалізації.

Недоліки:

а) дуже великий об’єм обчислень.

3.2.2 Метод гілок та меж

Метод гілок і меж — один з поширених методів дискретної оптимізації. Метод працює на дереві рішень та визначає принципи роботи конкретних алгоритмів пошуку розв'язків, тобто, є мета-алгоритмом.

По суті, метод є варіацією повного перебору з відсівом підмножин допустимих рішень, які заздалегідь не містять оптимальних рішень[3-6].

Переваги:

а) точний метод

б) відкидання неактивних гілок, тобто які заздалегідь не містять оптимального результату.

Недоліки:

а) не можна оцінити кількість задач, які потрібно вирішити.

б) при примусовій зупинці процесу рішення висока ймовірність отримання цілочисельного результату, але без встановлення його оптимальності.

в) велика кількість обчислень

3.2.3 Адитивний алгоритм Балаша

Адитивний алгоритм Балаша – метод неявного перебору. Для розв’язування задач такого типу Е. Балаш запропонував алгоритм часткового перебору, який отримав назву адитивного, оскільки завдяки умові (3.4) для обчислень за алгоритмом використовуються лише операції додавання. Формально процес пошуку оптимального розв’язку за адитивним алгоритмом може бути представлений у вигляді побудови деякого дерева варіантів, де кожна вершина (не кінцева) відповідає певному частковому розв’язку, а можливі його доповнення породжують

гілки цього дерева. Крім того, кожному частковому розв'язку відповідає деяка підзадача, що належить до списку задач, який називається основним.

Існує певний зв'язок між основною ідеєю адитивного алгоритму і принципом динамічного програмування [3-6].

Переваги:

а) висока швидкість роботи;

б) «зондування» кожного частинного розв'язку, що призводить до відкидання великої кількості варіантів.

Недоліки:

а) використання великої кількості оперативної пам'яті.

3.2.4 Динамічне програмування

Метод динамічного програмування – рекурсивний спосіб вирішення складних завдань шляхом розбиття їх на більш прості підзадачі. Він застосовується до завдань з оптимальною підструктури, Що виглядає як набір перекриваються підзадач, складність яких трохи менше вихідної Динамічне програмування зв'язане з багатокроковим процесом прийняття рішення[7].

Під цим багатокроковим процесом прийняття рішення розуміється діяльність, при якій приймаються послідовні рішення, які направлені на досягнення однієї цілі.

Переваги:

а) висока швидкість роботи;

б) простота реалізації.

Недоліки:

а) використання великої кількості оперативної пам'яті.

3.3 Висновки

Проведено огляд існуючих методів пошуку маршрутів. Проаналізовано методи вирішення задач і було вирішено використовувати метод динамічного програмування для задачі пошуку мінімального за часом шляху.

4 ЗАДАЧА ДИНАМІЧНОГО ПРОГРАМУВАННЯ

Динамічне програмування - один із потужних методів вирішення задач математичного програмування,розроблений американським математиком Р. Беллманом в кінці 50-х років. він спочатку орієнтований на оптимізацію так званих багатокрокових(багатостадійних, багатоетапних) процесів і використання ЕОМ.

Фундаментальною основою методу динамічного програмування є сформульований Беллманом принцип оптимальності,згідно з яким оптимальне управління визначається кінцевою метою управління і станом системи в розглянутий момент, незалежно від того, яким чином вона прийшла в цей стан. Інакше кажучи, при фіксованому стані системи подальше оптимальне рішення не залежить від її попереднього стану.

Принцип оптимальності Беллмана. Оптимальна поведінка має властивість, що які б не були початковий стан і рішення в початковий момент, наступні рішення повинніскладати оптимальну поведінку щодо стану, отриманого в результаті першого рішення.

4.1 Опис методу

Задана орієнтована мережа, що містить N точок (вузлів). Знайти найшвидший шлях з точки 1 в точку N , якщо задана матриця

$C = \{c_{ij}\}$ –Матриця часу(вартості) переходу між ребрами. Позначимо через W_j^* – мінімальний час з точки і, в точку N. Оптимальний маршрут з будь-якої точки, повинен володіти тим властивістю, що яким би не був

спосіб досягнення пункту і, подальше рішення повинне бути оптимальним для частини шляху, який починається в точці і (принцип оптимальності).

Нехай з точки і можемо перейти в току j, відстань між цими точками дорівнює c_{ij}. Точка j повинна вибратися таким чином, щоб шлях з j в N був частиною оптимального з 1 в N. Позначимо мінімальний шлях із j в N через W_j^*. Тоді і вибирається з умови мінімізації суми:

$$c_{ij} + W_j^* \tag{4.1}$$

Таким чином отримуємо рівняння Беллмана:

$$W_i^* = \min\{c_{ij} + W_j^*\} \tag{4.2}$$

Розділимо умовно всі точки мережі на n множин за кількістю кроків 1, 2, ...,n .

$$W_k^*(i) = \min\{c_{ij} + W_{k+1}^*(j)\} \tag{4.3}$$

Вектор c_{n-1} складається з n точок , з яких можна потрапити в N не більше ніж за один крок, тому

$$W_k^*(i) = \min\{c_{in}\} = c_{in}, \quad x_{in} = 1 \tag{4.4}$$

де x_{in} – умовне оптимальне рішення на n-му переході з точки і в N по найшвидшому шляху. У підсумку умовної оптимізації отримаємо сукупність рішень з матриці x_{ij}, використовуючи умову переходу в точку $x_{ij} = 1$ визначимо точки, відповідні оптимальним маршрутом.

4.2 Висновки

Було побудовано та описано математичну модель задачі динамічного програмування.

