Міністерство освіти і науки України

Житомирський державний технологічний університет

Факультет інформаційно-комп'ютерних технологій

Кафедра інженерії програмного забезпечення

Пояснювальна записка

до атестаційної магістерської роботи

на тему:

«Аналіз та оцінка роботи громадського транспорту»

Виконав студент 2-го курсу,

групи ПІ-48м спеціальності 121  
 «Інженерія програмного забезпечення»

Д.С. Сарно

Керівник Єфіменко А.А.

Зав. кафедри А. В. Панішев

Рецензент О. І. Грабар

Житомир – 2018

ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

Факультет інформаційно-комп'ютерних технологій

Кафедра інженерії програмного забезпечення

Освітньо-кваліфікаційний рівень: магістр

Спеціальність: 121 «Інженерія програмного забезпечення»

|  |  |
| --- | --- |
|  | «ЗАТВЕРДЖУЮ» |
|  | Зав. кафедри ІПЗ |
|  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.В. Панішев |
|  | “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_\_2018р. |

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ МАГІСТЕРСЬКУ РОБОТУ СТУДЕНТУ

Сарно Дмитро Сергійович

1. Тема роботи: «Аналіз та оцінка роботи громадського транспорту»,

керівник роботи: Єфіменко А.А., к.т.н.

затверджені наказом вищого навчального закладу від “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_р. №\_\_\_\_\_

2. Строк подання студентом: “\_\_\_\_”\_\_\_\_\_\_\_\_\_\_\_\_\_ 201\_р.

3. Вихідні дані до роботи: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

|  |
| --- |
|  |
|  |
|  |
|  |
|  |
|  |
|  |

6. Консультанти розділів проекту (роботи)

|  |  |  |  |
| --- | --- | --- | --- |
| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
| завдання видав | завдання прийняв |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

7. Дата видачі завдання “\_\_” \_\_\_\_\_\_\_ 2017р.

**КАЛЕНДАРНИЙ ПЛАН**

|  |  |  |  |
| --- | --- | --- | --- |
| № з/п | Назва етапів атестаційної роботи | Строк виконання етапів роботи | Примітка |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Студент** | **\_\_\_\_\_\_\_\_\_** (підпис) | **Сарно Д.С. \_**  (прізвище та ініціали) |
| **Керівник роботи** | **\_\_\_\_\_\_\_\_\_** (підпис) | **Єфіменко А.А. \_**  (прізвище та ініціали) |

Реферат

Магістерська атестаційна робота складається з програмного комплексу “Веб-сервіс оцінки та аналізу громадського транспорту”. Пояснювальна записка складається із 62 сторінок, 31 ілюстрації, та 10 таблиць.

Метою магістерської роботи є проектування архітектури, аналіз та розробка алгоритмів роботи веб-сервісу для оцінки та аналізу роботи громадського транспорту.

В магістерській роботі наведено завдання на основі який розроблявся веб-сервіс та обґрунтовано вибір інструментальних засобів для його реалізації. Було описано варіанти використання системи, наведено діаграму класів основних компонентів та схему бази даних.

Сервіс був реалізований із використанням MVC фреймворку Laravel та мови програмування PHP. У якості мови для аналізу даних була використана мова програмування Python.

КЛЮЧОВІ СЛОВА: ВЕБ-СЕРВІС, ОБРОБКА ЗОБРАЖЕНЬ, АНАЛІЗ ДАНИХ.

ABSTRACT

Master's attestation work consists of the program complex "Web service for evaluation and analysis of public transport". The explanatory note consists of 62 pages, 31 illustrations, and 10 tables.

The aim of the master's thesis is to design architecture, analyze and develop algorithms for web services to evaluate and analyze the operation of public transport.

The master's thesis presents the tasks on the basis of which the web-service was developed and the choice of tools for its realization was substantiated. The system usage options were described, a chart of the main components classes, and a database schema.

The service was implemented using MVC Laravel framework and PHP programming language. As a language for data analysis, the Python programming language was used.

KEYWORDS: WEB SERVICE, IMAGE PROCESSING, ANALYSIS OF DATA.

Перелік термінів та скорочень

БД – база даних.

RGB – адитивна кольорова модель.

HSV – колірна модель, в якій координатами кольору є: колірний тон, насиченість, яскравість.

ФВЧ – фільтр верхніх частот.

ФНЧ – фільтр нижніх частот.

БПФ – швидке перетворення Фур'є.

АіФ – аргументи і факти.

**ЗМІСТ**

[РЕФЕРАТ 4](#_Toc532174282)

[ABSTRACT 5](#_Toc532174283)

[ПЕРЕЛІК ТЕРМІНІВ ТА СКОРОЧЕНЬ 6](#_Toc532174284)

[ВСТУП 8](#_Toc532174285)

[РОЗДІЛ 1. АНАЛІЗ АЛГОРИТМІЧНОЇ СКЛАДОВОЇ ВЕБ-СЕРВІСУ 10](#_Toc532174286)

[1.1. Опис методів розпізнавання зображень 10](#_Toc532174287)

[1.2. Методи аналізу даних 20](#_Toc532174288)

[1.3. Постановка задачі 26](#_Toc532174289)

[1.4. Вибір інструментальних засобів 27](#_Toc532174290)

[Висновок до розділу 1. 30](#_Toc532174291)

[Роздід 2. ПРОЕКТУВАННЯ МОДЕЛІ ВЕБ-СЕРВІСУ 31](#_Toc532174292)

[2.1. Аналіз бізнес вимог 31](#_Toc532174293)

[2.2. Розробка бази даних сервісу 36](#_Toc532174294)

[2.3. Аналіз роботи алгоритма Віоли-Джонса 39](#_Toc532174295)

[Висновки до розділу 2 41](#_Toc532174296)

[Розділ 3. РЕАЛІЗАЦІЯ ОСНОВНИХ АЛГОРИТМІВ ТА ІНТЕРФЕЙСУ ВЕБ-СЕРВІСУ 42](#_Toc532174297)

[3.1. Реалізація алгоритму Віоли-Джонса 42](#_Toc532174298)

[3.2. Реалізація аналізу даних 46](#_Toc532174299)

[3.3. Інтерфейс сервісу 48](#_Toc532174300)

[Висновки до розділу 3 52](#_Toc532174301)

[ВИСНОВКИ 53](#_Toc532174302)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 54](#_Toc532174303)

[ДОДАТКИ 56](#_Toc532174304)

ВСТУП

**Актуальність теми.** На сьогодняшній день більшість українців користуються громадським транспортом. На даний момент перевізники беруть на себе певні забов’язання за надання хороших умов для перевезення своїх пасажирів та вимагають підвищення цін за перевезення. Ціни зростають, а якість обслуговування взагалі не змінюється, ще й в додачу перевізники все частіше вимагають перегляду цін. В наш час люди скаржаться на проблеми транспорту до влади та до преси, але з часом ці проблеми забуваються та на них не звертають увагу.

Актуальність обраної теми полягає в тому, що люди зможуть залишати свої відгуки про роботу громадського транспорту. Кожна людина матиме змогу зайти на цей сервіс на створити новий відгук і в подальшому переглянути роботу перевізників за весь або частковий термін, що в майбутньому надасть можливість впливати на зростання цін за проїзд та перегляд контрактів із перевізниками.

**Метою магістерської роботи** є проектування архітектури, аналіз та реалізацію алгоритмів роботи та реалізація загальної роботи веб-сервісу для оцінки та аналізу роботи громадського транспорту (надалі сервіс).

Встановлена мета обумовлює наступні завдання:

– відображення сторінки, де користувачі зможуть переглядати проблеми транспорту;

– перегляд графіків;

– вибір інструментів для реалізації сервісу;

– обгрунтування та вибір засобів реалізації системи;

– розробка інформаційного забезпечення системи;

– проектування структурних складових та алгоритмів роботи системи;

– програмна реалізація сервісу.

**Об’єктом дослідження** є інформаційні технології по роботі із створення веб-сервісу оцінки обслуговування громадський транспортом.

**Предметом дослідження** є використання веб-орієнтованих технології для створення веб-сервісу.

**Публікації.**

Д.С. Сарно, магістр, гр. ПІ-48м, Г.В. Марчук, ТЕХНОЛОГІЯ MAPREDUCE. Тези IX міжнародної науковотехнічної конференції «Інформаційно-комп’ютерні технології 2018» (20-21 квітня 2018р.). Житомир : Вид. О.О Євенок, 2018, - С.39-40.

РОЗДІЛ 1. АНАЛІЗ АЛГОРИТМІЧНОЇ СКЛАДОВОЇ ВЕБ-СЕРВІСУ

* 1. Опис методів розпізнавання зображень

Операції розпізнавання на зображеннях певних об'єктів, як правило, передує обробкою зображень для створення умов, що підвищують ефективність і якість виділення і розпізнавання шуканих або досліджуваних об'єктів. Методи попередньої обробки залежать від завдань досліджень, досить різноманітні і можуть включати, наприклад, розпізнавання якихось символів чи якихось певних елементів.

Основне завдання розпізнавання: зрозуміти, чи стосується дані на зображенні до класу шуканих об'єктів. Основні технічні складнощі, що виникають в даному випадку:

1. Зображення пред'являються на складному тлі;
2. Шукані області мають складну геометрію;
3. Вхідні дані мають шум або дезорієнтуючу інформацію;
4. Різні частини зображення мають різні характеристики (підсвічування, освітленість, перешкоди).

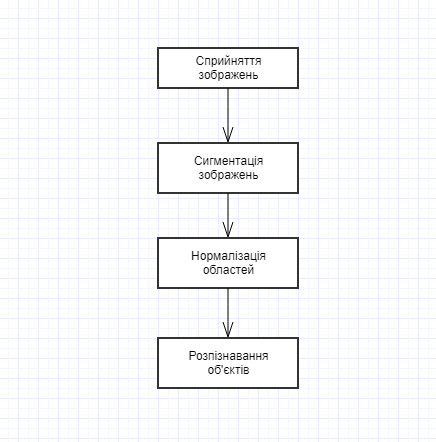


Рис. 1.1.1. Етапи алгоритмів розпізнавання

Попередня обробка - застосовується практично завжди після зняття інформації з відеодатчики (в нашому випадку у разі завантаження у веб-сервіс) і має на меті зниження перешкод на зображенні, що виникли в результаті дискретизації і квантування, а також придушення зовнішніх шумів. Як правило, це операції усереднення і вирівнювання гістограм.

Сегментація - процес пошуку однорідних областей на зображенні. Цей етап дуже важкий і в загальному вигляді не алгоритмізований для довільних зображень. Найбільш поширені методи сегментації, засновані на визначенні однорідних яркостей (кольорів) або однорідностей типу текстур.

Поліпшення / фільтрація - можуть використовуватися після проведення сегментації, і переслідує ту ж мету що і попередня обробка: зниження перешкод на зображенні, що виникли в результаті дискретизації і квантування, а також придушення зовнішніх шумів.

