Інформаційна система «EatForYou»

Команда «Політ»
м. Кременчук
Полтавська обл.

Технічне завдання

- Розробити систему обробки замовлень страв онлайн
- Розробити систему обліку товарів на складі
- Розробити алгоритм розподілу завдань із доставки між водіями
- Розробити систему взаємодії з водіями
- Розробити систему отримання статистичних звітів

Використані технології

Основні

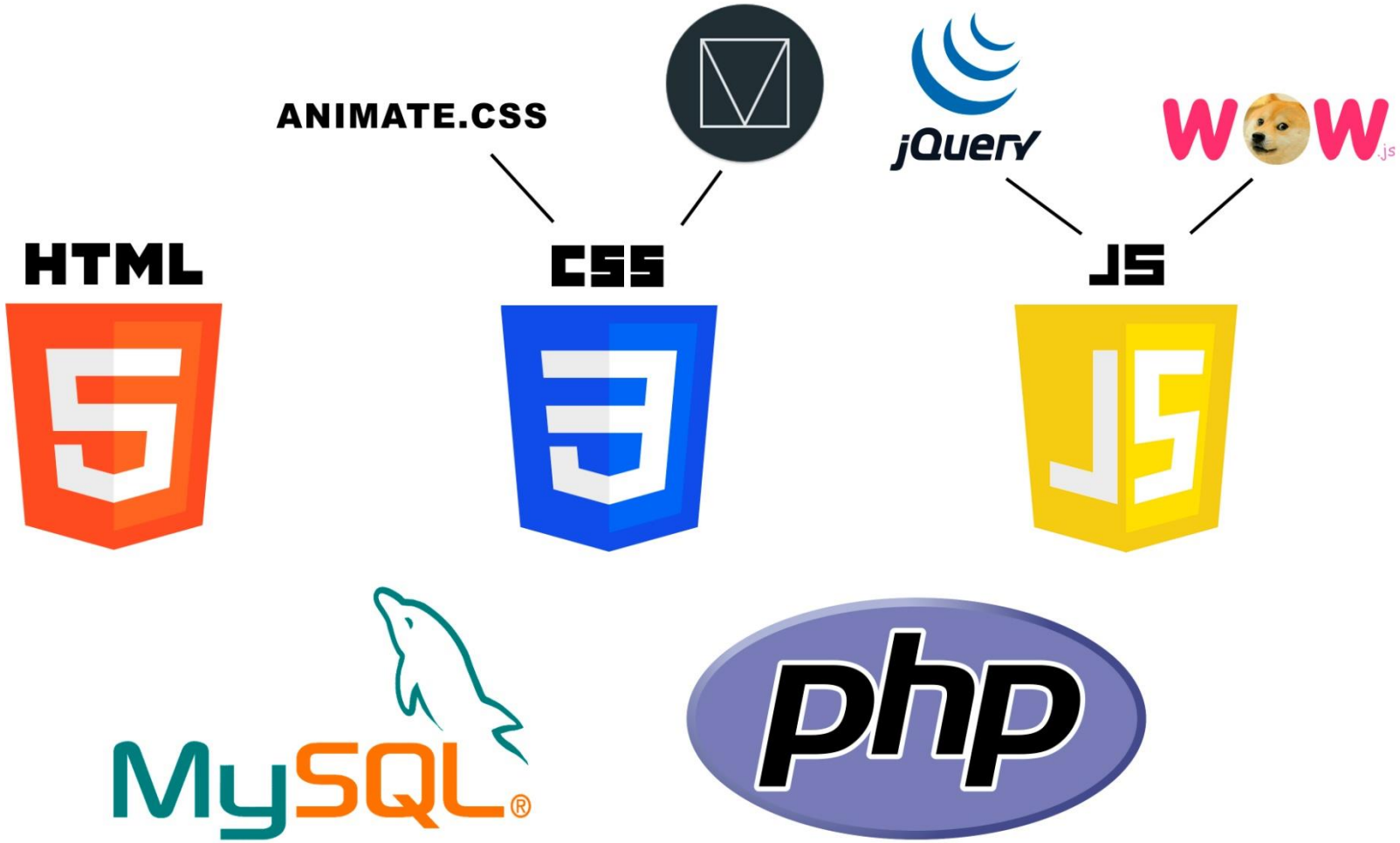
- HTML5, CSS3, JavaScript (ECMAScript 2018) – оформлення інтерфейсної частини
- PHP 7.2, MySQL 5.6 – управління базою даних, формування веб-інтерфейсу з використанням даних з бази