5 ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

5.1 Категорії користувачів

Для роботи з програмою користувач має авторизуватись за допомогою комбінації логіну та паролю.

Для забезпечення захисту інформації користувачі даного програмного забезпечення розділяються на дві групи:

а) звичайні працівники;

б) адміністратор відділу.

Адміністратори відділів мають повний доступ до даних проекту. Вони можуть виконувати наступні дії:

а) створювати або редагувати точки;

б) видаляти точки.

Звичайні працівники не мають доступу до зміни даних. Вони можуть лише обирати доступні точки та зробити запит на створення маршруту між ними(даний маршрут відображається на карті) .

5.2 Класи даних

Для повноцінного функціонування програмного забезпечення необхідні наступні вхідні дані:

1) Дані про точку маршруту:

а) Назва міста;

б) Назва вулиці;

в) Номер будинку та його географічні координати.

2) Дані про проміжки маршруту:

а) Завантаженість проміжків.

б) Можливість використання проміжку для проїзду з обох сторін.

5.3 Матриця елементарних подій

Матриця елементарних подій, яка відображає реакцію програмного забезпечення на різні дії користувача зображена в табл. 4.1

Таблиця 4.1 – Матриця елементарних подій

Опис події	Тип події	Реакція на подію
Користувач запускає програму	N	Надати форму авторизації
Користувач бажає реєструватись	N	Надати форму для реєстрації нового користувача
Користувач авторизується у програмі	N	Перевірка достовірності комбінації логіну та паролю у БД
Немає доступу до Інтернету	NN	Видати попередження про відсутність з'єднання
Користувач ввів існуючу комбінацію логіну та паролю	N	Запит до БД про тип користувача(користувач/адміністратор) та надання необхідної форми.

Продовження таблиці 4.1

Опис події	Тип події	Реакція на подію
Користувач ввів неіснуючу комбінацію логіну та паролю	N	Виведення помилки про неіснуючі дані та повернення до форми авторизації
Користувач натиснув кнопку «Look for routes»	N	Завантаження даних про пробки.Відображення у новій формі карти з зображеній на ній оптимального маршруту

5.4 Інфологічне проектування

Для побудови інфологічної моделі даних було використано модель типу «Сутність-зв'язок», яка складається з трьох конструктивних елементів: сутність, атрибут, зв'язок.

Сутності, а також їх атрибути, необхідні для виконання пошуку маршруту:

1. User – акаунт користувача

а) User_Id – унікальний ідентифікатор користувача

б) User_Login – логін користувача

в) User_Hash_password – пароль користувача

г) User_First_name – Ім'я користувача

д) User_Second_name – Прізвище користувача

2. House – дані про будинок

а) House_Id – Унікальний ідентифікатор будинку

б) House_Latitude – Координата широти будинку

в) House_Longitude – Координата довготи будинку

г) House_Number – Номер будинку

3. Street –дані про вулицю

а) Street _Name – Назва вулиці

4. City – дані про місто

а) City _Name – Назва міста

5. Interval – дані про точку

а) Interval_Id – унікальний ідентифікатор точки

б) House_Id – Унікальний ідентифікатор будинку

в) Street _Name – Назва вулиці

г) City _Name – Назва міста

6. User_point – дані про точку зі збереженням даних про користувача, який її застосував для пошуку маршруту.

а) Interval_Id – унікальний ідентифікатор точки

б) User_Id – унікальний ідентифікатор користувача

в) User_Point_Id – унікальний ідентифікатор точки користувача

7. Problem – дані про пробки

а) Problem_level – рівень затору

б) Problem_date_and_time –час та дата затору

в) Problem_Id – унікальний ідентифікатор затору

8. Problem_interval – дані про проміжок та його пробки

а) Interval_Id – унікальний ідентифікатор проміжку

б) Problem_Id – унікальний ідентифікатор затору

в) Problem_interval_distance–відстань інтервалу

г) Problem_interval_Id– унікальний проміжку з його пробками

9. Route – дані про маршрут

а) Problem_interval_Id– унікальний проміжку з його пробками

б) User_Point_Id – унікальний ідентифікатор точки користувача

в) Route_Id – унікальний ідентифікатор маршруту

г) Route_Distance – відстань маршруту

д) Route_Expected_time – час проходу маршруту

Результуючу концептуальну модель можна побачити на рис. 5.1.