Розпізнавання - найчастіше кінцевий етап обробки, що лежить в основі процесів інтерпретації і розуміння. Вхідними для розпізнавання є зображення, виділені в результаті сегментації і, частково, відреставровані. Вони відрізняються від еталонних зображень геометричними і яскравості спотвореннями, а також збереглися шумами. Для реальних завдань розпізнавання застосовуються, в основному, чотири підходи, які використовують методи: кореляційні, засновані на прийнятті рішень за критерієм близькості з еталонами; признакові і синтаксичні - найменш трудомісткі і нормалізації, що займають проміжне положення за обсягом обчислень. В рамках кожного підходу є свої конкретні алгоритми, які мають певну область застосування, яка залежить від характеру відмінностей вхідних і еталонних зображень, від поміхи обстановки в полі зору, вимог до обсягів обчислень і швидкості прийняття рішень.

В цілому, можна виділити три методи розпізнавання образів.

Метод перебору. У цьому випадку проводиться порівняння з базою даних, де для кожного виду об'єктів представлені різноманітні модифікації відображення. Наприклад, для оптичного розпізнавання образів можна застосувати метод перебору виду об'єкта під різними кутами, масштабами, зміщеннями, деформаціями і т.д. Для букв потрібно перебирати шрифт, властивості шрифту і т.д. У разі розпізнавання звукових образів, відповідно, відбувається порівняння з деякими відомими шаблонами (наприклад, слово, вимовлене кількома людьми).

Другий підхід - проводиться глибший аналіз характеристик образу. У разі оптичного розпізнавання це може бути визначення різних геометричних характеристик. Звуковий зразок в цьому випадку піддається частотному, амплітудному аналізу і т. д.

Наступний метод - використання штучних нейронних мереж. Цей метод вимагає або великої кількості прикладів завдання розпізнавання при навчанні, або спеціальної структури нейронної мережі, яка враховує специфіку даного завдання. Проте, його відрізняє більш висока ефективність і продуктивність.

Зображення, що підлягають розпізнаванню, можуть бути зашумленими. Причиною тому можуть бути спотворення, що вносяться до зображення предметами, активно відбивають світло (скло, залізо, водні поверхні), нерівномірна прозорість повітряного шару, пил, що потрапила в об'єктив, якість використовуваної апаратури і ін. Тому необхідно додатково попередня фільтрації зображення. Виділяють декілька методів фільтрації шумів.

**Гаусів шум.**

Гаусів шум є найбільш простим з математичної точки зору. Спектральні складові цього типу шуму рівномірно розподілені по всьому діапазону задіяних частот. Прикладами білого шуму є шум водоспаду або ефірні перешкоди. У природі і техніці «чисто» білий шум (тобто білий шум, який має однакову спектральну потужність на всіх частотах) не зустрічається (з огляду на те, що такий сигнал мав би нескінченну потужність), проте під категорію білих шумів потрапляють будь-які шуми, спектральна щільність яких однакова (або слабо відрізняється) в розглянутому діапазоні частот. Функція щільності розподілу гаусового шуму випадкової величини z має вигляд:

Де – середнє значення випадкової величини z, – її середньоквадратичне відхилення.

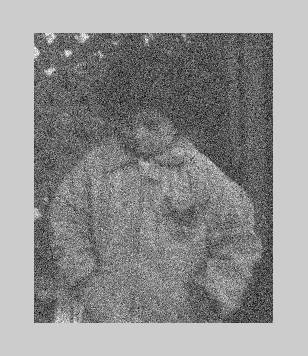


Рис. 1.1.2. Гаусів шум

**Шум Релея.**

Серед важливих і часто зустрічаються на практиці розподілів розглянемо розподіл Релея. Вперше воно було введено лордом Релеєм в 1880 році при розгляді обвідної суми великого числа гармонійних коливань різної частоти. Функція щільності розподілу задається виразом:

**Шум Ерланга.**

Функція щільності розподілу ймовірності шум Ерланга задається виразом

Де a>0, b – додатне число

Всі перераховані види шумів їх розподілу дають нам набір засобів моделювання реальних природних шумів. Наприклад, Гаусів шум зустрічається в електромагнітних ланцюгах, розподіл Релея зустрічається на різних фотознімках, шум Ерланга можна спостерігати на знімках, отриманих за допомогою засобів лазерної та радіолокаційної розвідки.



Рис. 1.1.3. Шум Ерланга

**Бінаризація по порогу, вибір області гістограми.**

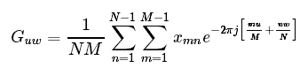
Саме просте перетворення - це бінаризація зображення по порогу. Для RGB зображення і зображення в градаціях сірого порогом є значення кольору. Зустрічаються ідеальні завдання, в яких такого перетворення досить.

Вибір порога, за яким відбувається бінаризація, багато в чому визначає процес самої бінаризации. В даному випадку, зображення було бінаризованими за середнім кольором. Зазвичай бінаризація здійснюється за допомогою алгоритму, який адаптивно вибирає поріг. Таким алгоритмом може бути вибір матожіданія або моди. А можна вибрати найбільший пік гістограми.

Бінаризація може дати дуже цікаві результати при роботі з гістограмами, в тому числі в ситуації, якщо ми розглядаємо зображення не в RGB, а в HSV. Наприклад, сегментувати цікавлять кольори. На цьому принципі можна побудувати як детектор мітки так і детектор шкіри людини.

**Фільтрації: Фур’є, ФНЧ, ФВЧ.**

Класичні методи фільтрації з радіолокації і обробки сигналів можна з успіхом застосовувати в безлічі завдань Pattern Recognition. Традиційним методом в радіолокації, який майже не використовується в зображеннях в чистому вигляді, є перетворення Фур'є (конкретніше - БПФ). Одне з небагатьох виняток, при яких використовується одновимірний перетворення Фур'є, - компресія зображень. Для аналізу зображень одновимірного перетворення зазвичай не вистачає, потрібно використовувати куди більш ресурсоємних двовимірне перетворення.



Мало хто його насправді розраховує, зазвичай, куди швидше і простіше використовувати згортку, що цікавить з уже готовим фільтром, заточеним на високі (ФВЧ) або низькі (ФНЧ) частоти. Такий метод, звичайно, не дозволяє зробити аналіз спектра, але в конкретному завданні видеонаблюдения зазвичай потрібен не аналіз, а результат.

Найпростіші приклади фільтрів, що реалізують підкреслення низьких частот (фільтр Гаусса) і високих частот (Фільтр Габора).

Для кожної точки зображення вибирається вікно і перемножується з фільтром того ж розміру. Результатом такої згортки є нове значення точки.

**Кореляція.**

Після такої вільного трактування вейвлетов з мого боку варто згадати власне кореляцію, що лежить в їх основі. При фільтрації зображень це незамінний інструмент. Класичне застосування - кореляція відеопотоку для знаходження зрушень або оптичних потоків. Найпростіший детектор зсуву - теж в якомусь сенсі різницевий коррелятор. Там де зображення не корелюють - був рух.



Рис. 1.1.4. Фільтр кореляції

Тепер потрібно розглянути основні методи для пошуку об’єктів на зображеннях. Серед найпопулярніших методів виділяють наступні:

* 1. Кольорові фільтри - якщо об'єкт суттєво виділяється на тлі за кольором, то можна підібрати відповідний фільтр.
  2. Виділення і аналіз контурів - якщо ми знаємо, що об'єкт має форму, наприклад, кола, то можна пошукати окружності на зображенні.
  3. Зіставлення з шаблоном - у нас є зображення об'єкта, шукаємо в іншому зображенні області збігаються з цим зображенням об'єкта.
  4. Робота з особливими точками - на зображенні з об'єктом шукаємо особливості (наприклад кути), які намагаємося зіставити з такими особливостями на іншому зображенні.
  5. Методи машинного навчання - навчаємо класифікатор на картинках з об'єктом, деяким способом поділяємо зображення на частини, перевіряємо класифікатором кожну частину на наявність об'єкта.

**Кольорові фільтри.**

Метод кольорових фільтрів можна застосовувати у випадках, коли об'єкт суттєво відрізнятися від фону за кольором і освітлення рівномірно і не змінюється.



Рис. 1.1.5. Приклад примінення кольорових фільтрів

**Виділення і аналіз контурів.**

Якщо об'єкт на тлі за кольором істотно не виділяється і / або має складну розмальовку, то застосування методу колірних фільтрів не дасть хороших результатів.

В цьому випадку можна спробувати застосувати метод виділення та аналізу контурів. Для цього ми виділяємо кордону на зображенні. Межі це місця різкої зміни градієнта яскравості, їх можна знайти наприклад за допомогою методу Canny. Далі ми можемо перевірити виділені лінії-кордону на відповідність геометричним контурам об'єкта, це можна зробити застосувавши метод Хафа, наприклад ми можемо пошукати в межах окружності.

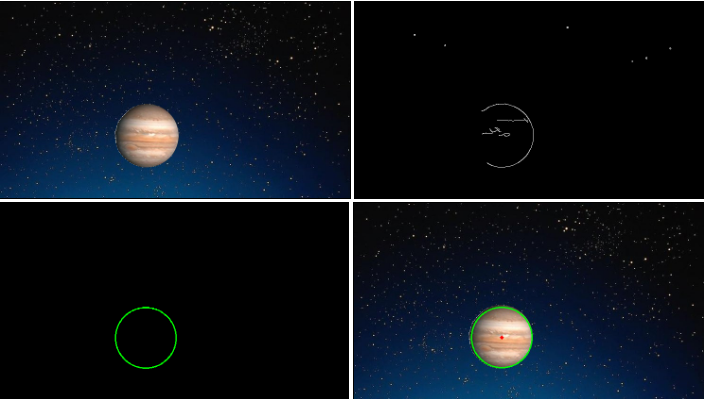


Рис. 1.1.6. Пошук окружності

**Зіставлення з шаблоном**

Якщо зображення має безліч дрібних деталей то аналіз контурів може бути затруднений. В цьому випадку можна застосувати метод зіставлення з шаблоном. Він полягає в наступному - беремо картинку з об'єктом (рис. 1.1.7.) і шукаємо на великому зображенні області збігаються із зображенням об'єкта (рис. 1.1.8.).



Рис. 1.1.7. Об’єкт пошуку



Рис. 1.1.8. Результати пошуку

**Робота з особливими точками**

Метод зіставлення з шаблоном, описаний в попередньому розділі, шукає точні співпадіння точок шаблону з точками зображення. Якщо зображення повернуто чи зі зміненим розміром щодо параметрів шаблону то цей метод працює погано. Для подолання цих обмежень застосовують методи засновані на так званих особливих точках.

Особлива точка це невелика область, яка істотно виділяється на зображенні. Існує кілька методів визначення таких точок, це можуть бути кутиабо БЛОБ (blob, крапля), тобто невеликі області однакової яскравості, досить чіткою кордоном, що виділяються на загальному тлі.