Додаткові бібліотеки

- jQuery 3.3.1 – опрацювання даних на боці клієнта
- Animate.css – забезпечення анімаційних ефектів
- WOW.js – анімація прокрутки сторінки
- Material Design Lite – розробка сайту в стилі Material Design

Зовнішні ресурси

- Hostpro (https://hostpro.ua) – хостинг для розміщення сайту
- OpenStreetMaps (http://openstreetmap.org.ua) – позначення на карті місцезнаходження водіїв


ANIMATE.CSS
jQuery
HTML
CSS
JS
MySQL
php

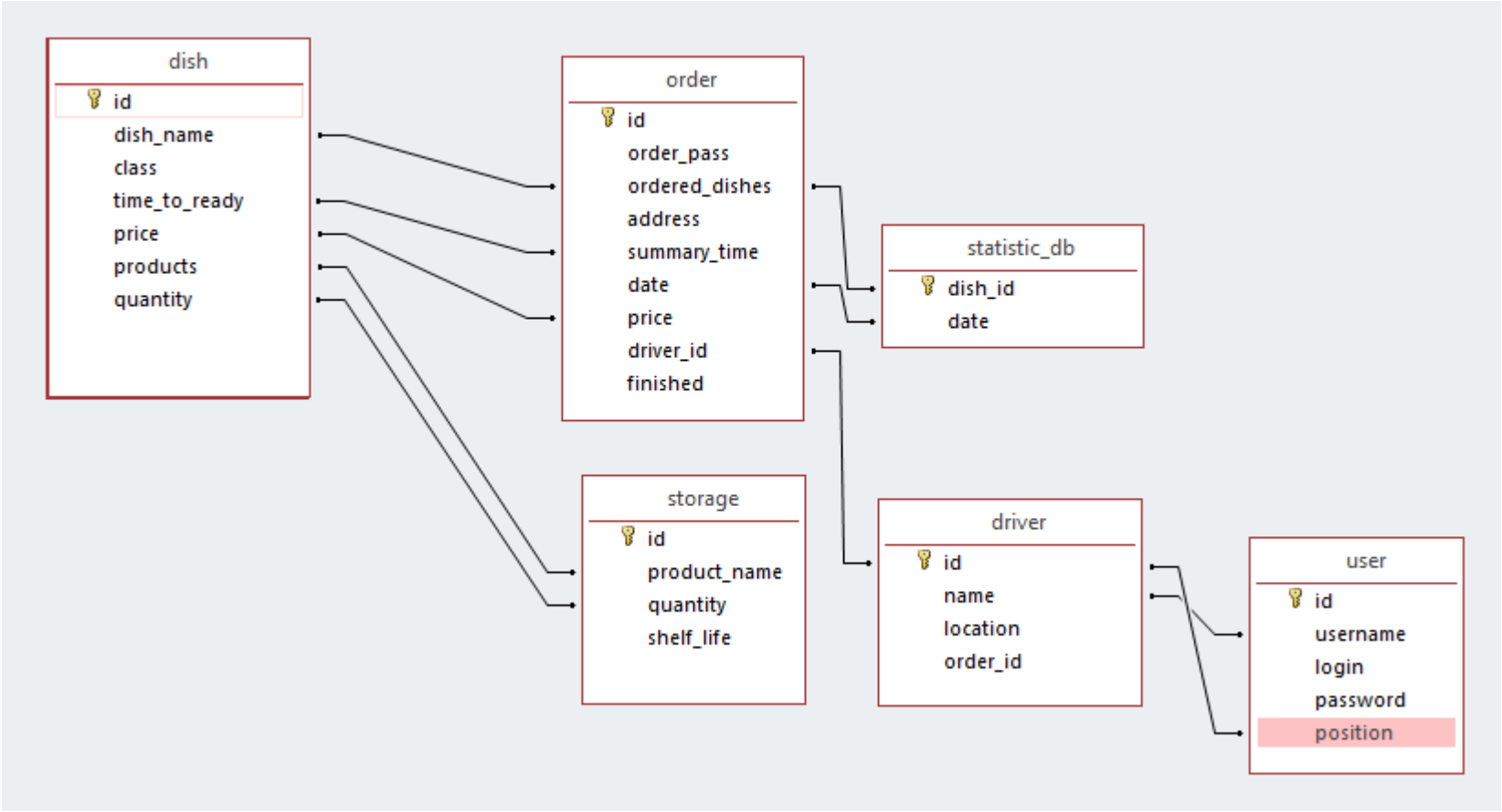
Складові системи

Веб-інтерфейс - забезпечує виконання основних функцій користувачами системи:

- **Клієнт**: створення замовлення та відстеження стану його виконання.
- **Оператор**: отримання замовлення, розрахунок часу виконання, контроль за наявністю продуктів на складі, розподіл водіїв на замовлення, контроль за доставкою замовлень.
- **Водій**: отримання замовлення на доставку, повідомлення про доставку, повідомлення про режим відпочинку.
- **Адміністратор**: контроль за роботою системи, управління персоналом, отримання статистичних звітів.

База даних - взаємодіє з веб-інтерфейсом, містить відомості про список страв (меню) ресторану, наявність та стан продуктів на складі, отримані та виконані замовлення, зареєстрованих користувачів інформаційної системи тощо.

Структура бази даних



Режими роботи

- В режимі онлайн - https://eatforyou.tk/
- В режимі офлайн:
  - на платформі PHP+MySQL (OpenServer, Denwer)
  - Інсталяція системи - EFY-installer.exe