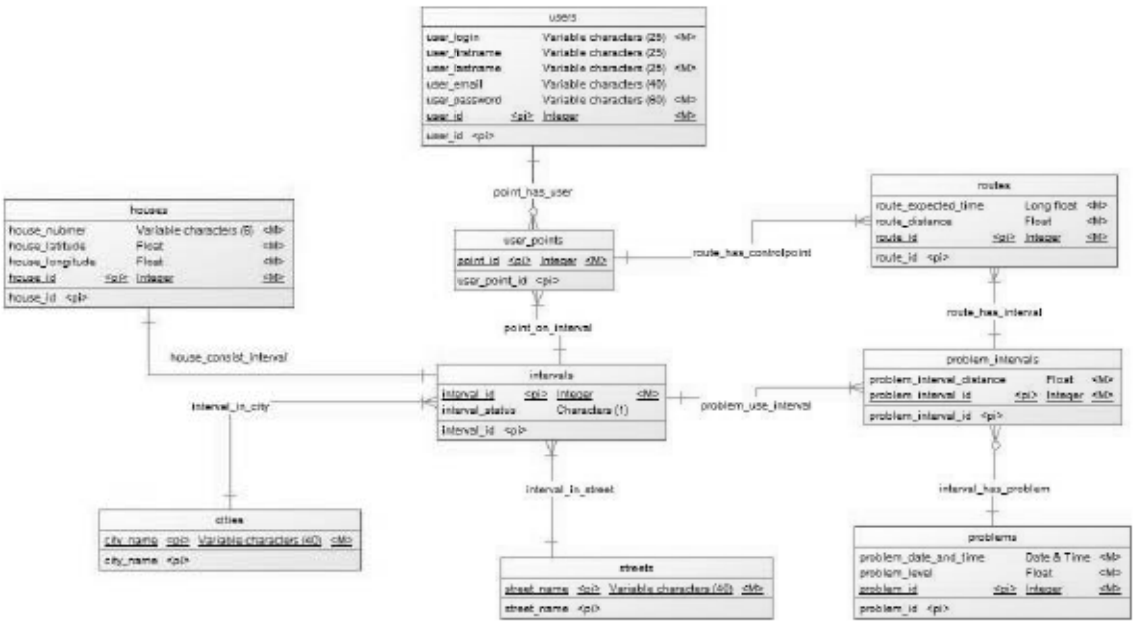

Рисунок 5.1 – Концептуальна модель сутність-зв’язок(ERD)

5.5 Моделювання бізнес процесів

Моделювання бізнес процесів передбачає виділення головних процесів системи, а також даних, необхідних для функціонування цих процесів. Взаємодію процесів з даними зображуються за допомогою DFD. Послідовність виконання процесів зображується за допомогою IDEF3. Основний процес системи – планування розкладу робіт на підприємстві. Дане планування виконується в межах одного проекту. З зовнішнього середовища процес отримує наступну інформацію:

а) дані про початкову та кінцеві точки;

б) карту місцевості;

в) дані про пробки.

Узагальнена схема потоку даних представлена DFD0 і зображена на рис. 4.2

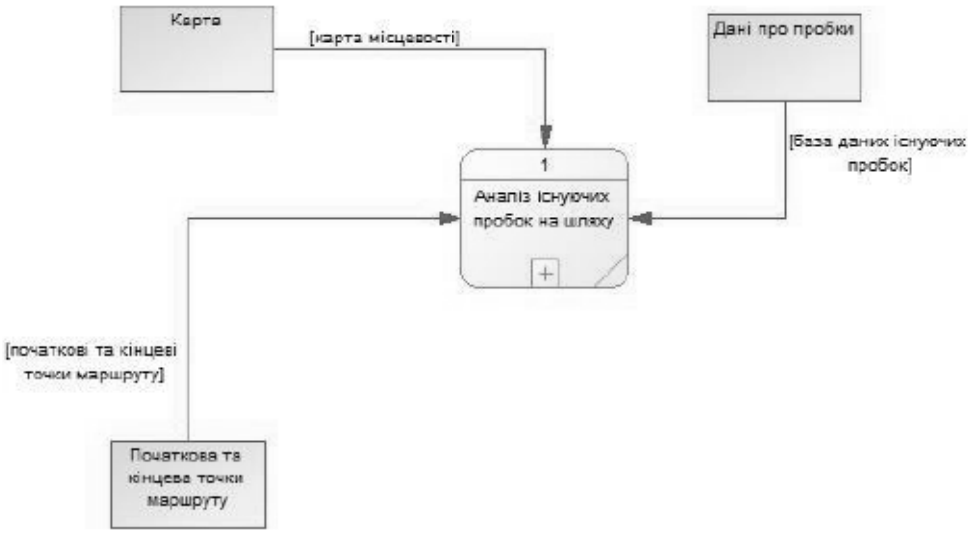

Рисунок 5.2 DFD0- Аналіз існуючих пробок на шляху

Для детальнішого опису процесу потрібно виконати декомпозицію процесу. Планування розкладу включає в себе наступні підпроцеси:

а) робота з Google Api;

б) збереження початкових даних у необхідному форматі;

в) вибір маршруту;

г) відображення результату.