Для особливої ​​точки обчислюють так звані дескриптор - характеристику особливої ​​точки. Дескриптор обчислюють за заданою околиці особливої ​​точки, як напрямки градієнтів яскравості різних частин цієї околиці. Існує кілька методів розрахунку дескрипторів для особливих точок: SIFT, SURF, ORB та інші. Треба зазначити, що деякі методи розрахунку дескрипторів є запатентованими (наприклад SIFT) і їх комерційне використання обмежене.



Рис. 1.1.9. Пошук об’эктів по особливим точкам

* 1. Методи аналізу даних

Аналіз даних - це напрямок досліджень, що включає комплекс методів обробки багатовимірної системи даних спостережень, яка характеризується багатьма ознаками. На відміну від класичних математико-статистичних методів, які передбачають відому вірогідну модель породження даних, методи аналізу даних використовують лише відомості, зафіксовані в цих звітах і даних.

Серед основних методів аналізу даних виділяють такі:

1. Первинний аналіз даних;
2. Інтелектуальний аналіз даних;
3. Статичний аналіз даних;
4. Тополоігчний аналіз даних;
5. Візуальний аналіз даних.

**Первинний аналіз даних.**

Первинний аналіз даних (описова статистика) - розділ математичної статистики, що вивчає способи обробки масивів статистичних даних з метою знаходження узагальнюючих характеристик елементів масиву, побудови компактного і наочного опису масивів даних, виявлення виявляються в масивах закономірностей і / або різко виділяються (з основної маси) спостережень . Закономірності, виявлені при первинному аналізі даних, необхідно перевіряти на нових масивах статистичних даних, щоб забезпечити їх об'єктивність.

**Інтелектуальний аналіз даних.**

Data mining (інтелектуальний аналіз даних, глибинний аналіз даних) - збірна назва, що використовується для позначення сукупності методів виявлення в даних раніше невідомих, нетривіальних, практично корисних і доступних інтерпретації знань, необхідних для прийняття рішень в різних сферах людської діяльності. Термін введений Григорієм П’ятецьким-Шапіро в 1989 році.

Англійське словосполучення «data mining» поки не має усталеного перекладу на російську мову. При перекладі українською мовою використовуються наступні словосполучення: ​​просівши інформації, видобуток даних, вилучення даних, а також інтелектуальний аналіз даних. Більш повним і точним є словосполучення «виявлення знань в базах даних» (англ. Knowledge discovery in databases, KDD).

Основу методів data mining складають всілякі методи класифікації, моделювання і прогнозування, засновані на застосуванні дерев рішень, штучних нейронних мереж, генетичних алгоритмів, еволюційного програмування, асоціативної пам'яті, нечіткої логіки. До методів data mining нерідко відносять статистичні методи (дескриптивний аналіз, кореляційний і регресійний аналіз, факторний аналіз, дисперсійний аналіз, компонентний аналіз, дискримінантний аналіз, аналіз часових рядів, аналіз виживаності, аналіз зв'язків). Такі методи, проте, припускають деякі апріорні уявлення про аналізованих даних, що виникає певна розбіжність з цілями data mining (виявлення раніше невідомих нетривіальних і практично корисних знань).

Одне з найважливіших призначень методів data mining полягає в наочному поданні результатів обчислень (візуалізація), що дозволяє використовувати інструментарій data mining людьми, які не мають спеціальної математичної підготовки.

Застосування статистичних методів аналізу даних вимагає доброго володіння теорією ймовірностей і математичної статистики.

**Статичний аналіз данних.**

Для аналізу даних можуть застосовуватися різні методи. Статистичні методи аналізу даних призначені для їх ущільнення, виявлення взаємозв'язків і структур.

Статистичні методи - методи аналізу статистичних даних. За своєю природою вони діляться на кількісні і категоріальні.

Кількісні (метричні) дані є безперервними за своєю структурою. Ці дані або виміряні за допомогою інтервального шкали (числова шкала, кількісно рівні проміжки якої відображають рівні проміжки між значеннями вимірюваних характеристик), або за допомогою шкали відносин (крім відстані визначено і порядок значень).

Категоріальні (неметричні) дані - це якісні дані з обмеженим числом унікальних значень і категорій. Існує два види категоріальних даних: номінальні - використовується для нумерації об'єктів і порядкові - дані, для яких існує природний порядок категорій.

Статистичні методи діляться на одно- і багатовимірні. Одномірні методи використовуються тоді, коли всі елементи вибірки оцінюються єдиним вимірником або якщо цих вимірників кілька для кожного елемента, але кожна змінна аналізується при цьому окремо від усіх інших.

**Топологічний аналіз данних.**

Топологічний аналіз даних - нова область теоретичних досліджень для задач аналізу даних (Data mining) і комп'ютерного зору.

Основні питання:

* Як з низькорозмірних уявлень отримувати структури високих розмірностей;
* Як дискретні одиниці складаються в глобальні структури.

Людський мозок легко будує уявлення про загальну структуру по приватним даними низьких розмірностей. Йому, наприклад, не складає труднощів отримати тривимірну форму об'єкта по плоским зображенням в кожному оці. Створення загальної структури також проводиться при об'єднанні дискретних в часі фрагментів в безперервний спосіб. Так, наприклад, телевізійне зображення технічно є масивом окремих точок, який, однак, сприймається як єдина сцена.

Основний метод топологічного аналізу даних:

* Заміна набору елементів даних деяким сімейством симпліціального комплексів відповідно до параметром близькості.
* Аналіз цих топологічних комплексів за допомогою алгебраїчної топології, а конкретно новою теорією стійких гомологий.
* Перекодування стійкою гомології набору даних в параметризрвані версію чисел Бетті, далі звану штрихкодом.

**Візуальний аналіз даних.**

Візуальний аналіз даних - це подання даних у вигляді, який забезпечує найбільш ефективну роботу людини по їх вивченню. Візуалізація даних знаходить широке застосування в наукових і статистичних дослідженнях (зокрема, в прогнозуванні, інтелектуальному аналізі даних, бізнес-аналізі), в педагогічному дизайні для навчання і тестування, в новинних зведеннях і аналітичних оглядах. Візуалізація даних пов'язана з візуалізацією інформації, інфографікою, візуалізацією наукових даних, розвідувальним аналізом даних і статистичної графікою.

Одні із основних представлень даних:

1. Графіки і діаграми.

Графіки та діаграми, напевно, самий звичний для нас вигляд візуалізації даних. Використовується як для презентації даних, так і для аналізу. Зустріти їх можна і на роботі, і в журналі і в науковому звіті. Зазвичай знання про існуючі типи діаграм і графіків ми отримуємо зі школи або з стандартного набору в екселя. Однак, мало хто знає, що світ графіків і діаграм не обмежується точковим графіком, столбиковой і круговою діаграмою. Існують близько 15 загальновідомих типів діаграм, а всього їх понад 60, при цьому їх кількість збільшується з кожним днем ​​- люди придумують нові типи для візуалізації складних і незвичайних даних. Розрізняють наступні види діграм:

* Стовпчасті діаграма ;
* Лінійні діаграми;
* Гістограми з декількома осями;
* Гістограми;
* Кругові діаграми;
* Географічні діаграми;
* Стрім-графік або графік потоку;
* Бульбашкова діаграма.



Рис. 1.2.1 Приклад графік

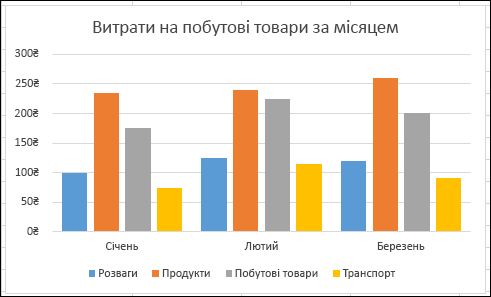


Рис. 1.2.2. Приклад діаграми

1. Інфографіки і схеми.

Інфографіка стала дуже популярна в останні роки, хоча існують вже давно. Інфографіка ставитися до журналістики даних, де графіки і схеми пояснюють будь-які факти з обраної теми. Зазвичай інфографіка статична і являє собою довге «простирадло» з картинками і текстом. Відмінною особливістю інфографіки є те, що в ній наводяться вже готові висновки, тобто читача проводять за руку з обраної теми і при цьому приправляють це все цифрами і картинками. Часто використовується мальований або мультяшний стиль. Деякі ЗМІ випускають інфографіку на щоденній основі, наприклад АіФ. Бум інфографіки викликав зниження загального рівня її якості. Часто використовується не до місця або «для краси», хоча звичайно ж є чудові і цікаві приклади.

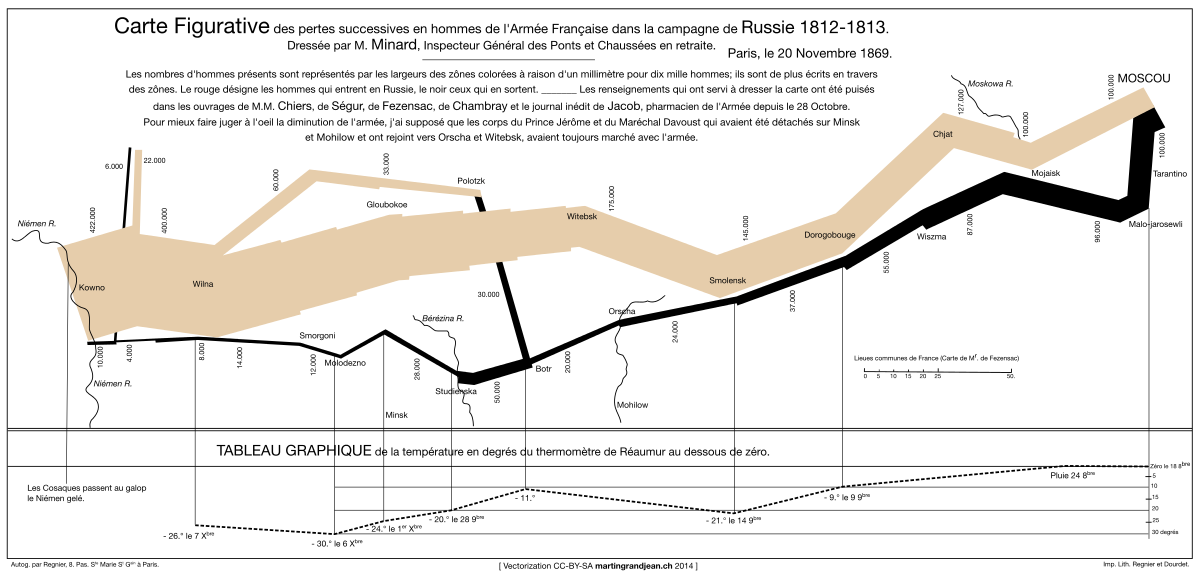


Рис. 1.2.3. Приклад інфографіки

* 1. Постановка задачі

Основною метою створення сервісу є надання людям можливості залишати відгуків про користування громадським транспортом. При залишенні оцінки про користування громадським транспортом ви маєте можливість вибрати проблему із якою ви зіткнулися, місто та транспорт, яким ви користувалися. Також є додатковий функціонал, який надасть можливість завантажити фото із транспортом, який вас обслуговував, а сервіс знайде транспорт в базі за цим номерним знаком. Також кожен користувач зможе переглянути порушення перевізників у вигляді повідомлень та графіків.