- Офлайн версія системи має обмеження при відсутності підключення до Інтернету

Захист системи

- Хостинг Hostpro дає можливість отримати сертифікат SSL
- Передавання даних забезпечує безпечний протокол https
- У кореневій директорії налаштовано файл .htaccess для заборони доступу сторонніх осіб до директорій інформаційної системи
- Уведення та виведення даних в полях форм екранується для захисту від шкідливого коду
- Захист сторінок зареєстрованих працівників забезпечено перевіркою авторизації користувача.
- До паролів додається випадково сгенерована сіль та застосовується алгоритм хешування SHA512.

Алгоритм роботи системи

Розподіл замовлень у мережі ресторану

- Вважаємо, що у мережі ресторану є кілька точок продажу (кафе).
- Кафе готує певну кількість страв одночасно (потоки приготування страв).
- На потоках можуть одночасно готуватися страви з одного або кількох замовлень.
- Страви на потоці можуть бути у стані опрацювання або у стані очікування.
- Критерій вибору кафе: тривалість приготування страв у сумі з тривалістю доставки повинна бути найменшою

Алгоритм обчислення тривалості приготування страв

- Новий заказ має певну кількість продуктів, що мають різну тривалість приготування.
- Продукти сортуються за тривалістю приготування та розподіляються між потоками приготування страв у кафе за жадібним алгоритмом:
  - продукт, що потребує найбільшого часу для приготування потрапляє на найменш завантажений потік.