Результат декомпозиції представлений DFD1 і зображений на рис. 5.3

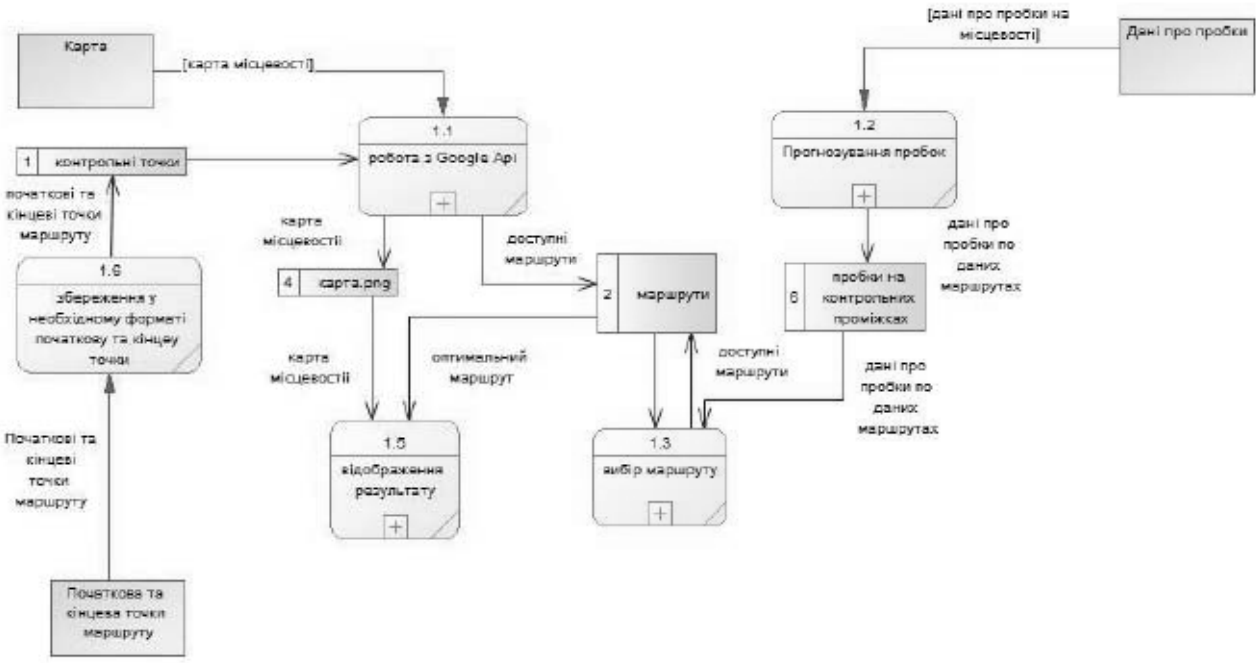

Рисунок 5.3 DFD1 Аналіз існуючих пробок на шляху

Послідовність виконання вибору маршруту зображена на рисунку 5.4 у вигляді діаграми IDEF3.

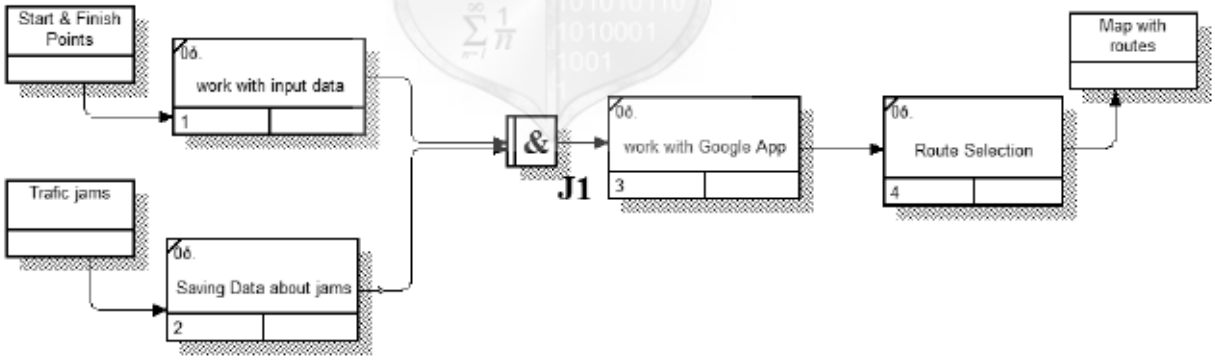

Рисунок 5.4 – IDEF3 процесу вибору маршруту

Опис процесів:

a) «Work with input data»;

Основна операція –Збереження вхідних даних у БД

б) «User registration»;

Основна операція – призначення дати ремонту.

в) «Choise points»;

Основна операція –вибір користувачем точок початку та кінця маршруту.

г) «Display routes»;

Основна операція –отримання маршрутів.

5.6 Даталогічне проектування

На базі концептуальної моделі, описаної в розділі 5.4 , можна створити логічну модель. Для цього потрібно:

а) для кожного атрибуту визначити тип;

б) для кожної сутності визначити первинні ключі (PK);

в) поєднати таблиці використовуючи зовнішні ключі (FK);

г) зв'язки типу «багато до багатьох» замінити допоміжними таблицями.

Для зручності програмної реалізації в кожній сутності було введено штучні первинні ключи цілочисельного типу. Два зв'язки типу «багато до багатьох» було замінено на допоміжні таблиці. Таблиці були поєднані за допомогою зовнішніх ключів і визначені типи кожного атрибуту. Результат виконання даних операцій можна побачити на рис. 5.5 Наступний крок – перехід до фізичної моделі даних. У фізичній моделі передбачається використання конкретної бази даних. В даній роботі використовується СКБД Oracle 10g. Архітектура програмного забезпечення «клієнт-серверна». Частина логіки розміщена на клієнті,

частина на сервері. Результуючу фізичну модель бази даних можна побачити на рис. 5.6