У магістерській роботі необхідно розробити функціонал для створення відгуку, розпізнавання номерного знаку, функціонал для побудови графіків. Для реалізації поставленого завдання, основними етапами є:

1. Реалізувати основний функціонал системи.
2. Написати програмні модулі для:

* Модуль для розпізнавання номерів;
* Модуль для побудови графіків;
* Модуль виконання сторонніх скриптів.

1. Реалізувати CRUD операції для всіх таблиць бази даних системи.
2. Створити демонстраційний та тестовий контент сайту.
3. Розробити план тестування системи управління контентом. Зокрема передбачити наступні види тестування:
   * Функціональне тестування;
   * Кросбраузерне тестування;
   * Тестування навантаження.

Результатом реалізації поставленого завдання є сервіс, що містить необхідний функціонал для створення веб-сервісу аналізу та оцінки роботи громадського транспорту.

* 1. Вибір інструментальних засобів

Для створення веб-сервісу, доцільно застосувати: мову розмітки гіпертекстових документів HTML, каскадні таблиці стилів CSS, браузерну мову програмування JavaScript, мову програмування РНР, в якості сервера використовується .

*HTML* − це мова розмітки гіпертекстових документів, яка визначає які елементи будуть відображатися на сторінці.

*CSS* − це каскадна таблиця стилів, яка визначає відображення HTML-документів.

*VueJs* − прогресивний фреймворк для створення користувацьких інтерфейсів. На відміну від фреймворків-монолітів, Vue створений придатним для поступового впровадження. Його ядро в першу чергу вирішує завдання рівня уявлення (view), що спрощує інтеграцію з іншими бібліотеками та існуючими проектами. З іншого боку, Vue повністю підходить і для створення складних односторінкових додатків (SPA, Single-Page Applications), якщо використовувати його спільно з сучасними інструментами та додатковими бібліотеками.

*PHP* − це широко використовувана мова сценаріїв загального призначення з відкритим вихідним кодом. PHP це мова програмування, спеціально розроблена для написання web-додатків (сценаріїв), що виконуються на Web-сервері.

Переваги використання:

* РНР є кросплатформеним. Він є сумісний з більшістю операційних систем, таких як Windows, Linux, Solaris, Mac OSX;
* Існування великої кількості різних фреймворків за допомогою яких можна створювати веб-сайти різної складності за більш швидший період часу ;
* Існування великої кількості різних бібліотек, які розширюють функціонал сайту;
* Найважливішою перевагою використання РНР є підтримка великої кількісті різних баз даних.

*Open Server -*  це портативний локальний WAMP / WNMP сервер, який має багатофункціональну керуючу програму і великий вибір підключаються компонентів.

*JetBrains PhpStorm* - це інтегроване середовище розробки на PHP з інтелектуальним редактором, яка глибоко розуміє код, підтримує PHP 7.1, 7.0, 5.6, 5.5, 5.4 і 5.3 для сучасних і класичних проектів, забезпечує краще в індустрії автодоповнення коду, рефакторинг, запобігання помилок нальоту і підтримує змішування мов.

Порівняльний аналіз СКБД дозволяє раціонально вибрати систему керування базами даних для проекту. У якості альтернатив будуть розглянуті наступні СКБД: MySQL, Microsoft SQL Server (далі MS SQL), PostgreSQL. Дві з них безкоштовні – MySQL, PostgreSQL.

Усі обрані системи керування базами даних підходять для проведення аналізу й порівняння тому що реалізують реляційну модель даних.

Таблиця 1.4.1

Максимально можливий обсяг збережених даних для кожної СКБД

|  |  |  |  |
| --- | --- | --- | --- |
|  | Розмір БД | Розмір таблиці | Розмір рядка |
| MySQL | ∞ | 256 ТБ | 64 KB |
| MS SQL | 524258 ТБ | 524258 ТБ | ∞ |
| PostgreSQL | ∞ | 32 ТБ | 1.6 ТБ |

За критерієм тригери й збережені процедури всі альтернативи ідентичні. Усі підтримують тригери, процедури й функції.

Таблиця 1.4.2

Аналіз підтримуваних операційних систем

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Windows | Linux | Unix | Android | Symbian |
| MySQL | + | + | + | + | + |
| MS SQL | + | - | - | - | - |
| PostgreSQL | + | + | + | - | + |

Інформаційна система організації містить у собі конфіденційну інформацію, для запобігання несанкціонованого доступу застосовуються різні способи захисту. Проведемо порівняння систем забезпечення безпеки даних у таблиці

Таблиця 1.4.3

Аналіз систем забезпечення безпеки даних

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Ідентифікація | Захист від brute-force | Шифрування | Сертифікація безпеки |
| MySQL | + | - | + | - |
| MS SQL | + | - | + | + |
| PostgreSQL | + | + | + | + |

Проведений аналіз таких СКБД, як MySQL, MS SQL та PostgreSQL показав, що для виконання поставлених задач найбільш підходящими є MySQL.

Переваги використання MySQL:

* Швидкодія. Завдяки внутрішньому механізму багатопоточності швидкодія MySQL є досить високою
* Для некомерційних цілей поширюється безкоштовно
* Є досить надійною
* Для роботи не потребує великої кількості ресурсів ком’пютера
* Досить легка переносимість на різні ОС

Під час розробки сервісу було використано OpenServer.

Отже нами було обрано для створення програмного комплексу такі інструментальні засоби та технології:

* мову розмітки гіпертекстових документів HTML;
* каскадні таблиці стилів CSS;
* мову програмування JavaScript;
* мову програмування РНР;
* середовище керування базами даних MySQL;
* пакет OpenServer.

В якості середи розробки було використано JetBrains PhpStorm - комерційне кроссплатформене інтегроване середовище розробки для PHP, яке значно поліпшує написання коду.

Висновок до розділу 1.

У цьому розділі булу дослідженно основіні методи по обробці зображень та аналізу даних.

Було проведено аналіз та вибір інструментальних засобів за допомогою яких буде розроблятися веб-сервіс.

На основі усіх необхідних вимог було сформовано постановку задачі.

Роздід 2. Проектування МОДЕЛІ веб-сервісу

* 1. Аналіз бізнес вимог

Сервіс для аналізу та оцінки роботи громадського транспорту створюється з метою надати жителям, які користуються громадським транспортом, залишити відгук про використання громадським транспортом та переглянути загальну кількість відгуків. Також в майбутньому сервіс надасть можливість міській владі переглянути відгуки та обдумати чи потрібно підписувати контракти із перевізниками, які є заядлими порушниками.

У розроблюваному сервісі передбачені такі користувачі, як адміністратор , внутрішній та зовнішній користувачі.

Внутрішні користувачі має можливість:

* переглянути відгуки;
* переглянути відгуки у вигляді графіків;
* залишити відгук;
* дізнатися номер транспорту та його власника за допомогою завантаженого фото

Адміністратор в загалом поповнює базу даними.

Внутрішні користувачі матимуть змогу переглянути відгуки.

Була розроблена діаграма варіантів використання (Рис. 2.1) в якій більш детально наведено функціональні можливості користувачів.

За результатами діаграми використання визначені наступні функціональні вимоги:

1. Реєстрація.
2. Авторизація.
3. Створення відгуку.
4. Перегляд відгуків.
5. Побудова графіків за відгуками.

6. Можливість працювати зі своїм профілем

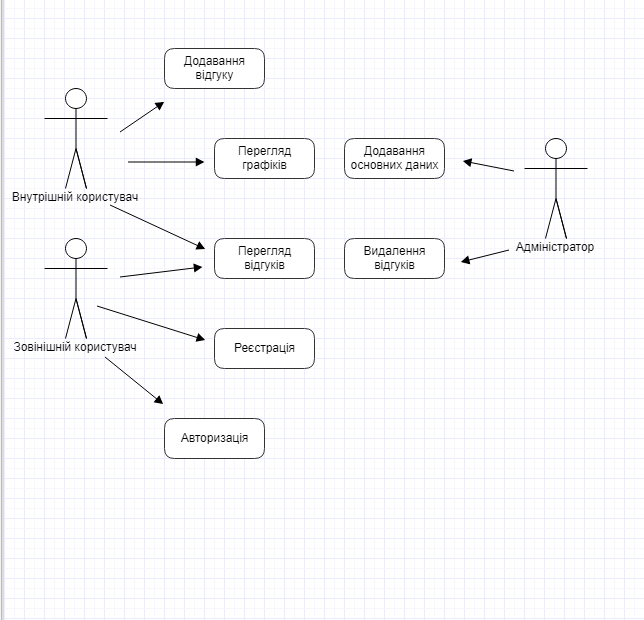


Рис. 2.1.1 Діаграма варіантів використання

Нефункціональні вимоги:

1. Навчання роботи з системою:

* час, потрібний для навчання з інструментами роботи з інформаційною системою для звичайних користувачів – 1 година, а для досвідчених – 0.5 годин;
* час відповіді системи для звичайних запитів неповинен перевищувати 1 секунд, а для більш складних запитів – 3 сек;
* інтерфейс представлення ІС повинен бути інтуїтивно зручним для користувача та не вимагати від нього додаткової підготовки;
* надійність;
* доступність – час, потрібний для обслуговування системи не повинен перевищувати 10% від загального часу роботи;
* середній час безперервної роботи – 20 робочих днів;
* максимальна норма помилок та дефектів в роботі системи – 1 помилка на 1000 запитів користувача.

1. Продуктивність.

Система повинна підтримувати мінімум 100 одночасно працюючих користувачів, пов’язаних з спільною базою даних.

1. Можливість експлуатації:

* масштабування – система повинна мати можливість збільшувати потужності (продуктивність), зі збільшенням користувачів таким чином, щоб це аж ні як негативно не відобразилося на її роботі.
* оновлення версій – оновлення версій повинно здійснюватися в автоматичному режимі залежно від вподобань користувачів та розширення списку запропонованих послуг.

Для реалізації проекту була вибрана технологія MVC та мова програмування PHP. Для реалізації системи було використано фреймворк Laravel, який працює на мові програмування РНР. Також в якості одного із js фреймоврків було використано vue, що на сьогодняшній день є досить популярним фреймворком. А якості реляційної бази даних було використано MySql.

Аналіз функціональних вимог дозволив виділити наступні сутності, що забезпечать реалізацію програмного комплексу системи. На рисунках нижче наведено діаграми класів.