Контрольний приклад

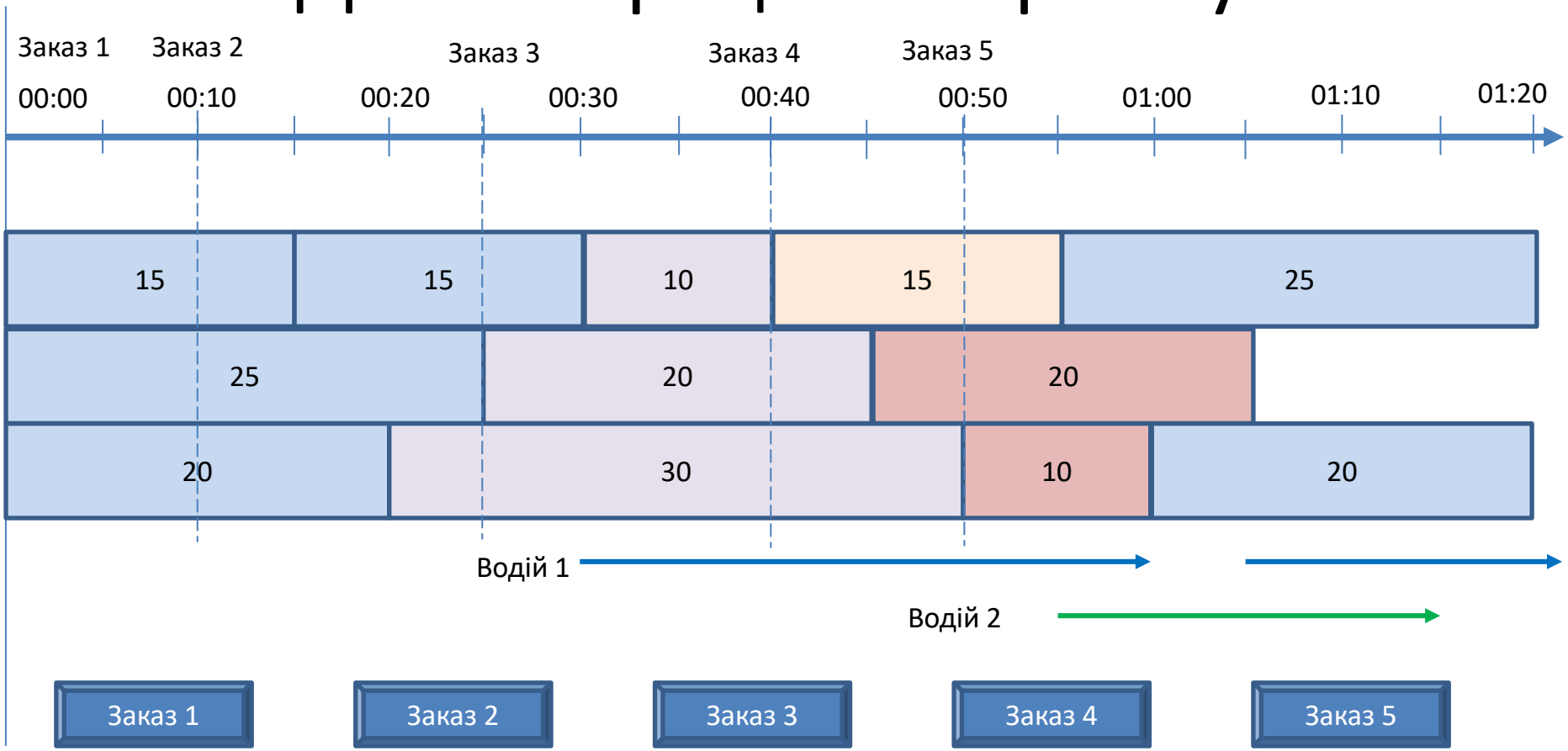
- Кількість кафе – 1
- Потоків приготування страв – 3
- Водіїв – 2
- Положення водіїв – у кафе
- Час очікування (за потреби) на завершення приготування наступного замовлення – 5 хвилин
- Час надходження першого замовлення – 00:00 на шкалі часу