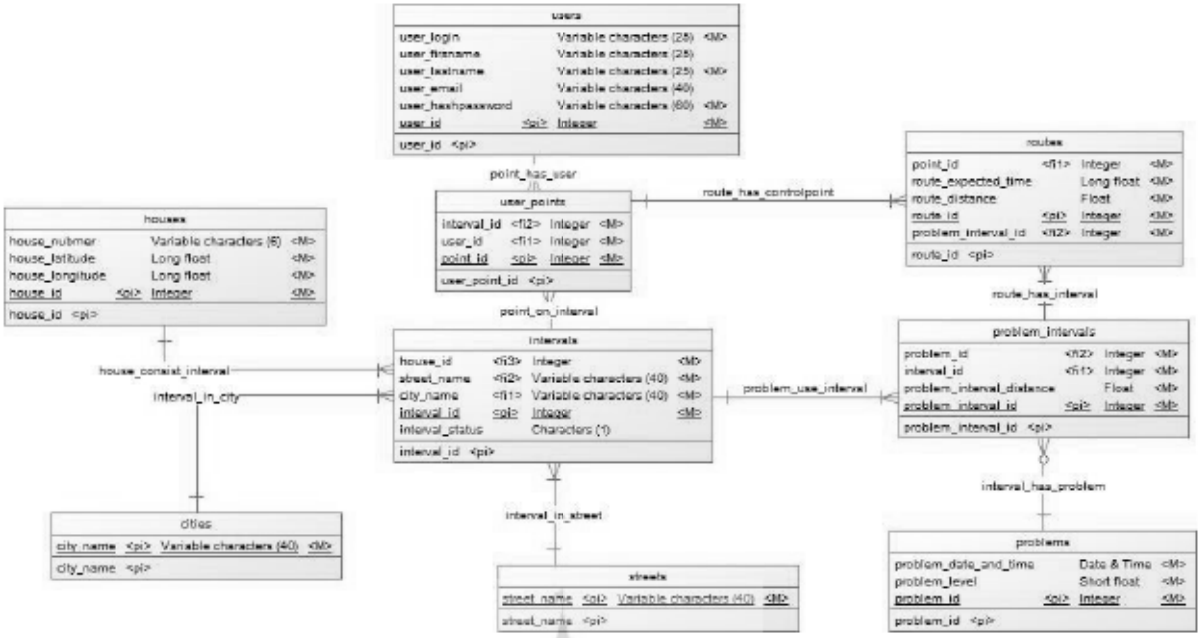

Рисунок 5.5 – Логічна модель

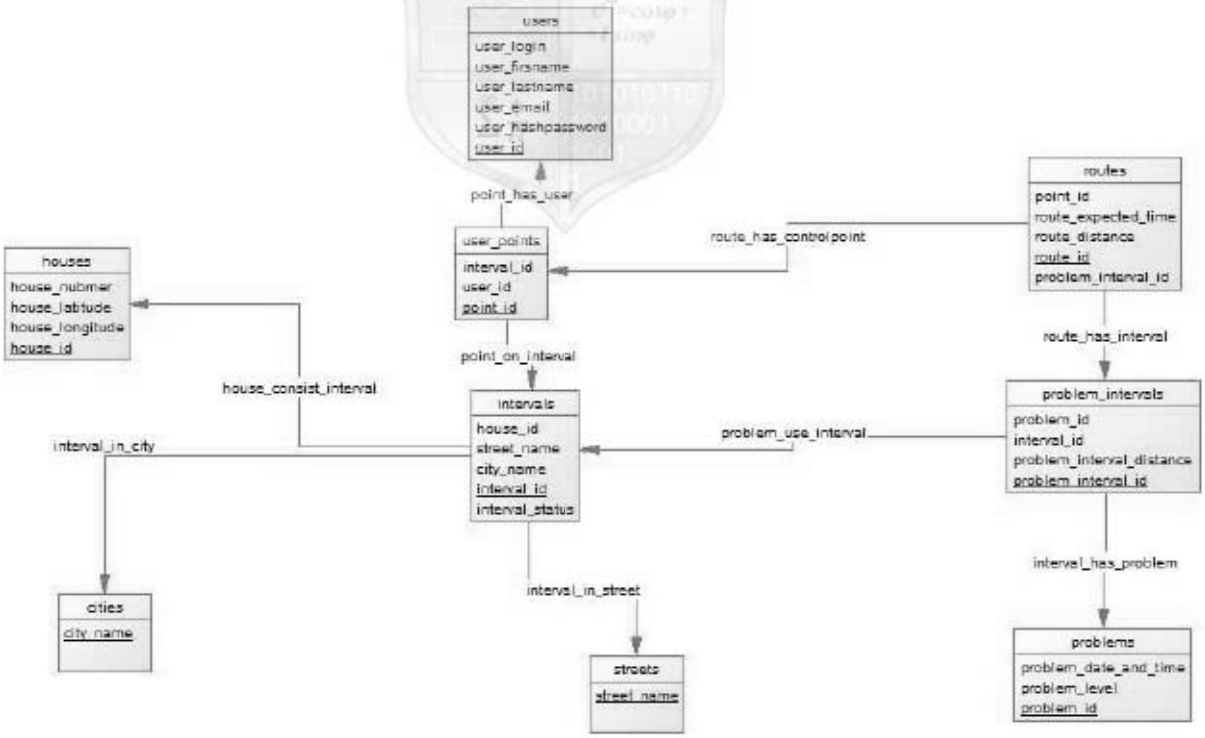

Рисунок 5.6 – Фізична модель

5.7 Реалізація Транзакцій

В даній програмі всі дії з базою даних, які використовують транзакції, можна розділити на наступні групи:

Послідовні SELECT-запити (процедура getusertype). В даному випадку використовується рівень ізоляції read comited, оскільки необхідно кількома запитами зчитати інформацію, не змінюючи її. А для такого рівня досить лише рівня ізоляції read comited.

Операції створення/редагування користувача (процедури create_city , create_street , registration, updatehouse, deletehouse). Тут йде послідовність рівень ізоляції read comited Операції редагування таблиць, які виконує адміністратор ІС. Використовуються операції commit() та rollback() при зміні даних таблиць.

Операції створення з використанням зв'язків (create_interval) використовують найвищий рівень ізоляції Serialazible. В них також використовуються методи commit() та rollback().

5.8 Висновки

Виконано інфологічне та даталогічне проектування. Розроблено базу даних, за допомогою СКБД Oracle 10g. Програмне забезпечення написанена мові C# з використанням фреймворку Visual Studio.

6 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

Розроблене програмне забезпечення було протестовано з декількома точками. Запити на пошук маршрутів виконано в різний час, для наглядного прикладу, як програмне забезпечення обирає маршрут відповідно до завантаженості доріг.

6.1 маршрут №1

Початкова точка : вул. Федори Пушиної ,19.

Кінцева точка : вул.Осіння ,115.

Географічні координати точки вул.Федори Пушиної ,19 – (50,45729;30,36528).

Географічні координати точки вул.Осіння ,115 – (50,44946;30,45045)

На рис 6.1 Зображено графічний результат роботи програмного забезпечення маршруту для маршруту №1. Запит щодо маршруту виконано о 19:00. Відстань маршруту 4,06 км. Час на подолання даного маршруту 8 хв.