Діаграма класу, що представлена на рис. 2.1.2 є спрощена. На діаграмі присутні основні моделі та контролери. Також на діаграмі класів представлені класи деякіх сервісів, які взаємодіють зі сторонніми системами.Моделі та котролери наслідуються від базових класі, які не будуть відображатися на діаграмі.

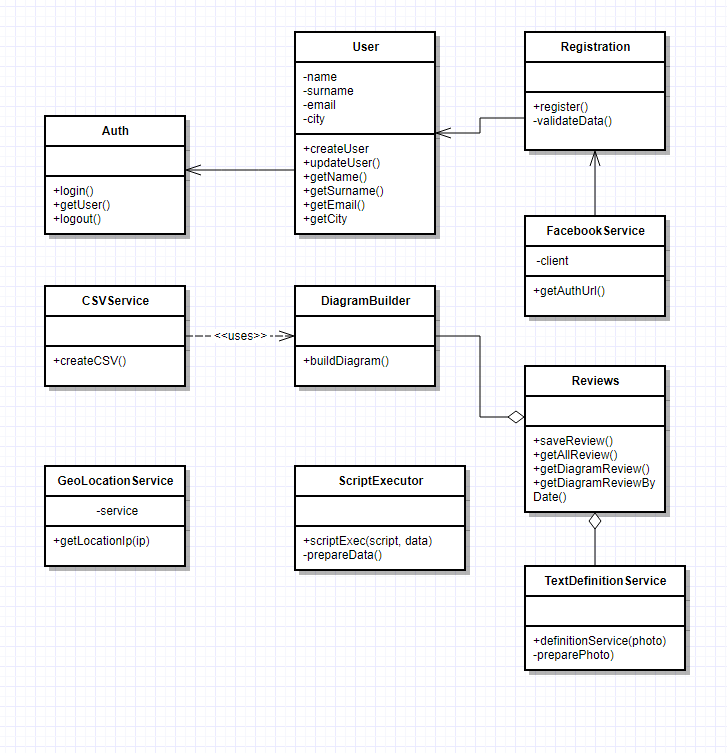


Рис. 2.1.2 Діаграма класів

Із діаграми, яка показана на рис. 2.1.2 можна виділити та описати наступні класи:

* Auth – клас роботи із користувачами. Основні його задачі – це авторизація користувачів у системі. Клас має наступні методи : login – авторизує користувача у системі; getUser – повертає авторизованого користувача; logout – розлогінює користувача;
* User – клас для роботи із користувачами сервісу. Клас маж наступні методи: getName – повертає ім'я користувача; getSurname – повертає прізвище користувача; getEmail – повертає емейл користувача; getCity – повертає місто проживання користувача; createUser – зберігає нового юзера; updateUser – оновляє інформацію про юзера;
* Registration – клас, який надає змогу реєструватись новим користувачам. Класс має наступні методи : register – реєструє користувачів; validateData – перевіряє коректність даних для реєстрації.
* CSVService – клас для роботи із збереженням усіх відгуків у файл csv. Клас має наступні методи : createCSV – зберігає усі дані у файл.
* DiagramBuilder – клас для роботи із створенням графіків. Клас має наступні методи: buildDiagram – метод для відправки даних, на основі яких будуть будуватися графіки.
* Reviews – клас, який призначений для роботи із відгуками. Клас має наступні методи: saveReviews –зберігає відгуки від користувачів; getAllReview – метод на отримання усіх відгуків; getDiagramReview– метод на отримання графіків по усім відгукам; getDiagramReviewByDate – мето на отримання графіків по певним датам вігуків.
* TextDefinitionService – клас, який надає змогу розпізнати номерний зна автомобіля на фотографії. Клас має наступні методи: definitionText – розпізнає номер на фото; preparePhoto – підготовлює фото.
* ScriptExecutor – клас призначений для виконання сторонніх скриптів. Клас має наступні методи: scriptExec – метод для виконання стороннього скрипта; prepareData – метод для підготовки даних, які будуть відправлятися на сторонній скрипт.
* FacebookService – клас, який надає змогу зареєструвати користувача у системі через соціальну мереже Facebook. Клас маж наступні методи: getAuthUrl() – метод генерує на основі даних facebook додатку посилання на реєстрування користувача.
* GeoLocationService – клас, який надає змогу отримати координати користувачі по ip-адреси. Клас має наступні методи: getLocationIp – метод повертає координати користувача на основі його ip-адреси.
  1. Розробка бази даних сервісу

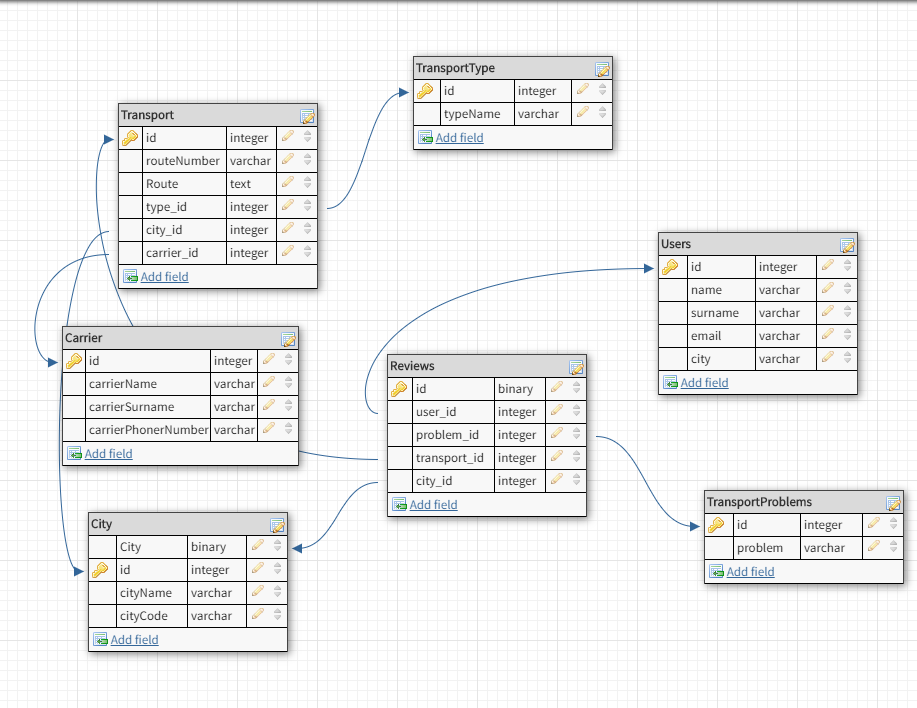


Рис. 2.2.1

Далі наведемо опис основних таблиць бази даних.

Таблиця «City» призначена зберігання даних по містах. Структура таблиці наведена нижче:

Таблиця 2.2.1

Опис полів таблиці «City»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип даних | ПК | ЗК | Опис поля |
| Id | Int | + | - | Первинний ключ |
| citeName | Varchar(100) | - | - | Назва міста |
| cityCode | Varchar(100) | - | - | Код міста |

Таблиця «Carrier» призначена для зберігання основної інформації, яка стосується міст. Структура таблиці наведена нижче:

Таблиця 2.2.2

Опис полів таблиці «Carrier»

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Назва | Тип даних | ПК | | ЗК | | Опис поля |
| Id | Int | + | - | | Первинний ключ | |
| carrierName | Varchar(100) | - | - | | Ім'я перевізника | |
| carrierSurname | Varchar(100) | - | - | | Прізвище перевізника | |
| CarrierPhoneNumber | Varchar(100) | - | - | | Мобільний номер перевізника | |

Таблиця «Review» призначена для збереження інформації по відгукам, які залишили користувачі. Ця таблиця є основною, на базі цієї таблиці будуються графіки по яким можна робити певні висновки про діяльності громадського транспорту. Також наоснові цієх таблиці генеруються документи для експорту даних по порушенням. Структура таблиці наведена нижче:

Таблиця 2.2.3

Опис полів таблиці «Review»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип даних | ПК | ЗК | Опис поля |
| Id | Int(11) | + | - | Первинний ключ |
| User\_id | Int(11) | - | + | Зовнішній ключ |
| City\_id | Int(11) | - | + | Зовнішній ключ |
| Problem\_id | Int(11) | - | + | Зовнішній ключ |
| Transport\_id | Int(11) | - | + | Зовнішній ключ |
| dateReview | Date | - | - | Дата відгуку |
| timeReview | Time | - | - | Час відгуку |
| Location | TEXT | - | - | Місце в якому було порушення |

Таблиця «TransportProblems» призначена для збереження проблем, які стосуються громадського транспорту. В цій таблиці зберігаються основні проблеми, які можна вибрати у випадаючому списку, коли залишаєш відгук про певний тип транспорту. Структура таблиці наведена нижче:

Таблиця 2.2.4

Опис полів таблиці «TransportProblems»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип даних | ПК | ЗК | Опис поля |
| Id | Int(11) | + | - | Первинний ключ |
| Problem | varchar(100) | - | - | Проблема |

Таблиця «Transport» призначена для зберігання даних про основні пробоблеми, які пов’язані із громадським транспортом. Структура таблиці наведена нижче:

Таблиця 2.2.5

Опис полів таблиці «Transport»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип даних | ПК | ЗК | Опис поля |
| Id | Int(11) | + | - | Первинний ключ |
| routeNumber | Varchar(100) | - | - | Номер маршруту |
| Route | TEXT | - | - | Маршрут транспорту |
| Type\_id | Int(11) | - | + | Зовнішній ключ |
| carNumber | Varchar(100) | - | - | Номер транспорту |
| Carrier\_id | Int(11) | - | + | Зовнішній ключ |
| City\_id | Int(11) | - | + | Зовнішній ключ |

Таблиця «TransportType» призначена для зберігання інформації про тип транспорту. Структура таблиці наведена нижче:

Таблиця 2.2.6

Опис полів таблиці «TransportType»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип даних | ПК | ЗК | Опис поля |
| Id | Int(11) | + | - | Первинний ключ |
| typeName | Varchar(100) | - | - | Тип транспорту |

Таблиця «Users» призначена для зберігання яка стосується внутрішніх користувачів систем. Структура таблиці наведена нижче:

Таблиця 2.2.7

Опис полів таблиці «Users»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Назва | Тип даних | ПК | ЗК | Опис поля |
| Id | Int(11) | + | - | Первинний ключ |
| Name | Varchar(30) | - | - | Ім'я користувача |
| Surname | Varchar(30) | - | - | Прізвище користувача |
| Email | Varchar(30) | - | - | Емейл користувача |
| City | Varchar(40) | - | - | Місто |

* 1. Аналіз роботи алгоритма Віоли-Джонса

В даний час метод Віоли-Джонса є популярним методом для пошуку об'єкта на зображенні в силу своєї високої швидкості та ефективності. В основу методу Віоли-Джонса покладені: інтегральне представлення зображення за ознаками Хаара, побудова класифікатора на основі алгоритму адаптивного бустінга і спосіб комбінування класифікаторів в каскадну структуру. Ці ідеї дозволяють здійснювати пошук об'єкта в режимі реального часу.