- Час відраховується по шкалі часу з 5-хвилинним інтервалом

Список замовлень

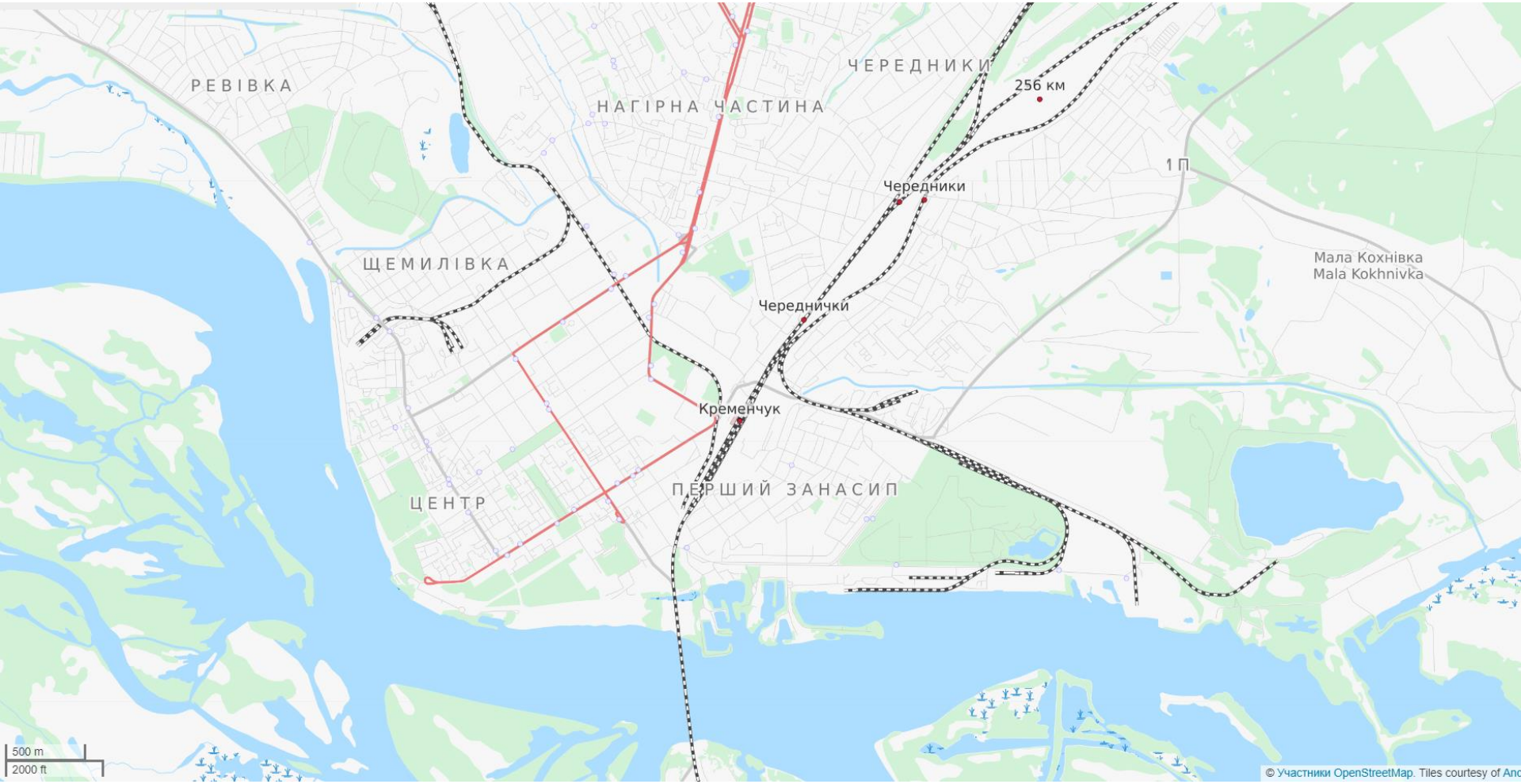
Параметри замовлення	Номер замовлення				
	1	2	3	4	5
Час надходження	00:00	00:10	00:25	00:40	00:50
Кількість страв	4	3	1	2	2
Тривалість приготування страв (хв.)	15, 25, 20, 15	30, 10, 20	15	20, 10	25, 20
Час початку приготування	00:00	00:20	00:40	00:45	00:55
Час завершення приготування	00:30	00:50	00:55	01:05	01:20
Час виклику водія	00:30, 1	00:50, 2	-	01:05, 1	01:20, 2
Час очікування водія	-	00:05	-	-	-
Тривалість доставки	00:15	00:10	-	00:10	00:15
Час доставки замовлення	00:45	01:05	01:05	01:15	01:35
Час повернення водія	01:00	01:15	-	01:25	01:50