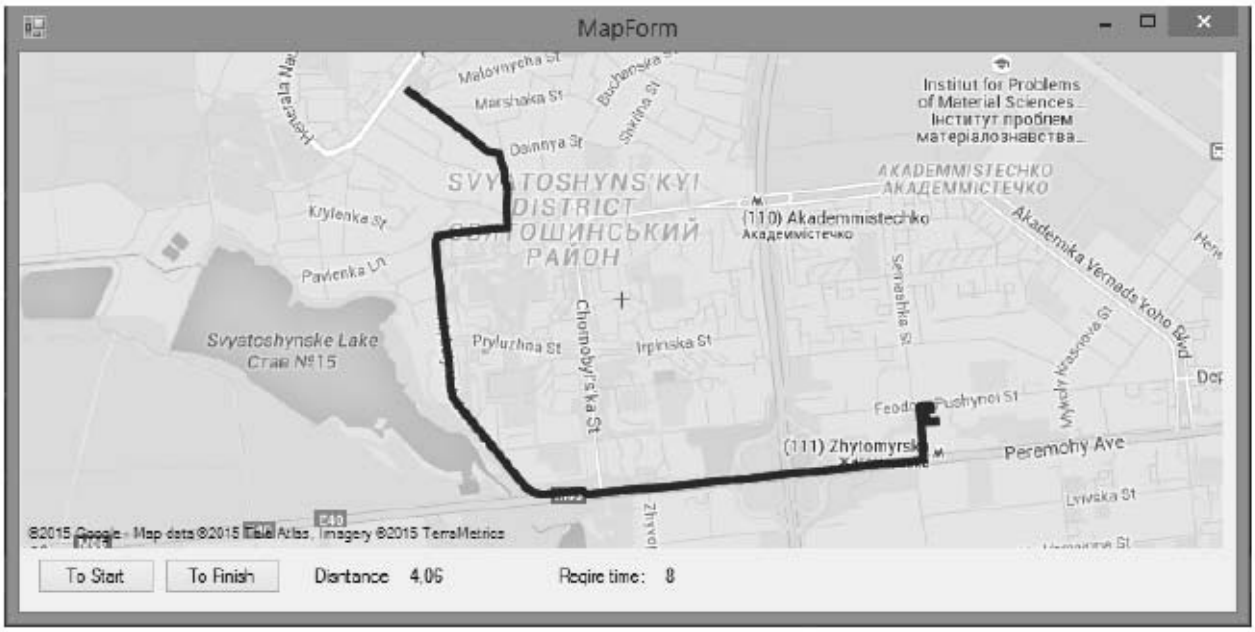

Рисунок 6.1 Графічне зображення маршруту №1, виконаного програмним забезпеченням о 19:00.

На рис 6.2 Зображено графічний результат роботи сервісу Google Maps для маршруту №1. Запит щодо маршруту виконано о 19:00. Відстань маршруту 4,0 км. Час на подолання даного маршруту – 8 хв.

Рисунок 6.2 Графічне зображення маршруту №1 за допомогою Google Maps, виконаного о 19:00

На рис 6.3 Зображено графічний результат роботи програмного забезпечення маршруту для маршруту №1. Запит щодо маршруту виконано о 23:00. Відстань маршруту – 3,8 км. Час на подолання даного маршруту – 7 хв.

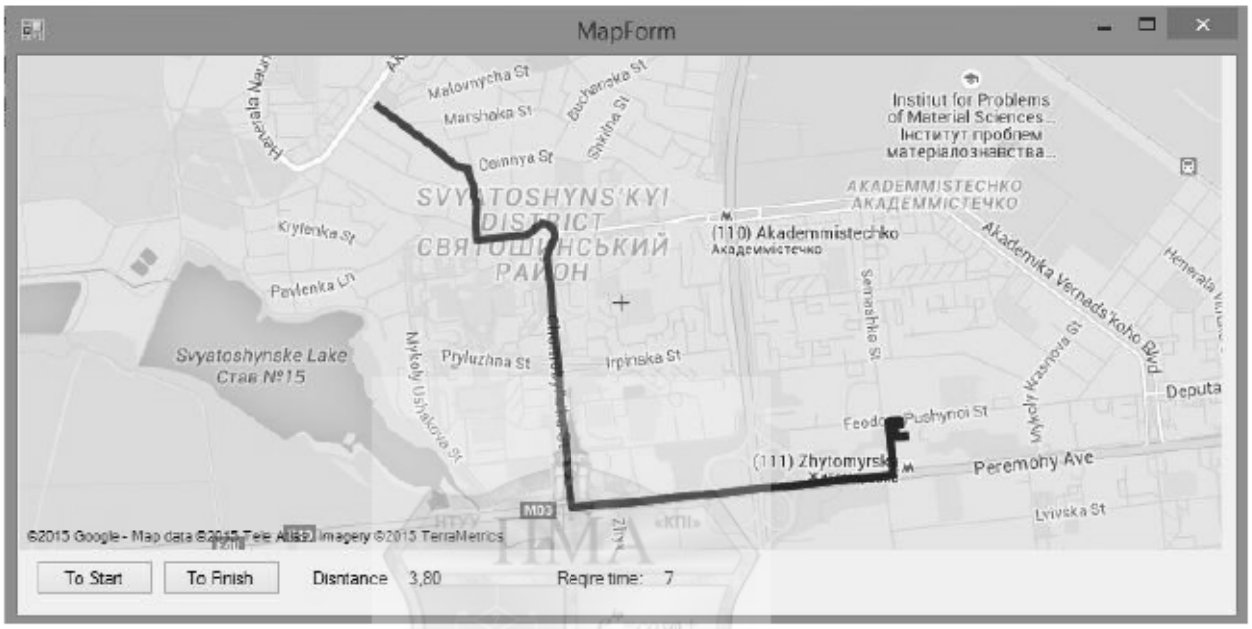

Рисунок 6.3 Графічне зображення маршруту №1, виконаного програмним забезпеченням о 23:00.