Інтегральне представлення зображення - це матриця, однакова за розмірами з вихідним зображенням. У кожному елементі матриці зберігається сума інтенсивностей всіх пікселів, що знаходяться лівіше і вище даного елемента - правого нижнього кута прямокутної області (0,0) до (x, y).

Розрахунок значень елементів матриці проходить за час, пропорційне числу пікселів в оригінальному документі, тому інтегральне зображення прораховується за один прохід.

За допомогою інтегрального представлення зображення можна швидко розрахувати сумарну яскравість довільної прямокутної області на зображенні.

На етапі виявлення об'єкта в методі Віоли-Джонса використовується вікно певного розміру, яке рухається по зображенню. Для кожної області зображення, над якою проходить вікно, розраховується ознака Хаара, за допомогою якого відбувається пошук потрібного об'єкта.

Ознака-відображення, де Df-безліч допустимих значень ознаки. Якщо задані ознаки f1, ..., fn, то вектор ознак x = (f1 (x), ..., fn (x)) називається признаковая описом об'єкта x. Прізнаковие опису допустимо зіставляти з самими об'єктами. При цьому безліч X = Df1 \* ... \* Dfn називають просторі ознак.

Ознаки поділяються на такі типи в залежності від безлічі Df:

* бінарний ознака, Df = {0,1};
* номінальний ознака: Df - кінцеве безліч;
* порядковий ознака: Df - кінцеве впорядкована множина;
* кількісний ознака: Df - безліч дійсних чисел.

Ознака Хаара обчислюється по суміжних прямокутним областям. У стандартному методі Віоли-Джонса використовуються прямокутні примітиви, зображені на Рис. 1.4.1.

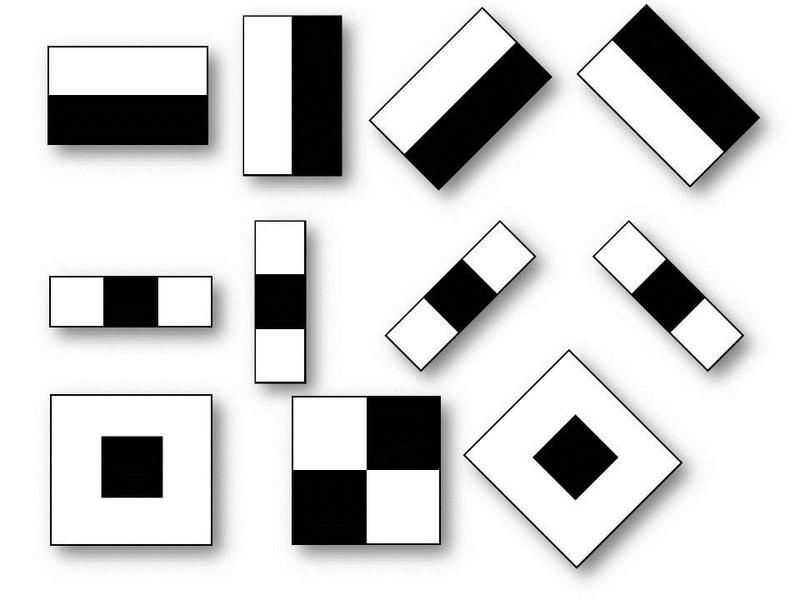


Рис. 2.3.1 Примітиви Хаара

Висновки до розділу 2

Реалізований сервіс здійснює збереження відгуків про громадський транспорт. Дає можливість користувачам отримати список усіх відгуків, отримати діаграми по кількістю порушень перевізниками або отримати їх у ексель форматі.

В даному розділі було розроблено бд, представлено сему бд та детальний її опис.

Було здійснено реалізацію оснвних компонентів системи та представлено діаграму класів.

Також у цьому розділі було дослідженно основні методи аналізу даних.

Розділ 3. Реалізація основних алгоритмів та інтерфейсУ Веб-сервісу

3.1. Реалізація алгоритму Віоли-Джонса

Для пошуку номерного знаку на зображенні було використано цей алгоритм, бо він добре справляється із пошуком об’єкта на зображенні за певними критеріями.

Тепер поетапно пройдемо по всім крокам пошуку об’єкта на зображенні. Для того щоб розпізнати зображення першочергово на вхід потрібно прийняти якесь зображення на якому будемо шукати об’єкт (рис. 3.1.1).



Рис. 3.1.1. Вхідне зображення

Далі для того щоб прибрати високочастотні шуми потрібно примінити до зображення фільтр, що надасть зображенню сірого відтінку (рис. 3.1.2.) і після цього приміняєтьяс Гаусівське розмиття (рис. 3.1.3.)

raw\_image = cv2.imread(name, 1)

gray = cv2.cvtColor(input\_image, cv2.COLOR\_BGR2GRAY)

gray = enhance(gray)

gray = cv2.GaussianBlur(gray, (5, 5), 0)

Лістинг 3.1.1. Примінкння сірого фільтру та Гаусівського шуму



Рис. 3.1.2. Примінення сірого фільтру



Рис. 3.1.3. Примінення Гаусівського розмиття

Після примінення вищезгаданих фільтрів потрібно опреділити усі края об’єктів для того щоб потім опреділити вейвлети для пошуку об’єктів. Для цієї задачі можно використовувати такі функції як Canny або Sobel. Було обрано Sobel, результат обробки на рис. 3.1.4. Метод Собель, іноді званий оператором Собель-Фельдман або Фільтр Собеля, використовується при обробці зображень і комп'ютерному баченні, особливо в алгоритмах виявлення кордонів, де він створює образи, що підкреслюють краю.



Рис. 3.1.4. Примінення Sobel методу

Були знайдені контури об'єктів на зображеннях. Однак, як ви бачите, деякі з контурів не закриті - між контурами існують проміжки. Щоб прибрати проміжки між білими пікселями зображення, ми застосуємо операцію «закриття», яка надалі надасть можливість знайти усі контури, або так названі вейвлети, серед яких будемо шукати потрбний контур

se = cv2.getStructuringElement(cv2.MORPH\_RECT, se\_shape)

gray = cv2.morphologyEx(sobel, cv2.MORPH\_CLOSE, se)

Лістинг 3.1.2. Примінення операції закриття конутів

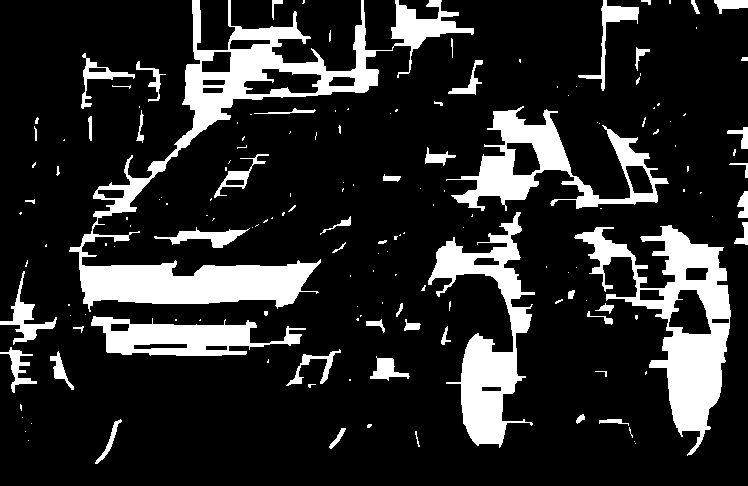


Рис. 3.1.5. Результат закриття контурва

Далі потрібно опреділити усі можливі контури за допомогою функції findContours. Для прикладу наведено 2 можливі контури на рис. 3.1.6.



Рис. 3.1.6. Пошук контурів

Після цього відбувається порівняння заданого по крітеріям контуру із тими, що були знайдені на зображенні. Після того як якісь контур співпадає, ми можемо його виділити (рис. 3.1.7).



Рис. 3.1.7 Знайдений номерний знак

3.2. Реалізація аналізу даних

Для реалізації аналізу даних було обрано візуальний аналіз даних, який надасть можливість користувачам більш детально переглянути усі необхідну їм інформацію у вигляді діаграми.

В загалом процес аналізу даних, які відбуваються на сервісі, проходить в 2 етами. Перший етап полягає в перенесенню даних в csv формат (рис. 3.2.1), для того щоб в подальшому можна було обробити цей файл на python – скрипті.

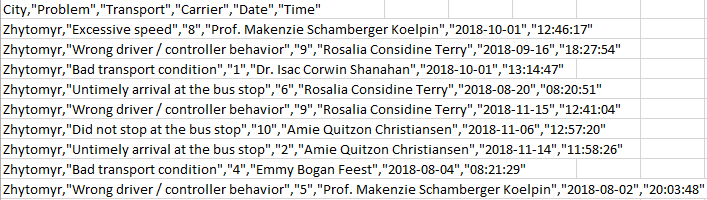


Рис. 3.2.1. Приклад csv файлу

Після того як файл було згенеровано файл python – скрипт має можливість завантажити, опрацювати необхідні дані та збудувати на основі них діграму (рис. 3.2.2.)

df = pd.read\_csv("C:/Program Files/OSPanel/domains/Urban-transport-reviewer/storage/app/review.csv", escapechar='`', low\_memory=False)

mask = (df['Date'] > dateStart) & (df['Date'] <= dateEnd)

country\_count = pd.value\_counts(df.loc[mask]['Carrier'].values, sort=True)

country\_count\_keys, country\_count\_values = dict\_sort(dict(country\_count))

TOP\_COUNTRY = len(country\_count\_keys)

plt.title('Reviews from ' + str(dateStart) + ' to ' + str(dateEnd), fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.bar(np.arange(TOP\_COUNTRY), country\_count\_values)

plt.xticks(np.arange(TOP\_COUNTRY), country\_count\_keys, rotation=90, fontsize=10)

plt.yticks(fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.ylabel('Count reviews', fontsize=PLOT\_LABEL\_FONT\_SIZE)

plt.show()

Лістинг 3.2.1. Приклад побудови діаграми

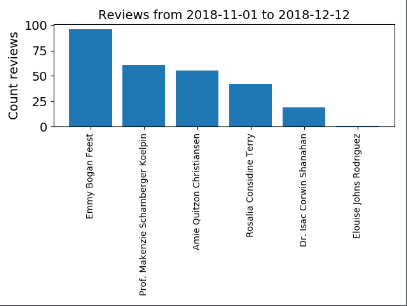


Рис. 3.2.2. Приклад рейтингу перевізників за кількістю порушень

3.3. Інтерфейс сервісу

В загалом сервіс матиме такі сторінки:

* Головна сторінка;
* Сторінка реєстрації;
* Сторінка авторизації;
* Сторінка перегляду відгуків;
* Сторінка додавання нового відгуку

Головна сторінка в основному матиме схему руху транспорту та графік їх руху (рис. 3.3.1)