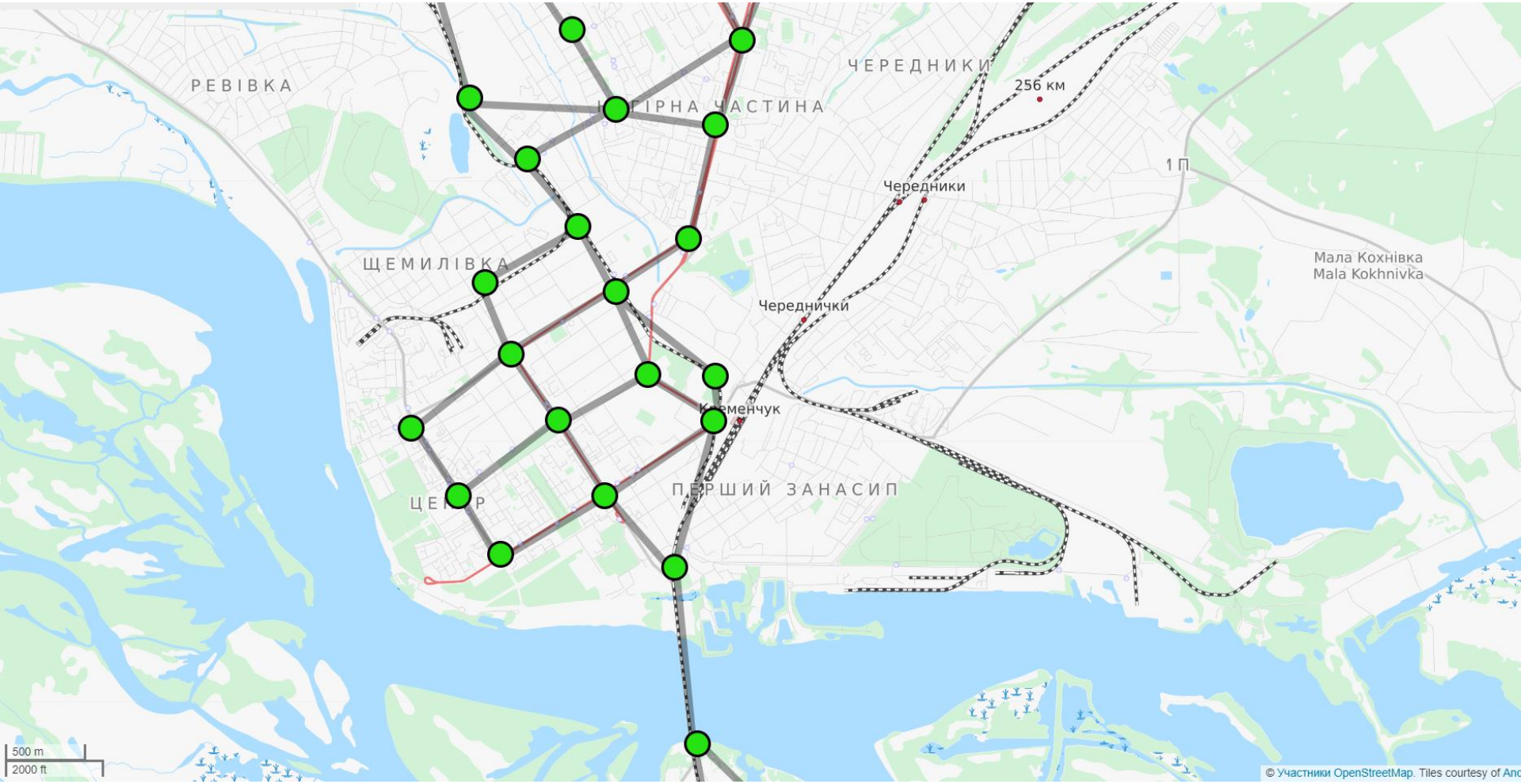
Демонстрація алгоритму



Алгоритм доставки замовлень

- Основа алгоритму - побудова орієнтованого зваженого графа по території міста.
- Вершини графа – ключові точки на карті міста
- Під час оформлення замовлення система отримує місцезнаходження кінцевої адреси та знаходить найближчу до неї ключову точку.
- Після цього відбувається розрахунок часу, необхідного для приготування страв замовлення та їх доставки


РЕВІВКА
НАГІРНА ЧАСТИНА
ЧЕРЕДНИКИ
256 км
1 П
Чередники
Мала Кохнівка
Mala Kokhnivka
ЩЕМИЛІВКА
Череднички
Кременчук
ЦЕНТР
ПЕРШИЙ ЗАНАСИП
500 m
2000 ft
© Участники OpenStreetMap. Tiles courtesy of Anc


РЕВІВКА
ЧЕРЕДНИКИ
256 км
Чередники
1 П
ЩЕМИЛІВКА
Мала Кохнівка
Mala Kokhnivka
Чередничкй
ПЕРШИЙ ЗАНАСИП
500 m
2000 ft
© Участники OpenStreetMap. Tiles courtesy of

Алгоритм доставки замовлень

- Між кожною парою точок зберігатимемо час, який потрібен, аби подолати відстань між ними для кожного водія.
- У кожен момент часу зберігаємо місцезнаходження усіх водіїв, час, після якого вони закінчать свою роботу, перелік замовлень кожного водія
- Для кожного водія зберігаємо впорядковану множину подій, які він повинен опрацювати впродовж наступних двох годин
- Нехай «стан» – це комбінація усіх множин подій усіх водіїв.
- Введемо функцію f від стану. $f(x) = \sum_{n=1}^{N} \Delta t_n{}^2$, де Δt_n– різниця між часом доставки та часом приготування замовлення під номером n.
- Назвемо деякий стан x оптимальнішим за стан y, коли $f(x) < f(y)$.
- Використовуючи **алгоритм імітації відпалу (Simulatedannealing)**, знайдемо усі множини подій