На рис 6.4 Зображено графічний результат роботи сервісу Google Maps для маршруту №1. Запит щодо маршруту виконано о 19:00. Відстань маршруту – 3,8 км. Час на подолання даного маршруту – 7 хв.

Рисунок 6.4 Графічне зображення маршруту №1 за допомогою Google Maps, виконаного о 19:00

6.2 маршрут №2

Початкова точка : Артилерійський пров., 13,

Кінцева точка : вул. Вільгельма Піка, 24.

Географічні координати точки Артилерійський пров., 13,– (50,46072;30,42672).

Географічні координати точки вул. Вільгельма Піка, 24– (50,46981; 30,41385).

На рис 6.5 Зображено графічний результат роботи програмного забезпечення маршруту для маршруту №2. Запит щодо маршруту виконано о 06:00. Відстань маршруту 3,33 км. Час на подолання даного маршруту 7 хв.

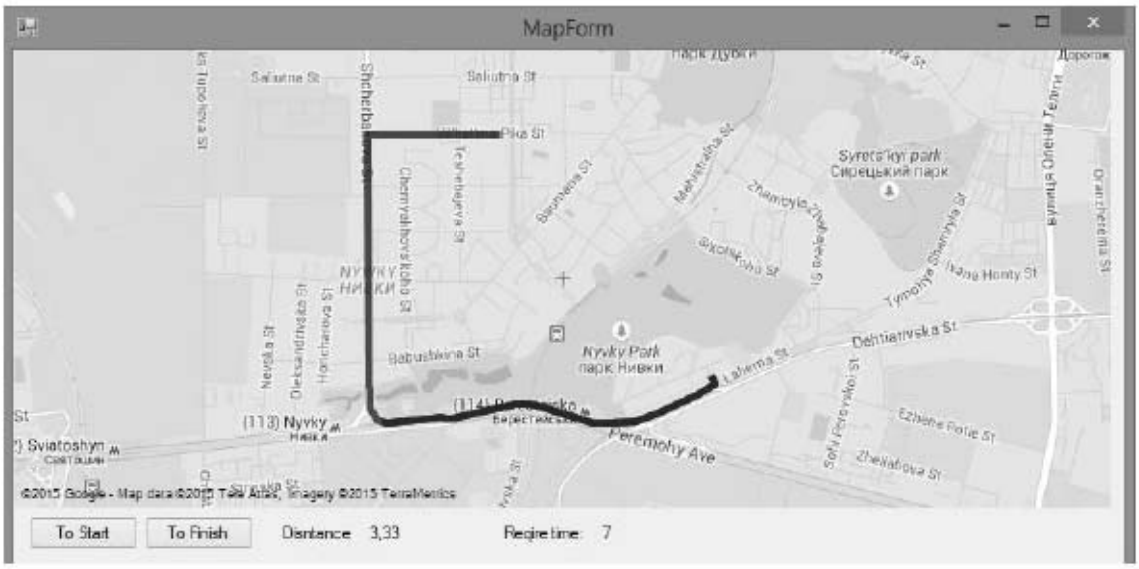

Рисунок 6.5 Графічне зображення маршруту №2, виконаного програмним забезпеченням о 06:00.

На рис 6.6 Зображено графічний результат роботи сервісу Google Maps для маршруту №2. Запит щодо маршруту виконано о 06:00. Відстань маршруту – 3,3 км. Час на подолання даного маршруту – 7 хв.

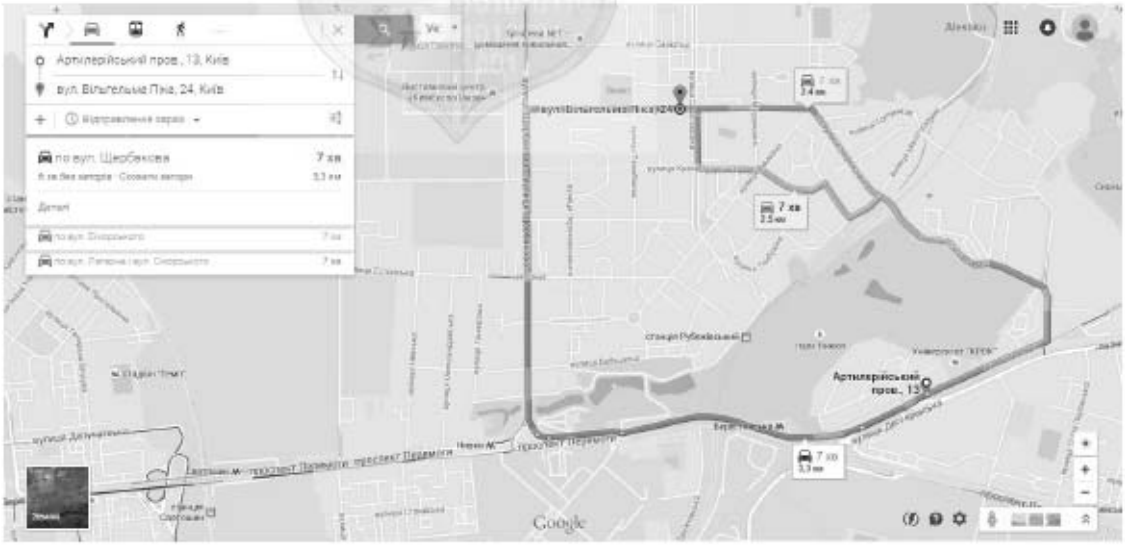

Рисунок 6.6 Графічне зображення маршруту №2 за допомогою Google Maps, виконаного о 06:00

На рис 6.7 Зображено графічний результат роботи програмного забезпечення маршруту для маршруту №2. Запит щодо маршруту виконано о 08:00. Відстань маршруту 2,35 км. Час на подолання даного маршруту 7 хв.