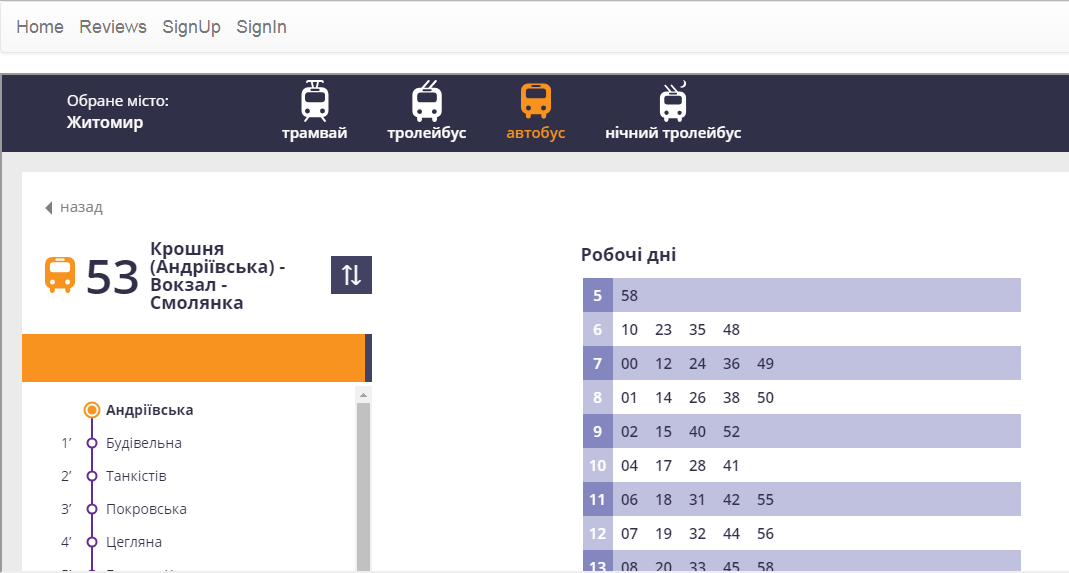


Рис. 3.3.1 Головна сторінка

Сторінка реєстрації представляє собою звичайну форму, що дає можливість зареєструвати нового користувача у системі. Також на сторінці реєстраціїї зареєструватися на сервісі за допомогою соціальної мережі фейсбук (рис.3.3.3). Майже аналогічно виглядає сторінка авторизації (рис.3.3.2).

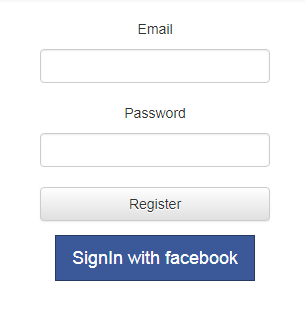


Рис. 3.3.2. Стрінка авторизації

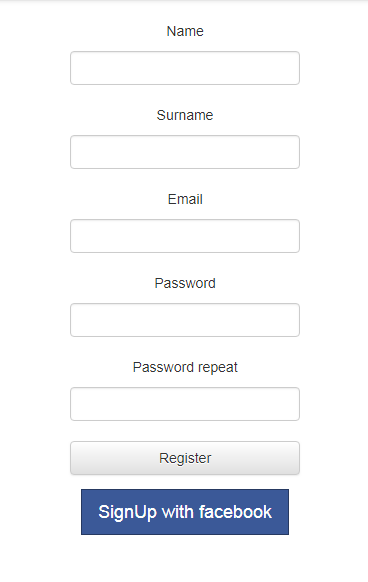


Рис. 3.3.3. Сторінка реєстрації

Наступна сторінка дає можливість переглядати усі відгуки по транспорту (рис. 3.3.4.). На сторінці є карта, де відображаються фотографії із порушеннями (рис. 3.3.5.).

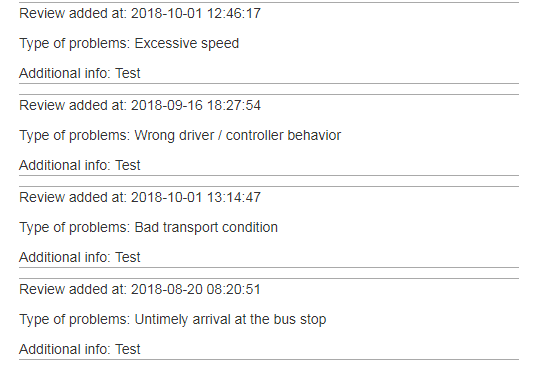


Рис. 3.3.4. Відображення відгуків

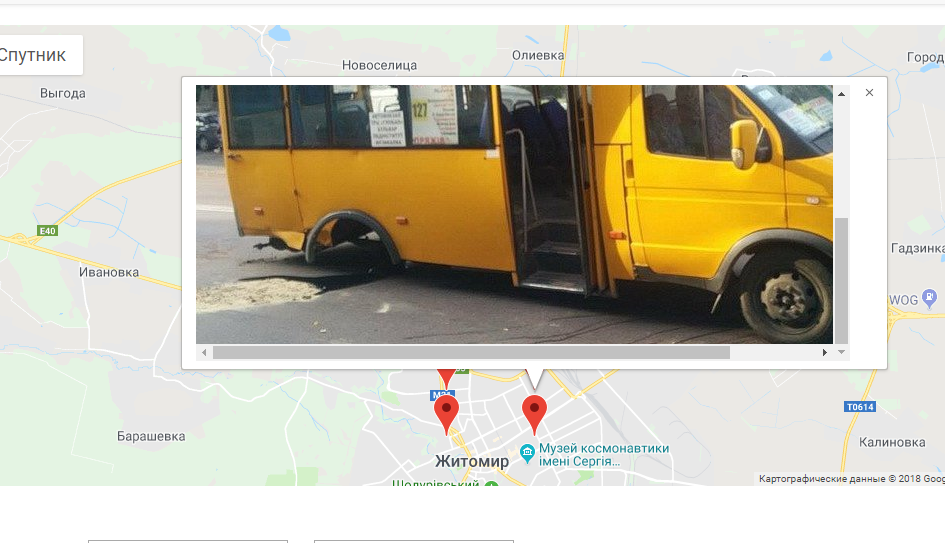


Рис. 3.3.5. Відображення фотографій на карті

Також на сторінці під картою я кнопки, які надають можливість отримати графіки по порушенням або отримати ексель файл (рис. 3.3.6.).

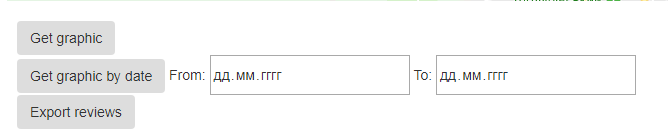


Рис. 3.3.6. Функціонал для отримання даних по порушенням

Остання сторінка представляє собою форму для збереження відгуку про певний транспорт (рис 3.3.7.). Користувач має змогу вибрати обрати тип проблеми, ввести якусь додаткову інформації та завантажити фото порушення, якщо вони були зафіксовані на фотокамеру. На сторінці присутній функціонал для вибору адреси, поле якого має інтеграцію із Google API. Також користувачі матимуть змогу завантажити фото номерного знаку, що дасть змогу розпізнати цей номер та отримати усі необхідні дані по цьому транспорту.

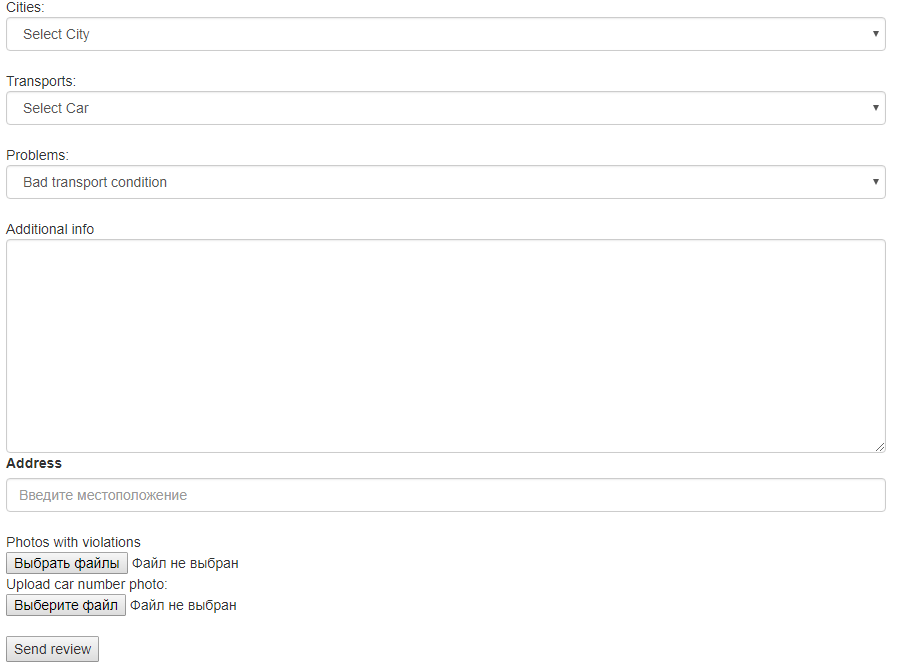


Рис. 3.3.7. Сторінка збереження відгуку

Висновки до розділу 3

В даному розділі було представлено поетапну реалізацію пошуку номерного знака на фото та реалізацію візуального аналізу даних.

Також було було описано основні компоненти візуального інтефейсу користувача.

Висновки

Під час виконання магістерської роботи було розроблено веб-сервіс для оцінки та аналізу роботи громадського транспорту. На основі поставлених задач було розроблено такий функціонал:

* Перегляд графіку руху транспорта;
* Перегляд відгуків;
* Перегляд фотофіксації порушення на карті;
* Можливість додавати відгук;
* Розпізнавання номерного знаку автомобіля на фото при збереженні відгуку;
* Можливість отримання даних по порушення перевізників у вигляді діаграми або excel-файлу.