- При додаванні нового замовлення або іншої події знову знайдемо оптимальне рішення.
- Після кожного опрацьованого замовлення оновимо уявлення про тривалість часу
- Щогодини будемо перераховувати всі найкоротші відстані між кожною парою точок.
- Алгоритм набуває ознак самонавчального, із часом точність передбачень буде збільшуватись.

Алгоритм імітації відпалу

- Згенеруємо початковий стан, використовуючи жадібний алгоритм, та не розглядаючи той випадок, коли двом водіям вигідно зустрітись. Назвемо цей стан «стабільним».
- Зафіксуємо початкову «температуру» T $= t_{max}$.
- Згенеруємо новий стан, помінявши у часі дві події деякого навмання вибраного водія.
- Порівняємо функцію f від новоутвореного стану та стабільного. Замінимо стабільний стан новим з вірогідністю $p = \exp\left(\frac{f(\text{стабільного}) - f(\text{нового})}{T}\right)$.
- Знизимо «температуру» T.
- Якщо «температура» вища за t_min, то повторимо ітерації III – V. До того ж, можна стверджувати, що для знаходження оптимальної оцінки у часі достатньо обмежитись прогнозуванням маршруту на півтори-дві години у майбутнє. Врахувавши це, орієнтовну кількість замовлень, а також швидкість знаходження f, то можемо надати оцінку, що цей алгоритм у змозі виконати 5 – 50 мільйонів ітерацій за секунду.

Демонстрація роботи

- Процес інсталяції:
  - Інсталяція платформи PHP+MySQL (OpenServer, Denwer)
  - Інсталяція системи - EFY-installer.exe
  - Створення та імпортування бази даних OSPanel\domains\restoran\restoran_db.sql
- Робота онлайн:

https://eatforyou.tk