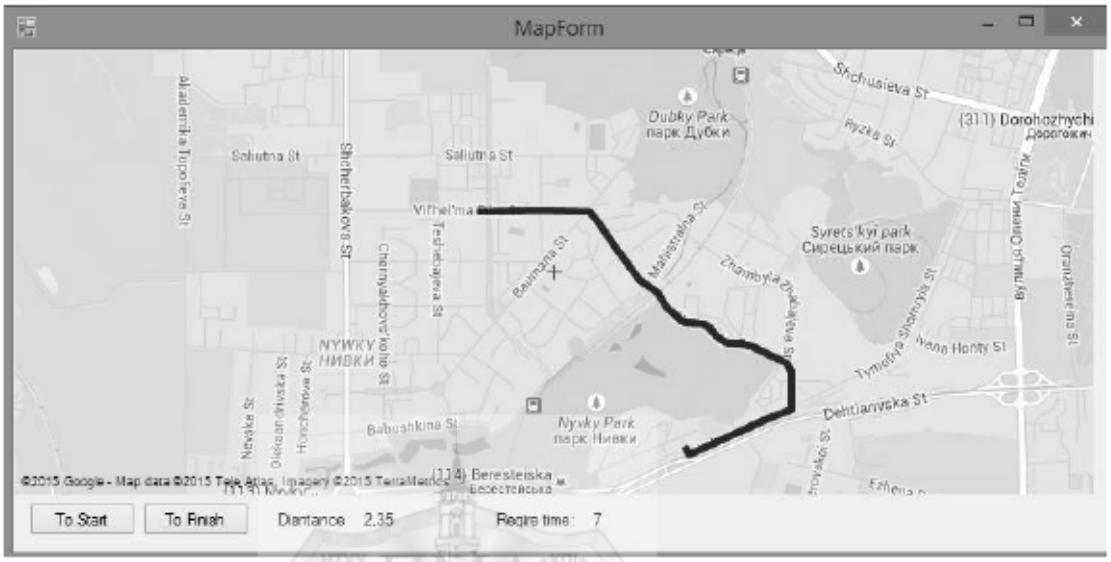

Рисунок 6.7 Графічне зображення маршруту №2, виконаного програмним забезпеченням о 06:00.

На рис 6.8 Зображено графічний результат роботи сервісу Google Maps для маршруту №2. Запит щодо маршруту виконано о 19:00. Відстань маршруту – 2,4 км. Час на подолання даного маршруту – 7 хв.

Рисунок 6.8 Графічне зображення маршруту №2 за допомогою Google Maps, виконаного о 08:00

6.3 Висновки

Проведено тестування програмного забезпечення. Програмне забезпечення успішно пройшло тестування. Показано залежність вибору маршруту від ситуацій на дорогах.

ВИСНОВКИ

В ході дипломної роботи, згідно поставленої задачі, було реалізовано програмне забезпечення пошуку мінімального за часом маршруту. Розглянуто та проаналізовано існуючі системи

Проведено огляд існуючих методів пошуку маршрутів. Проаналізовано методи вирішення задач і було вирішено використовувати метод динамічного програмування для задачі пошуку мінімального за часом шляху. Виконано інфологічне та даталогічне проектування. Розроблено базу даних, за допомогою СКБД Oracle 10g. Програмне забезпечення написане на мові C# з використанням фреймворку Visual Studio. Проведено тестування програмного забезпечення. Проведено тестування програмного забезпечення. Програмне забезпечення успішно пройшло тестування. Показано залежність вибору маршруту від ситуацій на дорогах.

Розроблене програмне забезпечення в подальшому може бути розширене шляхом введення нових модулів програми, розширенням бізнес-правил та функцій без зміни структури.

ВИКОРИСТАНА ЛІТЕРАТУРА

1. В.С. Лукинский, В.И. Бережной, Е.В. Бережная, Е.И. Зайцев, И.А. Цвиринько, Логистика автомобильного транспорта: Учебное пособие/Л84 В.С. Лукинский, В.И. Бережной, Е.В. Бережная, Е.И. Зайцев, И.А. Цвиринько - М.: Финансы и статистика, 2004. - 368 с.
2. Громов С. Методы адаптивного и генетического поиска в оперативном планировании производства [Текст] / Громов С. // Известия высших учебных заведений. Машиностроение. – 2011. - № 6
3. Кузнецов А. В. и др. Высшая математика: Мат. программир.: Учеб./ А.В. Кузнецов, В.А. Сакович, Н.И. Холод; Под общ. ред. А. В. Кузнецова. – Мн.: Выш.шк., 1994. – 286с.
4. Е. Н. Гончаров А. И. Ерзин В. В. Залюбовский Исследование операций Примеры и задачи учебное пособие – Новосибирск: Новосибирский государственный университет, 2005. – 78 с.
5. Сигал И. Х. Иванова А. П. Введение в прикладное дискретное программирование: модели и вычислительные алгоритмы: Учеб. пособие. – Изд. 2-е, испр. – М.: ФИЗМАТЛИТ, 2003. – 240с.
6. А. А. Корбут Ю.Ю. Финкельштейн. Дискретное программирование – М.: 1969 . – 368 ст. с илл.
7. Томас Х. Кормен Чарльз И. Лейзерсон Рональд Л. Ривест Клиффорд Штайн Алгоритмы: построение и анализ, 2-е издание: Пер. с англ. . – М.: Издательский дом «Вильямс», 2005. – 1296 с.