В ході роботи було дослідженно та реалізовано алгоритм для пошуку номерного знака на фото, а також було розглянуто основні методи по аналізу даних та реалізовано візуальний аналіз даних.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Kennedy B. HTML & XHTML: The Definitive Guide / B. Kennedy, C. Musciano. – Sebastopol: O'Reilly Media, 2006. – 680 с. – (6th Edition).
2. С. Хольцнер, РНР в примерах. М.: ООО«Бином-Пресс», 2007 г., 352 с
3. Duckett J. HTML & CSS: Design and Build Web Sites / Jon Duckett. – Hoboken: John Wiley & Sons, 2011. – 512 с.
4. М. Фаулер, К. Скотт. UML в кратком изложении. Применение стандартного языка объектного моделирования. М: «Мир», 1999, 191 с.
5. К. Канер, Дж. Фолк, К.Е. Нгуен, Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. Киев: ДиаСофт, 2001, 544 с.
6. А. Гончаров, Самоучитель HTML. СПб.: Питер, 2002, 240 с.
7. Д. Котеров, Самоучитель PHP4. СПб.: БХВ, Петербург, 2003, 576 с.
8. Э. Харрис, PHP/MySQL для начинающих, пер. с англ. М.: КУДИЦ–ОБРАЗ, 2005, 384 с.
9. Л. Криспин, Д. Грегори, Гибкое тестирование: практическое руководство для тестировщиков ПО и гибких команд, М.: «Вильямс», 2010, 464 с.
10. С.А. Мартишин, В.А. Симонов, М.В. Храпченко, Проектирование и реализация баз данных в СУБД MySQL c использованием MySQL Workbench, Учебное пособие. М.: ИНФРА-М, 2012, 160 с.
11. С.В. Синицын, Н.Ю. Налютин, Верификация программного обеспечения. М.: БИНОМ, 2008, 368 с.
12. Довідниково-інформаційний портал htmlbook [Електроний ресурс]. Доступ за посиланням: http://htmlbook.ru/.
13. PHPFAQ [Електроний ресур]. Доступ за посиланням: <http://phpfaq.ru/pdo>. [Дата звернення 22.05.2017 р.]
14. Офіційний мануал по PHP. Доступ за посиланням: http://php.net/manual/ru/index.php.
15. В.В. Кириллов, Г.Ю. Громов, Структуризованный язык запросов (SQL): учебн. пособ.. СПб: Санкт-Петерб. госуд. техн. универ., 1998. [Електроний ресурс]. Доступ за посиланням: http://www.citforum.ru/database/sql\_kg/.

Додатки

**Додаток А**

Клас для виконання python-скриптів

**class** PythonScriptExec  
{  
  
 **CONST *PYTHON\_SCRIPT*** = **"C:\Users\User\AppData\Local\Programs\Python\Python37-32\python.exe"**;  
 **public static function** exec(string $pythonScript, $data = **null**)  
 {  
 **if**(!*is\_null*($data)) {  
 $res = *exec*(**static**::***PYTHON\_SCRIPT*** . **'"'** . $pythonScript . **'" '** . **static**::*prepareData*($data) .**' 2>&1'**, $out);  
 } **else** {  
 $res = *exec*(**static**::***PYTHON\_SCRIPT*** . **'"'** . $pythonScript . **'" '** . **static**::*prepareData*([**'type'** => **'all'**]) .**' 2>&1'**, $out);  
 }  
 **return** $res*/\*json\_decode(exec($script))\*/*;  
 }  
  
 **public static function** prepareData($data)  
 {  
 **return** *base64\_encode*(*json\_encode*($data));  
 }  
  
}

**Додаток Б**

Знаходження координат за ip-адресою

**class** GeoLocationService  
{  
 **private $service**;  
  
 **public function** \_\_construct($service)  
 {  
 $this->**service** = $service;  
 }  
  
 **private function** setIp($ip) {  
 **if**(**empty**($ip)) {  
 **return false**;  
 }  
 **return** $this->**service**->city($ip);  
 }  
  
 **public function** getLocationByIp($ip)  
 {  
 **if**($record = $this->setIp($ip)) {  
 $location = [  
 **'lat'** => $record->**location**->**latitude**,  
 **'lng'** => $record->**location**->**longitude** ];  
 **return** $location;  
 }  
 **return false**;  
 }  
}

**public function** getLocation()  
{  
 $city\_db = public\_path() . **'/GeoLite2-City.mmdb'**;  
 $service = **new** GeoLocationService(**new** Reader($city\_db));  
 **return** $service->getLocationByIp($\_SERVER[**'REMOTE\_ADDR'**]);  
}

**Додаток В**

Алгоритм Віоли-Джонса

def validate\_contour(contour, img, aspect\_ratio\_range, area\_range):  
 rect = cv2.minAreaRect(contour)  
 img\_width = img.shape[1]  
 img\_height = img.shape[0]  
 box = cv2.boxPoints(rect)  
 box = np.int0(box)  
  
 X = rect[0][0]  
 Y = rect[0][1]  
 angle = rect[2]  
 width = rect[1][0]  
 height = rect[1][1]  
  
 angle = (angle + 180) if width < height else (angle + 90)  
  
 output = False  
  
 if (width > 0 and height > 0) and ((width < img\_width / 2.0) and (height < img\_width / 2.0)):  
 aspect\_ratio = float(width) / height if width > height else float(height) / width  
 if (aspect\_ratio >= aspect\_ratio\_range[0] and aspect\_ratio <= aspect\_ratio\_range[1]):  
 if ((height \* width > area\_range[0]) and (height \* width < area\_range[1])):  
  
 box\_copy = list(box)  
 point = box\_copy[0]  
 del (box\_copy[0])  
 dists = [((p[0] - point[0]) \*\* 2 + (p[1] - point[1]) \*\* 2) for p in box\_copy]  
 sorted\_dists = sorted(dists)  
 opposite\_point = box\_copy[dists.index(sorted\_dists[1])]  
 tmp\_angle = 90  
  
 if abs(point[0] - opposite\_point[0]) > 0:  
 tmp\_angle = abs(float(point[1] - opposite\_point[1])) / abs(point[0] - opposite\_point[0])  
 tmp\_angle = rad\_to\_deg(math.atan(tmp\_angle))  
  
 if tmp\_angle <= 45:  
 output = True  
 return output  
  
  
def process\_image(name, debug, \*\*options):  
 se\_shape = (16, 4)  
  
 if options.get('type') == 'rect':  
 se\_shape = (17, 4)  
  
 elif options.get('type') == 'square':  
 se\_shape = (7, 6)  
  
 raw\_image = cv2.imread(name, 1)  
 input\_image = np.copy(raw\_image)  
  
 gray = cv2.cvtColor(input\_image, cv2.COLOR\_BGR2GRAY)  
 gray = enhance(gray)  
 gray = cv2.GaussianBlur(gray, (5, 5), 0)  
 gray = cv2.Sobel(gray, -1, 1, 0)  
 h, sobel = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)  
 se = cv2.getStructuringElement(cv2.MORPH\_RECT, se\_shape)  
 gray = cv2.morphologyEx(sobel, cv2.MORPH\_CLOSE, se)  
 ed\_img = np.copy(gray)  
  
 \_, contours, \_ = cv2.findContours(gray, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)  
  
 font = cv2.FONT\_HERSHEY\_SIMPLEX  
  
 for contour in contours:  
 aspect\_ratio\_range = (2.2, 12)  
 area\_range = (500, 18000)  
  
 if options.get('type') == 'rect':  
 aspect\_ratio\_range = (2.2, 12)  
 area\_range = (500, 18000)  
  
 elif options.get('type') == 'square':  
 aspect\_ratio\_range = (1, 2)  
 area\_range = (300, 8000)  
  
 if validate\_contour(contour, gray, aspect\_ratio\_range, area\_range):  
 rect = cv2.minAreaRect(contour)  
 box = cv2.boxPoints(rect)  
 box = np.int0(box)  
 Xs = [i[0] for i in box]  
 Ys = [i[1] for i in box]  
 x1 = min(Xs)  
 x2 = max(Xs)  
 y1 = min(Ys)  
 y2 = max(Ys)  
  
 angle = rect[2]  
 if angle < -45:  
 angle += 90  
  
 W = rect[1][0]  
 H = rect[1][1]  
 aspect\_ratio = float(W) / H if W > H else float(H) / W  
  
 center = ((x1 + x2) / 2, (y1 + y2) / 2)  
 size = (x2 - x1, y2 - y1)  
 M = cv2.getRotationMatrix2D((size[0] / 2, size[1] / 2), angle, 1.0);  
 tmp = cv2.getRectSubPix(ed\_img, size, center)  
 tmp = cv2.warpAffine(tmp, M, size)  
 TmpW = H if H > W else W  
 TmpH = H if H < W else W  
 tmp = cv2.getRectSubPix(tmp, (int(TmpW), int(TmpH)), (size[0] / 2, size[1] / 2))  
 \_\_, tmp = cv2.threshold(tmp, 0, 255, cv2.THRESH\_BINARY + cv2.THRESH\_OTSU)  
  
 white\_pixels = 0  
  
 for x in range(tmp.shape[0]):  
 for y in range(tmp.shape[1]):  
 if tmp[x][y] == 255:  
 white\_pixels += 1  
  
 edge\_density = float(white\_pixels) / (tmp.shape[0] \* tmp.shape[1])  
  
 tmp = cv2.getRectSubPix(raw\_image, size, center)  
 tmp = cv2.warpAffine(tmp, M, size)  
 TmpW = H if H > W else W  
 TmpH = H if H < W else W  
 tmp = cv2.getRectSubPix(tmp, (int(TmpW), int(TmpH)), (size[0] / 2, size[1] / 2))  
  
 if edge\_density > 0.5:  
 cv2.drawContours(input\_image, [box], 0, (127, 0, 255), 2)  
 cropped = input\_image[304:358, 91:315]  
 cv2.imwrite("C:/Program Files/OSPanel/domains/Urban-transport-reviewer/public/cars/crops/cropped.jpg", cropped)  
 return input\_image  
  
data = json.loads(base64.b64decode(sys.argv[1]))  
path = str(data['img'])  
real\_path = 'C:/Program Files/OSPanel/domains/Urban-transport-reviewer/public/cars/' + path  
o1 = process\_image(real\_path, 0, type='rect')

**Додаток Г**

Функціонал для реєстрації через фейсбук

**class** FacebookService  
{  
 **public static** *$client*;  
  
 **public static function** getClient()  
 {  
 **if**(**isset**(**static**::*$client*)) {  
 **return static**::*$client*;  
 }  
 **return static**::*$client* = **new** FacebookClient();  
 }  
}

**class** FacebookClient  
{  
  
 **CONST *APP\_ID*** = **'2223172801342066'**;  
 **CONST *SECRET\_AAP*** = **'bae8e74c340d3e22852d9ac9b6beb4ae'**;  
 **CONST *BASE\_URL*** = **'https://www.facebook.com/dialog'**;  
 **private $provider**;  
  
 **public function** \_\_construct()  
 {  
 $this->**provider** = **new** Facebook([  
 **'clientId'** => FacebookClient::***APP\_ID***,  
 **'clientSecret'** => FacebookClient::***SECRET\_AAP***,  
 **'redirectUri'** => **'https://1f7a2c56.ngrok.io/register-token'**,  
 **'graphApiVersion'** => **'v2.10'**,  
 ]);  
 }  
  
 **public function** getProvider()  
 {  
 **return** $this->**provider**;  
 }  
  
 **public function** Url()  
 {  
 **return new** AuthUrlResource($this);  
 }  
  
}

**class** AuthUrlResource  
{  
 **private $client**;  
  
 **public function** \_\_construct(FacebookClient $client)  
 {  
 $this->**client** = $client;  
 }  
  
 **public function** getAuthUrl()  
 {  
 **return** $this->**client**->getProvider()->getAuthorizationUrl([  
 **'scope'** => [**'email'**],  
 ]);  
 }  
}